A Novel Clustering Approach for Minimizing Transportation Lanes of Complex Supply Chain Networks based on Economic Significance

Eardley, Matthew Peter
(Student number: 1891286)

School of Mechanical, Industrial and Aeronautical Engineering

University of the Witwatersrand

Johannesburg, South Africa.

Supervisors:
Dr Bührmann, Joke

A research thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering.

Johannesburg 2024

Declaration

I declare that this thesis is my own unaided work. It is being submitted for the Degree of Doctor of Philosophy to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

(Signature of Candidate)

7th day of April 2025 in Pretoria

Abstract

This work studied a business problem in transportation management identified as an area of future research by Caplice and Sheffi (2003) and Acocella and Caplice (2023) and presented it as a fully formulated mathematical optimisation problem. This new clustering problem, here named the lane elimination clustering problem (LECP), is fundamentally multi-objective. The formulation includes the individual principal objective functions of the LECP as well as extensions of these functions that may represent additional measures of utility to some decision-makers.

Known evolutionary algorithm methods were applied to solve the LECP for three problem instances using large-scale, privately owned datasets with up to 400 000 data points. Mutation and crossover operators, as well as constraint handling techniques, were developed as a general evolutional model of the LECP based on *a priori* knowledge of the problem domain.

The applied genetic algorithms included the aggregation selection solution approach to provide a decision-maker with a Pareto front approximation that represents suitable trade-offs with respect to the various fitness functions. Focus, however, was given to Pareto dominance selection approaches to generate these solutions. The Pareto-based NSGA-II, SPEA2, PESA and PAES were applied to the LECP. Parallelization strategies were incorporated into these algorithms to improve performance.

The performance data of these algorithms were empirically obtained and presented as a new comparative study of the efficacy of well-known multi-objective optimisation solution approaches applied to a constrained combinatorial problem. Results were compared and discussed as the basis for suggested approaches to solving the LECP. A single integrated pMOEA approach to optimize lanes was therefore provided to the field of transportation science. A new case study of evolutionary methods applied to a combinatorial multi-objective problem was also thereby contributed to the field of operations research.

Although the problem was initially framed in the context of transport procurement event design, this research acknowledged the applicability of the LECP and its algorithmic optimisation to other areas of transportation management. This thesis concludes with a discussion regarding the relevance of understanding the trade-offs of this multi-objective optimisation problem with respect to these additional operational and strategic activities. Notes regarding the implementation of the LECP are given for each scenario, both in terms of the specific approaches to be used and the process by which the algorithms should be deployed.

Acknowledgements

For my wife and best friend, Luyanda. In loving memory of my son, Huey. In hope of joy and protection for my daughter, Nguse. Because of my parents, Connal and Pamela. With love for my family, the Eardleys, Athertons, Nguses, Kriges and Booths. With gratitude to my supervisor, Dr Bührmann. With thanks to my mentors Messrs Larkens, Allen, Schubert and Gower-Winter. Through my Lord.

"Programming sticks upon the shoals
Of incommensurate multiple goals,
And where the tops are no one knows
When all our peaks have become plateaus
The top is anything we think
When measuring makes the mountain shrink.

The upshot is, we cannot tailor
Policy by a single scalar,
Unless we know the priceless price
Of Honor, Justice, Pride, and Vice.
This means a crisis is arising
For simple-minded maximising."

K. Boulding

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Meaning
ADM	Average distance measure
AHD	Averaged Hausdorff distance
AWCD	Average within cluster distance
BLNE	Botswana, Lesotho, Namibia and Eswatini
BSU	Basic spatial unit
CCGA	Cooperative coevolutionary genetic algorithm
CDMOMA	Cross dominant multi-objective memetic algorithm
CGA	Coevolutionary genetic algorithm
csv	Comma separated value
CVI	Cluster validity index
DCCEA	Distributed cooperative coevolutionary algorithm
DCI	Diversity comparison operator
dEA	Distributed evolutionary algorithm
DM	Decision-maker
DRSA-EMO	Dominance-based rough set approach evolutionary multi-objective algorithm
EA	Evolutionary algorithm
ER	Error ratio
ERMOCS	Elitist recombinative multi-objective genetic algorithm with coevolutionary sharing
FEMCOC	Fuzzy-based evolutionary multi-objective clustering for overlapping clusters
FTL	Full truckload
FMCG	Fast moving consumer goods
GA	Genetic algorithm
GD	Generational distance
GraSC	Graph-predicated sequence clustering
GRIST	Gradient-based interactive step trade-off
GSA	Genetic symbiosis algorithm
HV	Hypervolume
IGD	Inverted generational distance
LECP	Lane elimination clustering problem
LSOP	Large-scale optimisation problems
LTFE	Lifetime fitness evaluation
M-PAES	Memetic Pareto archive evolution strategy
MDR	Mutual domination rate
MEMOEA	Memetic multi-objective evolutionary algorithm

Micro-GA	Micro genetic algorithm
ML	Machine learning
MOC	Multi-objective clustering
MOCCGA	Multi-objective cooperative coevolutionary genetic algorithm
MOCDP	Multi-objective cluster density peak
MOCLE	Multi-objective clustering ensemble method
MOCK	Multi-objective clustering with automatic K-determination
MOEA	Multi-objective evolutionary algorithm
MOGA	Multi-objective genetic algorithm
MOGLS	Multi-objective genetic local search
MOKGA	Multi-objective k-means genetic algorithm
MOMGA	Multi-objective messy genetic algorithm
MOP	Multi-objective optimisation problem
MOSGA	Multi-objective struggle genetic algorithm
MSPC-GA	Multi-sexual-parents-crossover genetic algorithm
MSR	Mean of the squared residual
NFL	No free lunch
NOM	Neural optimisation machine
NPGA	Niched pareto genetic algorithm
NSCCGA	Nondominated sorting cooperative coevolutionary genetic algorithm
NSGA	Nondominated sorting genetic algorithm
OD	Origin-destinations
OR	Operations research
PAES	Pareto archived evolution strategy
PBM	Pakhira-Bandyopadhyay-Maulik index
PD	Partition diameter
PDMOSA	Pareto dominant multi-objective simulated annealing
PESA	Pareto envelop-based selection algorithm
pMOEA	Parallel multi-objective evolutionary algorithm
PSO	Particle swarm optimisation
RasID	Random search with intensification and diversification
RFP	Request for proposals
RFQ	Request for quotations
SA	Simulated annealing
SCA	Symbiotic coevolutionary algorithm
SCN	Supply chain network
SDM	Sigma diversity metric

SMOSA	Serafini's multiple objective simulated annealing
SOP	Standard operating procedure
SPEA	Strength Pareto evolutionary algorithm
SSR	Sum of the squared residuals
TMS	Transportation management system
UoM	Unit of Measurement
VAT	Value-added tax
VEGA	Vector evaluated genetic algorithm
WDP	Winner determination problem
XB	Xie-Beni index

Chapter 1

Introduction

The complex and dynamic nature of transportation systems adds significant difficulty to the task of determining the best possible operational and strategic courses of action for supply chain decision-makers (DMs). There are substantial benefits associated with the computational mathematical optimisation of these decisions, resulting in transport optimisation being a widely studied area of operations research (OR) (Grazia Speranza, 2016).

Exact and inexact optimisation methods have been successfully developed and applied to air, public, railway, road freight and intermodal transportation problems. However, the geographic aspect of every transport leg can complicate the optimisation and execution of even the simplest decisions in practice, including rules-based decisions for which algorithmic support is not needed. Fine-grained representations of a supply chain network (SCN) with multitudinous points (or *nodes*) stratify transactional data, which can give rise to over-fitted solutions or procedures that are clumsy to implement and maintain (Manout and Bonnel, 2019).

Aggregation of locational data is an important and common analytical practice among supply chain professionals who seek simpler representations of their networks to aid decision-making processes. This simplification is often achieved using arbitrarily government- defined boundaries or data clustering methods that generate partitions without identifying similarity in a manner that supports the optimisation effort (Catini, 2015). In this research a particular SCN simplification problem is studied for which a fit-to-purpose clustering algorithm is applied.

By addressing this problem, this research aims to provide supply chain modellers and analysts with an effective method of aggregating transport lanes in terms of pricing proximity and lane count reduction potential. This, in turn, can be used to provide transport managers, operators and analysts with geographic clusters that are specifically developed to support cost optimization decisions.

1.1 Background and motivation

1.1.1 Lane elimination

Procurement of truckload capacity is a significant challenge to both the buyers and sellers of transport as they seek to optimize their costs and resources (Adjavor *et al.*, 2020). Companies that use third-party logistics providers to transport freight frequently conduct procurement events to source information from the transportation market. These exercises are carried out with the intention to reduce transport

cost and/or supply risk, either by securing more competitive pricing from its current or new service providers or by sourcing quotations for new transport requirements arising from a cost-saving strategy to be implemented in the future. During procurement events, the company (or *shipper*) provides the pertinent detail of the required transport to the participating transport providers (or *carriers*). Carriers then submit their prices (or *rates*). Rates are often submitted per route (or *lane*) described by the shipper. A lane was defined by Basu *et al.* (2010) as a "unidirectional movement from an origin to a destination for a certain time period" and is used as the "basic unit of interest" for full truckload procurement. A *lane rate* is defined by Lindsey *et al.* (2013) as a transportation charge associated with an origin-destination pair.

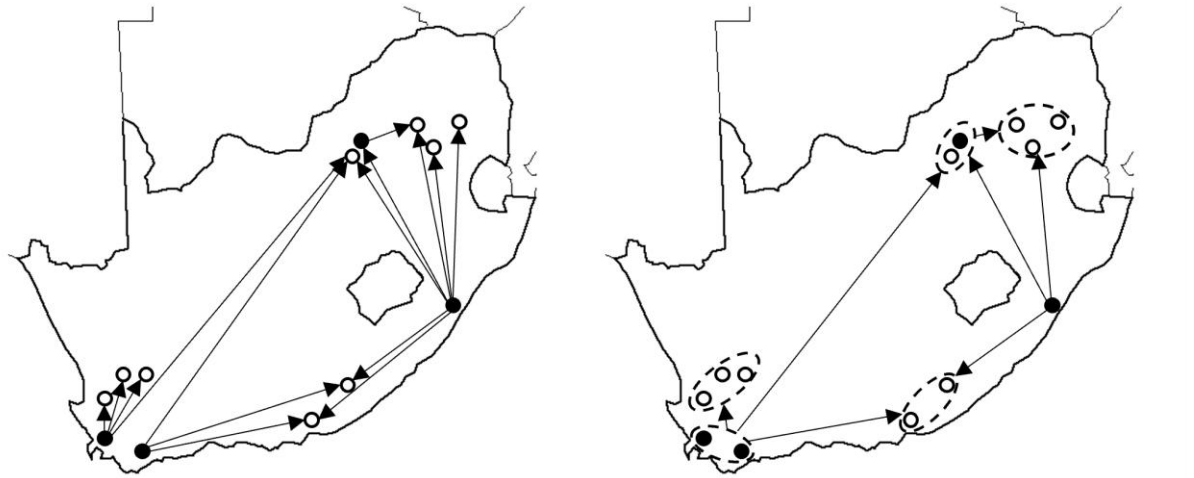
Caplice and Sheffi (2003) described various transport procurement event design dimensions that should be addressed by the shipper. One cited design element is the method by which the lanes are defined and communicated to participating carriers (Sheffi, 2004). Shippers can use discretion in terms of how the origin and destination of a lane are defined and communicated to bidding carriers (Caplice, 2007). Facility street addresses, suburbs, cities, regions, provinces, postal codes or countries can be used individually or in combination to describe origins and destinations of lanes. The approach chosen by the shipper therefore determines the specificity of the lane descriptions. This, in turn, influences the resolution of the bidders' view of the shipper's SCN and, by extension, the services for which they are quoting.

In real-world scenarios, a lane's origin and destination need not be defined at the same level of specificity (Liu and Miller, 2021). A lane may be classified as location-to-location, location-to-zone, etc., where a zone is a defined set of nodes (Caplice, 2007). Consistency is not required, and overlapping zones are allowed. For example, a shipper may define one lane as city-to-country (e.g. "Pretoria, South Africa, to Lesotho") and another as province-to-suburb (e.g. "Gauteng province, South Africa, to Sandton"). Furthermore, the shipper is not limited to standard geographic boundaries for defining lanes. For example, the origin of a lane could be defined as a group of warehouses in adjacent cities (Caplice and Sheffi, 2003). This flexibility results in a large decision space associated with the lane definition of an SCN of moderate scale and complexity.

The maintenance of a multi-carrier freight price dataset (or *rate card*) can be an onerous task and its complexity often "imposes high requirements for modelling freight rate costs functions for usage in transportation planning and optimisation" (Seiler, 2012). It is therefore desirable to define origins and destinations in broad terms (e.g. using provinces instead of cities, using cities instead of suburbs, etc.). In so doing, a shipper aggregates and eliminates lanes from its defined SCN. It thereby simplifies the network and consolidates volumes (Caplice, 2007). Collins and Quinlan (2010) noted that low-volume lanes are difficult to fit into shippers' existing networks, which complicates the development of transport

strategies. They also stated that shippers may deliberately seek a network representation with reduced lanes to lower the management cost of maintaining large numbers of rates.

Figure 1 illustrates an example of how clustering origins and destinations in a network can reduce the number of lanes from fifteen to seven, which, if used as the framework for defining the network, reduces the number of lanes communicated to the carriers. This can thereby expedite the bidding process, lessen the number of rates submitted, simplify subsequent rate analysis, and streamline carrier allocation decisions (Caplice and Sheffi, 2003). The latter consideration is particularly relevant as most shippers evaluate bids on a lane-by-lane basis (du Perron, 2008).



a) Unclustered network b) Clustered network

Figure 1: The effect of clustering on lane count (author's own illustration).

Caplice (2021) added that proliferation of lanes can complicate the configuration of the transport management system (TMS) used to execute daily transport planning and carrier allocation decisions, as well as payment and audit functionality. It follows that lane aggregation while considering transport rates, which are a key component of a TMS setup, can simplify these aspects of the shipper's transport management operations.

Furthermore, Collins and Quinlan (2010) found that the bundling of proximal lanes can provide tangible cost savings. This phenomenon is due to the reduced probability of a shipper having to revert to ad hoc or *spot* rates when aggregated lanes are used to supplement a shipper's point-to-point routes.

Acocella and Caplice (2022) showed that lanes defined on a granular level are more likely to become ghost lanes. These are lanes included in an original request for bids, but no transport on those lanes actualizes thereafter. This is an undesirable outcome, as it creates unnecessary rate management overhead, complicates carrier assignment optimisation and may lead carriers to capacitate in the

expectation of work along these lanes which ultimately does not materialise (Acocella and Caplice, 2022).

Simplification of an SCN also presents benefits for more holistic optimisation. These include the use of lane-covering optimisation methods to allocate carriers to lanes as proposed by Kuyzu (2007) and Ergun *et al.* (2005). The aspect of lane definition, however, is not discussed in their research. Basu *et al.* (2010) recognized that, for this type of optimisation, the magnitude of difficulty increases with the number of lanes. It follows that there is a carrier award optimisation benefit associated with defining SCN lanes using clusters as origins and destinations. This variable was manipulated by using differing numbers of zip-code digits to define lane origins and destinations. This research, however, aims to provide a more sophisticated and appropriate method of adjusting lane count by treating lane endpoints as configurable clusters unrestricted by pre-existing government boundaries.

Lane elimination also carries a volume consolidation effect, which allows the award of meaningful volumes per lane to carriers even in highly diverse networks (Holter *et al.*, 2016), as well as to determine optimal volume guarantees to ensure continuity of service during peak periods (Lim *et al*., 2008). For example, if 210 truckloads are transported weekly through the network shown in Figure 1(a), an average of 14 loads must be managed per lane. However, should the network be defined according to the aggregated lanes of Figure 1(b), 30 loads are transported per lane. This concept was supported by Caplice (2021), who presented four key dimensions of carrier rate and tender acceptance rates. These include total lane volume, variability and consistency, all of which can benefit from the improved depth of data that can be reasonably expected with the use of aggregated lanes.

However, this elimination of lanes may have a commercially negative influence (Collins and Quinlan, 2010). Bandaru and Dolci (2020) warn that the cost saving associated with lane aggregation is subject to the size of origin and destination regions. By reducing the number of lanes, a shipper denies the bidding carriers the opportunity to differentiate their submissions based on the otherwise visible specifics of the network. Caplice and Sheffi (2003) stated carriers hedge bids when uncertainty is present due to the quality of information provided by the shipper during the bidding stage of a procurement event. It follows that the uncertainty regarding stop locations and distances of potential routes defined using large geographic areas could have an escalating effect on transport rates. Location ambiguity due to large node definitions could, therefore, escalate rates where regional price sensitivity is high (Aemireddy and Yuan, 2019).

Figure 2 illustrates this phenomenon using the following example scenario. A carrier may quote R20 000 for transporting a truckload from Cape Town city to Johannesburg and R25 000 for a load from Cape Town city to Pretoria city. If the lane is defined as Cape Town city to Gauteng province, as shown in

Figure 2(b), which groups cities Johannesburg and Pretoria as a single provincial destination, the bidder is unsure of the true distance of the routes. Furthermore, uncertainty regarding the precise final destination of the vehicle would reduce the carrier's ability to assess the true value of the lane through the lens of its own network balance. Caplice and Sheffi (2003) specifically stated uncertainty around the availability of follow-on loads leads to carrier hedging of bids. It follows that the carrier is likely to bid a lane rate closer to R25 000 than R20 000 when required to submit a single price for the aggregated lane. This process is often referred to as *aligning to the top*. The shipper therefore incurs a higher cost overall by clustering the two destination cities for the purposes of SCN simplification.

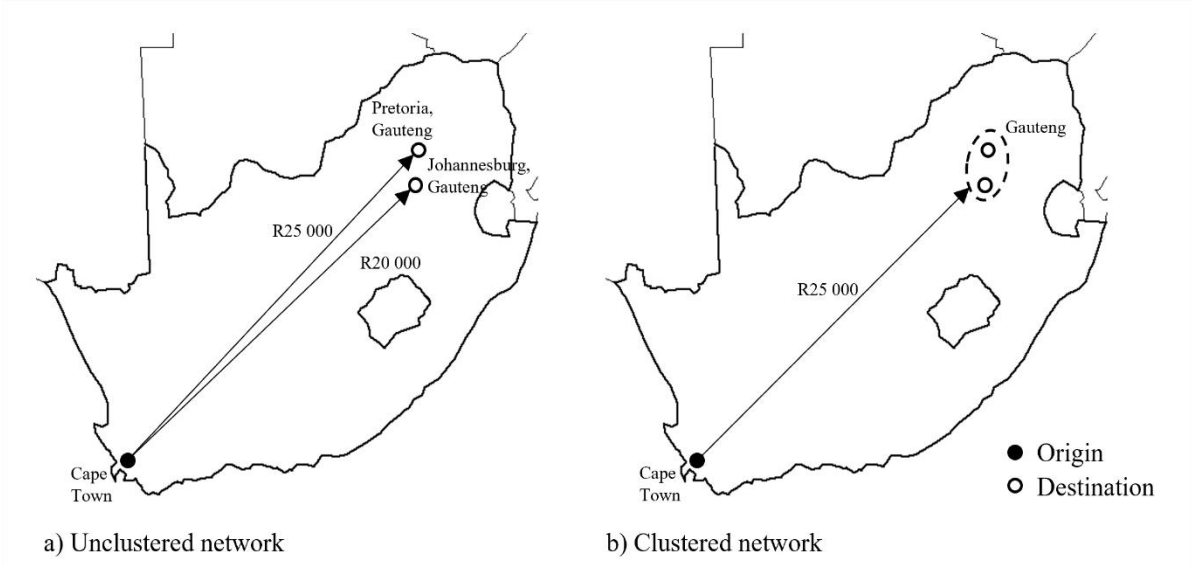


Figure 2: An illustration of the detrimental increasing effect of clustering on a transport rate (author's own illustration).

There can consequently be a cost associated with the combination of lanes due to this loss of information. A trade-off therefore exists between this *clustering cost penalty* and simplification of an SCN. Network simplification must, therefore, be balanced against economically relevant granularity due to the trade-off between the clustering cost penalty described above and lane elimination. Caplice (2007) described this as a trade-off between a network representation's effectiveness and coverage. The manner by which nodes are grouped can have significant and surprising effects on both these properties.

1.1.2 Manual lane definitions

No mathematical method or framework for the optimisation of this grouping of the shipper's network could be found in the literature. Opportunities to simplify the lane structure of a network are manually identified, while attempts are made to retain the necessary detail within the transport pricing structure (Caplice and Sheffi, 2003). In the absence of quantitative tools, practitioners define lanes for procurement events using rudimentary data analysis, administrative units and their own rules-of-thumb based on existing operational knowledge (Martinez *et al*., 2009).

There is considerable complexity associated with these manual methods. This can be attributed to:

1. Although important, geographic proximity is not the most important attribute when clustering freight demand (Mesa-Arango and Ukkusuri, 2015).
2. The number of available levels of geographic classification for defining an origin and/or destination.
3. The flexibility to inconsistently define lanes using different levels of classification.
4. Existing freight flows may not link every node in the SCN and not all flows are bi-directional. Figure 3 shows how the number of lanes eliminated by grouping nodes depends on their degree of linkage with other points. In both illustrated scenarios, two nodes are grouped into a single destination. However, grouping in Figure 3(c) resulted in the elimination of a lane, while the clustering shown in Figure 3(b) had no bearing on the lane count. This phenomenon can result in surprising interactions between clustering decisions and the extent of lane elimination achieved.
5. Clustering cost penalties are a function of the carrier rates to or from SCN nodes. They are therefore only partially dependent on geography and lane distance, as rates are significantly influenced by factors such as backhauls offered by the shipper, anticipated empty leg distances to other shippers' loading facilities (Caplice and Sheffi, 2003), the proximity of carrier depots (Caplice, 2007), cross border requirements and seasonal aspects. Figure 4 illustrates how special requirements can result in a node being geographically closer to one location, but economically closer to another.

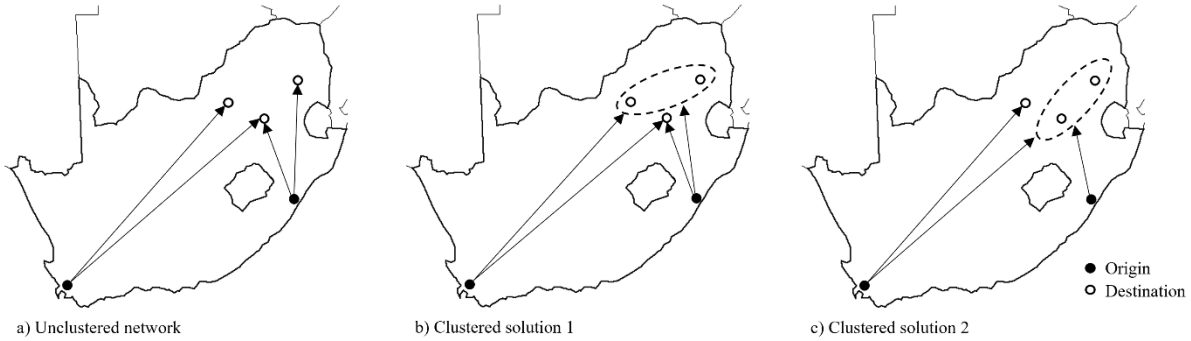


Figure 3: An illustration of the effect of the existing freight flows on the elimination of network lanes by clustering (author's own illustration).

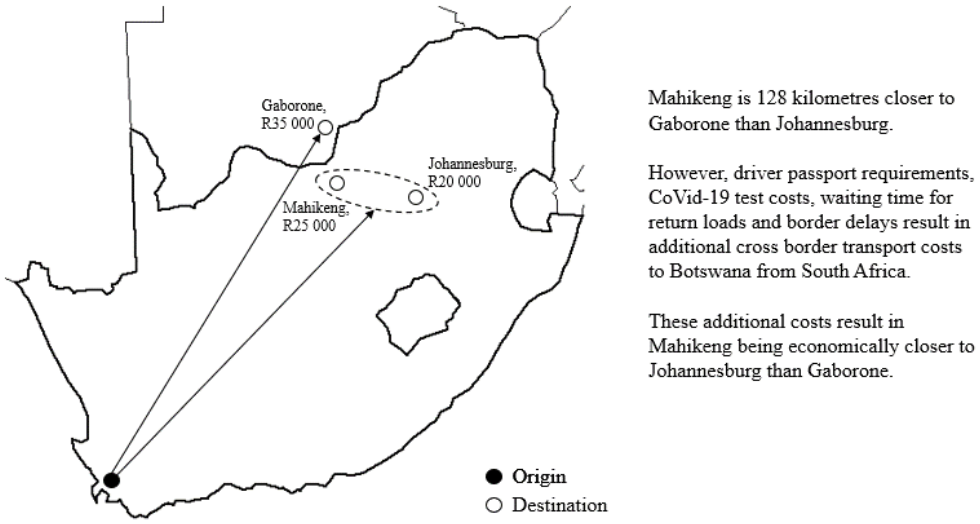


Figure 4: An illustration of the effect of cross border transport considerations on the cost of clustering (author's own illustration).

Further to the benefits of optimality, Jain and Dubes (1988) recognized the use of an algorithm to organise data introduces a systematic consistency and reliability to a clustering exercise that, if done manually, could otherwise be influenced by a person's background. In the context of transport economics, this could refer to a person's bias to group towns of an unfamiliar region, yet view suburbs of their home city as distinct areas. Considering these clusters have financial implications in many contexts, a systematic method for this grouping could be advantageous solely due to the removal of subjectivity from these exercises.

In summary, the complexity of this problem means replacement of manual exercises with a quantitative method that clusters nodes as a mathematical optimisation problem may produce more desirable solutions. The use of algorithms to solve this optimisation problem by finding good compromises between the simplicity of the SCN and retention of valuable pricing resolution should therefore be investigated.

1.2 Problem Statement

Clustering of nodes can simplify the definition of distribution networks and is used widely in practice to manage the number of lanes that comprise SCNs. However, the rates submitted by transporters for a lane, and by extension, a company's associated transport cost, are not solely dependent on loaded distance (Kuyzu, 2007). Shippers must also consider multiple carriers' overlapping networks and rates that may be uniquely formulated for the shipper. It can therefore be argued that transport procurement and management requires SCN nodes to be clustered from an economic, rather than a geographic, perspective (Acocella and Caplice, 2021).

A mathematical approach is required to solve the above-described problem, hereafter referred to as the lane elimination clustering problem (LECP). The generation of simplified clusters versus the retention of essential economic granularity discussed above must both be considered to calculate the optimal pricing schedule for the shipper. Due to the necessary looseness of dynamic transport contracting, in which prices are agreed but availability is not (Caplice, 2021), as well as the intangible nature of many of the benefits of load count reduction, the conflicting objectives of the LECP must be addressed such that optimal solutions are generated for retrospective (or *a posteriori*) selection according to the preferences of the DM.

The purpose of this thesis was to formally define the measurement of the objectives of the novel LECP and explore it as a multi-objective optimisation problem (MOP). This included the investigation of valid methodologies for solving this new problem, including novel heuristic adaptations uniquely designed to solve the introduced LECP. This research also introduced three new problem instances and an empirical evaluation of the performance of different multi-objective optimisation solution approaches on this new MOP.

1.2.1 Limitations and scope

The formulation of the LECP was limited to the following business scenario for this research:

- Rates are exclusively for full truckload transport priced on a contracted, flat, per-load basis for a standard equipment type from a pre-specified origin to a pre-specified destination. Loads comprise a maximum of one pick-up and one drop-off stop. En route diversions do not take place. Fixed costs and additional charges not reflected on rate cards are not considered.
- The lanes of the SCN are represented by the carrier lane rates. A rate only exists if cargo has been or is reasonable to expect to be transported for the shipper by a carrier from the rate's associated origin to its associated destination, however these may be defined. Any freight flows not reflected by the rate card are not considered part of the defined SCN and do not require an associated lane definition.
- The extent of resultant lane rate hedging expected by carriers when lanes are aggregated using clustering methods is unknown. The process of fully rounding costs upwards can be simulated and used as a proxy measure for the loss of economically significant information associated with the simplification of the defined SCN.
- The economic significance of the geographic position of individual SCN nodes is partially dependent on the distance of the routes to and from these locations. To incorporate this dependency, lanes are classified according to two distance brackets, namely *local* and *long-distance*, and can be aggregated according to a distinct cluster structure assigned to the bracket. For example, destination cities within a 100 kilometres radius may be clustered with little impact on the cost of deliveries from a town 1 500 kilometres away. However, the distance between

these nearby cities bears more relevance to the prices of transport originating from a more immediate vicinity. This is illustrated in Figure 5. A greater number of distance brackets would reduce the number of intrazonal loads, which aligns to the goal of minimization of intrazonal loads as presented in Martinez *et al*.'s (2009) summary of transport analysis zone definition guidelines. The use of more or fewer brackets can be implemented for further study. It is assumed Euclidean straight-line distances provide sufficient accuracy with regard to classifying lanes, although the use of road network distances can be explored as further research.

- The specific distance break value, expressed in kilometres, that is used to classify lanes in terms of local and long-distance categories is not imposed. This value is therefore a decision variable for optimisation.
- Separate consideration of origin and destination nodes is allowed when aggregating lanes by clustering. Figure 6 shows how identical groups of adjacent nodes can be clustered differently as origins and destinations. Distinct sets of optimal origin and destination clusters for each lane distance bracket are therefore required, resulting in a solution consisting of four cluster sets.
- The number of origin and destination clusters of each cluster set is not known *a priori*.
- The use of geographically overlapping or discontinuous clusters is prohibitively difficult to implement in practice and their use for transport analysis is discouraged (Martinez *et al*., 2009). Overlapping clusters within a cluster set are therefore not permitted.
- The inclusion of the same node in two different clusters of the same set would introduce ambiguity to transport pricing and is not permitted.
- Value-added tax (VAT) is billed to the shipper equal to the amount owed to the revenue authority by the carrier. All rates are analysed and negotiated exclusive of VAT. The Southern African Customs Union (SACU) Agreement allows for VAT exemption for cross-border transport from South Africa to certain neighbouring countries (i.e. Botswana, Lesotho, Namibia and Eswatini). However, this is not a consideration of the clustering solution, as it has no net effect on the carrier's profitability and, by extension, no bearing on the economic significance of the destination node.

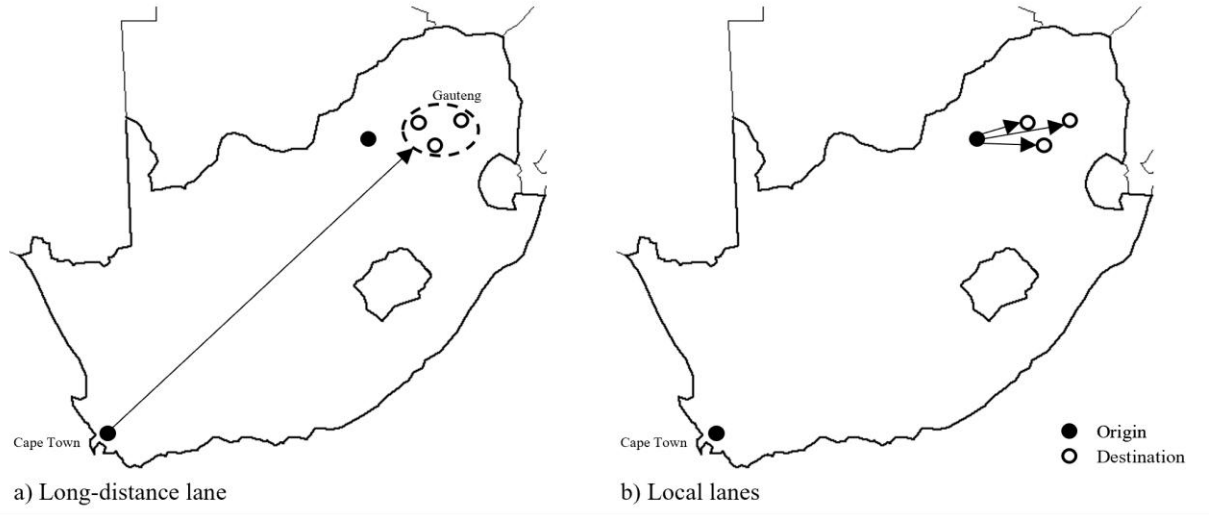


Figure 5: An illustration of distance considerations on the definition of transport lanes (author's own illustration).

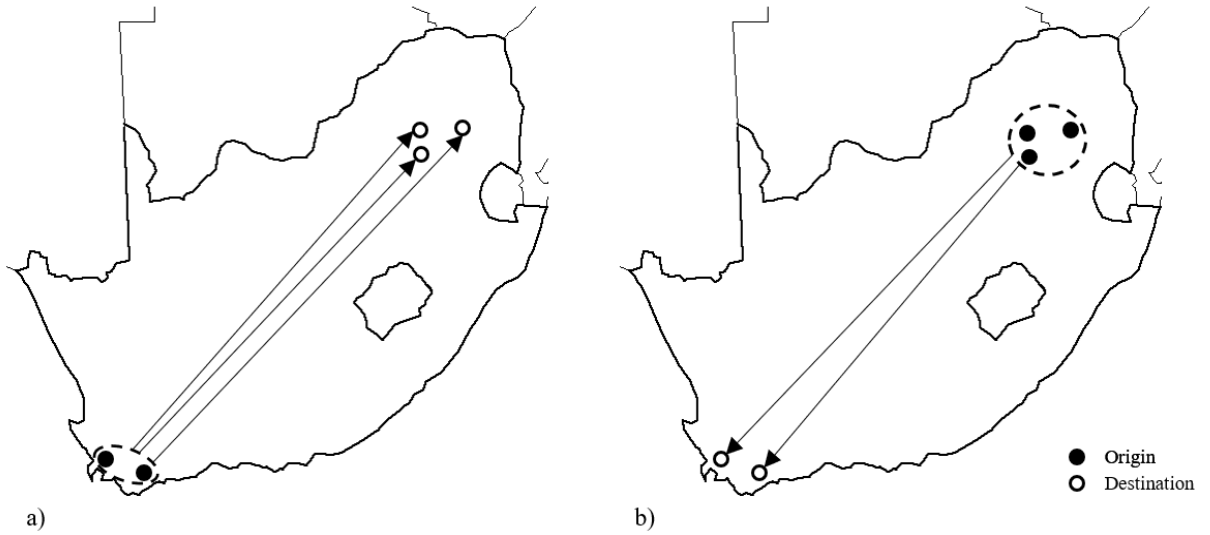


Figure 6: An illustration of directionality on the definition of transport lanes (author's own illustration).

This research positions the LECP as an MOP due to the incomparable and conflicting nature of the fundamental objectives of lane count minimization and rate inflation. MOPs have been addressed with a variety of optimisation techniques (Marti *et al.,* 2009). The optimisation approaches for addressing the LECP, studied for this research, were limited to evolutionary algorithms (EA), as they are known to be well-suited to MOPs (Khazaee, 2011). This is largely attributed to an EA's ability to simultaneously explore and exploit different regions of the objective space (Khazaee, 2011). EAs offer the added advantage of being able to address optimisation problems with discrete decision variables, such as the LECP, which is not the case for many optimisation techniques (Maier, 2019).

Several multi-objective evolutionary algorithms (MOEAs) in literature could be used to address the LECP. In their comprehensive review of many of these MOEAs, however, Coello Coello *et al.* (2007) analysed the advantages and disadvantages associated with each. Based on their analysis, they also provided a recommendation of five multi-objective MOEAs, or *solution approaches*, for use in an initial exploratory investigation of a new MOP. These are:

- Aggregation selection using linear combination of weights.
- Nondominated sorting genetic algorithm II (NSGA-II).
- Strength Pareto evolutionary algorithm 2 (SPEA2).
- Pareto envelope-based selection algorithm (PESA).
- Pareto archived evolutionary algorithm (PAES).

Coello Coello *et al.* (2007) also included the micro-genetic algorithm (Micro-GA) in the list of recommended *a posteriori* methods for consideration. The authors did, however, also acknowledge the number of parameters required by this technique as a disadvantage. Such algorithms were avoided during the scoping of this research to mitigate the influence of fine-tuning on the fairness of the comparative analyses.

The above solution approaches represent *a posteriori* techniques only, as these provide more information to the decision-maker (DM) than *a priori* and progressive methods (Mavrotas, 2009). This research specifically did not explore *a priori* methods under the assumption that a DM would have significant difficulty in articulating preferences for the LECP, which would severely negatively impact the quality of the generated solution. Furthermore, as it is clear that the introduction of pre-configured preferences would unnecessarily limit the search space of the problem and opportunities to find good solutions would be missed, *a priori* preferences were not accommodated.

Progressive approaches were not developed or implemented for this thesis since the interactive incorporation of the DM's preference makes testing highly subjective and impossible to compare to other approaches (Coello Coello *et al.,* 2007). Methods for the articulation of DM preferences for the LECP are therefore out of the scope of this research, although this is available as a topic for further study. Furthermore, for the purposes of this research, a LECP solution approach is limited to the generation of solutions and the determination of a good Pareto optimal approximation. Subsequent selection processes to prune the nondominated solution set or aid the DM in identifying a single solution are not in the scope of this research.

Coello Coello *et al.* (2007) stated MOEAs that use aggregation functions are widely considered as not true MOEAs. However, the additional computational cost associated with determining the Pareto optimality of solutions mean that methods that do not use Pareto ranking should not be ignored if the

LECP is to be solved within the wall clock constraint that might arise in a particular context. Aggregation selection using linear combination of weights is therefore included with four Pareto-based selection approaches as the set of algorithms applied to the LECP. For ease of reference, this method is described as one of the five in-scope MOEAs for the experimental component of this research. These approaches were tested with real-world datasets and their quality indicator values collected. Discussion is provided and a recommended approach is identified.

The five solution approaches to the LECP were selected based only on the literature reviewed for scoping of this research. This process incorporated problem domain knowledge introduction as recommended provided by Coello Coello *et al.* (2007) and supported by the NFL theorems (Wolpert *et al.*, 1997). As such, the selection of the assortment of tested MOEAs is not part of the research methodology. Rather, the methodology is designed to accommodate this assortment.

Coello Coello *et al.* (2007) indicated that MOEA experimentation typically does not include extensive analysis of the sensitivity algorithm behaviour to parameter values. They also acknowledged the no free lunch (NFL) theorems, which state that incorporating too much problem domain knowledge into an approach reduces its effectiveness on other problems, even within the same class (Wolpert *et al.*, 1997). Therefore, as a guiding principle, where *a priori* definition of parameters is required, values are selected to reflect the most generic modelling scenario imaginable. As such, parametric tuning is performed only at a coarse level to explore the problem domain, as well as evaluate and compare approaches with settings that do not unduly favour the performance of a subset of the tested MOEAs and thereby provide useful results when used across solution approaches and problem instances. These parameter values thereby provide only a starting point for implementation of the recommended approach in practice.

Evaluation of the approaches is performed using three shippers' historical data with the following delimitations:

- Although the approach is applicable in any single region, the problem instances were restricted to shippers operate within the same South African sector and industry.
- Rate data are for the full use of 36-ton tautliner trucks as at June 2022, as this was the prevailing equipment type used by the studied shippers at the time.
- Rate origins and destinations are inconsistently defined. Rates are primarily defined as "city-to-city". Each shipper does, however, maintain some rates defined using specific facility addresses, provinces or uniquely constructed zones (or *regions*).
- Rate data includes carrier prices for cross border transport to the neighbouring countries. However, exports via ocean and rail transport were not included.

The focus of this research is to provide the necessary theoretical understanding of the trade-offs of the

LECP, based on the above limitations, to address real-world instances of the problem, as well as to demonstrate the approach. While the modelling recommendations given in this thesis, based on the findings of the demonstration, can be used by practitioners looking to simplify their SCNs from an economic perspective, the characteristics of the specific problem instance to which the approach is applied may affect the efficiency and efficacy of the recommended approach. Although the methods outlined in this research are proposed as a universally *applicable* approach to the LECP, they should not be interpreted as the universally *optimal* approach. Nonetheless, it is suggested that the practitioner use the recommendations as a starting point, which can be followed by further experimentation in terms of the solution approach and associated parameters to optimise performance.

1.2.2 Thesis objectives

Primary Objective

To present and solve, with real-world datasets and practical modelling considerations, the multi-objective LECP.

Supporting Objectives

1. Fully formulate the new multi-objective LECP. This includes:
    1.1. Definition of decision variables.
    1.2. Formulation of objective functions.
    1.3. Formulation of constraints.
    1.4. Commentary on the formulation and the characteristics of the problem.
2. Code evolutionary algorithms to solve the LECP. This includes:
    2.1. Selection and adaptation of existing, or development of a novel suitable encoding scheme for chromosomal representations of solutions.
    2.2. Selection and adaptation of existing, or development of a novel appropriate population initiation method and termination criterion.
    2.3. Selection and adaptation of existing, or development of a novel problem domain-specific recombination and mutation operators.
    2.4. Adaptation and implementation of selected existing multi-objective evolutionary algorithm (MOEA) solution approaches to solve the LECP.
3. Test, analyse and discuss the performance of the MOEAs such that the findings can be used as recommendations to practitioners for solving the LECP. This includes:
    3.1. Design of a suitable experiment for testing and comparing the various MOEAs.
    3.2. Execution of the MOEA testing experiment to establish that an evolutionary approach can solve the LECP.
    3.3. Presentation of experimental results with comparative analysis and recommendations.

1.3 Approach

The research objectives were addressed in this thesis according to a general approach of using the output of each section as an input to the subsequent section. A high-level overview of the four main sections of this thesis is presented here such that the relationships between the chapters may be better understood. These relationships are also shown in Figure 7.

1.3.1 Define the LECP

The first section of this thesis introduces a new clustering optimisation problem. This includes the establishment of the uniqueness of the problem. A review and analysis of literature relevant to the LECP was therefore conducted and, in so doing, the new problem could be positioned within the existing body of knowledge.

A separate review of pertinent MOP concepts, including widely studied multi-objective optimisation approaches with an evolutionary algorithm focus, was also completed. Theoretical grounding for the subsequent LECP formulation, as well as the solving of this new problem with the MOEAs prescribed by the scope of the thesis, was thereby found. The literature reviews were followed by the development of a detailed methodology for the research, which outlines how the theoretical and experimental components of this thesis were carried out and interacted to satisfy the research objectives.

The formulation was then developed. This definition of the LECP underpinned the optimisation component of this research.

1.3.2 Develop the evolutionary approach

The reviewed literature and problem formulation were used to identify and design essential aspects common to all the pre-specified MOEA solution approaches. These aspects are collectively referred to as the evolutionary approach of the LECP and were developed and coded in full prior to the implementation of the specific EAs. The performance metrics, selected from literature based on the characteristics of the LECP and the approaches to be tested, were also coded as standalone functions.

1.3.3 Implement MOEAs to solve the LECP

The third section of this thesis represents the experimental aspect of this research. The developed LECP evolutionary approach was applied to real-world instances of the problem according to the five solution approaches prescribed by the scope of this thesis. Some cursory investigation of the behaviour of each algorithm was conducted to explore the problem domain and perform coarse algorithm modification and parameter tuning.

As the problem studied for this thesis is novel, benchmark data did not exist. Solving of the LECP is defined for this research as the provision of a nondominated set of solutions that improves upon those generated by purely spatial clustering methods. Solving the LECP was limited to techniques that support *a posteriori* decision-making without making any assumptions regarding the DM's preferences. The k-means algorithm was chosen as the spatial clustering process for benchmarking the MOEAs. The performance of each algorithm was compared to the k-means baseline results and a determination was made in terms of its status as a viable solution approach. Extreme value testing and comparisons to the use of government-defined areas were also used to validate the MOEAs. A final comprehensive experiment was then conducted using a standard set of parameters. The raw data was analysed as a comparative analysis. This analysis gave rise to an MOEA recommendation for consideration when solving this new problem in practice.

The existing literature presents several strategies that can be introduced to MOEAs to enhance performance. To avoid proliferation of experiments, these methods were implemented only for the recommended solution approach. The effectiveness of these methods was then analysed.

1.3.4 Reflect on the LECP and application of MOEAs

This thesis concludes with a discussion around the relevance of the LECP beyond the background context described presented in chapter 1. Alternative logistics management scenarios in which value can be derived by elimination lanes were considered. Furthermore, suggested procedures for the implementation of the recommended approach were developed for each of these scenarios. Limitations and opportunities for further research of this new optimisation problem were also identified.

1.4 Thesis contribution

The primary and supporting objectives of this thesis encapsulate the contributions that will be made to the fields of OR and transportation science. The primary contribution is the furtherance of the structurization of transportation for mathematical treatment of problems as discussed by Borndörfer *et al.* (1998). This includes:

- The introduction of a new multi-objective optimisation problem that represents a real business problem (i.e. the LECP).
- A full mathematical formulation of the new LECP.

Caplice (2021) stated that "data-driven analysis on shipments, lanes and rates is required to intelligently segment lanes to be procured and managed". The multi-objective treatment of the LECP provides an example of such an analysis. It builds upon the work Collins and Quinlan (2010), who developed a model for providing a DM with the estimated cost impact of aggregating lanes for easier rate management. This thesis provides a method by which the lane aggregations are specifically generated

for such a comparison such that the DM can make the selection using the optimal SCN definition at each level of aggregation.

Furthermore, Holland (2000) stated that the use of standard pedagogical problems for EA development and experimentation is limiting. A new real-world analysis of the effectiveness of known MOEAs for solving complex analysis problems will be a contribution to this field.

As recognized by the NFL theorems (Wolpert *et al.*, 1997), problem domain knowledge should be incorporated into the approach for solving MOPs. The adaptation of solution approaches to the LECP and subsequently derived insights based on experimentation are therefore included as a further contribution of this research. This includes:

- New or adapted evolutionary operators for addressing the LECP using MOEAs.
- A methodology for testing the effectiveness and efficiency of MOEAs for the new LECP.
- A new set of test results comparing commonly used MOEA approaches using real-world data. These include aggregation selection using linear combination of weights, NGSA-II, SPEA2, PESA and PAES for multi-objective optimisation.
- A recommended MOEA solution approach to the new LECP.

Simchi *et al.* (2013) stated that sourcing, verifying, and tabulating the required data for applying OR techniques to transportation problems is very problematic. This research contributes a solution to one instance of this problem by providing an efficient and effective method for tabulating a frequently used set of specific data that is important to distribution cost management activities.

1.5 Thesis overview

This research can be summarised as follows. Chapter 2 reviews and discusses existing literature to position the new LECP within the existing body of research as a new multi-objective optimisation problem (MOP). Chapter 3 represents the principal literature review for this thesis and contains a review of commonly used MOEA solution approaches. Other theoretical topics relevant to the remainder of this research are also presented in this chapter before the methodology is described in chapter 4.

The LECP is then mapped from the problem domain to the algorithm domain as a full problem formulation in chapter 5. This chapter includes some commentary regarding relevant or interesting characteristics of the LECP.

Chapters 6 to 8 discuss the solving of the formulated LECP. Chapter 6 addresses the design of the common MOEA elements used when applying any of the selected MOEAs to any instance of the LECP. Chapter 7 presents these specific solution approaches. Chapter 7 focusses on the empirical exploration

of the behaviour and validity of each MOEA as a solution to the LECP. Chapter 8 covers the quantitative comparison of the validated approaches, including the presentation of the computational results, discussion and conclusions. This analysis provides a recommended MOEA for the LECP. This chapter also includes the methodology and empirical results regarding the incorporation of an alternative generic MOEA strategy into the recommended solution approach, which, in turn, forms part of the overall recommendation.

Chapter 9 addresses the application of the recommended MOEA in real-world contexts. It explores the various use cases of the solution approach and provides notes regarding the processes required to implement the code effectively. Chapter 10 concludes this thesis with conclusions and identified areas for future research.

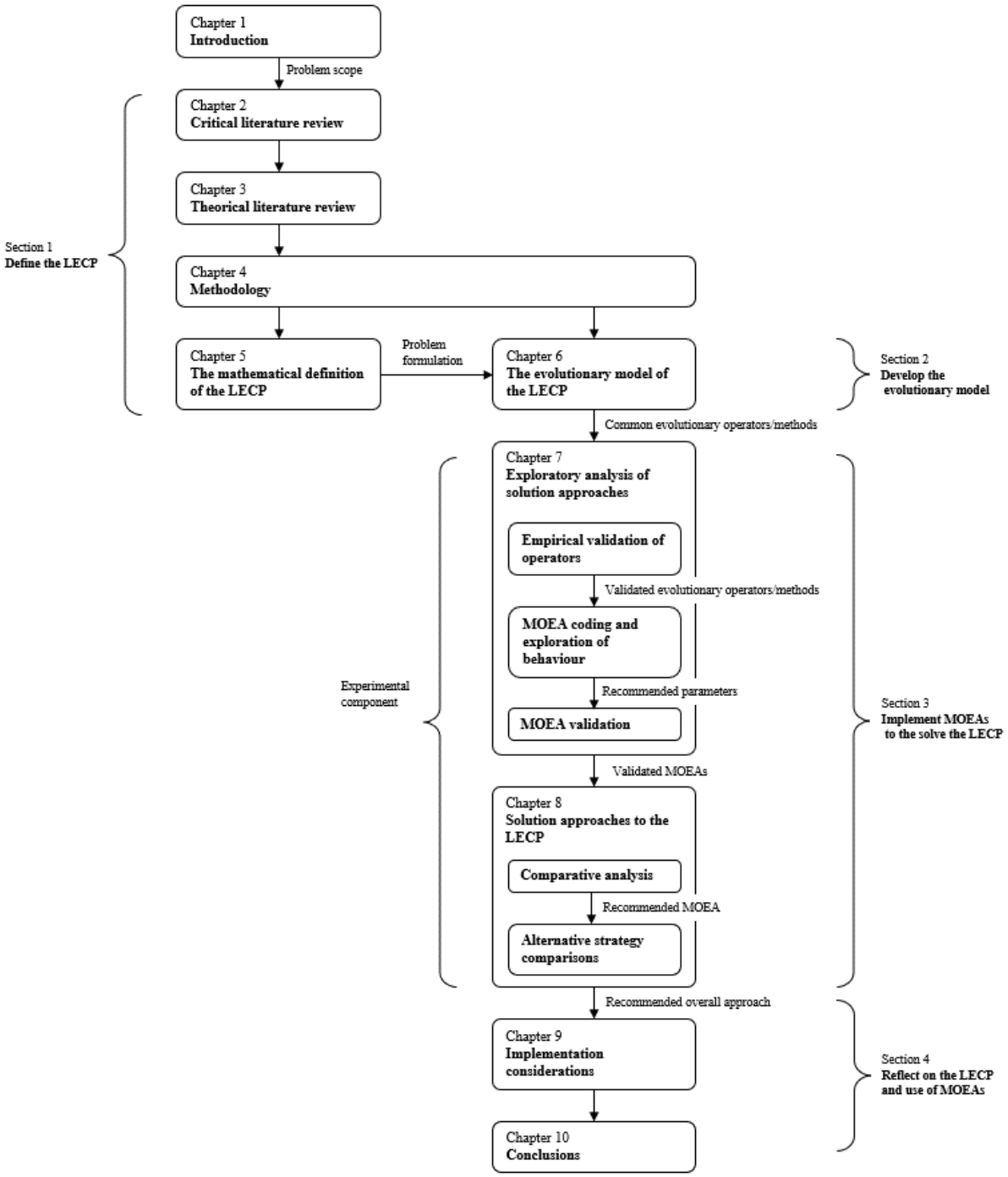


Figure 7: Thesis approach and chapter outline.

Chapter 2

Literature review – the LECP

In this chapter, the existing body of literature is critically reviewed to identify the gap in current theory and developed optimisation techniques with respect to the LECP. It thereby supports the uniqueness of this thesis' contribution to the body of knowledge.

2.1 Lanes and transportation optimisation

The term *lane* is ambiguous within the transportation context. The layman may associate it with a defined path on a road used to organise parallel traffic flow. However, in a supply chain and logistics management context, the term usually refers to a "unidirectional movement from an origin to a destination for a certain time period" as defined by Basu *et al.* (2010) or simply an "origin-destination pair" as described by Lee *et al.* (2007). This concept of a lane underpins several transport-related mathematical and process optimisation problems presented in the literature.

Clear and sensible definitions of lanes are required to implement solutions developed to address these problems. However, little consideration has been given by researchers to how these lanes are defined. This aspect of the solution is left to the modeller, who must typically work with real-world datasets that are often generated by processes with inherent quality issues (Arunachalam, 2017).

Examples of research that refer to transport lanes include Stank *et al.* (2000), who discussed several strategic and operational transportation activities and decisions that require distinct approaches for different lanes. These include lane-level carrier identification, outsourcing, functional cost trade-offs, synergistic use of inbound and outbound logistics, as well as vehicle and carrier consolidation. How these lanes might be defined such that these activities can be enhanced was not discussed.

Caplice (2021) alluded to the concept of carrier-lane assignment and noted that there is a lack of technology sophistication for strategic-level optimisation of this problem. Kuyzu (2007) provided in-depth discussion of lane covering problems and lane-based procurement of transport. However, topic of lane construction was not addressed. Kuyzu recognized the requirement for specialized techniques to manage carrier allocation optimisation when the number of lanes is large. He cited the use of problem decomposition and objective function approximation as possible strategies for overcoming this issue. He did not, however, identify alternative lane definitions as a method to reduce the number of lanes.

Like Kuyzu (2007), Yang *et al.* (2019) used the term *lane* in their formulation of the winner determination problem (WDP) without discussing how the origin and destination might be defined. For their formulation, they also define it simply as a combination of an origin and destination.

Some researchers do, however, give an indication in terms of how these origins and destinations are defined. The prevailing method for specifying lanes for mathematical optimisation problem formulations is the use of "cities" as origins and destinations. Examples include the formulation of a full truckload routing and scheduling optimisation problem given by Arunapuram *et al.* (2003), which was founded on a set of lanes defined in terms of starting and ending cities.

Although, Hao *et al.* (2016) did not explicitly refer to lanes in their formulation of a transport mode and routing optimisation problem, parameters specific to *routes*, such as transport costs and transit times, were central to the definition of the objective function and constraints. Here the term *route* was used to describe a single movement of a full equipment load of freight and is therefore synonymous with the term *lane*. Routes were defined for the formulation in terms of city-to-city movements. Triki *et al*. (2014) also used cities to describe lane origins and destinations for a formulation of the bid generation problem. For these formulations, the issue of how to overcome the ambiguity of the term *city*, the shortcomings of using cities as origins and destinations of relatively short lanes, and the benefit of aggregating cities into clusters for relatively long lanes, were not discussed.

Caplice (2021) referred to frequently referred to lanes in his discussion regarding the strategic aspects and pitfalls of the transport procurement process. He did not, however, address the method by which these lanes might be defined.

Kiser (2018) and Sheffi (2004), however, acknowledged lane definition as an aspect of transportation procurement and optimisation. Kiser (2018) briefly discussed the option available to shippers to disaggregate high volume lanes into smaller "individual lanes" during the post-auction carrier allocation activity. When discussing the pre-auction stage of this process, Sheffi (2004) recognized that a shipper defines a set of expected freight movements as lanes and that there is a "fair amount of discretion" available to the shipper in terms of how these lanes are defined. He also described the forecasting process for generating the expected movements for lane definition exercises as difficult due to historical flows being highly disaggregated.

Sheffi (2004) also noted that shippers are inclined to aggregate ship-to and ship-from locations to reduce the potential number of submitted rates that would need to be managed during the procurement process. Caplice (2007) described this more generally by stating that lanes can be classified as *point-to-point*, *zone-to-zone*, *zone-to-point* or *point-to-zone*, where a *point* is a specific address, city or postal code area

and a *zone* is anything larger than a point. These classifications were used by Dellbrügge *et al.* (2022) for their pre-auction stage design summary as shown in Figure 8.

Design feature	Design Options			
Classification of lanes	Point-to-point	Zone-to-zone	Zone-to-point	Point-to zone
Lane design approach	Threshold volume approach	Distance classes approach	Origin-destination pairs	
Demand forecasting approach	Estimation based on historic demands	Stochastic determination from historical data	Determination from future material flows	No demand planning
Distribution of forecasted demand	Based on day	Based on week	Based on month	Based on year
Lanes auctioned off	Higher volume lanes		Lower volume lanes	
	With provision of a volume forecast	Without provision of a volume forecast	With provision of a volume forecast	Without provision of a volume forecast
Agreement of back-up rates for lanes not auctioned off	Yes	No	Not applicable	

Figure 8: Part 1 of 3 of pre-auction stage design (Dellbrügge *et al.*, 2022)

Sheffi (2004), also recognized that rates defined using larger regions facilitate easier forecasting and analysis while simultaneously introducing greater uncertainty in terms of the number and distance of empty legs travelled within a region. Methods for optimally generating these regions while considering this trade-off were not cited in the research.

Only two methods for generating these regions were found in literature and were summarized by Dellbrügge *et al.* (2022). The first was proposed by Caplice (2007). He proposed a "threshold volume approach" by which aggregation of nodes is done such that lanes achieve a specified minimum volume. This approach requires a forecast of anticipated demand for each link of the SCN. This method focuses on aggregating volume and results in lanes that are fairly balanced from a volume perspective. This process eliminates lanes from the network but does not actively seek out clustering structures that minimise the number of lanes. It also ignores the pricing characteristics of individual nodes. As such, with this technique, adjacent nodes with different economic attributes may form a joint destination of a lane. This can result in significant information loss and prompt a carrier to hedge and quote the higher of the two rates that would have otherwise submitted separately.

The second approach is evident in the design of the Güterfernvekehrentgelte (GVE) that was described by Seiler (2012). The GVE is a standard rate structure used primarily for benchmarking in the German transportation industry. It uses a distance-based clustering structure for the purposes of rate design. Each lane is defined in terms of an individual origin and a cluster of all other nodes in the network within a defined distance range. Conversely, a lane can also be defined in terms of a cluster of origins, by distance bracket, linked to a single destination location. Dellbrügge *et al.* (2022) described this as a lane design approach "based exclusively on the point to zone or zone to point scheme". Although this scheme is

highly transferable and simple to implement, it assumes that transport distance is the only pricing factor and ignores specific economic attributes of nodes that might be inferred from historical pricing data. This approach can therefore significantly limit carriers' ability to price according to these factors. The method by which the specific distance values used to categorize the various lanes was not presented. It is also unclear whether the values determined using the input set of rates and some objective function, as is the case for the LECP.

2.2 Number of lanes and optimality

To date, no literature reference to the optimisation of SCNs from the perspective of minimization of the number of lanes could be found. However, there exist optimisation studies in the field of transport procurement in which the number of lanes were investigated as variables.

Basu *et al*. (2016) developed a weighted sum approach to the carrier allocation problem. Lim *et al.* (2012) presented a solution to the freight carrier allocation problem with the additional aspect of balancing the proportion of transport cost allocated to carriers. Both approaches were tested using varying numbers of lanes. Increases in run time with the number of lanes were observed in both studies, thus demonstrating the benefit of lane minimization from an optimisation performance perspective. However, the number of lanes were manipulated by creating subsets of the full SCN. The benefit of eliminating lanes through clustering was therefore not shown in these experiments.

Ergun *et al.* (2007), however, used nodal clustering to manipulate the number of lanes presented to an algorithm. The algorithm aimed to generate continuous transportation moves for carrier bidding, as opposed to individual one-way lanes. The results of the experiment indicated a strong direct relationship between the number of lanes and the optimality of the solution. This was attributed to increased opportunities to minimise the distances vehicles travel without cargo during continuous moves when a finer representation of the network is used. The underlying principle, although not explicitly stated in the paper, is that a reduction in the number of defined SCN lanes decreases the specificity of the network and thereby negatively impacts achievable optimality. This concept is fundamental to the LECP.

Like Basu *et al*. (2016) and Lim *et al.* (2012), Ergun *et al.* (2007) also noted a strong direct relationship between the number of lanes and the solution time. This alluded to the influence of the resolution of the SCN representation on the trade-off between the mathematical quality and implementation practicality of optimisation. This too represents a foundational idea of the LECP and strengthens its relevance to similar optimisation exercises.

The nodal clustering used by Ergun *et al.* (2007) used only geographic spatial considerations. Nodes were grouped according to "metropolitan areas and other geographic clustering of points". This alludes

to the tendency of modellers to control the complexity of an SCN by grouping locations based on physical proximity to each other or common positioning within existing geographical constructs, such as metropolitan areas, provinces, etc. As the algorithm was designed to reduce travelled distances as a proxy measure for cost, this approach was appropriate in the context of the investigation. Real transport costs, however, are not solely a function of distance.

Rozenfeld *et al.* (2009) remarked that this use of arbitrary administrative boundaries is generally "problematic". They continued to present a data-driven method for constructing boundaries for cities based on human population distribution. It is not possible to ascertain if the method used by Ergun *et al.* (2007) negatively impacted the results of the experiments, but it would be reasonable to concede that a fit-to-purpose mathematical approach using the problem instance's dataset to perform the clustering would have yielded better outcomes than the use of metropolitan areas.

Zhu and Guo (2014) agreed with the assertion made by Rozenfeld *et al.* (2009), stating that the use of predetermined geographic units, such as states, for defining flows causes significant information loss and introduces the risk of important patterns being undetected. This extended to extends to patterns in transport pricing. The problem statement of this research thesis is based on the principle described by Sheffi (2004) that nodal clustering in a transport procurement context should be done based on similarities of the costs associated with loading at or delivering to those nodes. It follows that the use of metropolitan boundaries, defined by governments without considering transport pricing, for defining nodes can result in the loss of important differentiation between lanes with materially different associated costs.

2.3 Geographic object clustering

Catini *et al.* (2015) recognized that economists had predominantly used government defined boundaries for geographic clustering and described administrative regions as economic "proxy clusters". The use of metropolitan areas for clustering by Ergun *et al.* (2007) discussed above is an example of this in a transport economics context. Dimária Silva *et al.* (2014) provided a further example of this approach in supply chain management by using municipality definitions to support the calculation of spatial concentration of different logistics activities in Brazil.

An alternative approach to predefined boundaries was described by Holmes and Lee (2008), which involved subdividing a country into a grid of equally sized cells. This technique was used by Maoh and Kanaroglou (2007) to support the definition of geographic clusters based on employment and population density.

Several techniques exist for grouping these geographic objects, whether defined by government boundaries or a grid system, into larger clusters. Kang *et al.* (2022) categorized them as attribute-based, regionalization-based and density-based clustering methods. The latter group of algorithms, however, group data points rather than geographic objects and is therefore discussed in section 2.4.

Attribute-based clustering of geographic areas is typically limited to seeking similarity with respect to characteristics other than the geographic position of these locations or areas. Jain and Dubes (1998) described data clustering as the grouping or classifying of objects by their measurements or relationships between other objects. Jain and Dubes (1998) also stated that clustering methods are widely used by scientists in many disciplines to group data points representing objects in a variety of scenarios using measures of proximity that do not include physical distance. General data clustering methods are therefore well-suited to attribute-based geographic clustering.

Data clustering is an important and frequently used tool in data mining in a wide variety of applications (Sakset and Pandya, 2016). They are therefore well-studied. Numerous techniques developed outside of a geographic analysis context have been leveraged for clustering geographic objects without the requirement to develop new algorithms. These include BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), developed by Zhang *et al.* (1996), as well as CURE (Clustering Using Representatives), developed by Guha *et. al* (1998). The self-organising map algorithm was specifically presented by Bação, *et al.* (2005) as an attribute-based method for clustering the districts of the Lisbon metropolitan area.

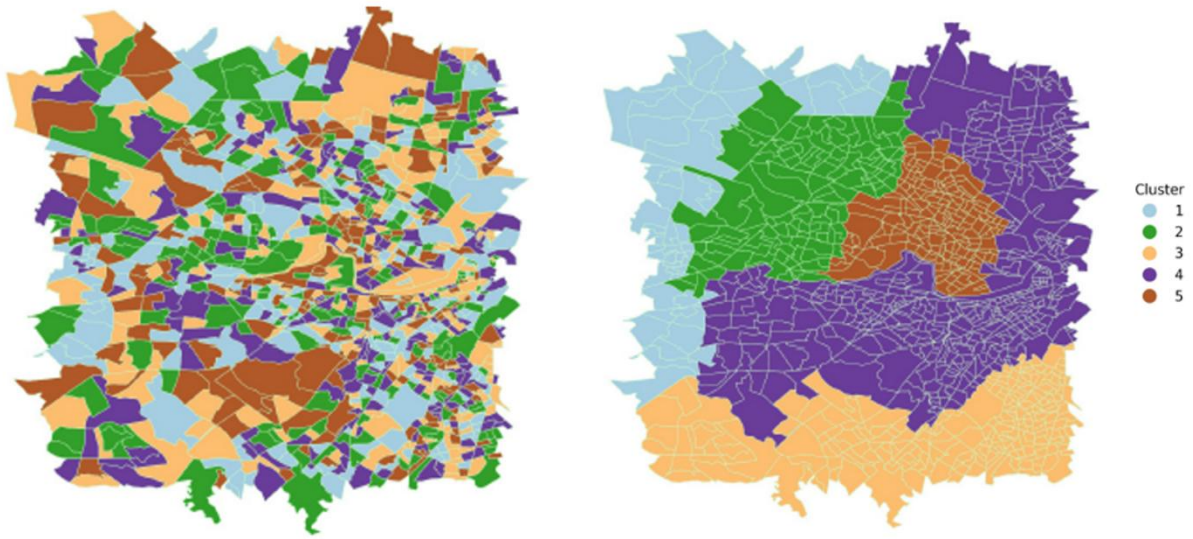
The geographic neutrality of attribute-based clustering is consistent with the LECP, which requires clusters to be based on pricing similarities rather than spatial position. However, some aspects to these approaches render them unsuitable as solutions to the LECP.

1. Spatial contiguity of clusters is not guaranteed (Kang *et al.,* 2022). While the LECP is geography-agnostic from an objective function perspective, it does include constraints that refer to physical position. The disjointed clusters that are often generated by attribute-based clustering methods would be considerably difficult to implement in practice as lane origins or destinations in a transportation management context (see section 1.2.1). Figure 9(a) shows an example set of generated clusters using one of these approaches. The use of any of the clusters, shown using different colours, as the start- or endpoint for a lane would arguably render the network more difficult to analyse and understand than the original network definition. The ongoing use of such lanes for transport rating purposes, which requires consensus across multiple parties, would be unwieldly. This discontinuity is sometimes addressed by including the latitude and longitude of the node as weighted attributes, although Kang *et al.* (2022) described several issues associated

with this strategy. Openshaw *et al.* (1998) also criticised the use of weighted measures as the various measures are not necessarily commensurable.

2. Defining transport rates to or from nodes within the network for multiple carriers as attributes of a geographic object would result in an exceedingly large number of attributes. It would not be uncommon for a node to be described by tens of thousands of attributes. The fact that not all carriers have rates for each lane, would also complicate the implementation of any method where rates are modelled as attributes of a node.
3. Incorporating the degree and manner of linkage of objects to the remainder of the SCN, such that the objective function could minimise the number of lanes, would be challenging to model using node attributes.

The spatial contiguity requirement is addressed by regionalization-based clustering methods. These are described by Kim *et al.* (2015) as methods that cluster small areas into a predefined number of regions. The areas assigned to a region should exhibit more similarity with others within the region than those without. These similarities are in terms of characteristics that are not necessarily directly related to the areas' positions, which is similar to attribute-based clustering. However, a key feature of these techniques is that distant regions with similar attributes are not grouped and remain distinct clusters. This is illustrated by Figure 9(b). These methods therefore solve the p-regions problem, described by Duque *et al.* (2009) as the agglomeration of *n* areas into *p* regions while optimising some function, with this function being a representation of intra-cluster homogeneity in the feature space (Assunção, *et al.*, 2006).



a) Attribute-based clustering b) Regionalization-based clustering

Figure 9: An illustration of the different clusters produced for the same dataset of geographic objects using different clustering methods (Kang *et al*., 2022).

Regionalization-based clustering methods include the Automated Zoning Procedure (AZP) presented by Openshaw (1977), Spatial 'K'luster Analysis by Tree Edge Removal (SKATER) technique developed by Assunção, *et al.* (2006), AUTOCLUST by Aldstadt (2010) and the enhanced Automated Zoning Procedure-Center Interchange (AZP-CI) presented by Kim *et al.* (2015).

The dual focus placed by regionalization-based clustering methods on area attributes and spatial contiguity enhances their applicability to the clustering of origins and destinations for lane elimination with simultaneous consideration of their associated transport rates. However, similar to attribute-based methods, the challenges of modelling and solving the LECP using these techniques with multitudinous transport rates as attributes would need to be overcome.

An additional consideration is that attribute- and regionalization-based methods group basic spatial units (BSU) that are areas already defined by their own existing boundaries. These methods fully tesselate the modelled landscape. Contiguity can therefore be defined as a binary value that indicates when areas share a boundary. This can be a fundamental element to the formulation of the problem and the algorithmic approach, as shown by Openshaw's (1977) use of binary contiguity matrix. This is not possible for clustering data points without a method of deriving these relationships, which would be fraught with high computational cost and complexity.

The consequence of clustering areas and not points is that the modeller is still required to choose the scale of the boundaries to use (e.g. the size of the grid partitions, wards, municipalities, districts, provinces, states, etc.) as the BSUs, which may not be optimally defined for subsequent identification of transportation rates. Multiple nodes contained within the shipper's rate card that, when used as origins or destinations of lanes, may have different prices associated for the same carrier and may be enclosed by a single BSU boundary and therefore may not be separated by these methods. The LECP must therefore be defined as a node-based clustering problem, as opposed to a geographic object clustering problem. It therefore should not be addressed by these attribute- or regionalization-based techniques without considerable modification.

2.4 Data point clustering

Regionalization-based methods make use of two-step solving approaches, weighted objective functions or adjacency constraints to achieve spatial contiguity (Assunção, *et al.*, 2006). Kang *et al.* (2022) cited the use of a geographic object's longitudes and latitudes nodes as weighted attributes for clustering to drive attribute-based cluster analysis methods toward solutions with enhanced spatial contiguity[1]. The

[1] Kang *et al.* (2022) indicated that there may still be a degree of spatial dependence of the input features that may result in some spatial contiguity be preserved when only non-spatial attributes are considered. This phenomenon

exclusive use of the longitude and latitude of a node when applying general data clustering methods provides an alternative approach that simplifies the clustering process while generally producing spatially continuous clusters. This is, essentially, a simplification achieved by a reduction of data features and allows for the application of *isotropic* algorithms that depend on distance as their main criteria for optimisation. The clustering of data points, representing SCN nodes, as opposed to BSUs, allows for easier implementation of existing data clustering algorithms to geographic clustering and broadens the scope of existing techniques that might be applied to these problems (Mai, 2017).

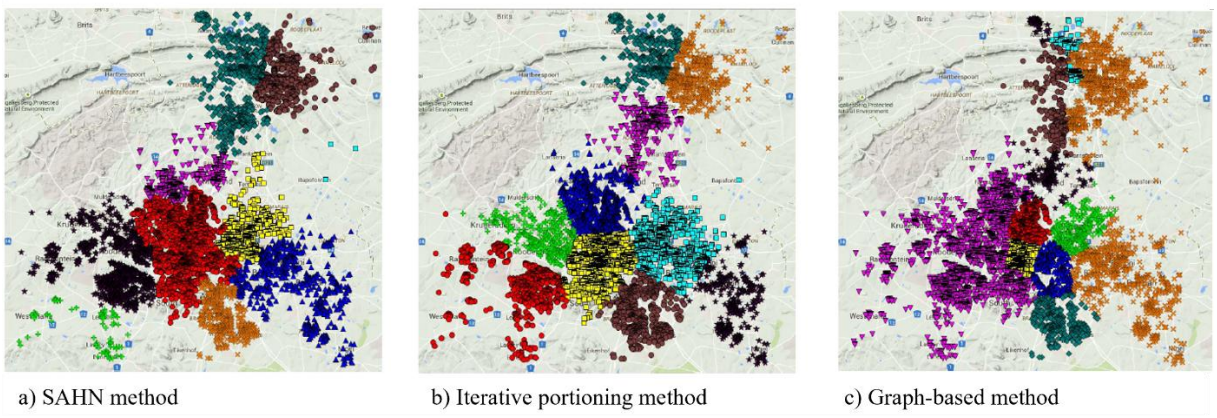
Density-based clustering methods are a subset of such general data clustering methods that have been applied to clustering geographic locations. They achieve spatial contiguity by primarily considering the geometric information associated with data points rather than their attributes (Kang *et al.*, 2022). These include the Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm developed by Ester *et al.* (1996), which was empirically tested using a dataset comprising of Californian landmark locations. Hu *et al.* (2015) used density-based methods to identify urban areas of interest within six different cities, while Liu *et al.* (2020) applied DBSCAN to define metro station areas in Shenzhen, China, according to attributes of places of interest. The Ordering Points To Identify the Clustering Structure (OPTICS) method was applied to television image clustering by Ankerst *et al.* (1999) and the DENsity-based CLUsteEring (DENCLUE) algorithm was tested using molecular biology by Hinneburg *et al.* (1998). Both authors referenced their techniques' applicability to geographic clustering. Catini *et al.* (2015) used a density-based method to define boundaries for economic and innovation zones by analysing the technological research output associated with nodes in the dataset. The technique generated innovation territories as "hubs" of nodes defined using polygons, which describes the method by which lane origins and destinations should be defined for the LECP.

These methods do not require the number of clusters to be predetermined and can generate clusters of arbitrary shape. Both these characteristics, in addition to the geographically contiguous clusters they produce, enhance these methods' applicability to the LECP.

Density-based methods were used by Bührmann (2016) as one of five general data grouping techniques applied to geographic locations that use only their coordinates. Other methods included sequential agglomerative hierarchical non-overlapping (SAHN) techniques, which are described by Kriege *et al.* (2014) as "classical clustering methods". Bührmann implemented a number of SAHN linkage methods that iteratively build and merge clusters. Iterative partitioning methods, which are distinguishable from SAHN methods by their requirement for a starting set of seed clusters, were also tested. These included the k-means method, which was described by Wu (2012) as one of the "oldest yet most widely used

would be expected for the LECP, as rates for transport to and from adjacent nodes should exhibit more similarity that those associated with distant nodes.

algorithms for clustering analysis". Graph-based, nearest neighbour and hybrid clustering methods were also evaluated by Bührmann. Examples of the results of these grouping methods are shown in Figure 10.



a) SAHN method b) Iterative portioning method c) Graph-based method

Figure 10: Example cluster sets generated by Bührmann (2016) using different clustering methods.

The performance of the methods was compared by Bührmann (2016) according to their sums of the squared residuals (SSR), means of the squared residuals (MSR), partition diameters (PD), averages distance measures (ADM), as well as their averages within cluster distance (AWCD). These evaluation measures quantify cluster homogeneity or cluster separation (Everitt *et al*., 2011). The former measures are maximised and the latter are minimised by the methods evaluated by Bührmann and other data clustering techniques like them. The inherent ambiguity associated with the question of what constitutes a "good" cluster continues to be problematic in the field of general data clustering (Jain, 2010). The LECP introduces a new measure of cluster quality that has not yet been explored in the literature.

Although these methods have a "long and rich history in a variety of scientific fields" (Jain, 2010) and have been applied to geographic clustering, no existing clustering technique found in the literature groups data points with the goal of minimising the number of inter-cluster links. Indeed, the length of the links may be considered by an algorithm to develop well-balanced, evenly spaced clusters, or to directly minimise a correlated cost such as energy consumption for data transmission between cluster heads of sensor networks (Fazackerley *et al.*, 2009). However, the *number* of links is not a factor in the logic of these algorithms. This "focus on single-location spatial events" was highlighted as a significant limitation of most cluster detection analysis by Tao and Thill (2016), who warned that the integrity of flow events is not preserved by general data clustering methods. Furthermore, the sole use of the longitudes and latitudes of the data points and the use of measures that only consider spatial relationships means that an alternative method to incorporate the economic attributes of the nodes and clusters would need to be developed to address the LECP.

2.5 Origin-destination flow clustering

Clustering methods that consider both the start- and endpoints of geographic movements, commonly referred to as origin-destination (OD) flows, have been developed to explicitly group lanes as the clustered entity. An OD flow was described by Fang *et al.* (2021) as a representation of an object's motion as a vector with spatial location.

A flow cluster groups the trajectory of an object from an origin to destination (which would be represented graphically as a line or arrow between two points in space) with other trajectories. It therefore fundamentally differs from a traditional geographic clustering that groups nodes (which would be represented as points or dots), that may or may not result in flow aggregation when used as lane starting or end positions. A transport lane, as defined for this research, is a type of OD flow and methods for clustering flows should reduce the number of lanes that describe an SCN. Most OD flow clustering methods in the literature were typically developed for aggregating trajectories that represent individual historical movements, such as taxi trips. The origins and destinations are usually captured using global positioning systems (GPS) and are unique per movement. In contrast, the lanes of an SCN are typically defined in a transportation management context using single static areas that encompass loading and offloading locations for all flows from or into these areas (Caplice, 2007). The datasets for which these flow clustering methods were developed therefore differ from those used as an input to transport procurement event design, as well other situations where the LECP may have relevance. Nonetheless, these methods may have applicability to the LECP and are therefore reviewed below.

Song *et al*. (2019) presented a method for defining arbitrability shaped flow clusters based on the spatial density of flows. Zhu and Guo (2014) also developed a clustering method that aimed to aggregate flows, using a spatial index to determine similar origins and destinations before clustering flows. Tao and Thill (2016) used state-of-the-art clustering methods to address OD flow clustering. Their method also addressed the issue of false positives for short-distance flows. This feature bears resemblance to the requirement of the LECP solution to provide different sets of clusters for lanes of differing distance categories.

Tao and Thill's method also requires manual tuning of parameters. Fang *et al.* (2021) addressed this limitation by developing an algorithm that clusters flows while simultaneously determining optimal parameter settings. One set of these parameters is the aggregation scale, which is a common setting for OD flow clustering algorithms and corresponds to distance ranges where dominant trajectory classes occur (Shu *et al*., 2020). The LECP also requires a distance band to be an optimised output of the algorithm as opposed to an input parameter.

While these algorithms were designed to simultaneously consider the pairing of origin and destination clusters and are therefore more applicable to the definition of transport lanes than other geographic or general data clustering methods, they share a common shortcoming that prevents their use for solving the LECP. These OD flow clustering methods, like the density-based geographic objects and the general data clustering methods reviewed in sections 2.3 and 2.4 respectively, only use spatial considerations to determine whether flows should be aggregated. They consider only two attributes of the origin and destination of a trajectory, namely the longitude and latitude. These methods are therefore well suited to aggregating large numbers of individual, unique movements in two-dimensional space and are appropriate for de-cluttering the representation of human, freight or vehicle trajectories. The LECP, however, requires simultaneous consideration of numerous transport rates from and to the origin and destination of each flow. The number of rates associated with each node can also vary significantly. While there are trade-offs that are considered by flow clustering techniques that are relevant to the LECP, such as the loss of spatial information (which corresponds with the loss of economic granularity for the LECP), their common focus on only two-dimensional spatial density recognition renders these methods unusable for clustering flows based on the transport costing attributes of nodes.

Xiang and Wu (2019), however, addressed the OD flow clustering problem with an additional measure of similarity. Their Tree-based and Optimum Cut-based Origin-Destination Flow Clustering (TOCOFC) algorithm considers not only spatial density of origins and destinations, but also how the set of trajectories change with time. This method considers the increasing similarity of flows over larger distances and the direction of the trajectory, both of which are important aspects of the LECP. Nonetheless, the additional dimension of time is not comparable with the modelling of transport rates for a diverse pool of carriers, many of which do not service all flows.

The TOCOFC shares additional pitfalls with other OD flow clustering techniques in its application to the LECP.

First, OD flow clustering methods have only one objective, that being the minimization of trajectory heterogeneity. Loss of information is minimised because of cluster homogeneity, not due to the algorithm's simultaneous consideration of the effect of the number of clusters to which it is aggregating the trajectories. The number of flow clusters is typically an optimisation input parameter, not a variable that is available to the algorithm for modification. As such, these techniques do not identify opportunities to cluster trajectories beyond the specified minimum number of clusters, nor do they decline the option to cluster flows when the loss of information is deemed too great. These are key aspects of the LECP.

Second, the output of an OD flow clustering algorithm is a set of clustered trajectories for which the orientations are a key input to the measurement of homogeneity. They are not a set of origin and

destinations for defining flows that can be defined regardless of direction. Flows stemming from the same geographic areas with significantly different directions are unlikely to be grouped and clustered flows can have overlapping starting or ending points. This phenomenon is illustrated by Figure 11. Commercial discussions in a transport management setting, however, require a set of non-overlapping origin and destination clusters that can be used to describe lanes regardless of direction[2]. The LECP must therefore be solved by clustering geographic locations with their associated flows in mind, not cluster the flows themselves.

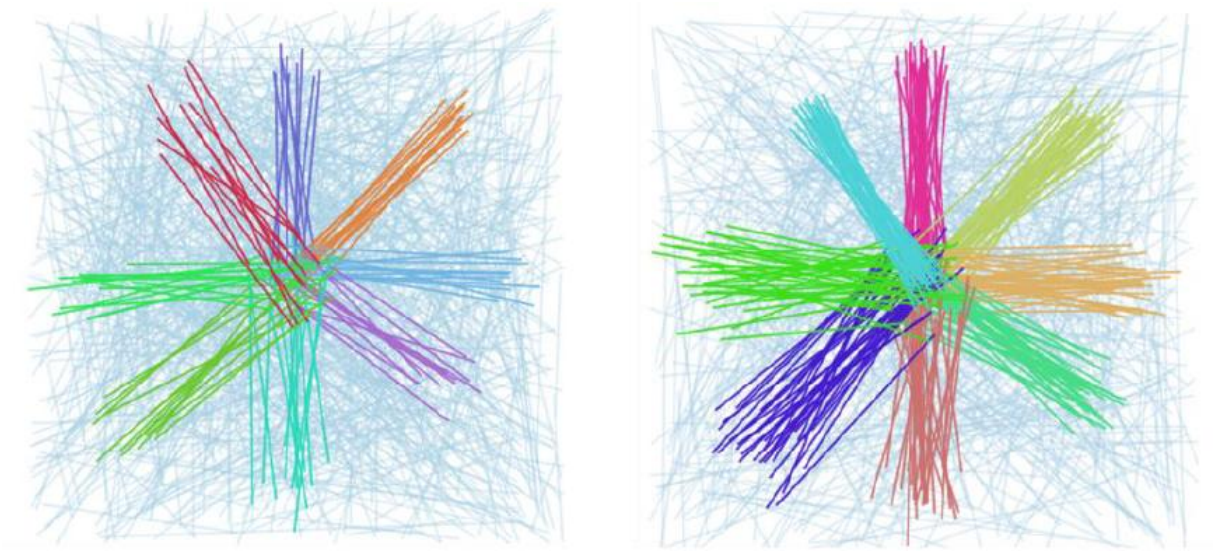


Figure 11: Example OD flow clusters (Fang *et al*., 2021).

Almannaa *et al*. (2020) addressed two important limitations of other location and OD flow clustering methods for solving the LECP. First, they proposed a novel clustering method that determined the optimal number of clusters. The algorithm is therefore multi-objective and its simultaneous minimization of the number of clusters while seeking cluster homogeneity bears resemblance to the requirement to minimise inter-cluster links to solve the LECP. Second, the method allows for simultaneous consideration of attributes of the start- and endpoints of trips additional to their geographic position, such as the hour, day or month on which the trip occurred. This consideration of additional data labels is congruent with the need to consider multiple inbound and outbound transport rates per location when addressing the LECP. However, this method was designed to cluster transportation data points, as opposed to clustering the flows between the clusters.

[2] Unless the shipper and carriers are receptive to a significant paradigm shift in terms of how lanes, rates and rate cards are designed.

2.6 Clustering according to transportation economics

Geographic areas are commonly used as a feature when transport pricing is compiled and analysed. Simchi *et al*. (2013) stated that, for North American full truckload transport, carriers typically define rates using zones. They also stated that zones are usually defined as states, although some large states are divided into two zones[3]. Collins and Quinlan (2010) stated that an origin or destination may also be defined as any geographic area, depending on the shippers' requirements. However, no solution is referenced to assistance shippers in determining the areas that best meet their needs. This presents the fundamental gap in the literature that this research aims to address.

The analysis by Gonzalez-Feliu *et al.* (2020) of freight rates and attributes of urban zones provided a real-world example of this practice. This study, as well as many like it, used pre-defined zones, like those described by Simchi *et al*., as an input. Pricing did not influence the definition of the zones.

Collins and Quinlan (2010) used three- and five-digit zip codes, as well as states, to quantify the economies of lane aggregation in the USA. This approach was also used by Acocella and Caplice (2022) for their analysis of the effect of lane aggregation on the emergence of ghost lanes. Transport rates were not an input to the structure of the resultant origins and destinations used for these exercises.

A study has been conducted in which locations were clustered while considering the costs of their associated inbound and outbound flows. Acocella *et al.* (2020) required lane definitions for an analysis of shipper-carrier relationships during periods of heightened transport demand. Instead of reverting to the use of administrative boundaries, "key economic regions" were used as load origins and destinations.

This concept of an economic region is applicable to the LECP's requirement to cluster SCN nodes from a transport pricing perspective. The reasoning for the clustering, that being for aggregation of data for subsequent analysis, also represents a use case for solving the LECP. Regrettably, the methods by which these regions were developed were not outlined in literature. Furthermore, it is unclear whether these economic regions were developed with the complexity of underlying SCNs in mind.

It was indicated by Acocella *et al.* that actual cost data, which was dependent on the offer and acceptance of loads to and by carriers, were used to define the regions. Furthermore, this data encompassed multiple shippers' operations. Solving the LECP must consider all carrier rates for each lane and generate

[3] Simchi *et* al. (2013) described this principle in the context of a network design optimization exercise that spanned large distances across North America. They also stated that transport rates are typically defined on a cost per kilometre basis. This mitigates the effect of using large zones, such as states, to define long distance lanes. This is, of course, not always the case when managing transport rates.

economic regions that simplify a shipper's specific unique SCN. It is also unclear whether Acocella *et al.*'s regions were defined based only on transport pricing similarities or if they also used the degree of connectedness of the nodes to the rest of the network and link elimination opportunities for lane minimization as direct considerations.

The method used by Acocella *et al.* to generate transport zones is arguably the most suitable technique for solving the LECP. However, it is the lack of a multi-objective approach to define economic regions to minimise lanes that is this, and the other review methods', most fundamental limitation with respect to this research's problem statement.

Acocella *et al.*'s work represents an example of "more sophisticated analysis and forecasting of transportation rates and flows" described by Caplice (2021). He continues to argue that these areas of research require further attention and that there is "a great need to be able to better segment networks" for transport contracting.

2.7 Concluding remarks

Chapter 2 positioned the LECP in the existing body of knowledge. It reviewed literature that supports the notion of lane definition as a key input to transportation optimization activities. It also showed that the concept of lane elimination, as a cluster analysis problem, provides several nuances that are not addressed by existing methods.

A new problem formulation and solution approach is therefore required to support practitioners in the transport management field by clustering supply chain nodes. This should be done by simultaneously solving for the objectives of SCN simplicity and alignment to existing carrier rate levers and, therefore, a multi-objective clustering approach to the LECP is needed. Chapter 3 explores existing multi-objective optimization theory and applications that should be considered during the design of such an approach to the LECP.

Chapter 3

Literature review – multi-objective optimisation

In this chapter, theory and previous works that are relevant to the formulation and solving of the LECP using MOEAs is presented. This literature review is not intended to serve as a critical analysis and few references to the LECP are made in this chapter. However, as the theory discussed in this chapter forms the basis of much of this thesis' research approach, subsequent chapters frequently reference the concepts discussed in this literature review.

3.1 Basic multi-objective optimisation concepts

3.1.1 Multi-objective optimisation problems

Many real-world problems, such as the LECP, require simultaneous optimisation of multiple criteria. Optimisation problems with this requirement are often referred to as multi-objective optimisation problems (MOPs). Saini *et al.* (2021) presented a general formulation of an MOP as follows:

Maximise $F(x) = (f_1(x),\ f_2(x), \dots, f_m(x))^T$ such that $x \in \Omega$ (3.1)

where x represents the decision variable within the decision space Ω, m indicates the number of objective functions for optimisation, $f_i(x)$ is the *ith* objective function, $i = 1, 2, \dots, m$, and $f_i: \Omega \rightarrow R^m$, where R^m is the objective space.

Saini *et al.* (2021) also indicated that an MOP may involve the minimization of $F(x)$ or a combination of maximisation and minimization of the problem's individual objective functions. Many similar definitions of MOPs, such as the formulation provided by Van Veldhuizen (1999), also explicitly state the maximisation or minimization is subject to $g_k(x) \leq 0, i = 1, 2, \dots, n$, where n is the number of problem constraints.

The solution to an optimisation problem often comprises multiple decision variable values. The decision x is therefore commonly represented as a vector. MOPs, however, differ from single-objective problems in that the target for maximisation or minimization is also represented as a vector. Two multi-dimensional Euclidean spaces are therefore relevant to an MOP, namely the decision and objective spaces (Gunantara, 2018). Each point in the decision space has an associated point in the objective space as shown in Figure 12. The activity of defining the relationship between these corresponding points is often referred to as mapping the problem from the decision to the objective space.

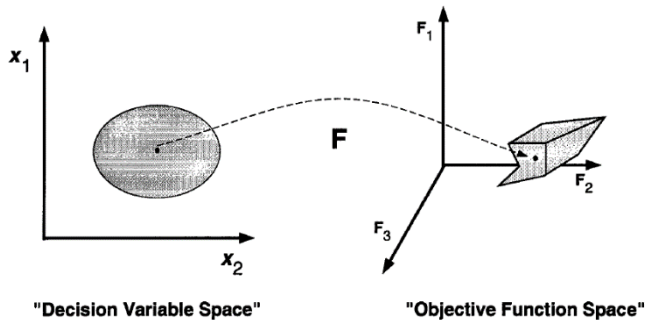


Figure 12: An illustration of the mapping of an example solution in two-dimensional decision space to a two-dimensional objective space (Van Veldhuizen, 1999).

The vector of objective function values that represents the most desirable achievable value for each function is referred to as the *ideal vector* (Deb *et al.*, 2006). Coello Coello *et al.* (2007) defined the ideal vector of an MOP as follows:

Let

$$x^{0(i)} = [x_1^{0(i)}, x_1^{0(i)}, \dots, x_n^{0(i)}]^T \tag{3.2}$$

be a vector of decision variables that optimises (either maximises or minimises) the *ith* objective function $f_{i(x)}$.

This can also be expressed as the vector $x^{0(i)} \in \Omega$ such that

$$f_i\left(x^{0(i)}\right) = f_i(x) \tag{3.3}$$

Then the vector

$$f^0 = [f_1^0, f_2^0, \dots, f_k^0]^T \tag{3.4}$$

where f_i^0 indicates the optimum of the *ith* function, is the ideal vector for an MOP. The point in R^n that determined this vector is the ideal solution and is the ideal vector. The terms *ideal* and *utopian* vector are often used interchangeably in literature.

The *worst point* (Wang, *et al.*, 2017) corresponds to the coordinates in objective space that represents the worst possible outcome for each objective of an MOP when evaluated individually. The objective function vector that corresponds to the coordinates on the trade-off curve representing the worst possible outcome for each objective is known as the *nadir point* (Audet, *et al.*, 2020). The position of the worst and nadir in an example two-dimensional objective space is illustrated by Figure 13.

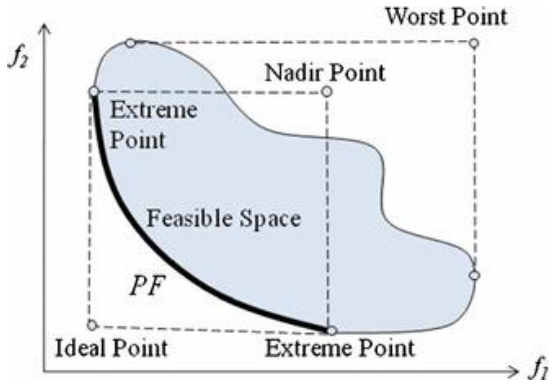


Figure 13: An illustration of how the ideal and nadir points are positioned in objective space for a bi-objective minimization MOP (Wang, *et al*., 2017).

Fleming *et al.* (2005) described two types of relationships that may exist between objective functions of MOPs. The interaction between two objective functions is considered *harmonious* if a change in the decision space usually has a simultaneous desirable or undesirable effect on both function values. A relationship in which the improvement in one objective's performance is typically achieved with a deterioration of the other is termed a *conflicting* relationship. The lane elimination and cost penalty objectives of the LECP can therefore be described as conflicting objectives.

Van Veldhuizen (1999) stated that the objectives of an MOP almost always conflict and, therefore, a solution with the best outcome in terms of all measures of utility seldom exist. Furthermore, these objectives are often incommensurable (Cui *et al.*, 2017). This means that the measures of the quality of a solution cannot be reduced to a common unit of measure (Magee, 2011). This complicates the quantitative comparison of the conflicting objectives of an MOP. For example, the trade-off between the commensurable objectives of number of warehouses and transport cost could possibly be analysed based on the single measure of cost, allowing for the existence of an optimal solution to the problem. However, an MOP may involve a trade-off between the incommensurable measures of cost and customer satisfaction, which are quantified according to different units of measurement and can therefore not be combined to a single function for optimisation.

It is therefore often not possible to find a unique optimal solution for an MOP. Focus, in this scenario, is rather given by the designer of the solution approach to incorporate the preferences of the DM into the optimisation process. An alternative approach is to generate a set of different compromise solutions with vector components that represent the trade-offs of the MOP in objective space. The DM is thereby provided the opportunity to select the solution that represents the best trade-off according to their preferences (Van Veldhuizen, 1999).

3.1.2 Pareto concepts

The concepts of Pareto dominance and optimality are commonly used to address the requirement to generate good compromise MOP solutions for a DM. A solution to an MOP is said to dominate another if it performs better with respect to all objectives of the MOP. Sahoo *et al.* (2022) provided the following definition of Pareto dominance:

Vector z^1 dominates vector z^2, denoted by $z^1 \prec_{pareto} z^2$, if $\forall i \in 1, 2, \dots, k: z_i^1 \leq z_i^2$ and $\exists i \in 1, 2, \dots, k: z_i^1 \leq z_i^2$.

Figure 14 illustrates how a solution to an MOP may or may not dominate another. Suppose that the MOP is defined such that both the objectives f_1 and f_2 should be minimised. Solutions *A*, *B* and *C* are positioned according to their performances with respect to the two objective functions. Solutions *A* and *B* are superior to *C* in terms of both objectives and can therefore be said to *dominate* solution *C*. However, while solution *A* is better than *B* with respect to objective f_1, this solution is inferior in terms of f_2. Neither of these solutions dominate the other.

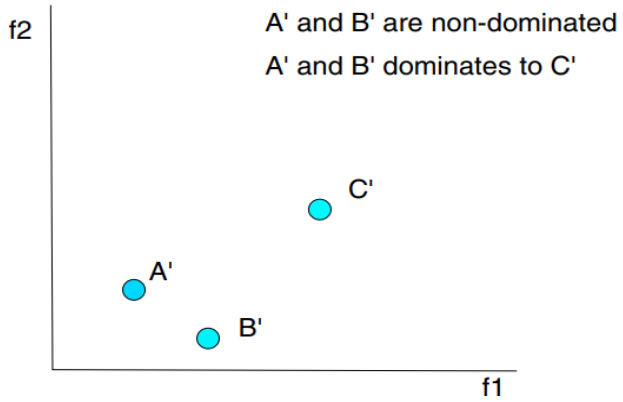


Figure 14: Examples of dominated and nondominated solutions of a minimization MOP represented graphically in objective space with respect to objectives f_1 and f_2 (Durillo, *et al*., 2011).

It follows that a solution to an MOP may not be dominated by any other known or unknown feasible solution. This is referred to as a *nondominated* or *Pareto optimal* solution. The terms *non-inferior* and *efficient* are also frequently used to describe these solutions (Coello Coello *et al.,* 2007). A solution is therefore only considered Pareto optimal if there is no possibly of there existing another feasible solution to the MOP that exhibits superior performance for one of the objectives without an associated performance deterioration with respect to any other objectives. It is therefore not possible to modify a Pareto optimal solution in the decision space and simultaneously improve its position in all dimensions of the objective space.

Pareto optimality can be formally defined as follows (Sahoo *et al.*, 2022):

A solution $x^* \in X$ is Pareto optimal if there does not exist another solution x $\in X$ such that $f(x) \prec f(x^*)$.

Coello Coello *et al.* (2007) differentiated between weak and strict Pareto optimality:

A point $x^* \in \Omega$ is weakly Pareto optimal if there is no $x \in \Omega$ such that $f_i(x) < f_i(x^*)$, for $i = 1, 2, \ldots, k$.

A point $x^* \in \Omega$ is strictly Pareto optimal if there is no $x \in \Omega, x \neq x^*$ such that $f_i(x) \leq f_i(x^*)$, for $i = 1, 2, \ldots, k$.

The Pareto optimal set is the subset of MOP solutions that satisfy the criteria for Pareto optimality. Figure 15 shows where Pareto optimal points are positioned relative to dominated solutions in an example two-dimensional objective space. Sahoo *et al.* (2022) provided the following definition of a Pareto optimal set, P^*:

$$P^* = (x \in X \mid \nexists y \in X : f(y) \preccurlyeq f(x)) \tag{3.5}$$

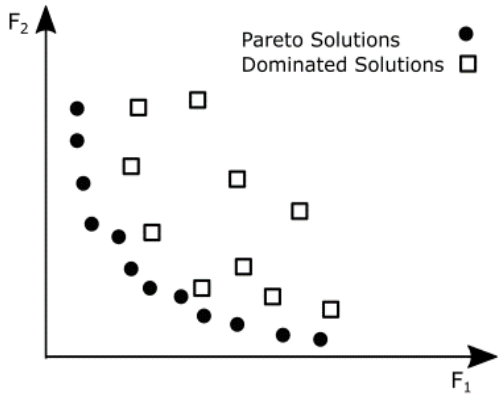


Figure 15: An example Pareto optimal set, shown by black-shaded data points, for a bi-objective minimization MOP (Rahnamayan *et al*., 2020).

Graphical representations of this subset of solutions are often accompanied by a curve fitted to the data points in objective space. This line represents the trade-off curve relating to the conflicting objectives of the MOP. This is also referred to as the Pareto front, curve or surface (Carmia and Dell'Olmo, 2008). Figure 16 shows the Pareto curve of an example two-dimensional objective space.

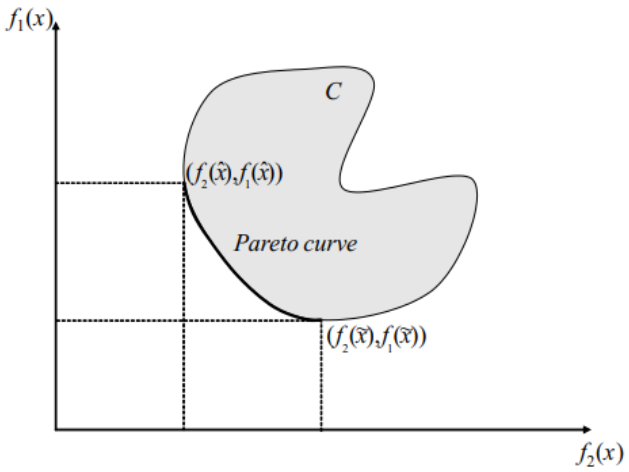


Figure 16: An example of a Pareto front for a bi-objective MOP (Carmia and Dell'Olmo, 2008).

This research adopted the symbolic notation presented by Coello Coello *et al.* (2007) for representing the Pareto front. The entire set of Pareto optimal solutions is represented by P^* and PF^*. The former represents the decision variable vectors of all Pareto optimal solutions, while the latter denotes these solutions in objective space.

P^* and PF^* represent the theoretical Pareto front of an MOP. However, the computational domain is limited in that the full exploration of a continuous feasible region is often not possible. Coello Coello *et al.* (2007) therefore also uses P_{true} and PF_{true} to denote the computational true Pareto front, which is a theoretically attainable set of solutions for computers.

The Pareto front found by a solution approach, either during the execution of the algorithm or upon completion, may not be equal to this theoretical solution set. It is commonly accepted as an assumption that the output of an MOP solving technique represents only an approximation of the true Pareto front. Furthermore, the true fronts, P_{true} and PF_{true}, of many MOPs are unknown (Kumar and Rockett, 2002). P_{known} and PF_{known} are used by Coello Coello *et al.* (2007) to denote Pareto front approximations by an algorithm in the decision and solution spaces respectively. $P_{known}(t)$ and $PF_{known}(t)$ are also used to represent the approximations found at time *t* specifically.

Ranking solutions in terms of Pareto-dominance is frequently used for solution selection during optimisation when evolutionary methods are utilised. However, the PF_{true} for certain MOPs cannot be found using these relationships due to lack of selection pressure when the number of objectives is large (Liu *et al.*, 2007). A relaxation of the Pareto-dominance relationship has sometimes been successfully used to fine-rank solutions and thereby overcome selection pressure difficulties often experienced when using Pareto-ranking (Aguirre and Tanaka, 2009). The relaxed relationship is commonly referred to as ε-dominance, or *epsilon dominance*. Figure 17 illustrates the position of the ε-dominance region in relation to the dominance region of a two-dimensional objective space.

Junfeng *et al.* (2015) presented the following definition of ε-dominance:

A give objective vector $x = (x_1, x_2, \dots, x_k)$ is said to be ε-dominate another objective vector $y = (y_1, y_2, \dots, y_k)$ if and only if

$$(1 - \varepsilon)x_i \leq y_i \qquad \forall i \in 1, 2, \dots, k \tag{3.6}$$

,which denotes $x \prec_\varepsilon y$.

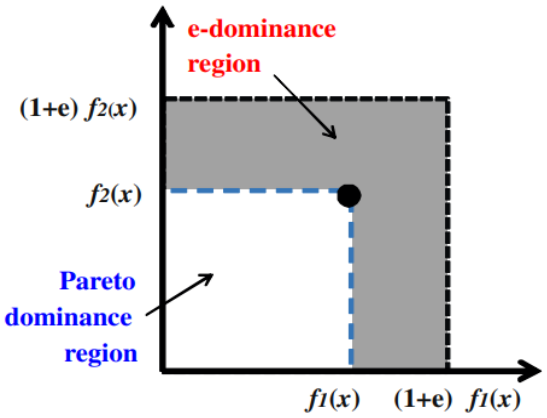


Figure 17: The Pareto dominance and ε-dominance regions of the solution space of a bi-objective MOP relative to a single solution (Aguirre and Tanaka, 2009).

Pareto optimality, the Pareto optimal set and a Pareto front can also be defined in terms of ε-dominance (Coello Coello *et al.*, 2007).

A solution $x \in \Omega$ is said to be Pareto epsilon optimal with respect to Ω if and only if there is no $x' \in \Omega$ for which $v = F(x') = (f_1(x'), \dots, f_k(x'))$ ε-dominates $u = F(x) = (f_1(x), \dots, f_k(x))$.

The phrase *Pareto epsilon optimal* is taken to mean with respect to the entire decision variable space unless otherwise specified (Coello Coello *et al.*, 2007).

For a given MOP, $F(x)$, the Pareto epsilon optimal set, P_ε^*, is defined as:

$$P_\varepsilon^* := \{x \in \Omega \mid \neg\exists\, x' \in \Omega\, F(x') \preccurlyeq F(x)\} \tag{3.7}$$

For a given MOP, $F(x)$, and Pareto epsilon set P_ε^*, the Pareto optimal Front (PF_ε^*) is defined as:

$$PF_\varepsilon^* := \{u = F(x) = (f_1(x), \dots, f_k(x)) \mid x \in P_\varepsilon^*\} \tag{3.8}$$

3.1.3 MOP techniques

Several methods have been developed and applied to optimisation problems with multiple objectives. Cohon and Marks (1975) originally classified techniques according to the stage of the optimisation process during which the preferences of the DM are articulated. According to this classification scheme, an MOP technique can involve either:

- Articulation of the DM's preferences prior to the execution of the method. This is known as an *a priori* approach (Emmerich and Deutz, 2018).
- Articulation of the preferences during the execution of the method. This is referred to as an *interactive* or *progressive* approach (Xin *et al.*, 2018).
- Articulation after the execution of the method. This is known as an *a posteriori* approach (Emmerich and Deutz, 2018).

Although *a posteriori* approach was pre-selected as part of the scope of this research, *a priori* and interactive methods are reviewed in this section to provide context and justification in terms of where the developed evolutionary approach is positioned in the MOP technique hierarchy.

***A priori* techniques**

A priori optimisation involves the reduction of all the objectives of an MOP into a single function that can be optimised using single-objective optimisation methods (Jain and Parashar, 2022). This reduction is often referred to as *aggregation* or *scalarization* of objectives (Braun, 2018).

The weighted sum method is the most common approach to solving an MOP (Chircop and Zammit-Mangion, 2012). Its scalarization strategy involves the construction of a weighted sum of the objective function values associated with a solution (Coello Coello and Christiansen, 2000).

The weighted sum function of an MOP with i objectives is defined as follows:

$$\sum_{i=1}^{k} w_i f_i(\underline{x}) \tag{3.9}$$

This value is then minimised or maximised, depending on the nature of the MOP. This solving of the scalarized MOP is as simple as the corresponding single-objective optimisation problem (Bérubé *et al.*, 2009).

Figure 18 graphically shows a scalarized MOP and how the weights selected by the DM affect the optimal solution found by the weighted sum method. The relative weights determine the gradient of line L. This line's angle of inclination directly affects the lowest possible position of intersection with the feasible region boundary Λ. This point represents the optimal solution to the scalarized problem.

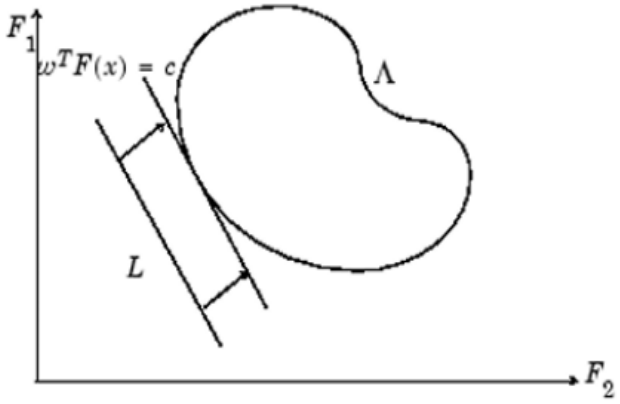


Figure 18: A graphical illustration of the determination of the optimal solution of a bi-objective MOP using the weight sum method (Coello Coello and Christiansen, 2000).

Das and Dennis (1997) presented a number of shortcomings of this approach. These include the failure of the method to generate solutions along nonconvex parts of the Pareto curve.

The ε-constraint method also provides a simple method of converting an MOP to a single-objective optimisation problem. By transforming all but one of the objectives into constraints, the problem can be solved as a single-objective optimisation problem (Chircop and Zammit-Mangion, 2013). This process, however, as noted by Konak *et al.* (2006), can be somewhat arbitrary.

Chircop and Zammit-Mangion (2013) provided the following general formulation of a bi-objective minimization MOP converted according to the ε -constraint method:

$$f_i(\vec{x}) \tag{3.10}$$

subject to: $\vec{x} \in \Omega$,

$f_j(\vec{x}) \leq \varepsilon_j, \quad i,j = 1,2\ i \neq j$

Goal programming represents a further *a priori* approach. It requires the assignment of targets for each of the MOP objectives. These targets should reflect the goals that the DM wishes to achieve with respect to each objective (Coello Coello *et al.*, 2007). The combined deviation of the solution's achievement from each of these goals is then minimised (Hussain and Kim, 2020). This deviation is defined by Ignizio (1981) as follows:

$$\sum_{i=1}^{k} |f_i(x) - T_i| \quad \text{subject to } x \in \Omega \tag{3.11}$$

where T_i is the target value for objective *i*, $f_i(x)$ is the value of objective function *i* for the solution *x*, and *k* is the number of objectives of the MOP.

The global criterion method is similar to goal programming, but involves the minimization of the relative distance of the solution to the ideal vector in objective space (Umarusman, 2019). This method is illustrated by Figure 19. The minimized distance is defined according to the following function:

$$f(x) = \left(\sum_{i=1}^{k}\left[\frac{f_i(x^*) - f_i(x)}{f_i(x^*)}\right]^p\right)^{\frac{1}{p}} \quad (3.12)$$

where $f_i(x^*)$ is the ideal vector coordinate value that corresponds to objective i, $f_i(x)$ is the value of objective function i for the current solution x, k is the number of objectives of the MOP, and p ($1 \leq p \leq \infty$) is a parameter that reflects the importance of the solution's deviations from the ideal vector.

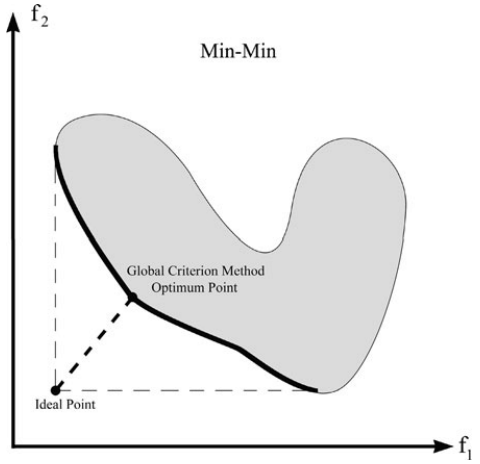


Figure 19: A graphical illustration of the determination of the optimal solution of a bi-objective MOP using the global criterion method (Karamooz *et al*., 2011).

Other popular MOP techniques include min-max optimisation, presented by Jutler (1967) and Solich (1969), the lexographic method described by Sarma *et al*. (1993), and the surrogate trade-off, proposed by Haimes and Hall (1974).

Interactive techniques

The interactive MOP methods presented in literature are diverse. Xin *et al.* (2018) provided a taxonomy of these techniques according to four design factors. The *interaction pattern* refers to whether the DM provides input after a complete run as part of an iterative process, or during the run. The *preference incorporation* describes the manner in which the DM articulates preferences. The *preference model* refers to how the articulated preferences are incorporated into the algorithm. The *search engine* describes the method by which the algorithm generates solutions to the MOP using the incorporated preference information.

Xin *et al.* (2018) also discussed a number of state-of-the-art interactive methods that could be categorised according to the proposed taxonomy.

Reference point-based methods involve the scalarization of the MOP using a reference point and the repeated resetting of this point by the DM until a satisfactory Pareto front is produced (Deb *et al.*, 2006). The generalised, classical reference point-based approach to MOP is formulated by Deb *et al.* (2006), as follows:

$$max_{i=1}^{M}\ [w_i(f_i(x) - \underline{z}_i)] \qquad \text{subject to } x \in \Omega \qquad (3.13)$$

where w_i, $f_i(x)$ and $\underline{z}_i$ are the *i*-th components of the chosen weight vector, current solution *x*'s objective vector and reference point respectively.

The reference point-based approach is shown in Figure 20. According to definition (3.13), the length of z_A is the value to be minimised. The point $\underline{z}$ is repositioned by the DM after review of the Pareto front after each run of the algorithm.

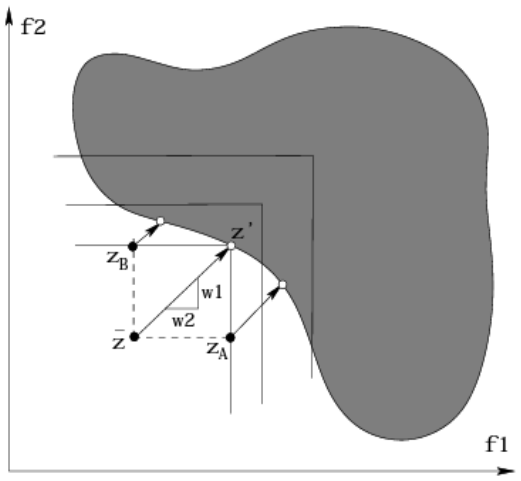


Figure 20: The classical reference point approach to solving a bi-objective MOP (Deb *et al*., 2006).

Specific interactive reference point-based methods include Fonseca and Fleming's (1993) modified ranking scheme and the R-NSGA-II method, proposed by Deb *et al.* (2006), which allows for progressive articulation of preferences during the execution of the NSGA-II MOEA.

Methods based on comparisons of objective functions include weights-based methods, such as Cvetković and Parmee's (2002) method of converting qualitative preferences into objective function weights, and trade-off methods, such as the GRIST method proposed by Yang (1999).

Interactive methods based on the comparison of solutions include pairwise comparison techniques, during which the DM is periodically prompted to select one of two generated solutions during the execution of the algorithm (Xin *et al.*, 2018). Comparison can also be provided by classification. This

is used by the dominance-based rough set approach evolutionary multi-objective algorithm (DRSA-EMO), which requires the DM to rate solutions qualitatively (Greco *et al.*, 2010).

A posteriori techniques

A posteriori methods of addressing MOPs generate a set of solutions for selection by the DM after the run of the algorithm. Miettinen (1998) presented the methods by which some *a priori* techniques can be iteratively executed in such a way that this Pareto front approximation is produced for *a posteriori* selection.

While the *a priori* application of the weighted sum function generates a single solution, it can be used to generate a set of solutions along the Pareto front by repeatedly and systematically optimising according to the aggregation function using a variations of weighting coefficients (Marler and Arora, 2010).

Similarly, the ε-constraint method can also be executed multiple times to produce a set of solutions that approximates of the Pareto optimal set. As per the corresponding *a priori* method, a single objective is optimised after the remainder of the objectives are converted to constraints. However, by repeating this process using varying bounds, solutions are generated that lie on different points of the Pareto front (Miettinen, 1998).

Figure 21 demonstrates the ε-constraint method as an *a posteriori* technique. When minimising objective z_1, the extent to which the z_2 function value is constrained influences the optimal solution to the transformed single-objective problem. By setting the z_2 upper boundary to ε_4, the feasible region is not affected and the point in the solution space that achieves the lowest possible z_1, shown as point z^4, is found as the optimal solution. However, as z_2's boundary is lowered, the feasible region is transformed and minimising z_1 for these scenarios generates alternative solutions that lie upon the Pareto front. Collectively these solutions provide the DM with a range of options for *a posteriori* selection according to their preferences.

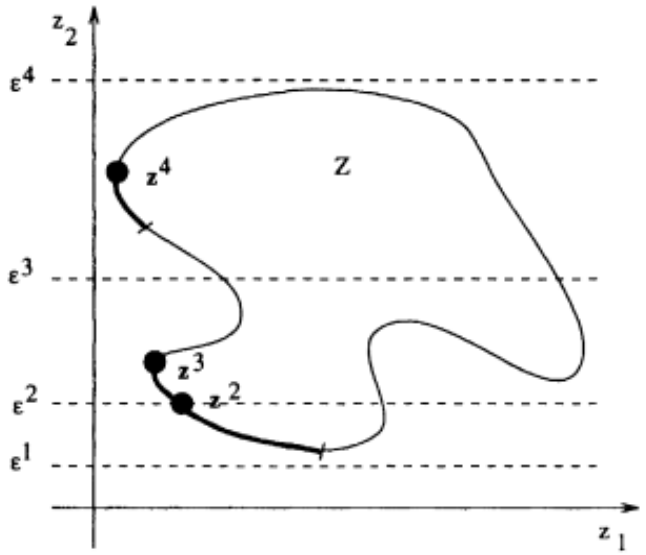


Figure 21: The use of different upper bounds to generate a Pareto optimal set of solutions of an MOP (Miettinen, 1998).

Evolutionary algorithms are widely used to generate the *PF* approximation of an MOP for an *a posteriori* process, which is required for addressing the LECP as scoped for this thesis. These, however, are reviewed in section 3.2. When applied to MOPs, these algorithms are referred to as multi-objective genetic algorithms (MOEAs). Figure 22 shows where these are positioned in the hierarchy of MOP techniques presented by Bakhsh (2013).

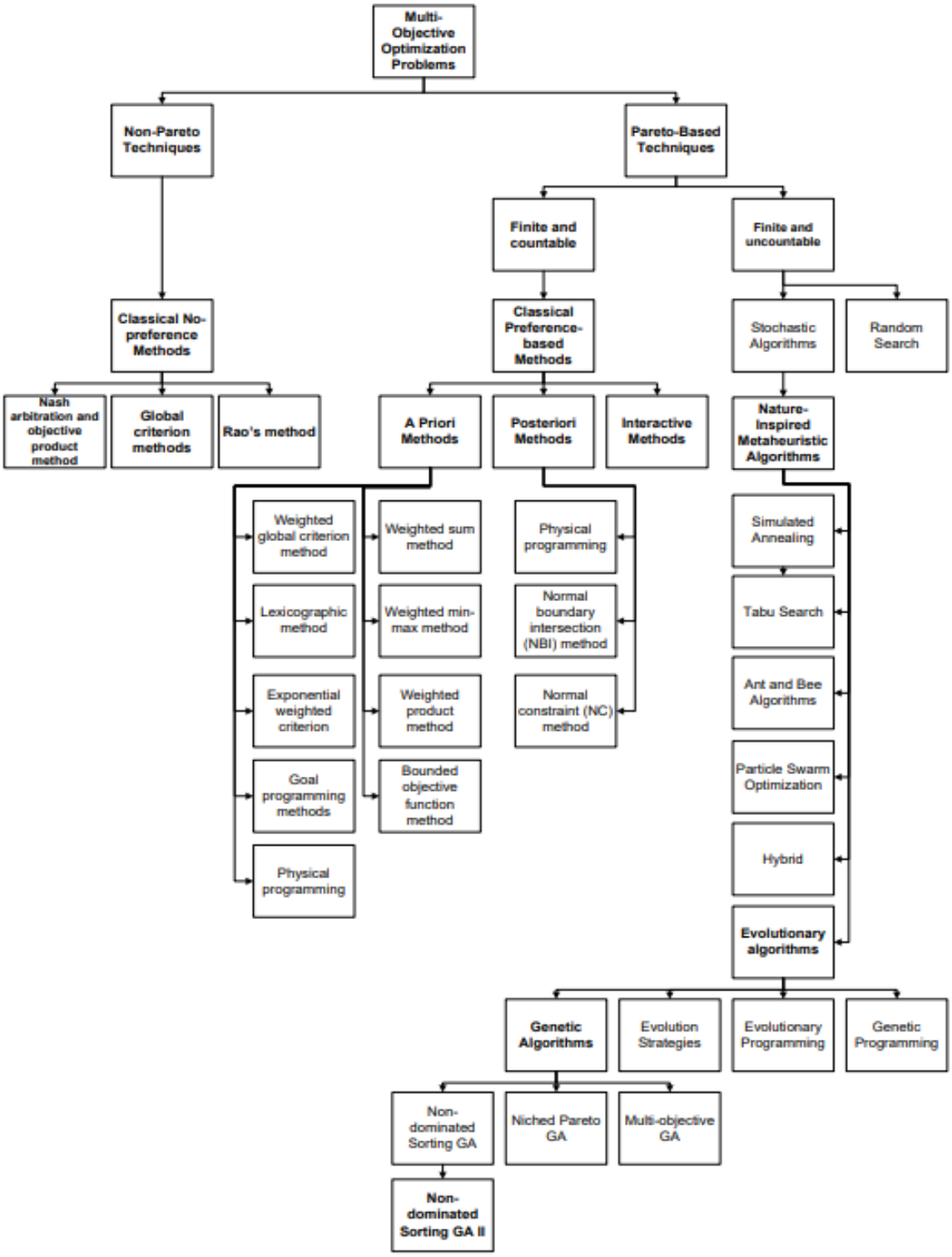


Figure 22: A taxonomy of MOP techniques (Bakhsh, 2013).

3.1.4 Quality measures

Unlike single-objective optimisation problems, or scalarized MOPs, that require only a single solution, the output of an *a posteriori* MOP solving process consists of multiple data points, namely the approximation of the Pareto optimal front *PF*. Quantification of the quality of individual solutions with respect to each objective is therefore insufficient for determining and comparing the effectiveness of

Pareto-based MOP solution approaches in generating options for a DM. The quality of the entire Pareto front must be evaluated. This measurement, however, is non-trivial (Audet, *et al.*, 2020). Several metrics have been developed and used for such measurement (Laszcyk and Myszkowski, 2019).

Selected quality metrics should reflect the goals of an MOP optimisation. Zitzer *et al.* (2000), however, described useful generic optimisation goals of an MOP as follows:

1. The generated Pareto front PF should be as close as possible to the true Pareto front PF_{true}. This is typically described as *convergence*.
2. The nondominated solutions should be well distributed in objective space. A uniform distribution is typically the most desirable and, as such, this goal is often referred to as *uniformity*.
3. The extent of, or range of values covered by, the PF approximation should be maximised. This is usually described as *spread*.

Uniformity and spread are often collectively referred to as *diversity* (Chand and Wagner, 2015).

Figure 23 illustrates the first and third of these goals by showing three sets of nondominated solutions to a minimization MOP. Each solution represented by a different symbol data point. The fronts in closer proximity to the true Pareto front, shown by the solid line, have achieved greater convergence and are more desirable to the DM. The fronts shown by circles and crosses are therefore higher quality outputs than the front shown with triangular data points.

Although both the fronts shown on the axes by circles and crosses appear to achieve approximately equal proximity to the true Pareto front, they are not equally desirable in terms of the third of the optimisation goals. The front shown by crosses is limited in terms of its extent and does not offer the DM with as wide a range of alternates across the objective space as the former (Talbi *et al*., 2012). It also does not offer as clear a view of the trade-off relationship between the two objectives, which may be a goal of the optimisation exercise.

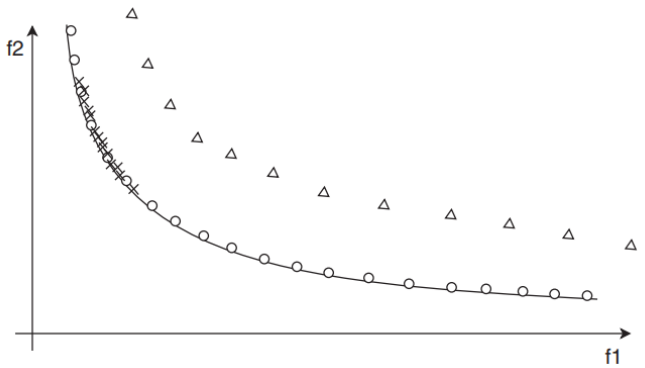
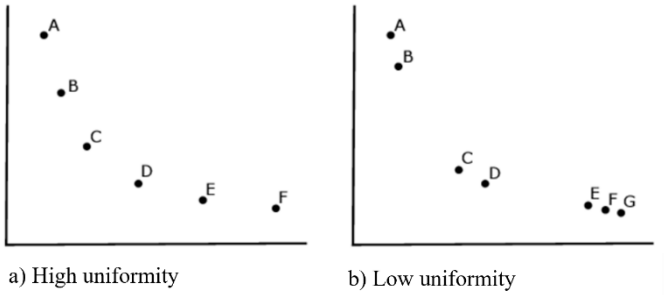


Figure 23: Examples of three Pareto front approximations relative to the true Pareto front of a bi-objective MOP (Talbi *et al*., 2012).

Figure 24 illustrates the second goal given by Zitzer *et al.* (2000). The front shown in Figure 24(a) would usually be considered a better front than the one in Figure 24(b) as the solutions are more uniformly distributed in objective space.



a) High uniformity b) Low uniformity

Figure 24: Examples of two Pareto fronts with similar convergence and spread, but displaying different degrees of uniformity (Scheepers and Engelbrecht, 2016).

Chand and Wagner (2015) summarised some important quality metrics of multi-objective optimisation with respect to convergence and diversity that have been developed and presented in literature. This summary is shown in Table 1. Note that the numbers in brackets correspond to the list of goals described by Zitzer *et al.* (2000).

Table 1: Quality indicators for multi-objective optimisation (Chand and Wagner, 2015).

Quality Indicator	**Measures**
Hypervolume (HV)	Convergence (1), Spread (3) and Uniformity (2)
Inverted Generational Distance (IGD)	Convergence (1), Spread (3) and Uniformity (2)
Hyperarea Ratio	Convergence (1), Spread (3) and Uniformity (2)
Averaged Hausdorff Distance (AHD)	Convergence (1), Spread (3) and Uniformity (2)
G-Metric	Convergence (1), Spread (3) and Uniformity (2)
Diversity Comparison Indicator (DCI)	Spread (3) and Uniformity (2)
Diversity Measure	Spread (3) and Uniformity (2)
R2 Indicator	Spread (3) and Uniformity (2)
Sigma Diversity Metric (SDM)	Spread (3) and Uniformity (2)
GD	Convergence (1)
Convergence Measure	Convergence (1)
Error Ratio	Convergence (1)

Laszcyk and Myszkowski (2019) presented a survey of MOP quality measures that provides precise descriptions and mathematical definitions, including the quality indicators shown in Table 1. Selected measures are presented below.

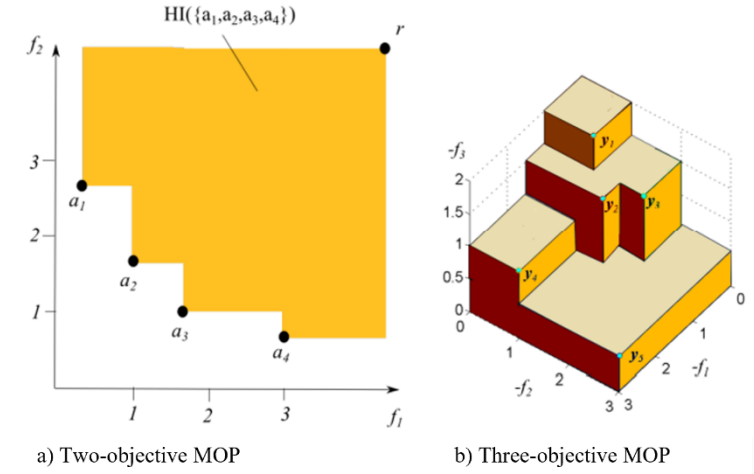
Hypervolume (HV)

The HV of a Pareto front is the volume of the hypercube defined by the front and some reference point, as shown in Figure 25. Laszcyk and Myszkowski (2019) explicitly state that this reference point is the nadir point. Other researchers, however, recommend the use of the "worst possible point", or *worst* point, which is the term adopted for this thesis. Laszcyk and Myszkowski's mathematical description of HV below has therefore been modified to align to the terminology used for this research:

$$HV\ (PF) = \Lambda(\ \cup_{s \epsilon PF}\{s' \mid s \prec s' \prec s^{worst}\}) \qquad (3.14)$$

where PF is an approximation of PF, s is the point of approximated PF, s^{worst} is the worst point of the MOP, and Λ is a Lebesgue measure, which is a generalisation of volume.

HV is a popular measure as it is an indicator of convergence, uniformity and spread. It is also the only metric that is strictly compliant with Pareto dominance (Jiang *et al.*, 2016). There is empirical evidence that shows that the maximisation of this metric produces subsets of true Pareto fronts (Coello Coello *et al.*, 2020). It is especially commonly used when only two objective functions are evaluated (Laszcyk and Myszkowski, 2019).



a) Two-objective MOP b) Three-objective MOP

Figure 25: Example HV calculations shown graphically in objective space for: (a) a two-objective MOP, in which the HV is represented by the area of the yellow-shaded region; (b) a three-objective MOP, in which HV is represented by the sum of the three-dimensional volumes of the cubes (Custódio *et al.*, 2012).

Jiang *et al.* (2016) described several limitations of the HV measure for quantifying convergence. The metric is highly sensitive to the reference point. It is also biased toward the knee-point of the Pareto curve. This results in improvements made to this region of the front resulting in more favourable HV measurements. The calculation of HV is also known to be computationally expensive.

Jiang *et al.* (2016) also explained a process of using HV as a more directed measure of the diversity of a Pareto front. *PF* is first projected onto a straight line. The HV of this transformed front is then calculated. This new HV, labelled HV_d, gives an enhanced indication of the spread and uniformity of the front. This is supported by the theorem proved by Auger *et al.* (2009) that the HV value for a front is maximised only if the points are evenly distributed (Jiang *et al*., 2016). The HV is also used to determine when to stop an EA based on the convergence of the solution from a quality perspective (Marti *et al.,* 2009).

Figure 26 shows the HV_d concept using two Pareto fronts with differing uniformity. The HV_d, shown by the shaded areas, of the more evenly spread front of Figure 26(a) is larger than that of Figure 26(b).

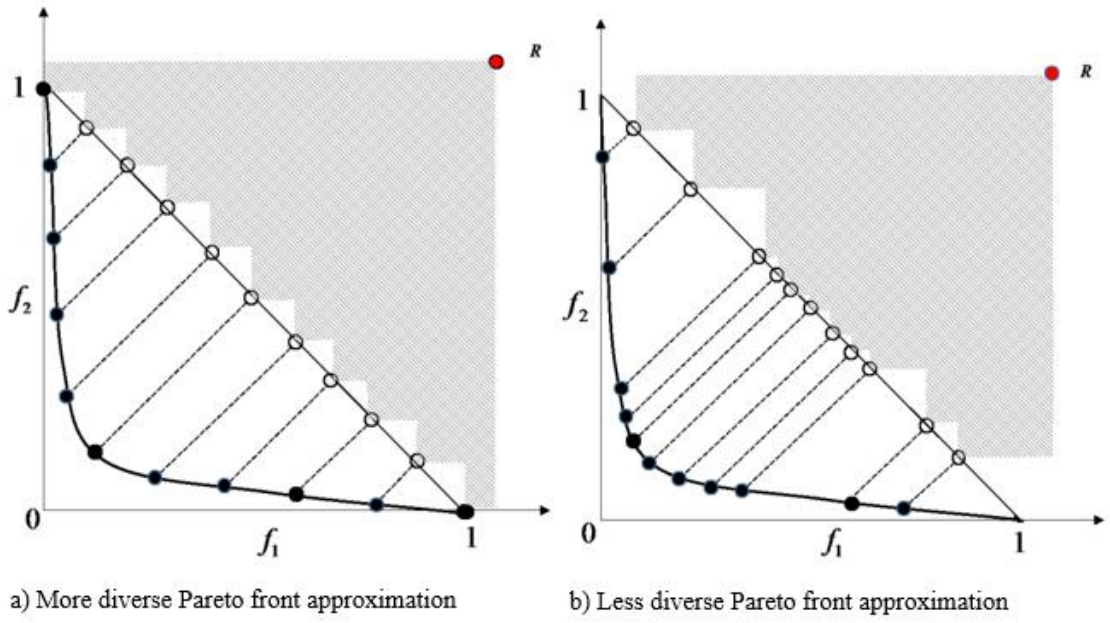


Figure 26: An illustration of how HV_d gives an indication of the diversity of an approximation of a Pareto optimal front (Jiang *et al*., 2016).

Inverted generational distance (IGD)

IGD is the average distance between each point on the true Pareto front PF_{true} to its closest point of the measured front *PF* (Laszcyk and Myszkowski, 2019).

$$IGD\ (PF) = \frac{\sqrt{\sum_{i=1}^{|TPF|} d(s_i,PF)^2}}{|TPF|} \tag{3.15}$$

where *PF* is an approximation of *PF*, *TPF* is the true Pareto front, s_i is a point of the true Pareto front, and *d* is the distance between the point s_i and the closest point in *PF*. Since the true Pareto front of an LECP problem instance is unlikely to be known, this quality measure has little applicability to the LECP.

Figure 27 shows the distances as solid black lines. It follows that this measure requires PF_{true} to be known, which is often not the case for real-world MOPs (Coello Coello *et al.*, 2007).

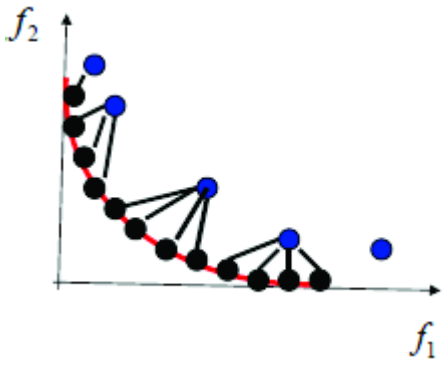


Figure 27: An illustration of the distances used for the IGD quality metric (Mashwani *et al*., 2015).

Diversity measure

The calculation of the diversity measure (DM) of a Pareto front *PF* involves dividing the objective space into a grid of equally sized cells. A predefined function is used to assign a value to each cell based on the existence of a nondominated solution within the cell or neighbouring cells. The ratio of the sum of the function values of each cell to the same sum for the true Pareto front PF_{true} is then calculated as the DM (Laszcyk and Myszkowski, 2019).

$$DM\ (PF) = \frac{\sum_{H(i,j,\dots)\neq 0} m(h(i,j,\dots))}{\sum_{H(i,j,\dots)\neq 0} m(H(i,j,\dots))} \tag{3.16}$$

where *i* and *j* are the indices of the grid, *h* is equal to 1 if a nondominated solution lies within the cell or 0 otherwise, *H* is equal to 1 if the corresponding *h* is 1 and the cell contains a nondominated solution from PF_{true}, and *m* is a function that returns the local diversity of a cell.

It follows that PF_{true} must be known for the DM calculation, which is a significant barrier to its implementation for evaluating a Pareto front approximation of an instance of the LECP. The *m* function, as well as the grid configuration and cell sizes, must be defined for the diversity measure. As this measure does not take the number of nondominated solutions in an occupied cell into account, the size of the grid partitions is critical to the meaningfulness of this metric (Laszcyk and Myszkowski, 2019).

Sigma diversity metric (SDM)

The sigma diversity metric (SDM) calculation requires the construction of straight reference lines emanating from the origin of the objective space. Figure 28 shows how these lines might be constructed. The angles between adjacent lines must be constant, although the angles relative to the axes are arbitrarily chosen. To calculate the SDM, the proximity of the Pareto front solutions to each of these reference lines must first be determined. The number of reference lines that have a solution positioned

within a distance d is then counted. The SDM is calculated by dividing this value by the number of reference lines.

$$D = \frac{C}{R} \tag{3.17}$$

where C is the number of reference lines with a PF solution positioned within a distance d of the line, and R is the number of reference lines (Mostaghim and Teich, 2003).

This metric uses several parameters that require tuning. A further disadvantage is that the positioning of the reference lines may also not fit the problem (Laszcyk and Myszkowski, 2019). Nonetheless, the SDM is relatively simple to implement and computationally light (Mostaghim and Teich, 2003).

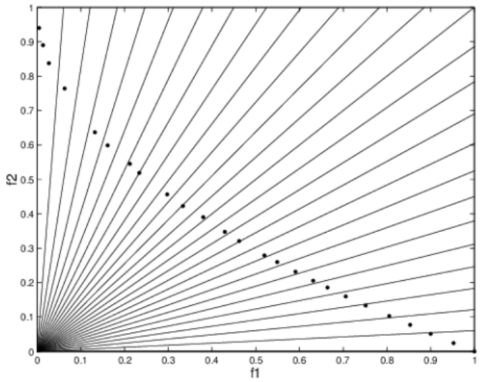


Figure 28: A representation of the reference lines, shown as black solid lines, constructed for calculating the SDM value of a Pareto front of solutions, shown as black data points (Mostaghim and Teich, 2003).

Error ratio

The error ratio (ER) is the ratio of the total number of dominated solutions contained in this set of visited solutions to the total size of the set. This calculation therefore requires the complete set of solutions generated by the MOP solving method during the exploration of the solution space.

The error ratio does not require the true Pareto front PF_{true} to be known. This is advantageous in the context of real-world MOPs, such as the LECP, as PF_{true} is usually unknown (Laszcyk and Myszkowski, 2019). This metric, however, gives an indication of the efficiency of the method rather than its efficacy.

The definition of error ratio given by Coello Coello and Pulido (2005), however, differs in that it is the ratio of solutions that are common to both PF and PF_{true} to the number of solutions of PF.

$$ER = \frac{\sum_{i=1}^{n} e_i}{n} \tag{3.18}$$

where n is the number of nondominated solutions in PF, e_i is 0 if the solution i is a member of PF_{true}, and $e_i = 1$ otherwise.

This adapted ER provides a better indication of convergence than the measure described by Laszcyk and Myszkowski (2019). It does, however, require PF_{true} to be known.

3.2 Multi-objective evolutionary algorithms

3.2.1 Optimisation approaches

Evolutionary algorithms (EAs) represent a subbranch of a diverse range of optimisation techniques studied in literature. Many optimisation methods are only suitable for solving problems with only a single objective. Some, however, can be adapted to address MOPs (Punia and Kaur, 2013). A basic understanding of these various types and subtypes of optimisation methods and their respective applicability to multi-objective optimisation is necessary to appreciate the merits of EAs with respect to the LECP.

Various taxonomies of optimisation methods have been proposed. Stork *et al.* (2020) studied these taxonomies and, from their analysis, it is evident that many researchers primarily categorise methods as either non-heuristic (or *exact*), or heuristic and metaheuristic methods (or *inexact*). Coello Coello *et al.* (2007) and Affenzeller *et al*. (2008) further distinguished between the stochastic (or *random*) and deterministic (or *enumerative*), heuristic and metaheuristic methods on the first level of their taxonomies. This differentiation is useful as these two types of inexact methods use fundamentally different strategies to search the solution space for global extrema. Figure 29 provides a graphic representation of the hierarchy presented by Affenzeller *et al*. (2008), which is used as the basis for this section of this literature review.

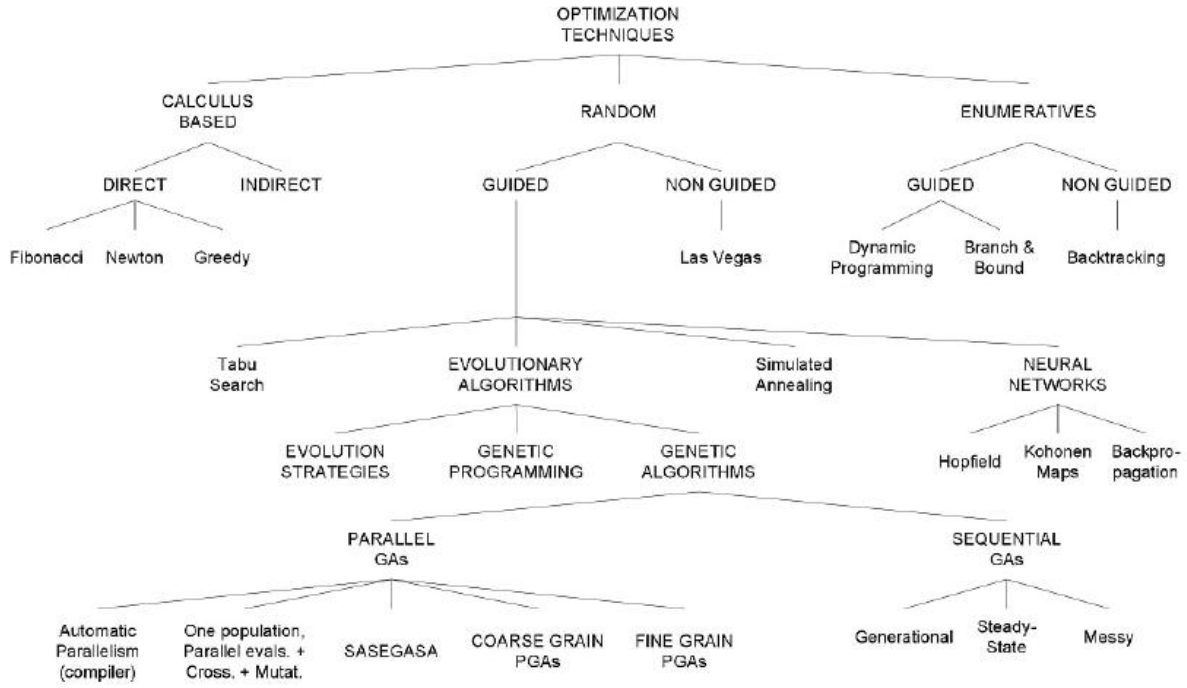


Figure 29: A taxonomy of optimisation methods (Affenzeller *et al*., 2008).

Calculus-based methods involve the determination of derivatives to calculate the gradient of an optimisation problem's objective function. The typical strategy is to iteratively generate solutions that "follow" the direction of the gradient toward a maximum or minimum objective function value. These methods are therefore often termed *hill-climbing* or *greedy* algorithms (Carr, 2014). Newton's gradient method, which uses second derivatives, is an example of a calculus-based strategy (Ravindran *et al*., 2006). Although simple to implement, these algorithms are associated with significant disadvantages. The objective function must be differentiable and therefore only continuous problems can be addressed using this method (Li and Rhinehart, 1998). Combinatorial problems are defined by Pirim *et al.* (2008) as problems that require a linear function to be minimised or maximised over a finite or countably infinite set of possible solutions[4]. These therefore require approximations or modifications of the objective function to render them differentiable and thereby allow calculus-based methods to be applied (Pogančić *et al.*, 2020). Calculus-based methods are also prone to converge on local maxima or minima and are thereby unable to find globally optimal solutions to problems with nonconvex surfaces (Kang and Kumar, 2007).

An optimisation problem is termed *convex* if the solution space can be described as a convex set. This means that the line segment between any two points in the region remains entirely inside the region. This principle is illustrated by Figure 30. The objective function of a convex problem is also convex, which requires that the line segment between any two points on the function graph should lie entirely

[4] The LECP is an example of a combinatorial problem.

below, above or on the graph. This is shown by Figure 31. The local optimum of a convex problem is also the global optimum (Nagahara, 2020). This is not necessarily the case for nonconvex problems. Convex problems are therefore widely regarded as easier to address than nonconvex problems (Luo and Yu, 2023).

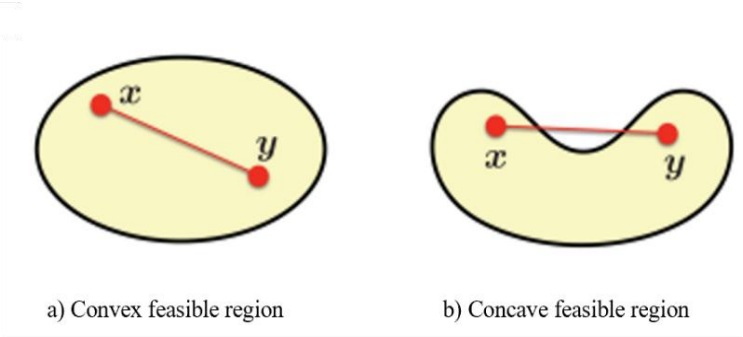


Figure 30: An examples different types of feasible regions (Nagahara, 2020).

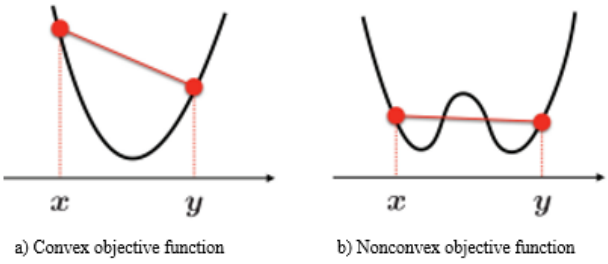


Figure 31: Examples of different types of objective functions (Nagahara, 2020).

Enumerative optimisation techniques are better suited to combinatorial problems as they involve the systematic evaluation of every possible solution to a problem to find the decision variable values that yield the best outcome (Curtin, 2005). They are therefore often termed “brute force” methods and are computationally expensive (Kulka *et al.*, 2022). This strategy is beneficial when the search space of the problem lacks structure (Nievergelt, 2000). Coello Coello *et al.* (2007) therefore note that enumerative and other deterministic methods are not well-suited to solving real-world engineering problems. Nonetheless, enumerative methods such as “branch and bound” and dynamic programming provide more efficient alternatives to simple exhaustive searches of the feasible region (Curtin, 2005).

An optimisation method’s ability to both explore and exploit the feasible region with the correct balance can allow for global optima to be found without onerous enumerative searches. Exploration is the process of generating solutions that lie in previously “unvisited” regions of the solution space, while exploitation is the refinement of a solution in the immediate “neighbourhood” to further improve upon a known, high-quality solution (Cuevas *et al.*, 2014). Figure 32 represents an objective function of a maximisation optimisation problem. The objective function curve is nonconvex and consists of both a local and global optimum. Should a simple hill-climbing method be used and the initial solution lie to the extreme left of the feature space as represented, the algorithm would iteratively generate new

solutions that produce the effect of a rightward "climbing" of the gradient and the eventual convergence to the local maximum shown by the red dot. No further exploitation is possible in that region and a hill-climbing algorithm would terminate, though it is visually clear that a more desirable solution to the problem exists elsewhere in the feature space. It is therefore necessary to mitigate the risk of this outcome by incorporating tactics that ensure that the solution space is adequately explored. Figure 32 represents such a tactic as a "jump" from the red dot position to the blue dot, which indicates the generation of a drastically different feasible solution. Once a new region of interest is found by the algorithm, it is then beneficial to exploit the region to determine the best solution within the neighbourhood. This exploitation is represented by the green dots, each indicating a relatively minor modification to the previous solution in the decision space as the best position in this locale of the objective space is sought.

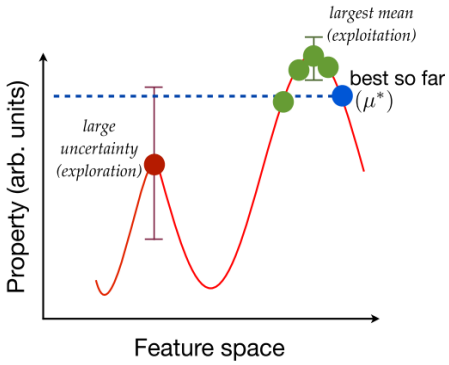


Figure 32: A schematic showing the trade-off between exploration and exploitation (Balachandran *et al.*, 2016).

Stochastic optimisation methods introduce elements of randomness to the search of the solution space to assist the process of exploration and exploitation (Li and Rhinehart, 1998). These algorithms are also known as non-deterministic, as the outcome cannot be known with precision (Cohen, 1979). A Las Vegas algorithm is defined as a non-deterministic algorithm that only returns correct solutions with a random run-time (Truchet *et al.*, 2012). This is a broad categorization of stochastic methods and, by definition, includes many guided search strategies. As such, several researchers, such as Hoos and Stützle (2013), consider Las Vegas algorithms to include guided search strategies such as simulated annealing (SA) and genetic algorithms. However, a Las Vegas algorithm, in its most basic form, does not necessarily employ tactics to specifically explore and exploit a problem's solution space and is therefore presented as an example of an unguided random optimisation technique by Affenzeller *et al.* (2008).

General stochastic strategies that guide the search of the solution space are often referred to as metaheuristics (Almufti *et al.*, 2023). These are designed to prevent premature convergence by including strategies to efficiently escape local optima and further explore the solution space for global optima.

Metaheuristics achieve this by allowing inferior moves with the hope that a better solution will be found at a later stage of the run of the algorithm (Taha, 2017).

The tabu search metaheuristic strategy allows non-improving steps to be taken, but references a list of solutions that should be avoided by the algorithm. This set, known as the *tabu list*, consists of the solutions that can lead to entrapment at a local optimum and is maintained by the algorithm during the run (Taha, 2017). This allows the tabu search to take advantage of the history of the run and explore the solution space more effectively (Hindsberger and Vidal, 2000).

Simulated annealing also allows non-improving moves during the execution of the algorithm. However, the probability of an inferior move being accepted is a function of an adjustable parameter referred to as the "temperature". The temperature value decreases with each iteration of the algorithm according to a schedule (Taha, 2017). The likelihood of an inferior solution being accepted therefore decreases as the solving process progresses. The search of the solution space thereby mimics the annealing process of metallic solids during which the probability of particles repositioning themselves within the metallic lattice decreases with a controlled and gradually lowered temperature (Pirim *et al.*, 2008).

SA provides a flexible method of finding globally optimal solutions for combinatorial problems if adequate randomness and a sufficiently gradual cooling schedule are used (Yang, 2014). The cooling schedule is particularly important (Pirim *et al.*, 2008). This, and other parameters can be difficult to fine-tune and requires modelling expertise and experimentation to define effectively. The requirement to have a highly gradual schedule can, in some cases, also result in SA being a relatively time-consuming technique (Ingber, 1993).

Neural networks are mathematical models that transform a set of input values to output variables according to nonlinear functions (Bishop, 1993). These have been primarily used as an alternative to regression techniques for determining interpolating functions for a set of data points or for cluster analysis (Gurney, 1997). However, neural networks can also be adapted to solve optimisation problems. For example, Hopfield and Tank (1985) applied neural network concepts to address the combinatorial travelling salesman problem. Chen and Liu (2022) also leveraged neural networks' built-in backpropagation algorithm as part of their Neural Optimisation Machine (NOM).

The taxonomy provided by Affenzeller *et al.* (2008) presents evolutionary strategies, genetic programming and genetic algorithms on the same level of the hierarchy. Kumar *et al.* (2010), however, indicate that genetic algorithms are methods that make use of evolutionary strategies. Furthermore, Koza (1995) explains that genetic programs are a type of genetic algorithm. As there is some ambiguity in literature regarding these techniques and their relationships to each other, which is a view shared by

Eshelman (1997) and Mitchell (1999), the general term *evolutionary algorithm* (EA) is preferred in this thesis for all categories of evolutionary optimisation techniques due to its prevalent use in the context of multi-objective optimisation. However, as literature often uses the term *genetic algorithm* (GA) to refer to methods that use evolutionary strategies in general, these terms are, at times, used interchangeably in this thesis. The subsequent sections review these algorithms in greater detail.

3.2.2 Evolutionary algorithms

EAs represent a branch of metaheuristic search methods that simulate the process of natural selection and genetics (Kumar *et al.*, 2010). Taha (2017) described them as algorithms that "mimic the biological evolution process of survival of the fittest". These methods, although varied, share the common concept of an active set of potential solutions, analogous to individual members of a population of the same species, undergo an interactive process of evolution toward better, or *fitter*, solutions (Mitchell, 1999).

In the context of evolutionary computation, the representation of a solution in the decision space is commonly referred to as the *genotype* of the individual. A solution's genotype is typically encoded as a string of bits, called *genes* or *alleles*, that is analogous to the genetic code of an individual member of a biological species population (Eshelman, 1997). The allele values are therefore sometimes referred to as the *genetic material* of the solution and, by extension, the collective set of allele values of all population members and their inherent patterns are often called the *genetic material* of the population.

The position of each allele is known as the *locus* (Mitchell, 1999). An individual population member is often referred to as a *chromosome*. An individual's associated position in objective space is known as its *phenotype*. The mapping of an individual's genotype to its phenotype is achieved using one or many objective functions[5] (Eshelman, 1997). The objective function is used as a *fitness function* that determines an individual's probability of "survival" during the evolutionary process.

To initialise the algorithm, a starting population, or *seed* is required. This differs from many other optimisation techniques, such as SAs, that require only a single seed solution. The population iteratively undergoes a sequence of transformations as sample solutions are selected from the set to recombine to generate new individuals. These samples represent *parents* in the evolutionary process, while the output of recombination can be viewed as their *offspring*. During this recombination process, stochastic unary and high order transformations of the solutions take place to ensure the resultant solution represents genetic material from both parent solutions which is also subject to an element of random modification to ensure sufficient exploration of the decision space. (Michalewicz, 1992). The likelihood of an

[5] Although the terminology differs in the evolutionary optimization context, this is tantamount to the more generally described process mapping of a solution from the decision space to the objective space.

individual being selected as a parent for recombination is a function of its fitness value(s), which thereby simulates the "survival of the fittest" phenomenon observed in nature. This achieves the evolution of the population by the retention and recombination of desirable genetic material through the repeated mating of parents with desirable attributes and subsequent production of improved generations of children (Eshelman, 1997).

Merkle and Lamont (1997) used some of these terms to define a general definition for an EA:

Let I be a non-empty set (the individual space), $\{\mu^{(i)}\}_{i \in \mathbb{N}}$ a sequence in $\mathbb{Z}^+$ (the parent population sizes), $\{\mu^{(i)}\}_{i \in \mathbb{N}}$ a sequence in $\mathbb{Z}^+$ (the offspring population sizes), $\Phi: I \longrightarrow \mathbb{R}$ a fitness function ι: $\bigcup_{i=1}^{\infty}(I^{\mu})^{(i)} \longrightarrow$ {true, false} (the termination criterion), $\chi \in$ {true, false}, r a sequence $\{r^{(i)}\}$ of recombination operators $r^{(i)}: \mathbb{X}_r^{(i)} \longrightarrow T\left(\Omega_r^{(i)}, T\left(I^{\mu^{(i)}}, I^{\mu'^{(i)}}\right)\right)$, m a sequence $\{m^{(i)}\}$ of mutation operators $m^{(i)}: \mathbb{X}_r^{(i)} \longrightarrow T\left(\Omega_m^{(i)}, T\left(I^{\mu^{(i)}}, I^{\mu'^{(i)}}\right)\right)$, s a sequence $\{s^{(i)}\}$ of selection operators $s: \mathbb{X}_s^{(i)} \times T(I, \mathbb{R}) \longrightarrow T\left(\Omega_s^{(i)}, T\left(\left(I^{\mu'^{(i)}} + \chi\mu^{(i)}\right), I^{\mu^{(i+1)}}\right)\right)$, $\Theta_r^{(i)} \epsilon \mathbb{X}_r^{(i)}$ (the recombination parameters), and $\Theta_s^{(i)} \epsilon \mathbb{X}_s^{(i)}$ (the selection parameters). Then the algorithm shown in Figure 33 is called an evolutionary algorithm.

t := 0;
initialize $P(0) := \{a_1(0), \ldots, a_\mu(0)\} \epsilon I^{\mu^{(0)}}$;
while (ι ($\{P(0), \ldots, P(t)\}$) $\neq true$) **do**
  recombine: $P'(t) := r_{\Theta_r^{(t)}}^{(t)}(P(t))$;
  mutate: $P''(t) := m_{\Theta_m^{(t)}}^{(t)}(P'(t))$;
  select:
    if χ
      then $P(t+1) := s_{(\theta_s^{(t)}, \Phi)}^{(t)}(P''(t))$;
      else $P(t+1) := s_{(\theta_s^{(t)}, \Phi)}^{(t)}(P''(t) \cup P(t))$;
    fi
  t := t+1;
od

Figure 33: Evolutionary algorithm outline (Merkle and Lamont, 1997).

Bandyopadhyay and Pal (1998) presented a simplified general form of a GA as shown Figure 34. The pseudocode is consistent with the definition of a GA given by Coello Coello *et al*. (2007), that being a class of EA that includes mutation, recombination and selection.

```
Begin
    t = 0
    initialise population P(t)
    compute fitness P(t)
    repeat
        t = t + 1
        select P(t) from P(t – 1)
        crossover P(t)
        mutate P(t)
        compute fitness P(t)
    until termination criterion is achieved
End
```

Figure 34: Generalised GA pseudocode (Bandyopadhyay and Pal, 1998).

It is preferable to generate *good* initial populations that consist of feasible individuals that already show relatively high fitness before the selection and reproduction cycle of the algorithm begins (Kazimipour *et al.*, 2014). Victer Paul *et al.* (2013) evaluated a number of seeding strategies, including random initialization, nearest neighbour, gene banks, selective initialization and sort population techniques. The strategy must be selected while considering the computational budget and the nature of the optimisation problem. Furthermore, the setting of the size of the initial population and other parameters used to effectively guide the initialization process are important variables that must be manually determined (Kazimipour *et al.*, 2014).

The fitness evaluation of an individual involves the decoding of the bitstring representation of the chromosome. This requires an encoding scheme that allows an individual's chromosome to unambiguously represent the solution in such a way that that the chromosome can be efficiently decoded, and new individual solutions encoded (Kumar, 2013). The concept of encoding and decoding solutions between the decision and objective space is illustrated by Figure 35.

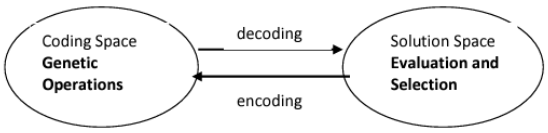


Figure 35: The relationship between the coding, or *genotype*, space, and the solution, or *phenotype*, space (Kumar, 2013).

The evolutionary cycle typically consists of three types of repetitive processes called operators. These are referred to as *selection*, *crossover* and *mutation* operators.

The selection operator is responsible for determining which members of the current population should serve as parents to the next generation (Mitchell, 1999). Shukla *et al.* (2015) reviewed commonly used selection operators. Roulette wheel selection is a process that generates and uses a random number to select a single chromosome from the set based on a probability that is proportional to each individual's relative fitness (Pencheva *et al.*, 2009). In the example illustrated by Figure 36(a), it can be inferred that individual *a1* has the highest associated fitness as it is assigned the largest angle of the roulette wheel and thereby has the highest probability of being selected as a parent.

Linear ranking selection also identifies individuals based on assigned probabilities. However, solutions are linearly assigned selection probabilities based on their fitness rankings within the set. In contrast, exponential ranking selection assigns selection probabilities to individuals using an exponential weighting mechanism. The most popular selection method for GAs is tournament selection (Shukla *et al.*, 2015). This involves pairwise comparison of randomly chosen individuals in which the solution with the highest fitness "wins" and is selected. This process is shown in Figure 36(b).

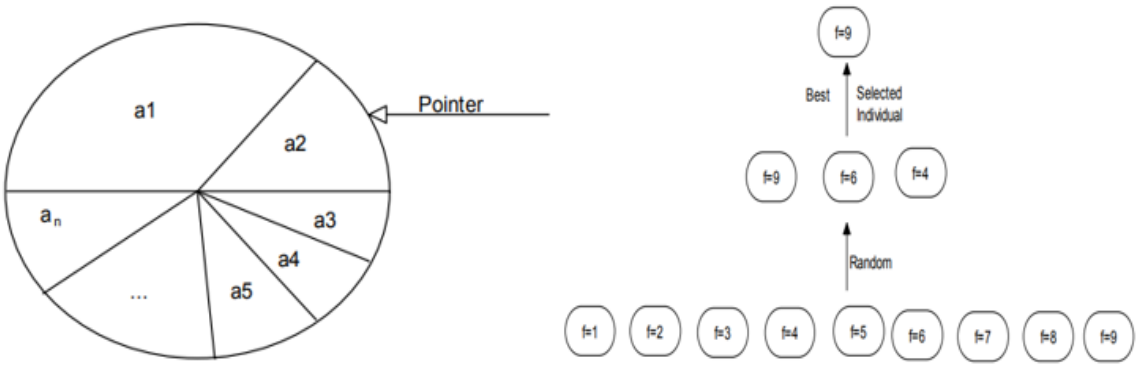


Figure 36: An illustration of different selection procedures (Shukla *et al*., 2015).

Shukla *et al.* (2015) noted that the choice of selection operator is an important aspect of the GA design. If this operator gives too little bias toward solutions with higher fitness values, it may provide too little *selective pressure* and the algorithm may fail to find optimal or near-optimal solutions. If the selective pressure is too great, the algorithm may prematurely converge to a poor-quality solution due to insufficient exploration of the solution space.

The crossover and mutation operators of a GA are responsible for achieving the balance between exploration and exploitation. The crossover operator creates new offspring using the genes of the parents identified by the selection operator (Kora and Yadlapalli, 2017). Crossover is also referred to as

recombination (Kumar and Panneerselvam, 2017). Although the more specific of these two terms, *crossover* prevails in much of the literature as recombination is often achieved by selecting a random locus and the subsequent exchange of the allele sequences before and after that position to create child solutions (Mitchell, 1999). This is known as 1-point crossover. In the example shown in Figure 37, the point between the fourth and fifth loci is used as the crossover position. There are, however, several other classical standard crossover operators. These include k-point crossover and shuffle crossover. The selection of the type of crossover operator depends on the application and can often be affected by the GA's encoding scheme (Umbarkar and Sheth, 2015).

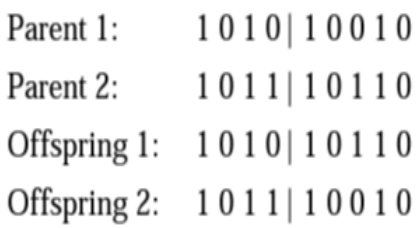


Figure 37: An example of 1-point crossover (Umbarkar and Sheth, 2015).

The mutation operator modifies the genes of offspring chromosomes. It thereby assists in preventing the establishment of a uniform population that is unable to evolve (Otman *et al*, 0212). It is therefore responsible for creating new genetic material (Konak *et al.*, 2006). The mutation operator can also serve as a means to recover good material (i.e. sets of decisions) that may be eliminated from the population during selection and crossover (Rani, *et al.*, 2013). Offspring are typically mutated based on a probability that is set as a parameter (Mitchell, 1999). Mutation can also be disruptive to the progress of the EA by destroying blocks of genes, or *memes*, that give rise to high quality solutions (Mitchell, 1999). As such, it is advisable to perform some experimentation to ensure that the mutation probability is set such that the desirable balance is achieved.

The process of selection, crossover and mutation repeats as shown in Figure 38 as the quality of the solutions that comprise the population improves. The best-known solution at the time of termination of the GA is typically taken as the output of the optimisation exercise. EAs therefore require a stopping criterion (Marti *et al.*, 2009). Wagner *et al.* (2011) presented a taxonomy of stopping criteria.

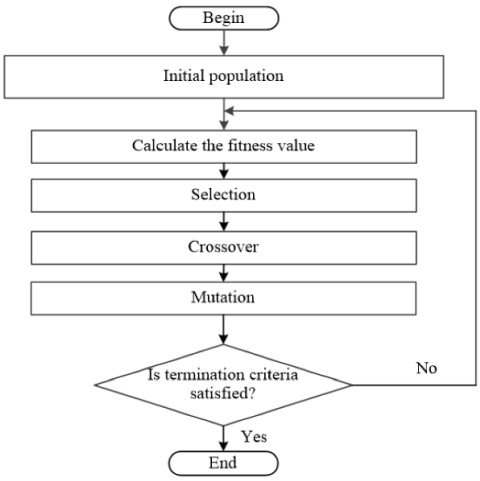


Figure 38: Flowchart of a standard GA (Albadr *et al*. 2020).

3.2.3 MOEA fundamentals

A multi-objective evolutionary algorithm (MOEA) functions in much the same way as the standard EA. Due to its population-based nature, MOEAs are able to generate a set of nondominated solutions in a single run of the algorithm. They have also been successfully used to solve problems with complexities such a multifrontality, discontinuity and disjoint solution spaces (Antion and Coello Coello, 2018). It acts as an *a posteriori* MOP solving technique, using evolutionary operators to constantly improve the approximation of the true Pareto optimal front by finding new nondominated solutions (Deb, 2011).

Coello Coello *et al.* (2007) presented a definition of an MOEA:

Let $\Phi: I \longrightarrow R^k$, ($k \geq 2$, a multi-objective fitness function). If this multi-objective fitness function is substituted for the fitness function in definition accompanying Figure 34, then the algorithm shown in Figure 34 is called a *Multi-objective Evolutionary Algorithm*.

The same MOP notation presented by Coello Coello *et al.* (2007) and discussed in section 3.1.2 may be used in the MOEA context. However, there is a requirement to differentiate between the set of nondominated solutions found at the end of a generation *t*, and a secondary population of all nondominated individuals found since the initiation of the algorithm and maintained as an external archive. The genotype of the former is termed $P_{current}(t)$, while the latter retains the symbol $P_{known}(t)$. The secondary population is required for an MOEA implementation as discovered desirable solutions are not guaranteed to remain part of the active population (Horn, 1997). The associated phenotypes are termed $PF_{current}(t)$ and $PF_{known}(t)$. In most MOEA implementations, $P_{current}(t) \cup P_{known}(t-1)$ (Coello Coello *et al.*, 2007).

There are cases in which it is desirable for the selection process of an MOEA to guarantee the survival of high-quality solutions as it can increase the convergence of the algorithm to good approximations of PF_{true} (Purshouse and Fleming, 2002). This is known as *elitism* and is usually achieved by simply identifying and copying these individuals to the next generation. Elitism is also used in single-objective optimisation, although its implementation with MOEAs is more challenging. There are two prevailing methods for incorporating elitism into an MOEA. The first is the maintenance of an elitist solution set within the active population $P_{current}(t)$. The second method is the storage of elitist solutions in an external archive and reintroducing them to $P_{current}(t)$ at strategic stages of the evolutionary process (Konak *et al*., 2006).

The goals of an MOEA are defined by Coello Coello and Lamont (2004) as follows:

1. Preserve the nondominated points found by the algorithm in objective space and their associated points in the decision space.
2. Continually move the known Pareto front $P_{known}(t)$ toward the true front P_{true}.
3. Maintain the diversity of points on the Pareto front in phenotype space and/or of the Pareto optimal solutions in the genotype space.
4. Provide the DM a sufficient but limited number of nondominated points for selection[6].

The achievement of goals (1) and (2) relate to the convergence of the approximation set to the true Pareto front and require selection processes that accommodate optimisation problems with multiple objectives. MOPs present scenarios where neither solution dominates the other and, as such, it may not be clear during the selection phase of the MOEA which individuals should survive to produce offspring. A weighted-sum of fitness values might be used to reduce the strength of an individual to a single value. However, this approach may fail to find good approximations of true nonconvex Pareto fronts (Konak *et al*., 2006). A number of ranking methodologies based on the dominance concepts have therefore been developed as alternative bases for selection in MOEAs.

Dominance count refers to the number of solutions in the population that the individual dominates. This is illustrated in Figure 39(a). It follows that higher dominance count values indicate stronger individuals. *Dominance-depth* ranking requires the classification of the population in terms of *fronts*, as shown in Figure 39(b). Individuals that form part of the Pareto front are assigned to the first front and then removed from the population. The remaining individuals that comprise the Pareto front of the new subpopulation make up the second front. The process should be until all solutions are assigned to a front. A lower dominance-depth value indicates a more desirable individual for selection. The *dominance rank*

[6] Section 1.2 states that the scope of this research excludes post-optimization pruning of the archive solution set to provide the DM with a limited range of options for *a posteriori* selection. As such, this thesis does not aim to address Coello Coello and Lamont's fourth goal of MOEAs for the LECP.

of a solution corresponds to the number of individuals that dominate it. This is illustrated in Figure 39(c). Lower dominance rank values signify stronger solutions, while individuals with a dominance rank of zero comprise the Pareto front (Reyes Fernández de Bulnes *et al.*, 2019).

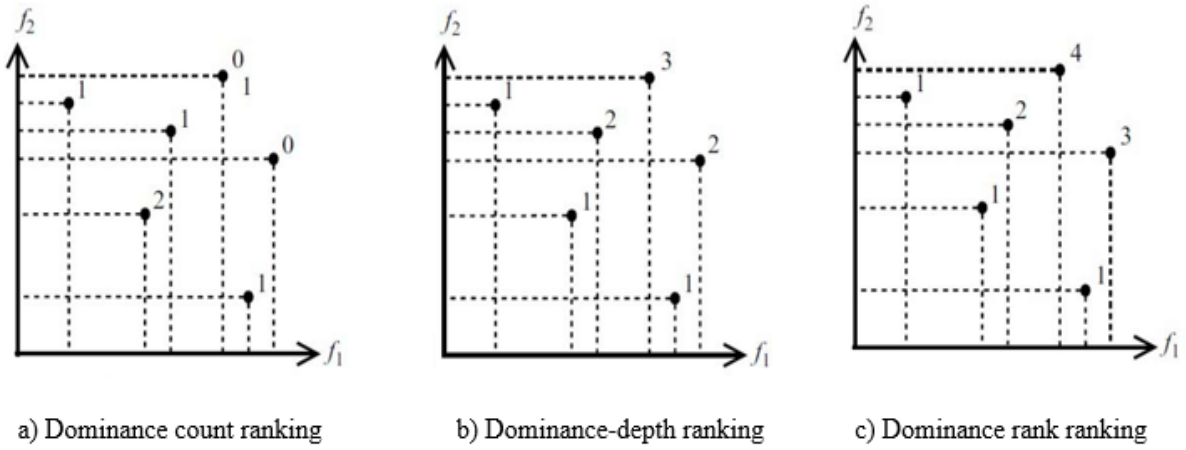


Figure 39: Examples of methods ranking solutions to a bi-objective minimization optimisation problem (Reyes Fernández de Bulnes *et al*., 2019).

The importance of the third and fourth of Coello Coello and Lamont's goals of lies in the fact that nondominated solutions tend to form clusters in phenotype space (Konak *et al*., 2006). This is an undesirable outcome, as it presents the DM with less information when making an *a posteriori* decision (Tian *et al*., 2019). Three methods for mitigating this clustering are illustrated in Figure 40. Fitness sharing, shown by Figure 40(a), involves the fitness penalization of individuals in densely populated regions of phenotype space. The number of solutions within an individual's neighbourhood, or *niche*, is used during the selection process to enforce this principle. The calculation requires an important parameter σ_{share} that defines the size of each niche, within which solutions must share their fitness scores. Cell-based density, illustrated by Figure 40(b), operates in a similar manner, except *niches* are replaced with regular cells into which the objective space is divided. The critical parameter for this approach is *K*, namely the number cells.

The crowding distance approach shown by Figure 40(c), however, removes the requirement to define σ_{share} or *K*. A crowding distance score is calculated for and assigned to each solution and used as a tiebreaker in the selection process of various MOEA solution approaches (Konak *et al.*, 2006).

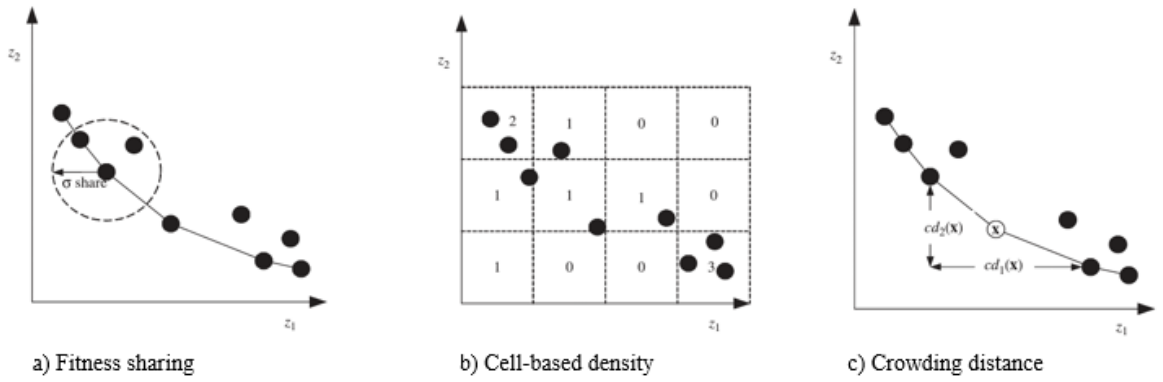


Figure 40: Examples of different approaches to diversity preservation (Konak *et al*., 2006).

An alternative approach to diversity preservation is *restricted mating*. This is a non-niching technique that involves the prevention of recombination of individuals that are beyond a pre-specified distance from each other (Zitzler and Thiele, 1999).

The implementation of niching techniques can give rise to sets of individuals existing and evolving in isolated regions of the solution space. This process is known *speciation* (Della Cioppa *et al.*, 2011). This outcome is generally considered desirable as it is viewed as a potential source of diversity (Garza-Fabre *et al.*, 2011). However, speciation sometimes prevents full exploration of the solution space. Speciation is more likely to occur when the selection scheme of the MOEA favours performance of *specialist* individuals that perform well only with respect to a single objective (Deb, 2003).

Standard models of an MOEA, such as those presented in Figure 34 and Figure 38, often omit the process of incorporating constraints to ensure the evolving population consists only of feasible solutions. As unconstrained MOPs are rare in real-world scenarios, it is necessary to build processes into an EA to enforce constraints in some way (Coello Coello, 2002). Real-world MOPs, such as the LECP, are often associated with small feasible regions. This can prove problematic to the algorithm designer and require special constraint handling methods to be incorporated into the algorithm (Tian *et al.,* 2020). Deb (2000) classified constraint handling methods as either generic methods that work for linear or nonlinear constraints or specific methods that can only address certain types of constraints.

Deb (2000) continued to state that most applications of GAs make use of a penalty function that is built into fitness functions to reduce the probability of infeasible solutions prevailing during selection. By extension, this strategy reduces the likelihood of the retention of the genetic material that results in constraint violations. Several types of penalty strategies exist, such as static, dynamic, annealing, adaptive and co-evolutionary penalties (Coello Coello, 2002). Ponsich *et al*. (2008) agreed the penalty functions are the most frequently used constraint handling method while outlining some guidelines for the development of penalization strategies. For example, it is important to consider that the global optimum of most problems is on the boundary of the solution space. If the penalty for constraint

violations is too strong, the MOEA may be restricted to searching well within the borders of the feasible region and fail to find good Pareto front approximations.

Ponsich *et al*. (2008) also reviewed other constraint handling techniques. Elimination, also described as the *death penalty method*, involves the complete removal of infeasible solutions from the active population. Coello Coello (2002) viewed this as an extreme case of a penalty function. While simple to implement, this method can be problematic for highly constrained problems. Dominance-based methods provide a less extreme strategy and involve the pairwise comparison of individuals in a tournament structure as part of a selection procedure. The logic is typically such that an infeasible solution will be considered inferior to a feasible solution, with the degree of constraint violation providing a tiebreaker if both are infeasible.

A further strategy is to prevent the production of infeasible solutions. Attempts can be made to preserve feasibility during crossover and mutation. Suitable operators must be designed to achieve this. If this is not possible, chromosomes can possibly be *repaired* to remove infeasibilities and give the nearest feasible version of the individual addressed (Ponsich *et al*., 2008). The repaired solution can be used either as a replacement for the original individual and thereby be returned to the active population or inserted to the archive population as a potential nondominated solution (Coello Coello, 2002). No GAs exist for this operation and, as such, these must be designed specifically for the optimisation problem being addressed (Ponsich *et al*., 2008).

The population-based approach of EAs make them well-suited to addressing multi-objective optimisation problems. EAs have therefore become the most popular approach to solving multi-objective optimisation problems (Konak *et al.*, 2006).

Talbi *et al*. (2012), however, also reviewed various non-evolutionary heuristic approaches to solving multi-objective optimisation problems. They reviewed the use of simulated annealing, tabu search, scatter search, path relinking, ant colony optimisation and particle swarm optimisation (PSO) to address multi-objective problems. They also discussed the combined use of multiple heuristic techniques in the form of hybrid metaheuristics. Following the review, Talbi *et al*. (2012) agreed with the assertion of Konak *et al.* (2006), concluding that GAs are the most popular approximate method for solving these types of problems. Furthermore, Stantana-Quintero *et al.* (2010) noted the popularity of GAs for solving multi-objective problems, such as the LECP, due to their ease of use and "wide applicability". Fonseca (1999) noted that the population of solutions that is maintained during the GA solving process makes this heuristic a desirable technique for solving multi-objective problems. Such a population is not maintained by other metaheuristic methods, such as tabu search and SA.

Xiujuan and Shi Zhongke (2004) agreed that GAs are comparatively easy to code, but warned that they are computationally intensive and difficult to tune. In contrast, SA was described as a more computationally efficient method than GAs.

3.2.4 MOEA solution approaches

Coello Coello *et al*. (2007) provided a generic formulation of an MOEA shown in Figure 41.

Initialise population P and P^{iv}
Evaluate objective $F(x)$ values over the populations
Assign rank based on Pareto dominance
Compute niche count
Assign shared fitness or crowding
While not terminal condition (number of generations or other)
    Selection of "good" individuals from $P \rightarrow P^{i}$
    Recombination, mutation of individuals in $P^{i} \rightarrow P^{ii}$
    Evaluate objective function values of children P^{ii}
    Rank $(P^{i} \cup P^{ii}) \rightarrow P^{iii}$ based on Pareto dominance
    Compute niche count
    Assign shared fitness or crowding
    Reduce $P^{iii} \rightarrow P$
    Copy $P^{iii} \rightarrow P^{iv}$ based on Pareto dominance
End while

Figure 41: Generic MOEA pseudocode (Coello Coello *et al*., 2007).

The majority of MOEA solution approaches follow this structure. Most variations of this definition are due to the use of differently designed selection procedures (Coello Coello *et al.*, 2007). The remainder of this section reviews some commonly used solution approaches that use specific selection operators, all of which are candidates for the evolutionary approach to solve the LECP. The set of MOEAs used to test the designed approach with the case study problem instances were chosen based on the recommendation of (Coello Coello *et al.*, 2007) and not on scrutiny of each solution approach's merits in the context of the LECP. This section therefore assumes that these MOEAs could be implemented to address the LECP and any solution approach not tested as part of the experimental component of this research could be employed as further research.

Criterion selection

Schaffer (1984) introduced a simple MOEA selection approach. His vector evaluated genetic algorithm (VEGA) involves the division of the population into subpopulations. Each subpopulation is assigned, at

random, one of the defined objective functions of the MOP. The members of each subpopulation undergo fitness assignment and roulette wheel selection according to only one objective function assigned to the group. The surviving individuals from each subpopulation are then combined and mating follows, as shown in Figure 42.

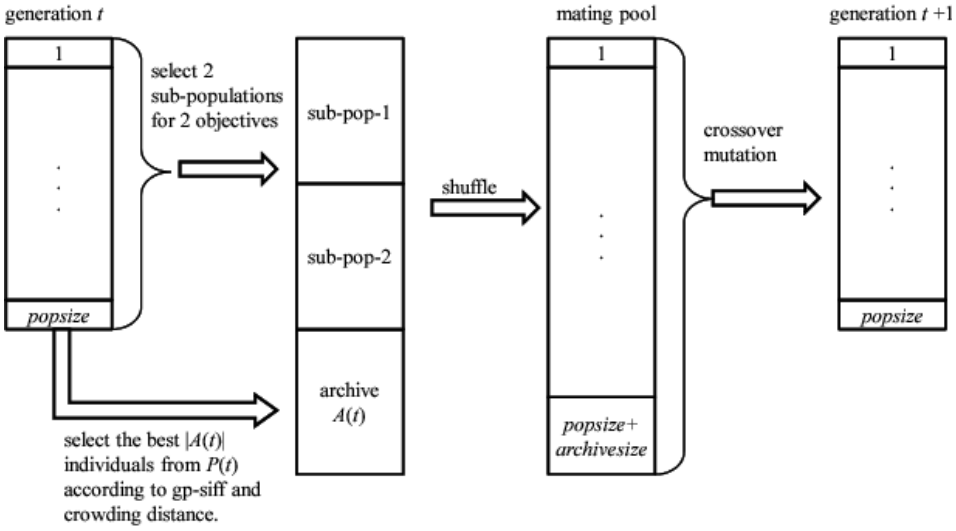


Figure 42: Schematic illustrating the VEGA process as part of an MOEA (Zhang and Fujimura, 2010).

Although, simple to implement, the selection pressure of VEGA is applied such that individuals that perform well in only one of the objectives, even at extreme detriment to their performance according to the remaining objectives, are favoured (Deb, 2003). This gives rise to undesirable speciation, as good compromise solutions, which may be of interest to the DM, are disadvantaged by the selection mechanism and their genetic material removed from the overall population (Horn, *et al.*, 1994). Konak *et al*. (2006) noted that this method strongly directs the algorithm to finding the nadir and ideal points of the Pareto front. This renders criterion sampling unsuitable for MOPs with obvious Pareto front endpoints, such as the LECP, as optimisation in such cases only adds value by finding good compromise solutions along the Pareto front between these points in phenotype space.

Aggregation selection

Van Veldhuizen and Lamont (1998) describes the use of scalarization, typically used as an *a priori* MOP technique, as part of the *a posteriori* MOEA process. Fitness is assigned to individuals based on a weighted linear or nonlinear combination of their respective objective function values.

The expression for the combination, presented by Van Veldhuizen and Lamont, is conceptually equivalent to that shown by equation 3.9. The weights assigned to each objective vary between generations or each fitness evaluation (Khare *et al.*, 2009). This single fitness is then used by the selection operator. This method can be used to reduce the multiple objectives of the LECP to a single

fitness function, after which traditional EA processes can be employed to evolve the population to a Pareto front approximation.

NPGA/NPGA 2

The niched pareto genetic algorithm (NPGA) for multi-objective optimisation was developed by Horn *et al.* (1994). For this approach, the prevailing concept of tournament selection for GAs was adapted for MOPs and introduced to the MOEA process. Its selection operator checks for a dominance relationship between randomly selected individuals. The dominant solution, should it exist, is retained, while the dominated solution is discarded. Diversity preservation is addressed by algorithm using a niche count as a tournament tie breaker if neither solution dominates the other.

Erikson *et al.* (2001) implemented an improvement to the NPGA. The new version, named NPGA 2, incorporates the Pareto ranking of the selected individuals into the tournament selection operator. Figure 43 shows the pseudocode for NPGA. The pseudocode for NPGA 2 differs only by the additional note that the specialised binary selection is done by dominance rank.

```
procedure MOGA (N′, g, f_k(x)) ▷ N′ members evolved g generations to solve f_k(x)
    Initialise population P′
    Evaluate objective value
    for i=1 to g do
        Specialised binary tournament selection
        Begin
            if Only candidate 1 dominated then
                Select candidate 2
            else if Only candidate 2 dominated then
                Select candidate 1
            Else if both are dominated or nondominated then
                Perform specialised fitness sharing
                Return candidate with lower niche count
            end if
        End
        Single point crossover
        Mutation
        Evaluate objective values
    end for
end procedure
```

Figure 43: NPGA pseudocode (Coello Coello *et al*., 2007).

MOGA

Fonseca and Fleming (1999) developed and presented a generalised multi-objective genetic algorithm (MOGA), the pseudocode for which is shown as Figure 44. The ranking mechanism corresponds to the dominance rank concept presented by Reyes Fernández de Bulnes *et al.* (2019) shown in Figure 39(c). This ranking type can introduce large selective pressure to the procedure and thereby excessively inhibit exploration of the search space. As such, Fonseca and Fleming proposed the use of a niche count to scale the fitness of individuals with the same rank scores.

```
procedure MOGA (N′, g, f_k(x)) ▷ N′ members evolved g generations to solve f_k(x)
    Initialise population P′
    Evaluate objective values
    Assign rank based on Pareto dominance
    Compute niche count
    Assign linearly scaled fitness
    Share fitness
    for i=1 to g do
        Select via stochastic universal sampling
        Recombine via single point crossover
        Mutate
        Evaluate objective values
        Assign rank based on Pareto dominance
        Compute niche count
        Assign linearly scaled fitness
        Share fitness
    end for
end procedure
```

Figure 44: MOGA pseudocode (Coello Coello *et al*., 2007).

NSGA/NSGA-II

The nondominated sorting genetic algorithm (NSGA) proposed by Srinivas and Deb (1994) makes use of layered solution classification system equalisation using the dominance-depth ranking method illustrated by Reyes Fernández de Bulnes *et al.* (2019) in Figure 39(b). Figure 45 provides the pseudocode for the NSGA solution approach. After all individuals are ranked according to their respective fronts, dummy fitness values, proportional to the population size and the front to which the individual belongs, are assigned. This is followed by a fitness sharing process based on a niche count. Individuals in sparsely populated regions of the phenotype space from the first fronts are thereby assigned higher fitness values and are more likely to be selected for reproduction via stochastic proportionate selection.

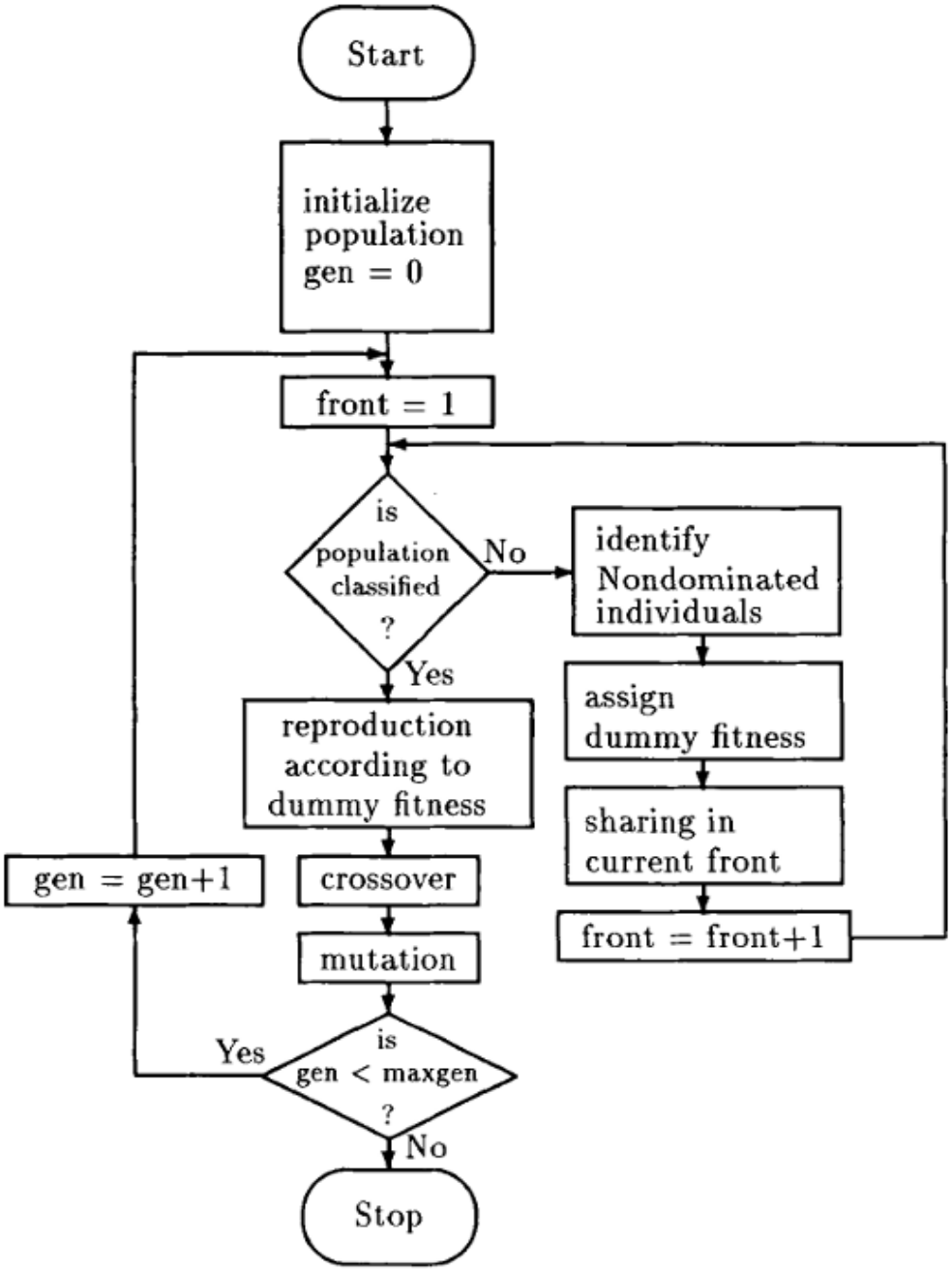


Figure 45: Flowchart of the NSGA solution approach (Srinivas and Deb, 1994).

Shortcomings of the NSGA approach include the high computational demand of nondominated sorting and the absence of elitism (Deb, 2002). The requirement to specify a sharing parameter is also problematic (Konak *et al.*, 2006).

Deb (2002) proposed an improved version of the NSGA approach, which was named NSGA-II. This approach, illustrated by Figure 46, incorporates elitism and does not require a sharing parameter, thereby overcoming two significant weaknesses of its predecessor.

An offspring population from a parent population of the same size, of pre-specified size *n*, is generated by crossover and mutation operators. The offspring and parents are combined and undergo dominance rank ranking. The next parent generation of size *n* is generated by iteratively including the individuals from each Pareto front of the combined population, starting from the first front. This achieves elitism, which significantly increases the rate of convergence of the algorithm to the true Pareto front (De Buck *et al*, 2019). The process is terminated when the number of individuals in the generation reaches *n*.

The final Pareto front that is included in this new generation is unlikely to provide exactly the number of individuals required to complete a population of *n* individuals. The final front therefore undergoes crowd-distance ranking to determine which individuals occupy the least crowded areas of the phenotype space. This front is sorted by the crowd-distance ranking and trimmed to provide the next parent population with the exact number of individuals required while favouring solutions that have the highest likelihood of enhancing the diversity of the next generation (D'Souza *et al.*, 2010). The pseudocode for this process was provided by Coello Coello *et al*. (2007) and is shown as Figure 47.

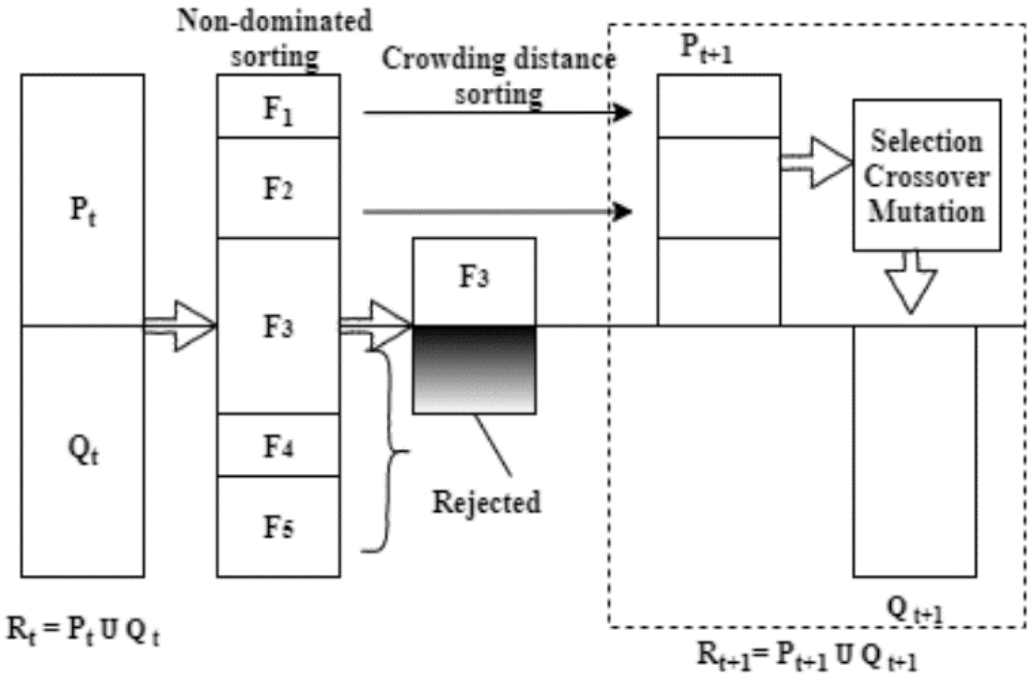


Figure 46: Schematic of the NSGA-II procedure (Verma *et al*., 2021).

procedure NSGA-II $(N', g, f_k(x)) \triangleright N'$ members evolved g generations to solve $f_k(x)$

Initialise population P'

Generate random population – size N'

Evaluate objective values

Assign rank (level) based on Pareto dominance - *sort*

Generate child population

Binary tournament selection

Recombination and mutation

for *i*=1 to *g* **do**

for each parent and child in population **do**

Assign rank (level) based on Pareto – *sort*

Generate sets of nondominated vectors along PF_{known}

Loop (inside) by adding solutions to next generation starting from the *first* front until N' individuals found determine crowding distance between points on each front

end for

Select points (elitist) on the lower front (with lower rank) and are outside a crowding distance

Create next generation

Binary tournament selection

Recombination and mutation

end for

end procedure

Figure 47: NSGA-II pseudocode (Coello Coello *et al*., 2007).

Konak *et al.* (2006) cites the fact that the crowding measure only considers proximity of solutions in phenotype space as a disadvantage of NSGA-II. This, however, can be said of most MOEA solution approaches. NSGA-II is a "powerful search method" (Wiem Mkaouer and Kessentini, 2014) and a popular solution approach to real-life MOPs (Yang, 2014). Furthermore, it is used extensively to solve combinatorial problems (Verma *et al.* 2021).

PAES

Knowles and Corne (2000) introduced the Pareto archive evolution strategy (PAES) MOEA, shown in Figure 48. The authors argued that this was "the simplest possible non-trivial algorithm capable of generating diverse solutions in the Pareto optimal set". It employs a (1 + 1) evolution strategy. This means that a single parent always produces a single child. PAES therefore can be described as an entirely local search procedure that relies solely on mutation to generate new solutions.

PAES' selection mechanism involves the comparison of only two candidates, namely the parent and mutant (Corne *et al.*, 2000). In the event of neither solution dominating the other, an external archive of solutions and a crowding measure that uses an adaptive grid is used to determine whether that mutant should be retained and added to the archive, or discarded.

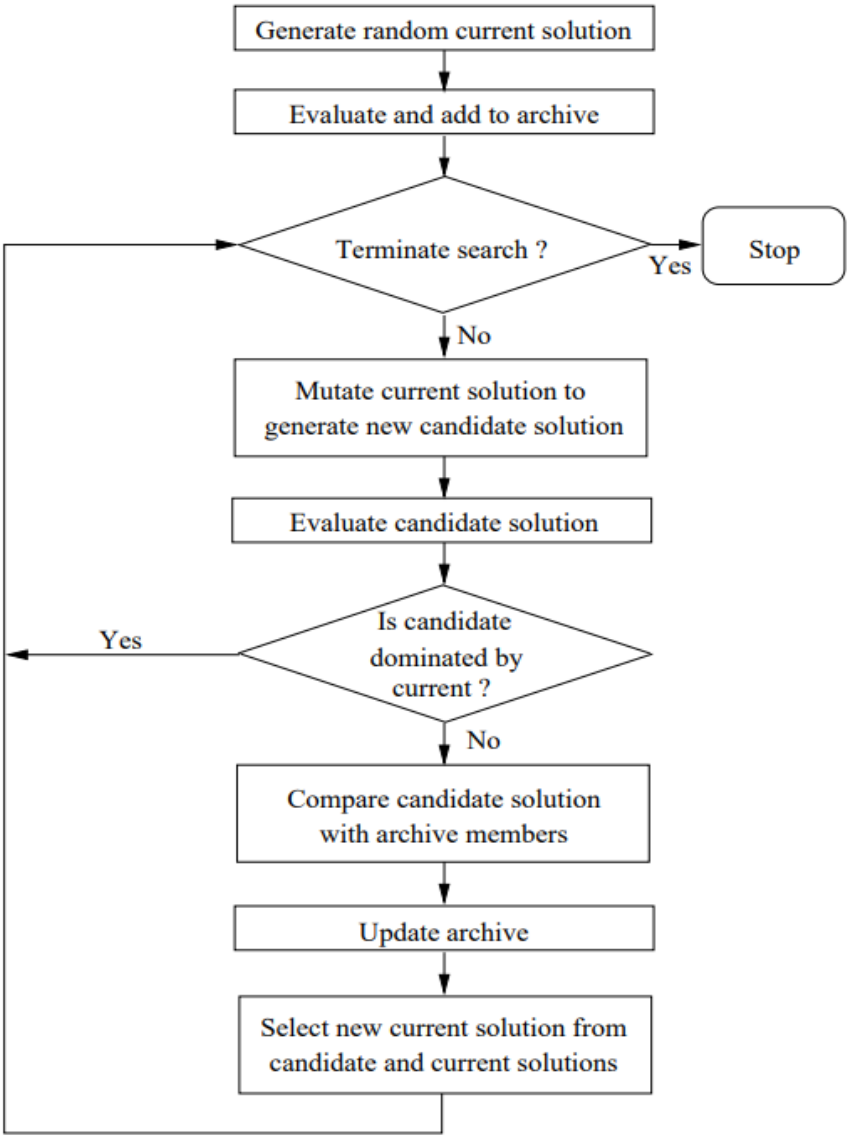


Figure 48: Flowchart of the PAES solution approach (Knowles and Horne, 2000).

The simplicity of this algorithm results in relatively low computational cost (Coello Coello *et al.*, 2007). However, the absence of recombination limits this approach to a type of random hill-climbing strategy with a relatively poor ability to search complex solution spaces (Konak *et al*, 2006). PAES is also known to display scalability problems (Hernández Aguirre *et al.* 2003). Although the complexity and scale of the problem studied for this thesis is dependent on the shipper's context, it is reasonable to expect most LECP problem instances to consist of relatively large numbers of decision variables that form complex search spaces, which would render PAES unsuitable for this MOP.

SPEA/SPEA2

The strength Pareto evolutionary algorithm (SPEA) was introduced by Zitzler and Thiele (1999). It is an MOEA solution approach that "combines several features of previous multi-objective EAs in a unique manner". SPEA is characterised by an evaluation procedure that actively refers to P_{true} in the operation of the MOEA (Van Veldhuizen and Lamont, 1998). It also includes a clustering algorithm that prunes the set of archived nondominated solutions in such a way that its characteristics are retained. While advantageous from an optimisation perspective, Konak *et al.* (2006) described the complexity of this procedure as a disadvantage of this solution approach.

In their subsequent paper presenting an improved version of SPEA, Zitzler *et al.* (2001) outlined further shortcomings associated with the approach that had since been discovered. These included the reduced selection pressure that results from the equal ranking of individuals and the removal of extreme solutions from the archive during the pruning procedure.

SPEA's successor, SPEA2, overcomes the former of these weaknesses by using a "fine-grained" fitness assignment method that makes use of density information. The *strength* of each solution in the active and archive populations is initially calculated as the number of solutions that they dominate in their respective sets. The *raw strength* of each solution in the active set is then calculated as the sum of the strengths of the individuals in both sets that dominate it. The inverse of the distance from the solution's k-th nearest neighbour is then added to the raw fitness to give the individual's fitness and thereby acts as a tiebreaker in scenarios where solutions do not dominate each other.

SPEA2 addresses the issue of extreme solution removal by replacing the clustering method with a truncation procedure to size the nondominated set of solutions to the archive size limit. This process involves the iterative removal of the individual that is positioned closest to another in the phenotype space until the desired number of individuals is reached. This process is illustrated by Figure 49. The crossed-out solid black dots represent the solutions that were removed during truncation. The numbers alongside the removed individuals represent the sequence in which the solutions were removed. They illustrate how, with each iteration, the individual with the closest neighbour is removed, with the distance to the second closest neighbour providing a tiebreaker. The use of this truncation is computationally cheaper than the original clustering process and preserves the extreme points of the front (Konak *et al.*, 2006). The pseudocode for this solution approach is included as Figure 50.

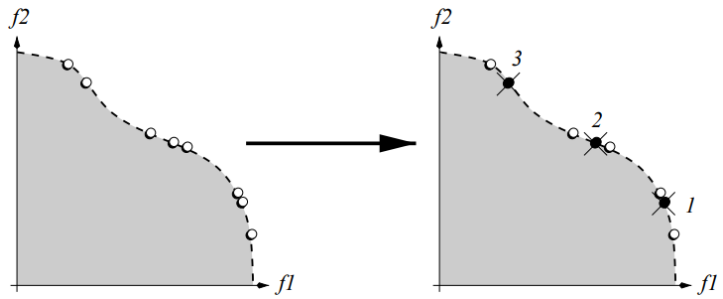


Figure 49: An illustration of the truncation procedure of SPEA2 showing the removal of nondominated individuals. (Zitzler *et al*., 2001).

procedure SPEA2 $(N', g, f_k(x))$ ▷ N' members evolved g generations to solve $f_k(x)$

Initialise population P'

Create empty external set E'

for i=1 to g **do**

Compute fitness of each individual in P' and E'

Copy all individual evaluating to nondominated vectors E' and E' to E'

Use the truncation operator to remove elements from E' when the capacity

If the capacity of E' has not been exceeded then use dominated individuals in E' to fill E'

Perform binary tournament selection with replacement to fill the mating pool

end for

end procedure

Figure 50: SPEA2 pseudocode (Coello Coello *et al*., 2007).

MOMGA/MOMGA-II

Van Veldhuizen and Lamont (2000) presented the multi-objective messy genetic algorithm (MOMGA), which is based on the GA concept of a *building block*. Gero (1995) described building blocks as subsets of the design space. In the EA context, building blocks are units of genetic material, defined by Goldberg (1989) as "short" and "highly-fit".

The MOMGA consists of three phases, which are shown as pseudocode by Figure 51. The first phase generates the initial population consisting of specifically constructed building blocks. This is achieved by a partially enumerative process. The next phase, named the *primordial phase*, typically includes tournament selection carried out upon the initial population. Finally, new solutions are generated during

the third and final *juxtapositional phase* using a "cut and splice" process of recombining the genetic material of the selected individuals.

```
procedure MOMGA (N', g, f_k(x))
    for i = 1 to epoch to g do
                                                    ▷ Initiation Phase
        Perform partially enumerative initialization
        Evaluate each population member's fitness w.r.t. k templates
                                                    ▷ Primordial Phase
        for i=1 to Max primordial generations do
            Perform tournament threshold selection
            if Appropriate number of generations accomplished then
                Reduce population size
            end if
        end for
                                                    ▷ Juxtupositional Phase
        for i=1 to Max juxtupositional generations do
            Cut-and-slice
            Evaluate each population member's fitness w.r.t. k templates
                P_Known(t) = P_Known(t) ∪ P_Known(t − 1)
        end for
        Update k templates
    end for
end procedure
```

Figure 51: MOMGA pseudocode (Coello Coello *et al*., 2007).

The MOMGA-II solution approach, developed by Zydallis *et al.* (2001), introduced efficiency improvements to the MOMGA MOEA. The deterministic initialization technique of the original MOMGA is replaced by a probabilistic process. Furthermore, the cut and splice recombination operator is replaced with a building block filtering process. They reported that these modifications remove "computational bottlenecks".

PESA

The Pareto envelope-based selection algorithm (PESA) was presented by Corne *et al.* (2000). It is described as an "unusual" MOEA approach due to its use of only elitism and a crowding measure for selection. Coello Coello *et al*. (2007) provided pseudocode for this solution approach, which is shown as Figure 52.

PESA uses an external archive of solutions to iteratively introduce parent individuals to an internal population for mating. The parents are selected according to a hyper-box crowding measure (Mishra *et al.*, 2011). Once the internal population reaches a predefined size, these individuals are evaluated against the external archive. Only the solutions that are not dominated by individuals of the external population are introduced to the archive. As such, this MOEA is highly elitist (Konak *et al.*, 2006).

procedure PESA $\left(N', f_k(x)\right) \triangleright N'$ members evolved to solve $f_k(x)$
    Initialise population P'_i of size N' randomly
    Evaluate each member of P'_i
    Initialise the external population P'_e to the empty set
    repeat
        Incorporate individuals evaluating to nondominated vectors from P'_i into P'_e
        Delete the current contents of P'_i
        repeat
            With probability p_c, select two parents from P'_e
            Produce a single child via crossover
            Mutate a single child via crossover
            Mutate the child created in the previous step
            With probability (1 - p_c), select one parent
            Mutate the select parent to produce a child
        until P'_i is filled
    until termination criteria
    Return (P'_e)
end procedure

Figure 52: PESA pseudocode (Coello Coello *et al*., 2007).

Although this algorithm is simple and computationally efficient, there is a requirement to set the population size parameters. The number of cells to use to create the hyper-box is an additional required parameter, which requires some *a priori* knowledge regarding the phenotype space (Konak *et al.*, 2006).

Micro-GA

Coello Coello and Pulido (2001) developed the micro-genetic algorithm (micro-GA) solution approach. This technique makes use of a population "memory" comprising a "replaceable" and "non-replaceable" set of solutions as shown in Figure 53. This population provides the initial population for repeated runs of a simple GA on a small population, with the replaceable portion of the memory updated with the

output of each run. The non-replaceable section of the memory remains unchanged throughout the entire micro-GA procedure to provide diversity.

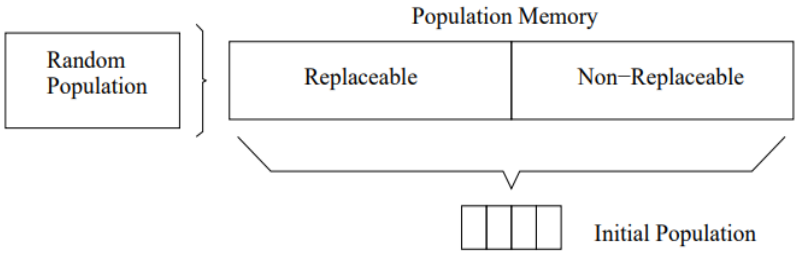


Figure 53: An illustration of how the population memory of a micro-GA is used to provide the starting population for each run (Coello Coello and Pulido, 2001).

The GA is restarted when nominal convergence is achieved, with the initial population sourced from the newly updated population memory (Coello Coello and Pulido, 2005). This process is illustrated in Figure 54.

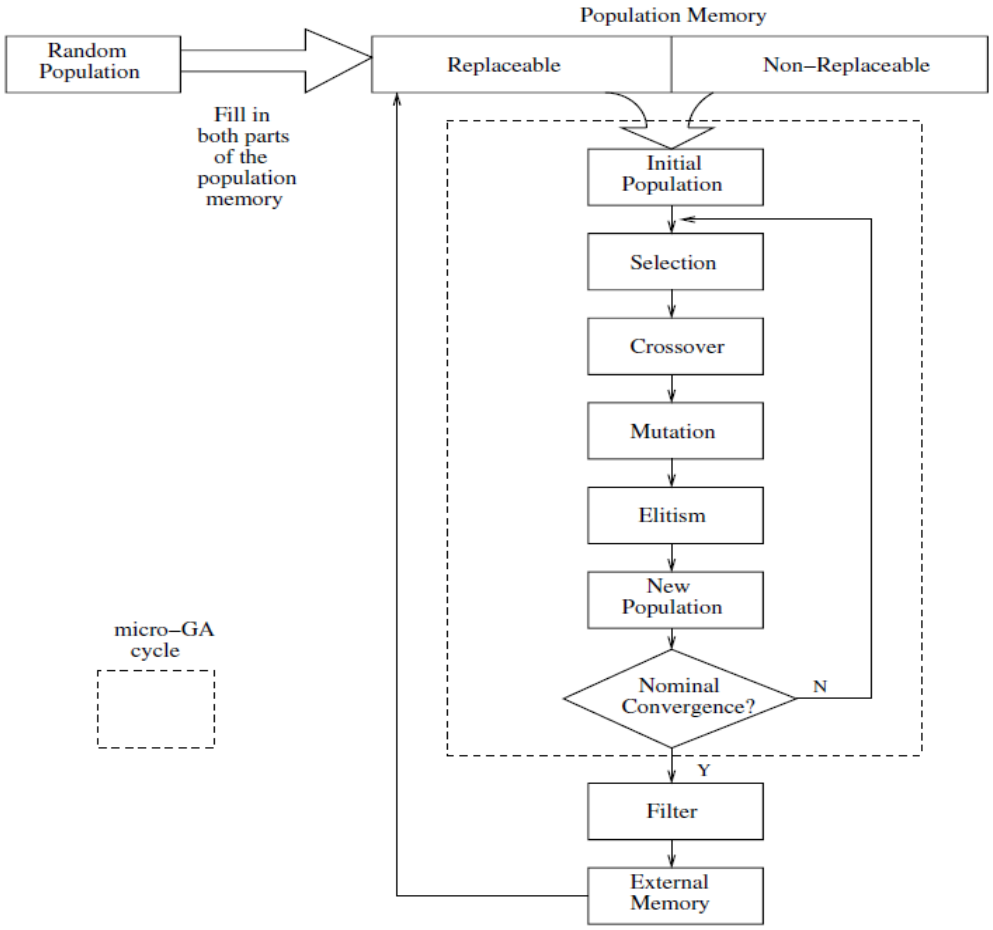


Figure 54: Flowchart of the micro-GA solution approach (Coello Coello and Pulido, 2001).

Although a highly efficient MOEA, Coello Coello *et al.* (2007) conceded that the relatively high number of parameters associated with the micro-GA is a significant disadvantage. This is addressed, to an extent, by the micro-GA's successor, the μGA^2, also developed by Coello Coello and Pulido (2003).

MOSGA

The multi-objective struggle genetic algorithm (MOSGA) was presented by Andersson and Krus (2001). It uses the prevailing GA method of child generation of random parent selection, crossover and mutation. However, after reproduction, the child is compared to existing individuals in the population, as shown by the pseudocode of Figure 55. Only the children that outcompete the individuals in their own respective regions of the phenotype space are retained and used for subsequent generations.

```
procedure MOMGA (N', g, f_k(x))
    Initialise population P_i
    repeat
        for i=1 to g do
            Randomly Select p parents from P_i
            Apply EVOPs to create a child
            Calculate the rank of the child
            Rank the entire population with the new child
            Locate the most similar individual
            if New child's ranking is better than the similar individual then
            Replace the similar individual with new child
            Update the ranking of the entire population
            end if
        end for
    until Stopping criteria is met
end procedure
```

Figure 55: MOSGA pseudocode (Coello Coello *et al*., 2007).

The MOSGA can be described as a simple variation of the MOGA (Coello Coello *et al.*, 2007). However, the replacement strategy of MOSGA is a noteworthy advantage, as it assists in preventing genetic drift (Andersson and Krus, 2001). Genetic drift is the tendency of an algorithm to converge to the production of individuals consisting of limited genetic material (Rogers and Prugel-Bennet, 1999). A strategy that limits the occurrence of this phenomenon, such as MOSGA replacement method, therefore assists in preserving population diversity.

3.3 Evolutionary approaches to multi-objective clustering problems

3.3.1 Multi-objective cluster analysis

Data clustering, often referred to as *cluster analysis*, is the process of classifying observations, data points or vectors into groups, or *clusters* (Jain *et al.*, 1999). Bilmes *et al.* (1997) presented the clustering problem as follows:

INSTANCE: A finite set X, a distance measure $\delta(i,j) \in R_0^+$ for $j, j \in X$, a positive integer K, and a criterion function $J(C, \delta(.,.))$ on a K-partition $C = \{C_1, \dots, C_K\}$ of X and measure $\delta(.,.)$.

OPTIMISATION: Find the partition of X into disjoint sets $C_1, \dots, C_K$ that maximises the expression $J(C, \delta(.,.))$.

Traditional partitioning methods, such as k-means, are widely employed to address the clustering problem. However, these algorithms tend to converge to local optima and require *a priori* specification of the number of clusters. Guided search methods such as GAs, SA and PSO have been used to overcome these issues (Maulik *et al.* 2011).

The goals of cluster analysis problems, such as the LECP, and real-world applications are varied. This provides a further challenge to both traditional and search approaches. The exact definition of the criterion functions $J(C, \delta(.,.))$ by which clustering solutions are measured are diverse. These criteria include *homogeneity*, which is the extent to which the members assigned to the same cluster should resemble each other, and *separation*, which is the degree to which items in different clusters differ from each other (Hansen and Jaumard, 1997). Measures of homogeneity include intra-cluster dissimilarity, the extent to which members of a cluster are represented by the cluster's centroid, as well as the convexity of the clusters' boundaries. Separation criteria include between-cluster dissimilarities and cluster size variation (Wierzchoń and Kłopotek. 2017).

It follows that a clustering algorithm's relative performance depends on the criterion by which it is evaluated. Most clustering algorithms seek the best possible structure according to a single criterion (Handl and Knowles, 2006). A single cluster quality metric, however, has limited applicability across datasets with varied characteristics (Saha and Bandyopadhyay, 2012). Multi-objective clustering (MOC) addresses this shortcoming by decomposing a dataset while simultaneously seeking optimality according to multiple clustering quality measures (Jiamthapthaksin *et al.* 2009).

The advantages of population-based methods over SA and PSO for solving MOPs have resulted in its emergence as the preeminent method for addressing MOCs (Maulik *et al*., 2009). Mukhopadhyay *et al*. (2015) presented a generalised MOEA for clustering that can be used to describe many MOEAs that are not specific to cluster analysis. However, Mukhopadhyay *et al*. (2015) continued to present an in-depth review of the nuances to the design considerations for the individual elements of the algorithm that must be considered when MOCs, such as the LECP, are addressed. These include the solution representation scheme, objective function selection, design of evolutionary operators, nondominated solution management strategy and the manner in which the final clustering solution is obtained.

3.3.2 Solution representation

The chromosomal representation of a clustering solution can be either prototype- or point-based.

Prototype-based encoding schemes assign meaning to each allele value of a chromosome such that the value directly represents a cluster or attribute of a cluster.

Centroid-based prototype representations of cluster sets use real numbers to denote the coordinates of points in genotype space that correspond to the centroids of the clusters. These are represented as star-shaped points in the example solution shown in Figure 56, which are not themselves members of the clusters. Mukhopadhyay *et al*. (2015) provided an example chromosome for a clustering problem of two dimensions. The chromosome (2.4 5.9 0.36 2.7 5.3 10.2) represents three clusters with centroid coordinates of (2.4, 5.9), (0.36, 2.7) and (5.3, 10.2). It follows that d x k genes are required to represent k clusters in d-dimensional space.

The relative shortness of chromosomes generated under this scheme simplifies the execution of genetic operators and facilitates quick convergence to optimality (Maulik *et al.* 2011). They also require no additional decoding if the objective function requires only a prototype cluster. However, this encoding does not capture non-convex clusters.

The medoid- and mode-based encoding scheme also involve the assignment of loci to each cluster. They thereby cater for categorical clustering problems by using an actual datapoint from each cluster as a representative (Mukhopadhyay *et al*., 2015). Figure 56 shows cluster medoids using emboldened outlines as representatives of their respective clusters. The label for each medoid would comprise the solution chromosome.

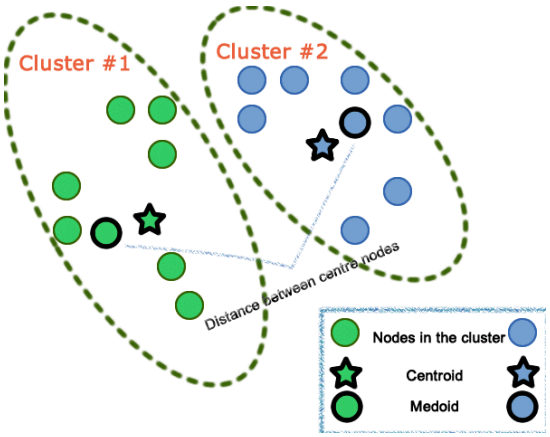


Figure 56: Example prototype representations of solution clusters for a two-dimensional clustering problem with $k = 2$ (Cheong and Si, 2018).

Point-based encoding schemes involve assignment of a position on the chromosome to each data point. Chromosomes generated using these schemes are therefore typically significantly longer than those constructed using prototype-based encoding.

Cluster label-based encoding is a straightforward method of solution representation in which each locus represents a data item and the corresponding allele value signifies the cluster to which the item is assigned (Handl and Knowles, 2004). There are multiple advantages associated with this scheme. Decoding the genotype is simple and efficient. The number of clusters does not need to be specified *a priori*. These chromosomes also provide sufficient detail to encode cluster structures of arbitrary shape.

However, the relatively lengthy chromosomes produced by cluster label-based encoding can reduce crossover and mutation operator efficiency (Maulik *et al.* 2009). It can also give rise to redundant solutions, which is generally considered an undesirable outcome (Handl and Knowles, 2004). Consider the example chromosomes (1 1 1 2 2 3 3 3) and (2 2 3 3 1 1 1) provided by Mukhopadhyay *et al*. (2015). These allele values, although different, describe the same cluster structure. Handl and Knowles (2004), however, argued that this shortcoming is not significant provided that well-designed genetic operators are used. A renumbering procedure can also be used to overcome this issue (Falkenauer, 1998).

Locus-based adjacency graph encoding provides an alternative point-based solution representation method by using an allele value per data point to describe links between grouped points. Should the locus assigned to data point i contain an allele value of j, data points i and j should be considered linked. Figure 57 illustrates how the assignment of self-links terminate the propagation of the data point linkage to form discrete clusters.

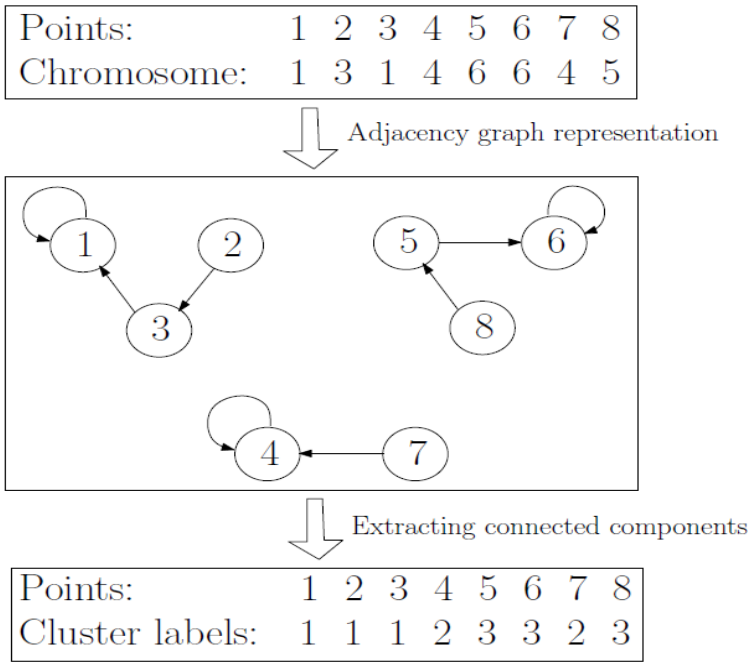


Figure 57: Example of the locus-based adjacency graph encoding and decoding procedure (Mukhopadhyay *et al*., 2015).

Like cluster label-based encoding, locus-based adjacency graph encoding allows a GA freedom to determine the optimal number of clusters. Handl and Knowles (2006) highlighted, however, that the latter method contributes the added benefit of allowing parents to undergo standard crossover processes without undergoing the disruption experienced when cluster label-based encoded chromosomes are recombined by similar operators. There is no indication in the literature that either of these encoding methods are incompatible with the LECP. Their corresponding advantages and disadvantages can therefore be weighed during the design of the evolutionary approach while considering the impact of each on other aspects of the approach, such fitness evaluation and recombination, to ensure that the most computationally robust and efficient algorithms can be implemented.

3.3.3 Objective function selection

Arbelaitz *et al*. (2013) described cluster validation as the process of evaluating how well cluster partitions fit the underlying data structure. Various cluster validation indices (CVI) are used to quantify the quality of this fit according to commonly recurring goals of cluster analysis. These indices provide a basis for developing the objective function for a clustering problem and, by extension, the fitness function for a GA that generates cluster partitions (José-García and Gómez-Flores, 2021).

José-García and Gómez-Flores (2021) described 22 distinct CVIs that measure the homogeneity or separation achieved by a clustering solution. CVIs that measure homogeneity, which is referred to as *cohesion* in this context, include the Dunn index, the Calinski-Harabasz index, the C-Index and the

Silhouette index. Separation can be quantified using the Xie-Beni index, the Modified Davies-Bouldin index and the Score Function index.

These and other CVIs have been used in combination as objective functions when addressing a clustering problem as an MOP. Handl and Knowles' MOCK algorithm provide an example of such an approach by using cluster deviation and cluster connectedness measures as objective functions, both of which are minimised by the algorithm.

The selection of the specific CVIs is subject to the discretion of the MOEA designer. Mukhopadhyay *et al*. (2015) did, however, provide some guidelines for objective function selection. Firstly, objective functions should be selected from different CVI categories. For example, a CVI that measures homogeneity should be paired with another that quantifies separation, as opposed to using it in conjunction with another homogeneity-based measure. Secondly, it is advisable to limit the selection to two CVIs where possible. This is due to unavoidable overlap of CVI categories when three or more are used, as well as the typical degradation of performance of known MOEAs as the number of dimensions of an MOP increase.

The existing CVIs in literature address the quality of a cluster in terms of the data points' relationship with others within the same or adjacent clusters. They therefore have little application to the LECP, which concerns itself with the best clusters for lane definitions that interact with the clusters in significantly more complex ways.

3.3.4 Evolutionary operator design

Evolutionary operators include parent selection, chromosome crossover and child mutation. MOC does not typically present significant unique challenges to the selection process of an MOEA (Mukhopadhyay *et al*., 2015). As such, only crossover and mutation operator design considerations are discussed in this section.

The eligibility and effectiveness of a crossover operator for a clustering MOEA is dependent on the selected chromosomal encoding scheme. Both types of encoding allow for single-point crossover, which is commonly used. However, each presents challenges for this type of crossover. When prototype-based encoding is used, care must be taken to ensure that the crossover point does not lie within the components of a prototype. For example, crossing over chromosomes between allele values that describe a centroid's coordinates would destroy building blocks and likely degrade the solution. The variable string lengths associated with prototype-based encoding can also be problematic when performing single-point crossover (Mukhopadhyay *et al*., 2015).

Point-based encoding ensures equal length strings, which is beneficial from a crossover perspective. Inconsistent cluster labelling between parents, however, can be problematic when employing single-point crossover if cluster label-based encoding is used (Mukhopadhyay *et al*., 2015). The ordering of the loci assignments to data points also becomes nontrivial when this solution representation and recombination are used simultaneously. This is not the case when locus-based adjacency graph encoding is used, which is a relatively robust scheme that accommodates single-point and uniform crossover (Handl and Knowles, 2006). Figure 58 illustrates how a uniform crossover mating process produces a sensible clustering solution child chromosome when locus-based adjacency graph encoding is used.

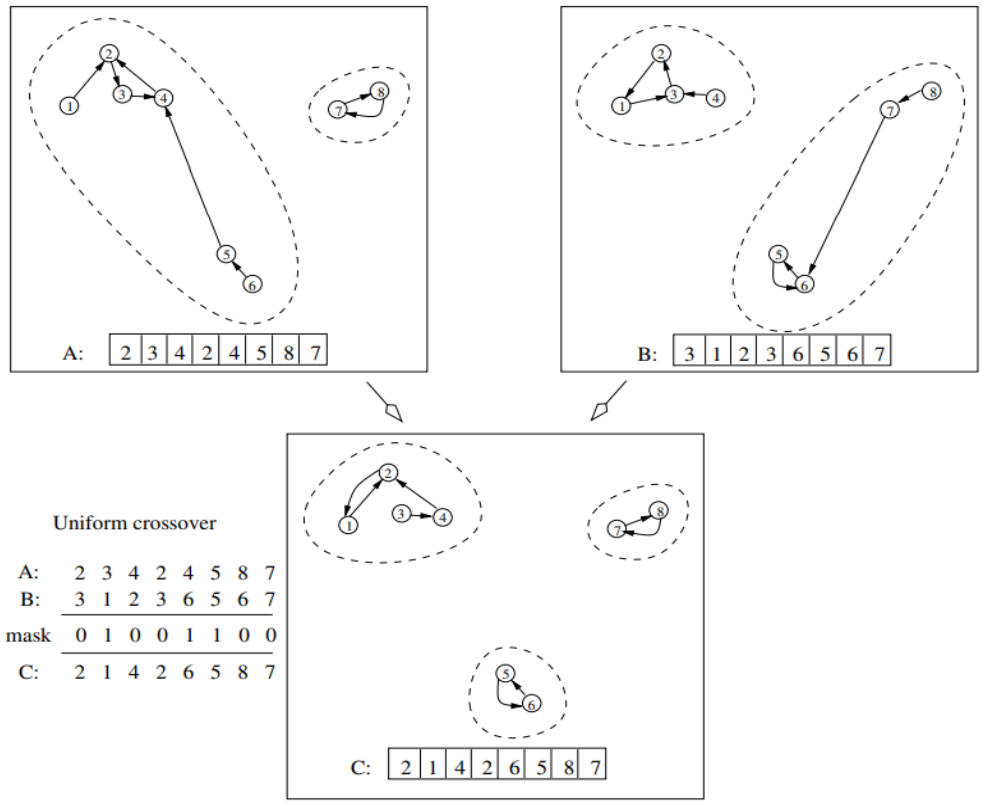


Figure 58: Example of uniform crossover used to recombine two parent chromosomes defined using locus-based adjacency graph encoding (Handl and Knowles, 2006).

Alternative recombination methods have been devised and successfully implemented to specifically overcome the challenges of producing a representative child clustering partition during the evolutionary process. Chen and Wang (2005) introduced a second crossover point to accommodate variable string lengths. Kirkland *et al.* (2011) addressed the issue of differing chromosome lengths by designing an operator that uses the largest cluster from the smallest string as the basis for the child. The remainder of the child chromosome is completed using clusters retrieved from the larger of the parent strings. The graph-predicated sequence clustering (GraSC) developed by Demir *et al*. (2010) includes a crossover method that can be used in conjunction with cluster label-based encoding. Initially all data points are represented as blank allele values and are considered "uncovered". Labels are repeatedly copied from parent chromosomes randomly until all loci are assigned a label from either parent.

Mutation operators must also be designed, selected or implemented with the solution representation scheme in mind. Mutation of child chromosomes represented by prototype-based encoding is arguably easier to achieve without creating invalid mutants. Real-valued prototypes, such as centroids, can be perturbed by modifying the component values according to a statistical distribution. These include uniform, Gaussian, polynomial and log-normal distributions (Mukhopadhyay *et al*., 2015). Medoid and modal cluster representatives can simply be replaced by other available data points (Hruschka, *et al.*, 2009).

Such perturbations and replacements are incompatible with point-based encoding. However, the variability of the number of clusters introduced by this solution representation scheme is advantageous from a mutation perspective, especially for clustering problems, like the LECP, d not specific or constrain the number of clusters. For example, Cole's (1998) mutation operators for general clustering for unknown k can be applied. These include the *split*, *merge* and *move* operators that are shown in Figure 59. Falkenauer (1998) also proposed a process of eliminating a cluster by reassigning its constituent data points to their closest respective neighbour clusters. Handl and Knowles (2004) generated new clusters by simultaneously assigning a new cluster label to nearby data points.

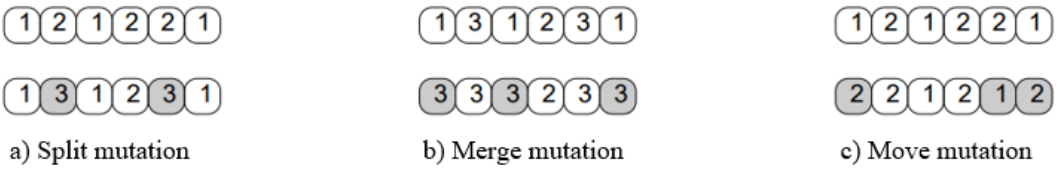


Figure 59: Examples different mutation methods of cluster label-based encoded chromosome (Cole, 1998).

3.3.5 Management of nondominated solutions

The ongoing management of discovered nondominated solutions by MOEA is a significant aspect to the solution approach. There is little in literature that suggests MOC-specific nuances complicate the management of nondominated solution. General MOEA principles apply when selecting a strategy for MOC and most solution approaches that have been used for clustering are reviewed in section 3.2.4. Literature shows that the same MOC-specific operator is often tested using different solution approaches. For example, Faceli *et al*. (2007) showed that their multi-objective clustering ensemble (MOCLE) method is effective when either the NSGA-II or SPEA solution approaches are used, which handle nondominated solutions differently. This thesis provides further examples of operators being tested using different MOEAs. As such, the detailed workings of the nondominated management solution strategies used in an MOC context are not discussed in this section. However, some review is provided of the applications of the solution approaches that are discussed in section 3.2.4.

Mukhopadhyay *et al*. (2015) provided a summary of nondominated solution handling techniques that have been applied to MOC. The preeminent MOEA solution approach is NSGA-II, which maintains nondominated solutions within the active population of the MOEA. Examples of its application to MOC include the genetic clustering approach for pixel classification presented by Bandyopadhyay *et al*. (2007), the approach used by Őzyer *et al*. (2004) to cluster m-RNA data, Chen and Wang's (2005) implementation of a cluster centre and number optimisation method, as well as Ripon and Siddique's (2009) fuzzy-based evolutionary multi-objective clustering for overlapping clusters (FEMCOC) approach.

Like NSGA-II, NPGA retains nondominated solutions within the active population. NPGA has also been used to address MOC, providing the basis of the multi-objective genetic k-means algorithm (MOKGA) presented by Liu *et al.* (2005). Mukhopadhyay *et al*. (2015) argued that the absence of elitism in this algorithm renders it less effective than NSGA-II.

PESA-II and SPEA2 have also been used to solve clustering problems. These apply an alternative different solution management strategy, as they make use of external archives of nondominated solutions. SPEA2 was used by Li and Wong (2019) to benchmark their (MOCDP) multi-objective cluster density peak method and was found to perform favourably. Examples of PESA-II approaches to MOC include Handl and Knowles' multi-objective clustering with automatic K-determination (MOCK) algorithm. MOCK has been used as the basis for many MOC studies. Matake *et al.*'s (2007) adaptation to this method made it more scalable to large datasets. Shirakawa and Nagao (2009) subsequently applied this technique to image segmentation.

3.3.6 Final MOC solution approach

Most MOEAs aim to produce a set of nondominated solutions that represent a good approximation of the true Pareto front. They typically do not include a final solution selection procedure (Branke *et al.*, 2004). Figure 60 provides an example of a framework that considers the final solution selection procedure to be separate from the MOEA. Studies also often do not address solution selection, as noted by Jing *et al.* (2019) in their review of decision-making methods for energy system design. This is not the case for MOC endeavours, which typically do include selection procedures, as shown by Mukhopadhyay *et al*. (2015).

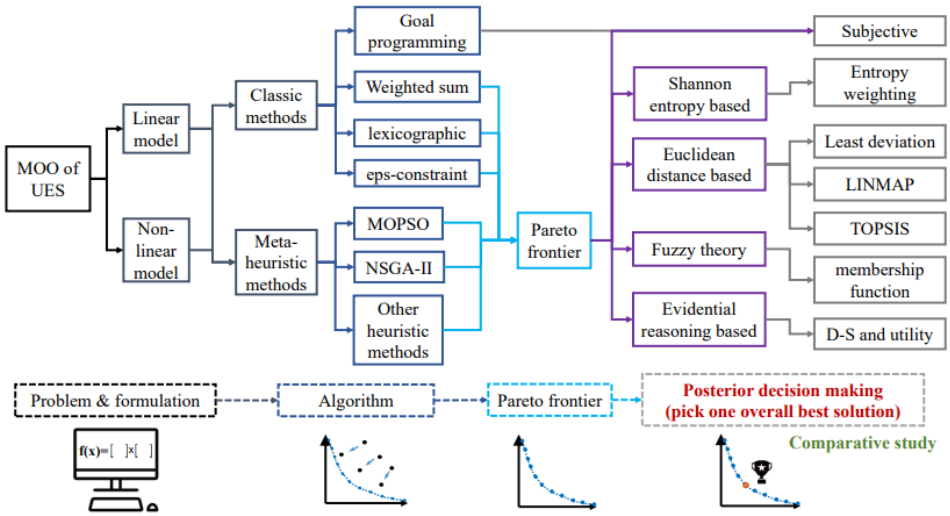


Figure 60: The problem-solving paradigm of MOPs (Jing *et al*., 2019).

A popular approach to final MOC solution selection the evaluation of the quality of the nondominated solutions using a CVI that was not used as an MOEA objective function (Mukhopadhyay *et al*., 2015). Examples include Bandyopadhyay *et al*.'s (2007) use of the I-index to select a solution after the J_m and *XB* indices were used as objective functions. Mukhopadhyay and Maulik (2011) also used the I-index for solution selection after using fuzzy cluster compactness and separation as objective functions.

An alternative approach is to reduce the number of nondominated solutions by isolating the *knees* of the Pareto approximation set. Knees are regions of a Pareto front that exhibit relatively large changes in the sensitivity of the conflicting objectives in a relatively small region of the objective space (Branke *et al.*, 2004). This is observed as a significant change in the gradient of the front. Figure 61 shows an example Pareto optimal front with a pronounced knee.

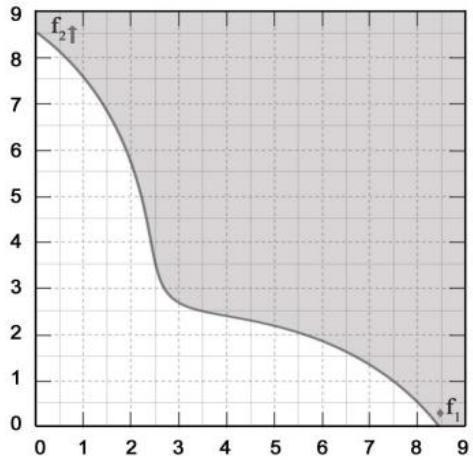


Figure 61: A representation of a Pareto front for a bi-objective minimization MOP with a knee (Branke *et al*., 2004).

Algorithms that find the knees of Pareto front approximations are computationally expensive. This is especially true for MOPs with more than two objectives (Mukhopadhyay *et al*., 2015). MOC typically does not involve three or more objectives, which makes the knee-based approach a viable solution selection technique even though more efficient methods may exist. It was used by Handl and Knowles (2007) for their MOCK algorithm, with their knee determination procedure enhanced by Matake *et al*. (2007) for a subsequent implementation of MOCK.

A *cluster ensemble* is a single representation of different clustering results combined according to a consensus function (Akbari, *et al.* 2015). Figure 62 shows how a set of clustering solutions, represented as L_1 *to* L_m in similarity matrix-form, are transformed to a single consensus solution.

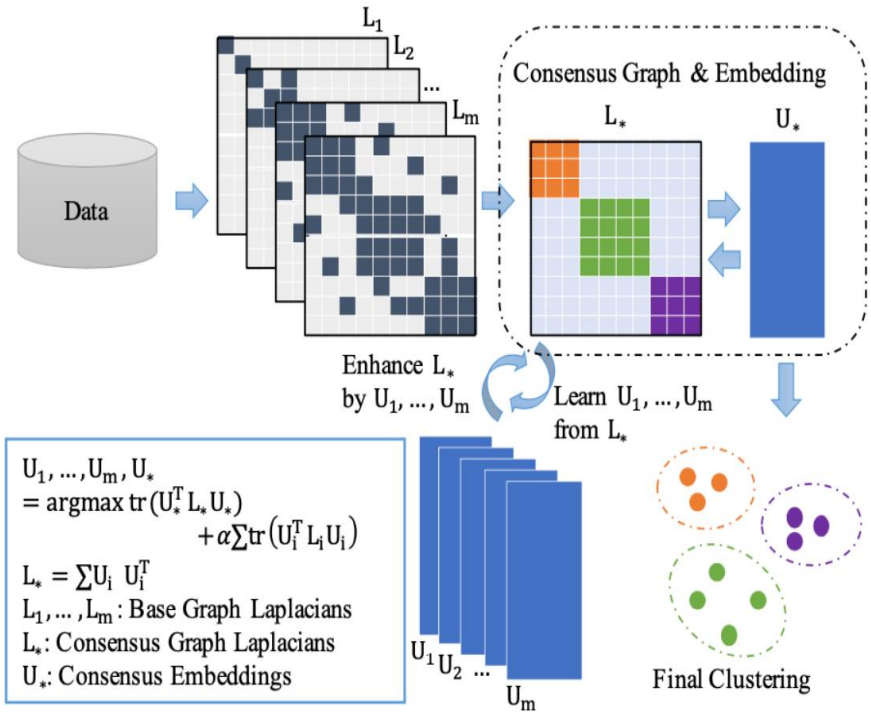


Figure 62: A representation of an example process for generating cluster ensembles (Li *et al*., 2020).

Cluster ensembles are often used as part of the MOEA process, such as the technique presented by Faceli *et al.* (2007). Ensembles can, however, be used only to reduce the end set of nondominated solutions to a limited number of good candidates for consideration as final clustering solutions to the MOP (Mukhopadhyay *et al*., 2015). For example, Maulik *et al.* (2009) used a cluster ensemble to generate a final optimal solution after using the J_m, XB and PBM CVIs to find the Pareto front approximation. Trimming the Pareto front approximation is out of the scope of this research. The I-index, knees and cluster ensembles therefore, while having the potential to assist practitioners solving the LECP, have no application to the methods developed, tested and compared in this thesis.

3.4 Improvement strategies for MOEAs

3.4.1 Local search techniques

Although the population-based approach to solving MOPs is well-suited to exploring the solution space, the mutation intensification process of MOEAs is often inaccurate. Opportunities to further improve good candidates are therefore often forfeited by the solution approaches reviewed in section 3.2.4 (Hakimi-Asiabar *et al.*, 2009). Local search methods, however, can be incorporated to these MOEAs to better exploit the solution space. In doing so, these methods locally improve individuals to produce enhanced nondominated solutions and thereby generate better Pareto front approximations (Huang *et al.*, 2019).

Once a local search technique is introduced to an MOEA, the solution approach is typically referred to as a *hybrid* MOEA (Huang *et al.*, 2019). Hybrid approaches are also often referred to as *memetic* MOEAs (MEMOEA) (Zhou *et al.*, 2011). Coello Coello *et al*. (2007) provided pseudocode for a generic memetic MOEA shown as Figure 63.

procedure MEMETIC MOEA $(N, g, f_m(x))$
  Randomly initialise population P_g with N individuals
  Evaluate fitness $f_m(x)$ of each individual x in P_g
  while Termination condition false **do**
    $g = g + 1$; number of generation
    Select P'_g from $P_{(g-1)}$ based on fitness $f_k(x)$ (k = # of objectives)
    Apply genetic operators to $P'_g \rightarrow P''_g$
    Local search in P''_g neighbourhood; $P''_g \rightarrow P'''_g$
    Evaluate fitness $f_m(x)$ of each individual x in (P''_g, P'''_g)
    Select P_g from $(P_{g-1}, P'_g, P''_g, P'''_g)$
  end while
end procedure

Figure 63: Generic memetic MOEA pseudocode (Coello Coello *et al*., 2007).

The local search operator of an MEMOEA achieves exploitation of the immediate phenotype neighbourhoods of selected individuals of the population. Simulated annealing has been a popular local search method applied to MOPs. Examples of the use of SA for this purpose include Leiva's (2000) multi-sexual-parent-crossover genetic algorithm (MSPC-GA), as well as Jaskiewicz's (2001) comparative study of various multi-objective SA algorithms as local search operators.

Adra *et al.* (2005) provided an additional local search operator that stochastically perturbs the population by randomly changing individuals' associated decision variable values. It follows that the effectiveness of this local search method is dependent on the degree of correlation or *connectedness* between the genotype and phenotype spaces of the specific MOP. Perturbations to decision variables do not necessarily translate into the desired movements in terms of the individual's characteristics if the decision space and the objective space are mapped nonlinearly (Coello Coello *et al.*, 2007). This is the case for the LECP, for which the relationships between the decision variables and objective functions are complex. Minor random perturbations of an LECP genotype would therefore likely bring about disproportionate movement in the phenotype space, thereby disrupting the local search process and yield low probability of improvement.

Further to the selection of a local search operator, of which many have been developed and shown to be successful by experimentation, the incorporation of a memetic strategy to a MOEA requires critical additional design decisions to be made. These were summarised by Lara *et al.* (2010) in terms of the following three questions:

1. At which point in the MOEA process will local search be introduced?
2. Which individuals will be subjected to local search and to what extent will they be fine-tuned via the procedure?
3. How will the refinement of individuals be applied?

Lara *et al.* (2010) provided a summary of hybrid techniques that describe alternative methods of addressing these three MEMOEA design aspects. Most of their reviewed hybridization techniques use either a scalarization function or Pareto dominance concepts to select the population members for localised fine-tuning. The selection of which type of selection mechanism depends on problem domain-specific considerations such as the ruggedness of the solution space and characteristics of the Pareto front (Coello Coello *et al.*, 2007).

Ishibuchi *et al.*'s (2003) multi-objective genetic local search (MOGLS) and Jaszkiewicz's Pareto memetic algorithm (2003) provided early examples of the use of scalarisation functions as part of the local search process. Both use a weighted vector function to select parents for local search that subsequently undergo crossover recombination. However, Jaszkiewicz's method of pre-selecting a small temporary population of good candidate parents prior to selection represented an improvement on MOGLS.

Figure 64 shows how the weighting of the scalar fitness vector, which for the example is $\boldsymbol{w}$ = (-0.1, -0.9), determines the local search direction, shown as black arrows, for selected individuals. The desired search areas for these parents, which would move the selected solution in the general direction of the

expected closet point of the true Pareto front, are shown encircled. Solution B provides an example of a solution that is not locally guided in the desired direction in phenotype space (Ishibuchi *et al.*, 2003).

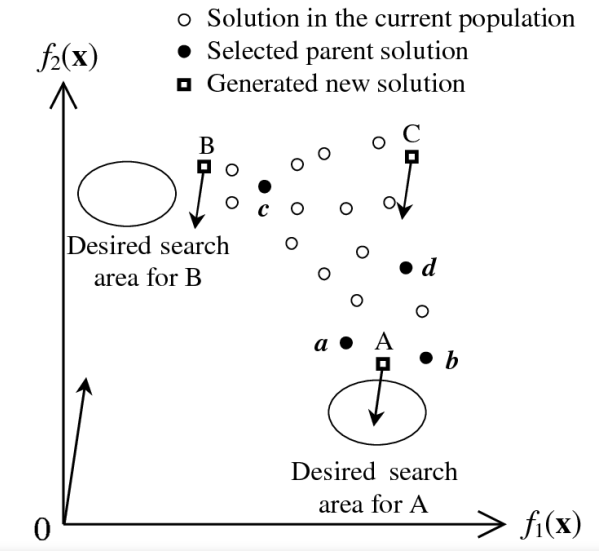


Figure 64: Illustration of the local search direction for parent solutions (Ishibuchi *et al*., 2003).

Examples of MEMOEAs that make use of Pareto dominance relationships during local search include the memetic Pareto archive evolution strategy (M-PAES) presented by Knowles and Corne (2000), which is based on their PAES MOEA and uses Pareto-ranking for selection. Murata *et al.* (2003) also focussed on the use of dominance relationships. They, however, used these relationships for the replacement strategy that governs how locally modified solutions are reintroduced to the active population.

Caponio and Neri (2009) used Pareto dominance as part of the search operator itself. Their cross dominant multi-objective memetic algorithm (CDMOMA) uses Suman's (2004) Pareto dominant multi-objective simulated annealing (PDMOSA) algorithm and the Rosenbrock algorithm to generate local improvements to selected parents. It also uses dominance relationships between solutions of consecutive generations to determine the probability of accepting local movements.

Continuous MOPs lend themselves to another general method of applying local search if their objective functions are differentiable. Hu *et al*. (2003) developed a local search algorithm that makes use of the gradients of the fitness functions. Their method was incorporated into the NSGA-II and SPEA MOEAs and, for both cases, demonstrated faster convergence with no associated loss in solution diversity. Bosman and de Jong (2005) also made use of gradient information for their experimentation on MEMOEAs. They concluded that the use of gradients based on the improvement directions of nondominated solutions is more effective than using the gradients of randomly selected objective functions. However, the objective functions of the LECP are not continuous or differentiable, which eliminates this type of local search as part of a solution approach to that particular MOP.

3.4.2 Coevolution techniques

Traditional MOEAs, such as those reviewed in section 3.2.4, manage individuals that compete as members of a single species. Coevolutionary methods, however, can be introduced to these MOEAs to enhance their performance by dividing the population into *subpopulations* that represent different species of solutions. These subpopulations interact according to various paradigms observed in nature (Daglia and Mirchev, 2020), which results in an exchange of information between populations that is often advantageous (Li *et al.*, 2021).

Coevolutionary strategies evaluate the fitness of a solution in terms of its interactions with members of other subpopulations. Interactions between these individuals could either be negative or positive. Negative interactions are used by competitive coevolution methods, while positive interactions characterise cooperative models (Dorronsoro *et al.*, 2013).

Jiao *et al*. (2012) expanded on this point by presenting it as one of three key considerations when a coevolutionary strategy is designed and integrated with an MOEA. These are:

1. How the total population is divided into sub-populations.
2. How fitness is assigned to individuals within the various subpopulations.
3. How individuals interact with others from other subpopulations.

Paredis' lifetime fitness evaluation (LTFE) method (1995) is an early example of an evolutionary algorithm that uses competitive coevolution. It models a population of *predator* solutions and a separate population of *prey* that interact and coevolve. This relationship is simulated by comparing each population member to random solutions drawn from the other population. The outcome of these encounters is used to assign fitness to the individual, which is subsequently used for parent selection. This method was described by Paredis as "closer to the reality" of natural selection than traditional fitness assignments. He also reported that this algorithm is particularly tolerant of noise due to the averaging out of the noisy fitness evaluations. Although LTFE was developed for single objective optimisation, its predator-prey concept was extended to MOP optimisation by Yao *et al.* (2015) and could therefore be tested on the LECP.

Neef *et al*.'s (1999) elitist recombinative multi-objective genetic algorithm with coevolutionary sharing (ERMOCS) also uses a competitive coevolutionary technique. The pseudocode for this algorithm is provided as Figure 65. It creates a population of *customers* that create niches in phenotype space and a population of *businessmen* that adapt to locate themselves in these regions. The goal is for the businessmen to maximise *profits* by *servicing* the customers.

```
procedure ERMOCS (N, g, f_k(x))
    Initialise customer population C'
    Initialize businessman population B'
    Generate random populations (C' & B') – size N
    Evaluate customer objective values
    Assign business rank based on customers served
    if Termination criteria not met then
        for Each parent in C' do
            Generate C' child population
                Random selection (uniform probability)
                Recombination and mutation
            Assign rank to children based on Pareto dominance with C'
            Assign businessmen to children
            Calculate shared fitness of children
            Compare fitness of children with their parents
            if Child is fitter than a parent then
                Child replaces parent in C'
                Adjust affected businessman fitness in B'
            end if
        end for
        Perform imprint operation with C' and B'
    else
        Stop
    end if
end procedure
```

Figure 65: ERMOCS algorithm pseudocode (Coello Coello *et al*., 2007).

The competing populations of Coello's coevolutionary MOEA (2003) are rewarded with an increase in population size according to their respective number of contributions to the known Pareto front. It specifically aims to search promising areas of the phenotype space while applying a relatively high selective pressure. This relatively simple concept would allow this competitive strategy to integrate well with other common MOEAs for solving the LECP.

Although cooperative coevolution methods also use multiple populations, they operate substantially differently to competitive algorithms. Where the former offer alternative methods of applying selective pressure or diversity preservation, the latter are typically used to decompose a complex MOEA. Cooperative strategies thereby allow the evolutionary process to separately search significantly smaller

solution spaces (Maneeratana *et al.*, 2004). They are therefore well-suited to large-scale optimisation problems (Li *et al.*, 2021). Cooperative methods have therefore received the most research focus as an avenue for addressing problems with large decision spaces (Coello *et al.*, 2020).

The cooperative coevolutionary genetic algorithm (CCGA) developed by Potter and De Jong (2000) provides an apt example of such a strategy. Widely cited as the first meaningful contribution to the area of cooperative coevolution, the CCGA was developed with only single objective optimisation in mind. It subdivides the total population according to sets of decision variables. The members of each subpopulation separately undergo evolutionary operations according to a standard GA. However, they *collaborate* with *representatives* from the other subpopulations to form complete solutions for fitness assignment. Representatives can be selected deterministically as the best members of their respective subpopulations, or stochastically according to their relative fitness values. The LECP's multi-objective nature means that this approach would need to be adapted for the simultaneous consideration of multiple fitness functions.

Figure 66 shows how the CCGA partitions a problem and allows for each population to evolve separately, with fitness assignment, according to a single objective function, occurring on a domain-level. Keerativuttitumrong *et al*. (2002) extended this concept to multi-objective optimisation by integrating the CCGA with the MOGA, giving rise to the multi-objective cooperative coevolutionary genetic algorithm (MOCCGA). Tan *et al.* (2007) expanded upon the MOCCGA by proposing their distributed cooperative coevolutionary algorithm (DCCMEA), which uses an adaptive niching scheme.

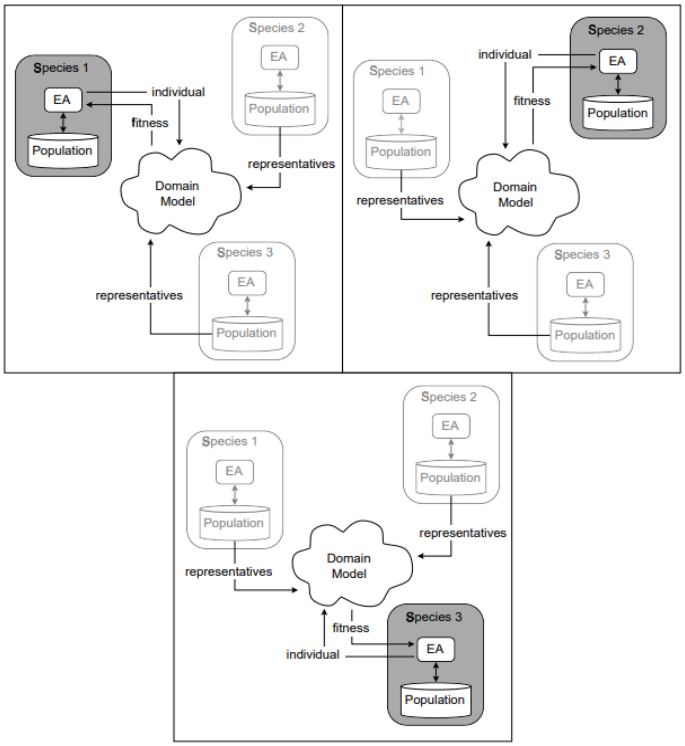


Figure 66: Illustration of the CCGA (Potter and De Jong, 2000).

The CCGA and MOCCGA approaches lend themselves to problems that can be logically decomposed into subproblems that can, in turn, evolve as subpopulations. Li *et* al. (2021) described these problems as separable large-scale optimisation problems (LSOP). The LECP bears this characteristic due to the four distinct cluster sets that could evolve separately, but must combined with representatives as a single solution for evaluation.

The symbiotic coevolutionary algorithm (SCA) proposed by Wallin *et al*. (2005) provides a further example of cooperative coevolution. It models the relationship between hosts and parasites of different species observed in the natural process of endosymbiosis. The SCA models each solution as a *host* that is paired at each generation with a partial solution that represents a *parasite*. The hosts are combined with their parasites prior to evolutionary operations, while the parasite population continues to reproduce asexually with each generation to provide new genetic material and prevent the host population from stagnating. This method applies only to single objective optimisation problems.

The genetic symbiosis algorithm (GSA) presented by Mao *et al*. (2000), however, illustrates a symbiotic approach to MOPs. This method, as shown by Figure 67, uses symbiotic parameters that are calculated by fuzzy logic and inference rules. These parameters describe the type and extent of the relationship between individuals of the same population. Relationships can be competitive, predatory, altruistic, mutualistic, damaging, benefiting or neutral. The symbiotic parameters are used to modify the fitness of the individuals during the execution of the MOEA. GSA makes use of a random search with

intensification and diversification (RasID) to train these parameter calculations during the execution of the algorithm.

```
procedure GSA for MOP (N, g, f_k(x))
    Initialise population P'
    Generate N individuals for population P'
    for i = 1 to g do
        Perform recombination and mutation
        Calculate symbiotic parameters
        Modify fitness with symbiotic parameters
        Evaluate objective values
        Rank individuals in P' based on Pareto dominance
        Generate children
    end for
    if Sufficient learning of symbiotic relation then
        Stop
    else
        Train symbiotic relation with RasID
        Repeat for loop
end procedure
```

Figure 67: GSA pseudocode (Coello Coello *et al*., 2007).

Further cooperative coevolutionary techniques include Parmee's coevolutionary MOEA (1999), which evolves one subpopulation per objective function. This method, however, is not useful for addressing MOPs with obvious extreme points of the Pareto front, such as the LECP. Lohn's coevolutionary genetic algorithm (CGA) and the nondominated sorting cooperative coevolutionary genetic algorithm (NSCCGA) proposed by Iorio *et* al. (2004) provide two additional examples of cooperative strategies.

Coevolution strategies are widely applied (Li *et al*., 2021). Coevolutionary MOEAs, in general, have been described by Jiao *et al*. (2012) as more effective than their standard MOEA counterparts due to their enhanced diversity, while cooperative coevolution provides the added benefit of allowing the simplification of complex problems by decomposition (Antonio and Coello Coello, 2016), such as the separate optimization of the four cluster sets of an LECP solution. This has resulted in greater focus being given to cooperation, as opposed to competition. Tan *et al.* (2007) tested and compared the efficacy of cooperative and competitive coevolution by introducing these strategies to SPEA2. They found that the cooperative approach outperformed competitive coevolution methods, which justifies, to an extent, the favouritism shown in research toward cooperation.

3.4.3 Parallelization

While MOEAs have been proven effective methods for generating good approximations of true Pareto fronts, they are, however, often relatively computationally demanding (Falcón-Cardona *et al*., 2021). MOEAs have been parallelized to mitigate this shortcoming. This has been achieved by taking advantage of the developments made in parallel computing, including processing architectures such as multi-core GPUs, workstation networks, process clusters, grids and clouds (Talbi, 2019).

Talbi (2019) described four goals of adapting MOEAs to make use to these parallel processing architectures and others like them:

1. Reduce the wall clock time required to approximate the true Pareto front.
2. Enhance the properties of the Pareto front approximation, such as convergence and diversity.
3. Solve large MOP instances, such as problems involving big data and databases that cannot be managed on a single processing unit.
4. Enhance the robustness of the MOEA by reducing problem domain knowledge dependencies and sensitivity to parameters.

Practitioners addressing the LECP could potentially benefit from the achievement of any of these goals through parallelising the search of the solution space of their problem instances.

An MOEA that makes use of one or more parallel processing strategies is often referred to as a parallel MOEA (pMOEA) or distributed EA (dEA). The method by which parallelization of an MOEA is achieved is known as the pMOEA or dEA *model*. In most cases, the underlying MOEA solution approach, such as those reviewed in section 3.2.4, is largely unchanged when parallelism is incorporated, although some elements may be introduced or modified to accommodate the parallelization strategy.

Talbi (2019) provided a classification of pMOEA models based on the level of the algorithm hierarchy that is parallelized. Table 2 summarizes this classification. Algorithm-level parallelization involves the simultaneous execution of independent or cooperative MOEAs. The latter alludes to the fact that coevolutionary methods often represent a parallelization strategy. Iteration-level pMOEAs focus on parallelizing the main loop of a single MOEA. Falcón-Cardona *et al*. (2021) proposed a further distinction of this model between *parallelism of population* and *parallelism of mechanism*. The former involves allocation of different subsets of the population to different processors, while the latter refers to the distribution of evolutionary operators to processors. Solution-level pMOEAs parallelize the evaluation of a single solution, either by simultaneously calculating individual objective functions or constraints using different processors, or by decomposing the functions themselves and assigning subfunctions to various processors (Falcón-Cardona *et al*., 2021).

Table 2: Classification of pMOEA models (Talbi, 2019).

Parallel Level	MOP Dependent	Behaviour	Granularity	Objectives
Algorithm	No	Modified	MOEA algorithm	Quality and speed up
Iteration	No	Non modified	MOEA iteration	Speed up
Solution	Yes	Non modified	MOEA solution	Speed up

Falcón-Cardona *et al*. (2021) described six additional attributes of a pMOEA model:

1. **Pareto front**: The management of the Pareto front approximation can be executed centrally or distributed to different processing units.
2. **Decision space**: The decision space can be explored as a single global region or partitioned and explored separately.
3. **Objective space**: The objective space can be explored globally or partitioned.
4. **Nodes**: The processors, or *nodes*, can run homogeneously or heterogeneously configured algorithms.
5. **Distribution**: Nodes can be statically assigned to tasks, subpopulations, etc. or can be dynamically reallocated during the execution of the algorithm.
6. **Communication**: Nodes can exchange information and communicate synchronously or asynchronously.

The master-slave model is an example of algorithm-level parallelization. This model's utility is predicated on the fact that fitness evaluation is often a significant source of computational cost for an MOEA and that parallelization of this process reduces solve time. This approach therefore involves the delegation of this operation to periphery processors while the master node performs the evolutionary operations such as selection, crossover and mutation (Zavoianu *et al*., 2013).

This process speeds up the MOEA's search of the solution space without affecting the search behaviour (Falcón-Cardona *et al*., 2021). The master-slave strategy therefore achieves Talbi's first goal. Figure 68 shows how slave nodes are used by the master processor as fitness evaluation engines while the master manages the MOEA.

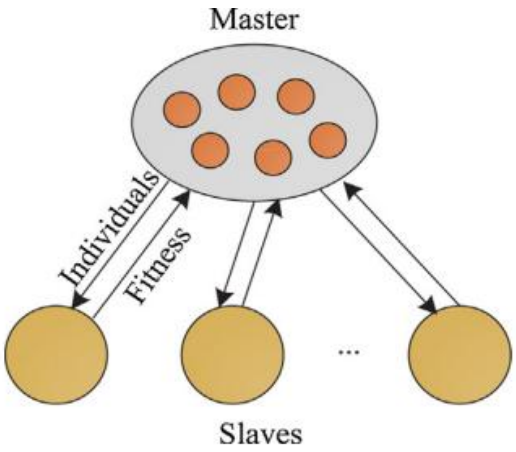


Figure 68: Illustration of the master-slave concept for EAs (Gong *et al*., 2015).

When considering parallelization, careful consideration should be given to the technical overhead associated with opening and closing communication channels between processors (Coello Coello *et al*, 2007). This effort, often referred to as *communication cost*, offsets the efficiency gained by parallel processing and is particularly relevant when master-slave approaches are implemented due to the relatively high frequency of information exchange between the nodes (Gong *et al.*, 2015). This pMOEA model should therefore only be considered for MOPs with relatively large evaluation times (Branke *et al.*, 2004). The relative duration of LECP fitness evaluation would depend on the problem instance and, as such, it would be difficult to anticipate whether the communication cost associated with parallelisation without some empirical testing per problem instance.

The scalability of a master-slave pMOEA is limited by the capacity of the master node. Should this processor reach full utilisation, it becomes the bottleneck of the optimisation process. The addition of slave nodes then serves only to increase communication cost and reduce the overall efficiency of the algorithm (Gong *et al.*, 2015).

The island model allocates a separate population, or *island*, to each processing unit. Each population is then evolved in parallel according to a MOEA solution approach (Gong *et al.*, 2015). The island model is therefore an example of algorithm-level parallelization. It is also, by definition, a coevolutionary technique, as these different populations cooperate by exchanging individuals during the evolutionary process (Branke *et al.*, 2004). This exchange process is referred to as *migration*.

Figure 69 shows the migration of individuals between populations. It can be inferred that the architecture used for the illustrated example includes four separate processing units.

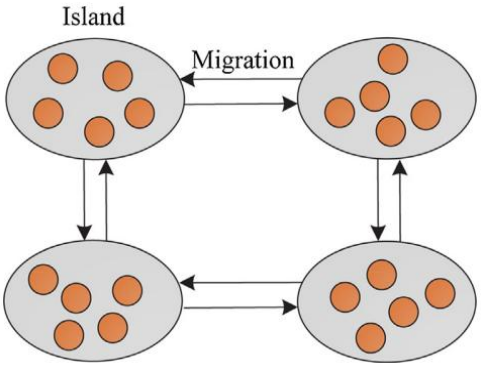


Figure 69: Illustration of the island EA concept (Gong *et al*., 2015).

Migrant solutions can move between islands at set intervals, with all nodes pausing operations until migration is fully complete, or as they are produced and meet the criteria for propagation. The former method is referred to as *synchronous* migration, while the latter is known as *asynchronous* (Kucukyilmaz and Kızılöz, 2018). Gong *et al*. (2015) indicated that synchronous island pMOEAs are easier to implement, while asynchronous algorithms are more efficient. As the LECP would likely be solved on an ad hoc basis with relatively low time sensitivity, the ease of implementation associated with synchronous models would probably make them more desirable to practioners.

Island populations can be evolved according to the same set of parameters, operators, etc., or different MOEA setups. If the former is implemented, the island model is referred to as *homogenous*, while the latter gives rise to *heterogeneous* models. Heterogeneous models are often preferred as the different parameter profiles can be used to balance exploitation and exploration, thereby achieving the second of Talbi's stated goals for parallelization (Gong *et al*., 2015). The opportunity to configure islands differently for the same run of the algorithm also reduces the reliance of the performance of the pMOEA on individual parameter settings, which satisfies Talbi's fourth goal.

The *topology* of the island model specifies the neighbours of each island and therefore maps the migration paths of individuals between populations (Laili *et al*., 2019). Figure 70 presents several common topologies used in island model design. Dots represent islands, while lines represent connections along which selected migrants can transfer.

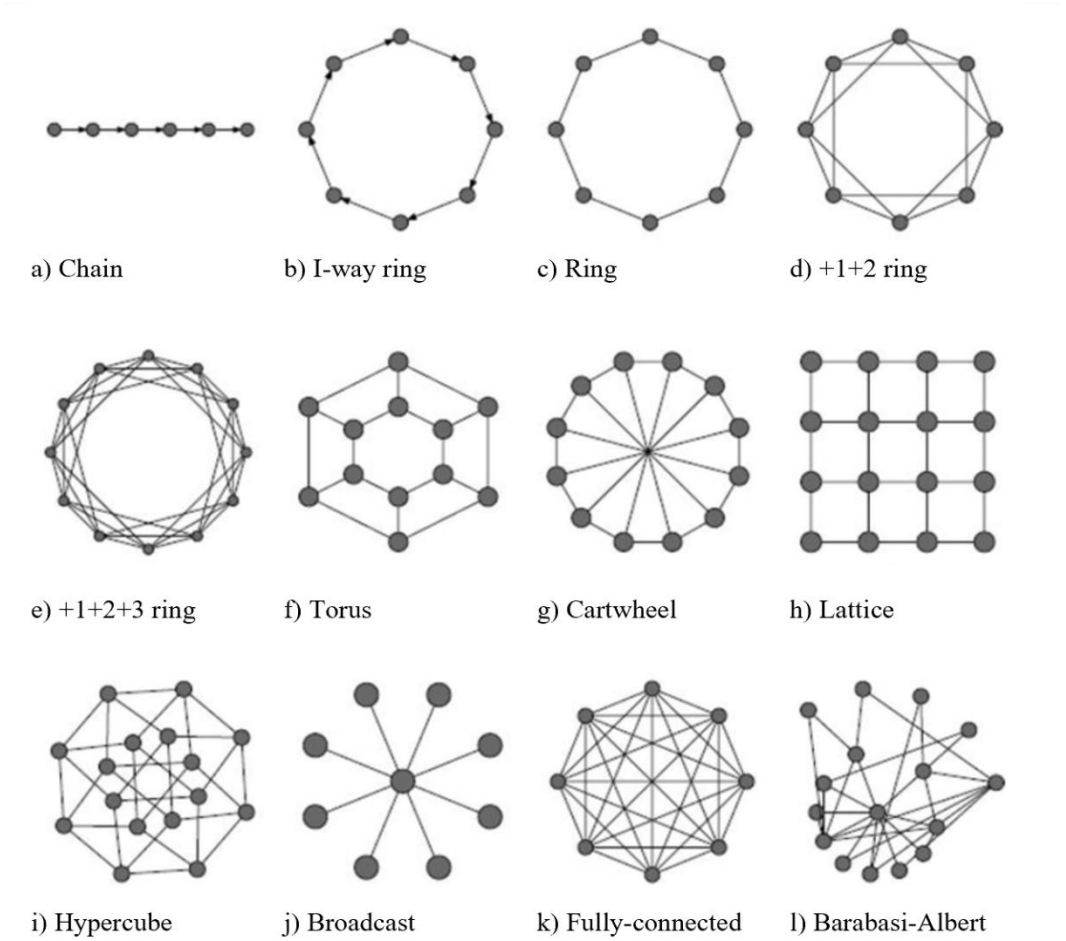


Figure 70: Examples of common migration topologies (Izzo *et al*., 2012).

A migration policy is also a requirement of the implementation of an island model. This includes the method by which migrants are selected from island populations, and the procedure for assimilating these individuals into their new populations. The latter is often referred to as the *replacement policy* (Izzo *et al.*, 2012). The *migration rate* is typically an important parameter used to control the number of individuals that transfer between islands at each cycle.

Cellular pMOEAs also create subpopulations that are assigned to various processors. A grid structure is imposed on the nodes such that each subpopulation has a spatial position relative to others. Neighbourhoods are then defined using a predefined geometry, as shown by Figure 71. Example geometries include square, rectangular and cube shapes (Falcón-Cardona *et al*, 2021). Subpopulations may only interact with nodes positioned within their respective neighbourhoods. However, since the neighbourhoods overlap, good solutions may transfer from node to node and thereby diffuse throughout the entire population (Gong *et al*., 2015). These pMOEAs are therefore also referred to as *diffusion* pMOEAs (Falcón-Cardona *et al*, 2021).

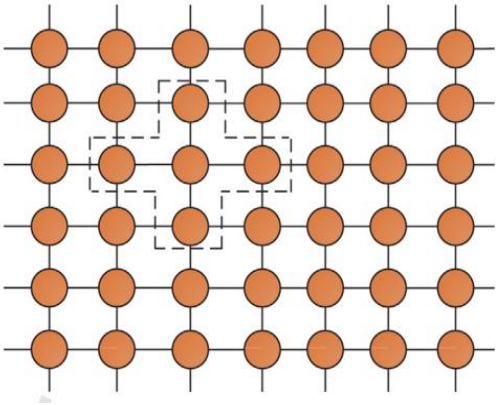


Figure 71: Illustration of the cellular EA concept (Gong *et al*., 2015).

pMOEA approaches have been widely implemented. They have significantly benefited different classes of MOPs from a computational cost perspective (Mishra *et al.*, 2011). Falcón-Cardona *et al*. (2021) therefore recommend that parallelization of MOEAs be explored when large scale problems are addressed or when dynamic problems must be solved in operationally time-sensitive circumstances.

Although solving the LECP can add value in a number of real-world scenarios, few are likely to be time sensitive. The scale of the LECP is specific to the problem instance. However, it can be reasonably assumed that most shippers seeking to simplify their SCN definitions would have sufficiently complex networks and rate cards that the LECP would be considered a large-scale MOP and parallelization would enhance the performance of the chosen MOEA solution approach in terms of convergence time. Algorithm-level models provide the added advantage of mitigating the risk of premature convergence as populations are allowed to evolve in different directions, which experimentation may show to be beneficial for the LECP (Gong *et al*., 2015).

3.5 Concluding remarks

Chapter 3 covered a wide scope of multi-objective optimization concepts and techniques. Many of the theoretical aspects that were reviewed, such as Pareto-optimality and evolutionary methods, would be central to any MOP solving approach addressing the LECP. Others, such as final MOC solution selection, are not directly applicable to the evolutionary approach developed for this thesis, but would, nonetheless, be useful to consider when applying the approach in practice.

The diversity of MOEAs and auxiliary strategies that could be applied to the LECP is such that methodical integration and testing of all alternatives would be impractical for this thesis. This literature review, however, did give a view of further methods that could be applied by operations researchers looking to further optimize results. Chapter 4 addresses how five of these reviewed solution approaches were integrated into a general LECP evolutionary approach and then tested.

Chapter 4

Methodology

In this chapter, the general approach outlined in section 1.3 is presented in further detail. Although attention is given to the end-to-end research process from the development of the LECP formulation to the conclusion, focus is given to the experimental component of this research. In so doing, this chapter addresses research objective 3.1.

This research methodology is based on the thesis objectives and business problem outlined in chapter 1. The research approach allows reference to theoretical concepts discussed in the literature review, while additional literature not included in the review may be introduced to support an explanation or discussion in this or the subsequent chapters as deemed appropriate.

4.1 Research process

Figure 72 illustrates how the methodology was designed to address the supporting objectives of this research. A full mathematical formulation of the LECP as described in section 1.2.1 was developed. This included mathematical expressions of the decision variables, objective functions and constraints that describe the LECP. The formulation, which is presented in chapter 5, represents the final component of the first research section as shown in Figure 7 and addresses the first supporting objective.

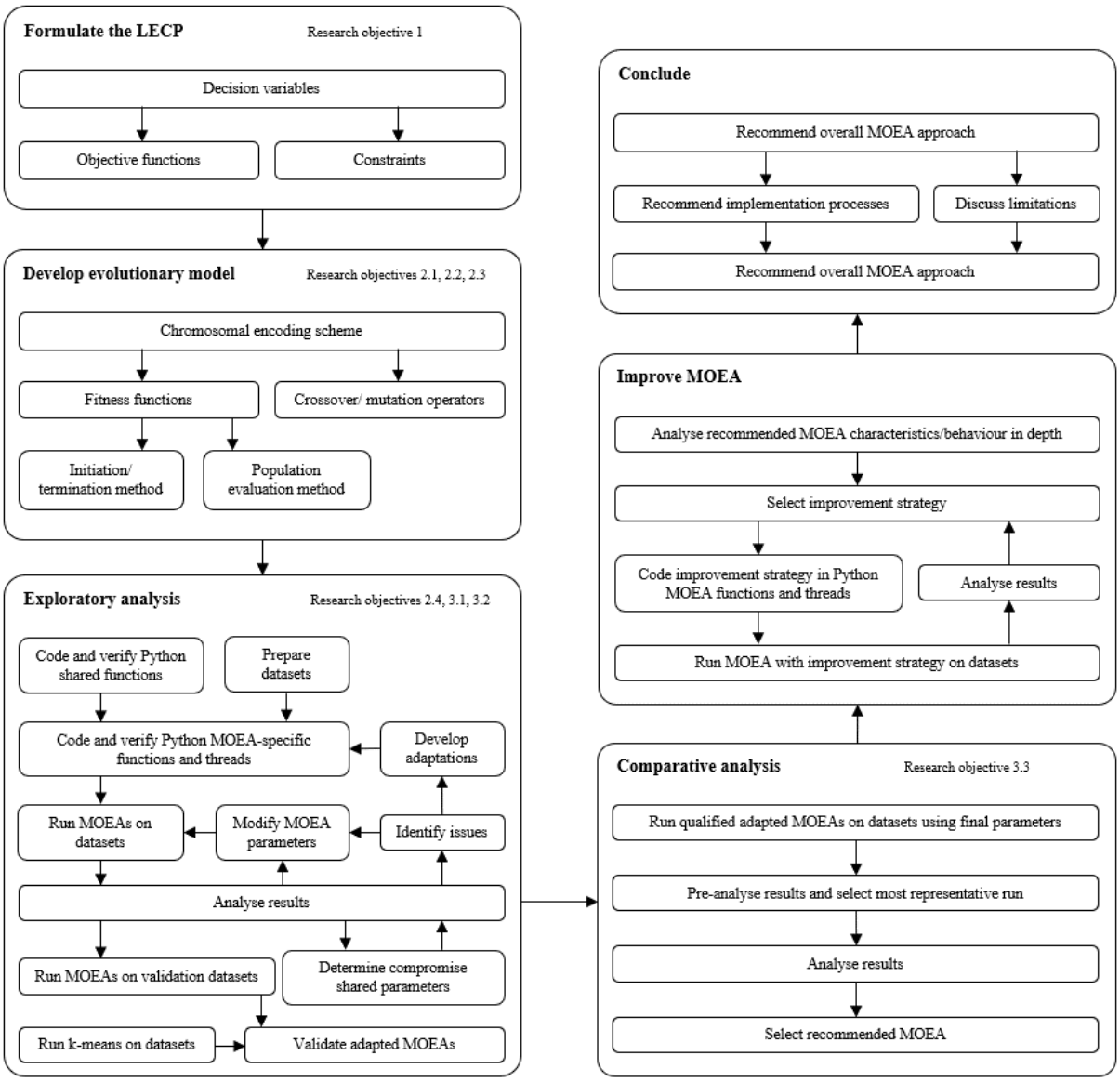


Figure 72: Research methodology.

Due to the multi-objective nature of the LECP, several objective functions were developed. These were categorised as primary and secondary objective functions. The set of primary functions provide an unambiguous definition of the fundamental trade-off of this MOP. The secondary objective functions are derivations of the primary functions that were developed for this research as suggested adaptations or additions to the LECP. These account for anticipated DM preferences that may be encountered in practice. Only the primary functions were used for the experimental component of this research to retain the most neutral view of the problem as presented in the research scope and avoid the proliferation of MOP dimensions. Further assumptions and methods that were used to develop a single formulation for the LECP are presented as commentary alongside the formulation itself.

The scope of this thesis imposes an evolutionary method of solving the LECP, as well as the specific MOEA solution approaches to be used for the experimental component of the research. The approaches

were selected by following the recommendation of Coello Coello *et al.* (2007), who suggested six MOEAs as suitable candidates for initial investigations into new problem domains[7]. This methodology therefore does not include any processes for the selection of solution approaches for testing. Nonetheless, it recognizes that MOEAs, as defined in literature, often only include high-level methods for handling multiple objectives and populations during the execution of an EA. MOEA designs do not typically address lower-level operations that should incorporate problem domain-specific considerations, such as encoding and decoding of solution chromosomes, quality measurement, population initiation, constraint handling, crossover and mutation, and algorithm termination. These procedures were individually designed or adapted for the LECP, encoded, verified and validated.

The evolutionary operators, once developed for the LECP in accordance with the mathematical formulation, were deemed transferable between MOEAs with minimal need for adaptation. As such, they are presented upfront in chapter 6 as an overarching MOEA-agnostic evolutionary approach to the LECP.

The general evolutionary approach, as presented in this thesis, was then applied using the in-scope MOEA solution approaches. This included the design of processes for integrating the generic LECP operations to each MOEA as presented in literature. The EAs were then coded and executed with real-world data. The procedure underpinning these activities, as well as the analysis of the collected output data, constituted the experimental component of this research and are explained in detail in sections 4.2, 4.3 and 4.4. The selection, development, experimentation and evaluation of MOEA improvement techniques was also considered part of the experimental research and is presented as such. The output of this component of the research is therefore a single recommendation in terms of an MOEA, with or without integrated improvement strategies, with accompanying commentary.

4.2 Investigation of solution approaches

4.2.1 Coding language

evolutionary algorithm developed for this research was tested using the five in-scope solution approaches and three sets of real-world data[8]. In so doing, research objective 2.4 was satisfied. Python was selected as the programming language for ingesting this data and executing the MOEAs.

Python bears several advantages over compiled languages, such as C, C++ or Java, for the implementation of evolutionary algorithms. Kim and Yoo (2019) presented a number of these benefits,

[7] One MOEA, the Micro-GA, was excluded due to the relatively large requirement for parameter tuning.

[8] The scope of this data, as well as the data management procedure, is outlined in detail in section 4.3.

including the ability to execute evolutionary operators as function calls[9]. They also cited the freely available Python data processing libraries as advantageous in this context. Function libraries such as pandas, NumPy and PyMoo were used extensively for the experimental component of this research. The data used for developing and validating the applied algorithms were maintained as comma separated value (csv) files and accessed by the Python scripts during optimisation runs. Input data was read from, and output data written to, csv files using the csv Python library.

4.2.2 Exploratory analysis

The collection and analysis of data for the experimental component of this research was conducted in two phases, namely the *exploratory* and *comparative* analysis phases. The initial exploratory phase followed an unstructured, nonlinear approach. Each MOEA was run on a single shipper dataset to explore the behaviour of the evolutionary processes[10]. Although specific parameter recommendations are not explicitly included as an objective of this research, sensible values were determined for any MOEA-specific parameters through this experimental process, as recommended by Kazimipour *et al.* (2014), to provide each solution approach a reasonable opportunity to perform well when applied to the LECP using the three datasets and compared to the other approaches.

This exercise yielded important insight into behavioural peculiarities of each MOEA when applied to the LECP. It also facilitated the detection of performance barriers inherent to the specific combinations of problem, algorithm and data domains that this experimentation produced. Discretion was used to determine whether these barriers represented valid limitations of interest that should be retained for the subsequent phase of experimentation, or if these shortcomings were fundamentally destructive to the applicability of the solution approach to the LECP and the meaningfulness of the comparative analysis. If deemed the latter, a remedy was developed and integrated with the algorithm on the condition that the corrective mechanism did not fundamentally change the MOEA process.

Only "MOEA-neutral" adaptations were considered. Strategies that are presented in literature as a feature of another MOEA could not be implemented for other solution approaches. Any adaptations that modified the shared evolutionary functions or contradicted the principles that underpin the relevant solution approach were also disqualified for implementation and testing. Once a correction was applied, the adapted MOEA formed the basis for all subsequent experimental activities.

[9] Section 4.2.3 outlines how the evolutionary operators, as designed for the evolutionary approach to the LECP, were coded as functions.

[10] Although exploring the algorithms' behaviours using all available problem instances would add depth and confidence to the findings, this benefit was deemed disproportionate to the required additional experimentation and data analysis.

Furthermore, the exploratory process simultaneously allowed for coarse tuning of parameters. These parameter values were also used to individually benchmark each adapted MOEA against the k-means clustering algorithm, which did not require these parameters. If a solution approach produced more desirable results than the k-means method, as defined by the population evaluation method of the evolutionary approach of the LECP, it was accepted as a viable solution to the LECP and qualified for the comparative analysis phase. The exploratory analysis exercise thereby satisfied research objective 3.2.

4.2.3 Comparative analysis

Each of the five encoded MOEAs were run multiple times using each shipper dataset and the corrective mechanisms and parameter values finalised as part of the exploratory phase of testing. The best, worst and most representative runs for each MOEA and dataset were determined according to each of the population evaluation indicators and subsequently visualised and compared. The definition and method of identification of a most representative run was deferred to the analysis exercise of chapter 8 as this was expected to be influenced by the statistical characteristics of the output data.

A single MOEA was then selected as the recommended solution approach to the LECP. The limitations and assumptions associated with this selection that surfaced during the analysis are discussed alongside the recommendation and explained in further detail in the conclusion of this thesis.

4.2.5 MOEA improvement

The recommended solution approach, including any adaptations developed during the exploratory analysis, as well as the parameters used for the comparative analysis, was subsequently used as the baseline algorithm for testing the effectiveness of general MOEA enhancement techniques. The inherent characteristics of the MOEA and the LECP, as well as the behavioural traits of the algorithm observed during the previous testing with the three shipper datasets, were considered for the selection specific improvement techniques. Candidate improvement techniques were limited to parallelization and coevolutionary methods as presented by Coello Coello *et al.* (2007) based on *a priori* understanding of the LECP and the literature review.

Parallelization methods were selected and sequentially incorporated into the recommended MOEA Python code. The algorithm was run multiple times to solve each of the three problem instances after the introduction of each enhancement. The generated data was then compared to the performance data collected during the preceding comparative analysis.

The Python *threading* module was used to initiate four separate threads of the adapted NSGA-II MOEA. These threads run in parallel and maintain their own local current and archive subpopulation dataframes and variables, such as fitness evaluation and duplicate solution counters.

However, upon the completion of a generation of a thread's EA and the evaluation of its own archive, it subsequently wrote the archive to a dedicated Excel .csv file. The thread then read the latest versions of the remaining three threads' files, merges the four archive dataframes and performed a population evaluation process on this combined set of solutions. This included dominance ranking and removal of dominated solutions to yield a set of all nondominated solutions yet found by all threads. The quality metrics for the combination archive were logged separately such that the progress of the overall Pareto front approximation PF_{known} was tracked alongside each thread's own independent progress. The number of fitness evaluations recorded with the combined population HV and HV_d was equal to the local fitness counter value of the thread that prompted the aggregated archive assessment. The overall performance of the pMOEA was therefore tracked and reported such that the efficiency of parallel use of processing nodes is reflected.

Figure 73 shows the topology of this model. The solid black nodes represent the subpopulations managed by each thread, while the white node indicates the combined archive. Nondominated solutions from each thread are transferred inwardly at the completion of each generation for possible inclusion in the combined archive.

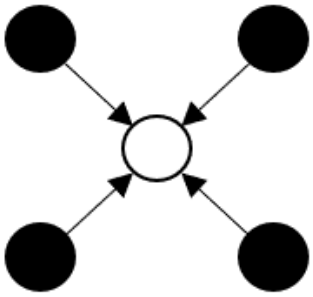


Figure 73: Topology of the simple LECP pMOEA (author's own illustration).

This simple pMOEA model, which was the basis of all the parallel approaches used to solve the LECP for this research, could therefore be described using Falcón-Cardona *et al.*'s (2021) six attributes as follows:

1. Pareto front: Each processing unit managed a Pareto front approximation separately, although a unified PF_{known} was temporarily created for evaluation of the pMOEA's overall progress.
2. Decision space: The decision space was explored as a single global region.
3. Objective space: The objective space was explored globally.
4. Nodes: The processors, or *nodes*, ran homogeneously configured algorithms.

5. Distribution: Subpopulations were statically assigned to nodes.
6. Communication: Nodes exchanged information and communicated asynchronously. However, this exchange was limited to the combination and evaluation of the nodes' Pareto front approximations. Data transfer therefore did not affect the behaviour or direction of the separate algorithms' searches of the decision space.

Three variations of this pMOEA model were developed and tested. They are described in this section in the sequence of their developmental evolution. As such that it could be assumed, that each incorporated the design complexities of the preceding models. The effect of the parallelism introduced to the adapted NSGA-II model was evaluated in terms of efficiency and effectiveness. Efficiency was measured by the number of fitness evaluations required by the pMOEA to achieve the same or greater degree of convergence to the true Pareto front PF_{true} as that achieved by the best run of the serial MOEA after the same number of fitness evaluations during the comparative testing. Effectiveness was measured as the difference between the extent of convergence to PF_{true} achieved by the pMOEA and that achieved by the same best run from the comparative analysis.

Variation 1 - pMOEA with heterogeneous starting populations

The evolutionary cycle of this variation is identical to the base pMOEA. The initiation process, however, involved the generation of four unique subpopulations. This strategy thereby enhanced the genetic diversity with which the pMOEA began its process.

Variation 2 - pMOEA with migration

This variation of the NSGA-II pMOEA solution approach to the LECP included an additional process at the completion of each generation of each thread's evolutionary cycle. A percentage of individuals from the archive of another randomly selected thread were added to a thread's current population. This was accessed as a .csv population file saved by the other thread at the conclusion of its last generation.

Figure 74 illustrated the migration pathways this island model. Nondominated solutions of each subpopulation could be retained in the central combined archive after being inwardly transferred. A sample of archive individuals, originally members by any of the four subpopulations, were subsequently transferred outwards to the threads as migrants upon completion of each island generation.

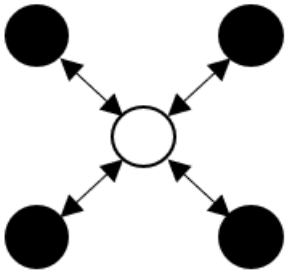


Figure 74: Island topology of variation 2 of the LECP pMOEA (author's own illustration).

This asynchronous communication thereby introduced randomized migration to the model and exposed each subpopulation to high-quality genetic material found by other threads. This model therefore differed from the simple pMOEA in terms of Falcón-Cardona *et al.*'s (2021) sixth attribute, as this asynchronous migration process was expected to influence the behaviour of each thread. A migration rate of ten percent was chosen for the tests conducted for this research.

Variation 3 - Heterogeneous pMOEA

The adapted NSGA-II MOEA was executed in parallel while each thread made use of a different set of parameters. These parameters included the primary and secondary mutation rates, as well as the cluster-blend and cluster-swap crossover probabilities. The initial population size could also be varied between threads, although this was not tested in this research.

4.3 Experimental data

4.3.1 Data sources

The input data for the experimental component of this research consisted primarily of the rate cards of three different shippers that manage their own full truckload (FTL) transportation using the same transportation management system (TMS). This rate card data was extracted from the TMS and included the following fields:

- Carrier name. Company names were obfuscated with anonymous labels to protect the identity of companies trading with the shippers.
- Origin name, defined either as an individual stop location, city or existing cluster as it is defined on the shippers' TMS payable rate master data.
- Destination, defined either as an individual stop location, city or cluster.
- Payable amount, defined as a flat price in South African rand excluding VAT. Rate values were multiplied by a single randomly-generated factor to distort the market pricing information of the dataset while maintaining the relative differences between rates.

The data was extracted in July 2022 and was therefore not current at the completion of the research.

Table 3 summarises the anonymized shipper datasets used for testing and validating the solution approaches. All rates defined using specific facility locations were excluded from the test data under the assumption that they represented specific shipper-carrier agreements that would have remained active and unaffected by any lane aggregation.

Table 3: Shipper datasets. [11]

Shipper	Carriers	Active Origin Cities	Active Destination Cities	Rates (Regionalized)	Rates (Deregionalized)
A	66	25	613	5 976	12 640
B	25	193	317	7 849	283 657
C	61	132	337	8 116	416 665

Rates defined using larger geographic constructs than cities (i.e. regions and provinces) were deregionalized. This involved duplicating the rate record for each city within the applicable region or province definition. This represented a reversal of the manual clustering process already performed by the respective shippers on the data such that the tested algorithms were presented with an entirely "un-clustered" set of rates to a city level. As such, it was inferred that datasets that exhibit a high ratio of deregionalized rates to original regionalized rates, such as shipper C in Table 3 provide an MOEA a rate card with high degree of rate uniformity by geographic area and a greater opportunity to find clusters that eliminate lanes with minimal loss of pricing information. This assertion was tested during the exploratory analysis of the experimental component of the research process.

Although the LECP requires that the clustering of SCN nodes must primarily consider transport pricing and lane elimination opportunities, there are aspects to the problem outlined in section 1.2.1 that also require consideration of the physical location of the nodes. These aspects include the separate treatment of local and long-distance lanes, as well as the requirement to cluster nodes without producing geographical overlap. As such, the longitude and latitude of each node were also required for solving the LECP and therefore needed for the experimentation component of this research.

Although some applications of the LECP may encounter the predominant use of specific locations or "street-level addresses" to define nodes in the rate card data, the nature of primary transport results in rates often being defined primarily on a "city-to-city" level. As this was the case for the three problem instances of this research's experimentation, a single representative longitude and latitude was assigned

[11] Approval was given to use the data for this research under conditions of anonymity. The raw data has not been made publicly available.

to each city node. This ensured an approach that could be consistently applied and avoided the complexities of managing polygons as nodes.

As such, names that describe large geographic areas were assigned single sets of longitudes and latitudes that are representative of the area using the geocoding engine of the shippers' TMS. This engine used the node name to determine the most accurate available set of coordinates. As such, it should be noted that the rate was applicable to transport picked up or dropped off anywhere within the commonly accepted area of the node. It also follows that a rate may be defined using the same description for the origin and destination node (e.g. a "Johannesburg to Johannesburg" rate). The distances of these lanes are uncertain as they are highly dependent on the exact pick-up and drop-off locations of the transport. However, it is usually understood that the rate covers any transport within the commonly accepted boundaries of the node description. It is also noteworthy that any calculated lane distance using longitudes and latitudes would also be zero for these special types of lanes.

These coordinates were also used to develop a straight-line distance matrix for the combined set of origin-destination pairs present in the shippers' rate cards. As such, the local and long-distance designation of rates was inferred from an existing dataset. This matrix was retained as a static input dataset to alleviate the computational burden of generating the matrix with each run of an algorithm.

4.3.2 Data scrubbing

Data scrubbing is generally regarded as a critically important step of optimisation exercises. The usefulness of any approach that solves the LECP, however, lies in the efficient and objective simplification of raw rate card data and resilience to outliers. It was therefore deemed desirable for this research to fully simulate this real-world situation to test each MOEAs effectiveness in this regard.

Activities involve the cleansing of LECP rate cards, such as the manual inspection of the rates to standardise and align origin and destination descriptors, or exclude unusually high or low rate values, would bear resemblance to the subjective manual lane elimination cluster exercise the mathematical approach to the LECP aims to replace. Furthermore, the categorization of extreme rate values as outliers is particularly problematic. The background of this thesis establishes that valid unusual values arise due to varied and complex transport pricing considerations that would not be reflected in or reliably inferred from the data. The exclusion of unusually high or low rates from the dataset would therefore not only be subjective, but also fundamentally incorrect if an onerous rate verification process with the shipper and carrier is not followed.

While it may be appropriate under certain circumstances for a practitioner to subjectively perform some rate cleansing to prevent algorithms from identifying clusters based on obviously incorrect data points,

it is not possible in an academic context to determine the degree to which the data would be cleansed in all cases. Furthermore, the operational and commercial importance of rate cards can generally be deemed sufficient to ensure that this data is maintained accurately at its source.

As such, to ensure an objective and consistent approach for the experimental component of this research, the rate card data was intentionally excluded from any cleansing process. For example, the names "Johannesburg" and "Joburg" can be considered to refer to the same geographical location. However, these values were not normalised if and when detected as rate destinations within the same dataset. These were therefore managed as distinct SCN nodes by the algorithms to be clustered or left unclustered. As such, each MOEA's resilience to data quality and outliers was also tested.

An auxiliary benefit of this approach was that the retained data anomalies provided further opportunities for the techniques to identify clusters based on transport pricing considerations. This allowed for more distinct differentiation of MOEAs on the basis of their ability to find such clusters. For example, as rates for transport to both "Johannesburg" and "Joburg" would be expected to be similar, the retention of their original names in the dataset preserved a point in the decision space where substantial improvement in objective space could be achieved by an MOEA. Normalisation of the data in this regard would forfeit these enriching elements of the dataset. This, in turn, would result in more homogeneous output generated by the various techniques and decrease the probability of finding statistical significance in the comparison of the MOEAs' performance. Similarly, an erroneously large or small rate value within the data would provide a further differentiator in terms of algorithm's resistance to the clustering of adjacent nodes that exhibit substantial economic peculiarities that are common in real-world scenarios, albeit exaggerated by these data.

The accuracy of the SNC nodes' geographic coordinates, as determined by the TMS geocoding engine, was assumed to be less critical to the shippers' operations and therefore more prone to quality issues than the rate card data. Furthermore, unlike rates, these longitudes and latitudes were not the subject of the decision space search, but rather used for the secondary processes of pre-categorization of rates and constraint checking. The impact of outliers in this data on the validity of the experimental results was therefore expected to be entirely negative and largely avoidable. As such, the coordinates were subjected to a scrubbing exercise using geographic visualisations of the longitudes and latitudes and manual inspection.

The scope of the sample shippers' transport operations included only road transport. This ensured that the MOEAs would detect pricing similarity across a similar vehicle type, as variations in the stated payload capacities of these vehicles on rate cards typically do not indicate any variation in value of the transport services. These variations typically only indicated the expected pallet stock density for the

shipper and lane combination, as transport of loads exceeding 30 tons was predominantly executed using non-refrigerated articulated tautliner vehicles with 36 standard pallet spaces.

4.4 Evaluation criteria

The literature review established that the effectiveness of an *a posteriori* approach to an MOP is typically determined by evaluating the quality of the approximation of the Pareto optimal front *PF* found upon termination of the algorithm. The research methodology supports the notion that the characteristics of the MOP should be considered when the quality measures are selected or developed. As such, the selection of evaluation criteria for the experimental component of this research, insofar as the efficacy of the solution approaches is concerned, is discussed as part of the evolutionary approach to the LECP as the population evaluation method.

Although analysis in terms of the elapsed computational run time is not a stated objective of the research, the primary objective of this thesis explicitly calls for real-world modelling aspects to be considered. CPU time is therefore considered for some MOEA design and parameter tuning decisions. As such, it is worth noting that tests were performed using an 11th Gen Intel(R) Core(TM) i7-11390H @ 3.40GHz 2.92 GHz processor with 16 GB RAM and a Windows 11 Pro operating system.

4.5 Concluding remarks

Chapter 4 described the approach to the experiment component of this research. Specifics regarding coding languages and data management were discussed, as well as a testing process that allowed sufficient flexibility to pivot, in moderation and only when required, based on empirical observations made of the behaviour of the algorithms.

The prior development of a clear formulation of the LECP, which is included in this thesis' scope, would be fundamental to any valid attempt to solve this MOP, and was therefore also included in the process. The formulation is covered in Chapter 5. However, the methodology focussed on how existing MOEAs would be integrated with a single evolutionary approach to the formulated LECP and then tested on three real-world case studies. The methodology described in Chapter 4 is therefore more recognizable in Chapters 6, 7 and 8, which cover the general evolutionary approach and the two phases of testing.

Chapter 5

The lane elimination clustering problem formulation

In this chapter, the optimisation problem as described in section 1.1 is fully mathematically formulated for the scenario outlined in section 1.2.1. A new MOP in the field of transportation management is thereby introduced. The decision variables, object functions and constraints are discussed separately. This chapter thereby addresses research objective 1 of this thesis. Table 4 provides an inventory of symbols used in expressions of the subsequent formulation.

Table 4: Summary of LECP formulation symbols.

Symbol	Description
s	Cluster set
i	Origin node
j	Destination node
G_i	Set of longitude and latitude coordinates associated with origin node j
G_j	Set of longitude and latitude coordinates associated with destination node k
k	Carrier
x_{si}	Cluster to which origin node i of cluster set s is assigned
y_{sj}	Cluster to which destination node j of cluster set s is assigned
D	Distance break value (in the selected distance UoM) for classifying rates as local or long-distance
X_{sp}	Set of origin nodes assigned to cluster p in cluster set s
Y_{sq}	Set of destination nodes assigned to cluster q in cluster set s
$G_{X_{sp}}$	Set of longitude and latitude coordinates associated with nodes of cluster set X_{sp}
$G_{Y_{sq}}$	Set of longitude and latitude coordinates associated with nodes of cluster set Y_{sq}
c_{ijk}	Payable rate value quoted by carrier k for transporting freight from origin node i to destination node j
x'_{ij}	Cluster to which origin node j is assigned when considering the distance of lane ij
y'_{ij}	Cluster to which destination node k is assigned when considering the distance of lane ij
d_{ij}	Distance (in the selected UoM) between nodes i and j
f_1	Lane elimination ratio

L	Set of original lanes (i.e. unique combinations of origin nodes i and destination nodes j)
L'	Set of aggregated lanes after clustering
f_2	Cost penalty ratio
c'_{ijk}	Replacement rate value (i.e. maximum original rate value for carrier k for lane i to j within a group of aggregated lane rates)
ϕ_{ijk}	Cost of clustering (i.e. amount of increase of the rate value for carrier k for lane i to j due to lane aggregation)
f_3	Prime-only cost penalty ratio
$e(c'_{ijk})$	Rank of the clustering replacement rate r^c_{ijk} per lane i to j in ascending order
n	Prime rate definition
C_{ijk}	Rank adjusted rate
f_4	Cost penalty equity
$h^x(G_i, G_{X_{sp}})$	Number of intersections of the horizontal ray emanating from a node i with the concave polygon line segments defined by origin cluster X_{sp}
$h^y(G_j, G_{Y_{sq}})$	Number of intersections of the horizontal ray emanating from a node j with the concave polygon line segments defined by origin cluster Y_{sq}

5.1 Decision variables

The LECP is concerned with the grouping of origins and destination nodes of an SCN. A full clustering solution for this problem requires four individual sets of clusters. These are referenced by the objective functions during evaluation according to certain logic. For this formulation, the cluster set is denoted by s. The meaning assigned to each set value is defined in Table 5.

Table 5: Cluster sets.

Set(s)	Node type	Distance category
A	Origins	Local
B	Origins	Long-distance
C	Destinations	Local
D	Destinations	Long-distance

Each cluster set is defined as a list of integer values. These four lists, when combined, represent the majority of individual decision variables of this MOP. Each position in the list represents a specific node of the SCN. The length of an *origin* cluster set is equal to the number of distinct origin descriptions in the rate card, while the length of *destination* sets equals the number of destinations in the rate card. As such, the extent of the decision variables is dependent on the instance of the LECP.

A repeated value in the list indicates that the corresponding nodes are assigned to the same cluster. The specific values of the integers are therefore not significant. Rather, it is the repetition of values and their positions in the list that indicate clustering structures and thereby bear significance. This method of defining the clusters simultaneously enforces the constraint of mutual exclusive assignment of nodes to clusters. It can also be viewed as a cluster label-based solution representation that directly translates to a simple chromosomal representation of the solution.

The number of unique integer values assigned to these variables is not stipulated or constrained. The number of clusters is therefore not a parameter of the problem, which is often the case for existing clustering algorithms such as the k-means method (Nanjundan, 2019).

A further decision variable is required to define the distance bracket value to be used by the objective functions to classify rates as local or long-distance during the lane elimination process.

We define these decision variables

x_{si} = the cluster to which node j of cluster set s is assigned $\quad \forall\, s \in \{A, B\}\,, \forall\, i \in I$ (5.1)

y_{sj} = the cluster to which node k of cluster set s is assigned $\quad \forall\, s \in \{C, D\}\,, \forall\, j \in J$ (5.2)

D = distance break value (in the selected distance UoM) for classifying rates as local or long-distance (5.3)

$$x_{ab} \neq x_{cd} \qquad \forall\, a, c \in \{A, B\}\,, \forall\, b, d \in I \tag{5.4}$$

$$y_{ab} \neq y_{cd} \qquad \forall\, a, c \in \{C, D\}\,, \forall\, b, d \in J \tag{5.5}$$

where $x_{si} \in Z$, $y_{sj} \in Z$ and $D \in R^2$.

I and J represent the set of unique origin and destination nodes described by the shipper rate card respectively. Expressions (5.4) and (5.5) ensure that the integers chosen to represent clusters are not repeated between local and long-distance cluster sets. This prevents subsequent lane counting operations treating two independent clusters as a single cluster due to their sharing of the same cluster integer.

The discreteness of the decision variables and, by extension, the decision space, means that the LECP can be classified as a combinatorial problem (Papadimitriou, 1998).

The variables x_{si} and y_{jk} give rise to $|x_{si}|$ and $|y_{skj}|$ sets of clusters. We define these consequential decision variables

$$X_{sp} := \{i: x_{si} = p\} \qquad \forall\, s \in \{A, B\}\,,\ \forall\, i \in I, \forall\, p \in \{x_{si}\} \tag{5.6}$$

$$Y_{sq} := \{j: y_{sj} = q\} \qquad \forall\, s \in \{C, D\}\,, \forall\, j \in J, \forall\, q \in \{y_{sj}\} \tag{5.7}$$

5.2 Objective functions

Coello Coello *et al.* (2007) stated that the number of objective functions is often limited only to the imagination of the algorithm designer. This arguably applies to the LECP, as the utility of a clustering solution to an analyst seeking a simplified SCN lane structure can be measured according to nuanced metrics. It therefore follows that a myriad of objective functions can be formulated for this optimisation problem and implemented. Coello Coello *et al.* (2007) added that these objective functions should be selected according to the specific context in which the MOP is being solved. However, this research does not refer to any single context and, as such, a precise set of objective functions cannot be exactly defined for all LECP applications.

Nonetheless, Coello Coello *et al.* (2007) also recommended that "MOEA application to a given MOP should begin with two or three primary objectives in an effort to gain problem domain understanding". This approach was taken for this LECP formulation. As this problem is primarily concerned with the trade-off between the reduction of lanes of a defined SCN and the cost associated with this simplification, these metrics, defined in their simplest form, were selected as the main objectives of the problem. Their formulations are therefore presented as primary objective functions, with additional suggested functions, which are not used for experimentation for this thesis, formulated as *secondary* objective functions. The secondary objective functions presented here include a variant of a primary function when only a subset of cheapest rates is evaluated, called the prime-only cost penalty ratio.

The shipper's rate card is central to the evaluation of each clustering solution as it describes the existing SCN lanes and specifies the carrier prices affected by clustering. In this formulation, the rate card is transformed from a table structure with four columns (namely "Carrier", "Origin", "Destination" and "Rate") to a variable described with three indices. This variable is defined as c_{ijk} ($c_{ijk} \geq 0$), where the payable rate value c was quoted by carrier k ($k \in K$) for transporting freight from node i ($i \in I$) to node j ($j \in J$).

It follows that $I \times J \times K$ rate values are defined, which would tend to be higher than those described by the shipper rate card. This is due to carriers not necessarily servicing or providing rates for all lanes, as the fact that freight seldomly moves between each origin and destination of a SCN. However, where a rate does not exist for carrier k for lane ij, a zero should be assigned as the c_{ijk} value.

The domains of rate values are typically limited by the transacting currency. For example, all rates in the South African context should be rounded to the nearest hundredth of a South African Rand (i.e. to the nearest cent). c_{ijk} is therefore expressed in the discretized lowest available unit of currency and is limited to the domain of natural numbers ($c_{ijk} \in N$).

The objective functions may represent the process of lane elimination given the clustering solution provided as the decision variable values. To achieve this, two additional variables, x'_{ij} and y'_{ij} are derived from the decision variables x_{si} and y_{sj}. These represent the alternative origin and destination descriptors for every lane (i.e. the clusters to which the origin *i* and destination *j* are assigned per lane *ij*). The selection of the specific cluster as the alternative node descriptor depends on the physical distance of the lane *ij* relative to the distance break decision variable *D*. A further variable d_{ij} was therefore introduced to specify the distance (in the selected UoM) that must be travelled to transport freight from origin *i* to destination *j* for this comparison[12].

We define these variables

$$x'_{ij} = \begin{cases} x_{Ai} \ if \ d_{ij} \leq D: \ \ i \in I \wedge j \in J \\ x_{Bj} \ if \ d_{ij} > D: \ \ i \in I \wedge j \in J \end{cases} \quad (5.8)$$

$$y'_{ij} = \begin{cases} y_{Cj} \ if \ d_{ij} \leq D: \ \ i \in I \wedge j \in J \\ y_{Dj} \ if \ d_{ij} > D: \ \ i \in I \wedge j \in J \end{cases} \quad (5.9)$$

d_{ij} = travel distance (in the selected distance UoM) from origin *i* to destination j

$$\forall \ i \in I, \forall \ j \in J \quad (5.10)$$

where $x'_{ij} \epsilon Z$, $y'_{ij} \epsilon Z$ and $d_{ij} \epsilon R^2$.

5.2.1 Primary functions

Lane elimination ratio

The degree to which a clustering solution eliminates lanes from the SCN reflected in the rate card is termed *lane elimination ratio*[13]. For this formulation, this lane elimination objective function is referred to as f_1.

This function requires counts of the unique number of lanes described in the rate card using the original origins and destinations *i* and *j*. This count is named *original lanes* and represented by the symbol L . It also requires the number of lanes defined using the alternative origin and destination descriptors x'_{ij} and y'_{ij}. This value is termed *aggregated lanes* and is represented by L'. To determine these counts, a set of original lanes L and clustered lanes L' are generated.

[12] d_{ij} should equal zero if the origin and destination of the lane are the same (i.e. if $i = j$) to ensure that rates defined as from and to the same SCN node are always handled as local lanes by the objective functions. Section 4.3.1. describes how the distance travelled within the same defined node may vary depending on the exact locations within the generally accepted boundary of the node description.

[13] The use of the term *fitness* to describe objective functions is specifically avoided to ensure separation of the problem from the solution approach, although this thesis uses an evolutionary approach to address the LECP

$$L := \{i \,||\, j : r_{ijk} \neq 0\} \qquad \forall\ i \in I,\ \forall j \in J,\ \forall k \in K \qquad (5.11)$$

$$L' := \{x'_{ij} \,||\, y'_{ij} : r_{ijk} \neq 0\} \qquad \forall\ i \in I,\ \forall j \in J,\ \forall k \in K \qquad (5.12)$$

K represents the set of unique carriers contained within the shipper rate card respectively.

The LECP seeks to minimise the ratio of original lanes to clustered lanes. The lane elimination ratio is therefore defined as

$$\text{Minimise } f_1 = \frac{|L'|}{|L|} \qquad (5.13)$$

The number of unique elements of set L' will always be less than or equal to the number of unique values of L. $|L'|$ will therefore always be less than or equal to $|L|$. f_1 will therefore never exceed one, with a value of one signifying that no lanes in the rate card have been eliminated when the cluster sets.

$|L'|$ will also always be greater than or equal to one, with a value of one signifying that all the rates of the rate card are described by a single lane. f_1 will therefore always be greater than zero. As such, the range of this function can be defined as $0 < f_1 \leq 1$.

$|L'|$ and $|L|$ are discrete variables. It therefore follows that f_1 is a discrete function.

Cost penalty ratio

The degree to which a clustering solution and its associated lane elimination adversely affects carrier rates is termed *cost penalty ratio*. For this formulation, this function is referred to as f_2.

This function sums the differences between the original lane rate value c_{ijk}, expressed in the unit of unit measure as per section 4.3.1, with the clustering replacement rate value c'_{ijk} across the entire rate card. The use of the former value as the basis for comparison is similar to Collins and Quinlan's (2010) use of point-to-point rates as their base model. To derive the latter value, this formulation models the process of "aligning to the top" as described in section 1.2.1, whereby the maximum original rate value for a carrier within a group of aggregated lane rates is applied to all the lanes as the replacement rate value.

We define the clustering replacement rate

$$c'_{ijk} = \begin{cases} \max\{r_{ijk} \mid x'_{ij} || y'_{ij} = l\} \\ 0, \text{where } r_{ijk} = 0 \end{cases} \qquad \forall\ i \in I, \forall\ j \in J, \forall\ l \in L_{agg},\ \forall\ k \in K \quad (5.14)$$

where $c_{ijk} \in Z$, $x'_{ij} \in Z$, and $y'_{ij} \in Z$.

The cost of clustering, or *cost penalty*, per rate can therefore be defined as

$$\phi_{ijk} = c'_{ijk} - c_{ijk} \qquad \forall\, i \in I, \forall\, j \in J,\ \forall\, k \in K \qquad (5.15)$$

ϕ_{ijk} will always be greater than or equal to zero. A zero ϕ_{ijk} results represents a clustering solution that does not have an effect on the rate card from a pricing perspective and results from an identical c'_{ijk} and c_{ijk} value for all $i \in I$ $j \in J$, and $k \in K$. It does not necessarily represent an entirely unclustered SCN or the absence of lane elimination.

The LECP seeks to minimise the normalised sum of the inflationary effect of the elimination of lanes on rates. The cost penalty ratio is therefore defined as

$$\text{Minimise } f_2 = \frac{\sum_{i\in I, j\in J, k\in K} \phi_{ijk}}{\sum_{i\in I, j\in J, k\in K} c_{ijk}} \qquad (5.16)$$

ϕ_{ijk} and c_{ijk} are limited to the domain of natural numbers. It therefore follows that f_2 is discrete and, as f_1 has been shown to also be a discrete function, it follows that the LECP has a discrete objective space.

Utopia and nadir points

Although the LECP primary functions both represent normalised measures of the effect of clustering on both the number of lanes and cost penalty, the base values of these functions are expressed in different units of measurement. It is not possible to reduce these to a common measure. The primary objectives of the LECP MOP are therefore incommensurable.

The lowest, most desirable possible cost penalty ratio value for any LECP is zero. Although the lowest, and most desirable number of lanes represented by the clustering solution is always two, the number of aggregated lanes presented in the rate card may be as low as one[14]. The lowest possible lane elimination ratio value, however, is dependent on the original number of lanes represented in the rate card, as well as the clustering solution. The ideal vector (or utopia point), for a LECP therefore depends on the specific input dataset.

The highest, least desirable cost penalty ratio value possible is also dependent on the specific input dataset. This value corresponds to a solution that has each node of each cluster set assigned to the same

[14]By assigning all nodes of the SCN to the same cluster for all cluster sets, a single local and single long-distance lane result. However, it is possible that all rates on the rate card are classified as either local or long-distance according to the distance bracket value of the solution. This would result in only one aggregated lane represented in the rate card. As such, the lane elimination ratio coordinate of the utopia point should be calculated using an arbitrarily high distance bracket value to ensure all rates draw their origin and destination cluster values from the local cluster sets.

single cluster. The effect of this clustering structure on the cost penalty depends on the composition of the shipper's rate card. The LECP's nadir point is therefore a function of the specific rate card used. The highest possible lane fitness value, however, is always one. This ratio is achieved when no lanes are eliminated from the SCN, according to the rate card, due to the clustering scheme of the solution. This is always achievable, as this would occur when all nodes of a cluster set are assigned their own unique cluster values. It is also possible for nodes to be clustered with no lane elimination resulting. As such, a solution may have certain nodes grouped into clusters and still achieve the nadir lane elimination ratio.

Table 6 shows the empirical utopia and nadir point coordinates in the primary objective space for the three example shipper datasets used for the empirical testing component of this thesis. It can be seen that the lane elimination coordinate of the utopia point and the cost penalty ratio component of the nadir point are problem instance-dependent as expected.

Table 6: Ideal and nadir points of the example shipper datasets.

Shipper	Utopia Point		Nadir Point	
	Lane Elimination Ratio	**Cost Penalty Ratio**	**Lane Elimination Ratio**	**Cost Penalty Ratio**
A	6.16×10^{-4}	0	1	3.11
B	8.32×10^{-5}	0	1	2.82
C	9.65×10^{-5}	0	1	2.39

5.2.2 Secondary functions

Prime-only cost penalty ratio

A DM looking for a simplified SCN lane structure may wish to optimise only according to the *n* cheapest carriers per lane, as these carriers are most likely to be utilised for the transportation of freight along these lanes. This would also have the effect of removing outliers from the cost penalty evaluation as exceptionally large rate values would not prevent the clustering of nodes.

An additional objective function is therefore included in this formulation such that only the *n* lowest carrier rates per aggregated lane, or *prime* rates, are considered for the calculation of the cost penalty ratio. In so doing, the cost penalty ratio assumes that cargo is only transported using the cheapest carriers available and can therefore be considered a *prime-only cost penalty ratio*. For this formulation, this function is referred to as f_3. Since the LECP is already an MOP, this function can be used instead of[15] or in addition to the cost penalty fitness function defined by equation (5.14).

[15]Should it be desirable to use this function in lieu of the primary cost penalty ratio, it is arguably easier to achieve the same result by removing the non-prime rates from the rate card and using the original cost penalty function.

A rank function $e(c'_{ijk})$ is used to assign an integer value to a clustering replacement rate. The function assigns a rank value of one to the lowest non-zero c'_{ijk} for $i \in I,\ j \in J,\ k \in K$ where $i = x'_{ij} \wedge j = y'_{ij}$. The second lowest non-zero c'_{ijk} is assigned a value of two, and so on.

This rank function is used to define the variables C_{ijk} and C'_{ijk}. These reflect the original rates c_{ijk} and clustered replacement rates c'_{ijk} with the non-prime rates nulled according to the parameter n, which is used by the DM to define how many rates per aggregated lane should be considered "prime" and used to calculate this fitness value.

$$C_{ijk} = \begin{cases} c_{ijk}, \text{where } e(c'_{ijk}) \leq n \wedge i = x'_{ij} \wedge j = y'_{ij} \\ 0, \text{otherwize} \end{cases} \qquad i \in I,\ j \in J, k \in K \tag{5.17}$$

where $n \epsilon \mathbb{N}$.

The objective function is then defined as per the original cost penalty objective function, with the exception that C_{ijk}, C'_{ijk} and Φ_{ijk} are referenced instead of c_{ijk}, c'_{ijk} and ϕ_{ijk} respectively.

$$C'_{ijk} = \begin{cases} \max\{c'_{ijk} \mid x'_{ij} || y'_{ij} = l\} \\ 0, \text{where } C_{ijk} = 0 \end{cases} \qquad \forall\, i \in I, \forall\ j \in J,\ \forall\, k \in K\ \forall\ l \in L', \tag{5.18}$$

$$\Phi_{ijk} = C'_{ijk} - C_{ijk} \qquad i \in I,\ j \in J, k \in K \tag{5.19}$$

The prime-only cost penalty ratio is therefore defined as

$$\text{Minimise } f_3 = \frac{\sum_{i \in I, j \in J, k \in K} \Phi_{ijk}}{\sum_{i \in I, j \in J, k \in K} C_{ijk}} \tag{5.20}$$

Cost penalty equity

Both the cost penalty, as well as the prime-only cost penalty objective functions, only consider the overall effect of clustering. However, the DM may be interested in an "equitable" solution that considers the evenness of distribution of the total cost penalty across the rates affected by a particular clustering.

A further objective function is therefore formulated should the equitable spread of cost penalty be a consideration for optimisation. There are several measures of dispersion that may be used for this purpose. For example, Lopes (2011) sought to minimise the maximum value of social rejection of semi-obnoxious facilities. Such an approach may work for the LECP, wherein equity is achieved by minimising the maximum individual rate cost penalty. However, this measure is highly sensitive to outliers and extreme values, which are expected for a shipper rate card. For this formulation, equity is

measured using the standard deviation of the individual cost penalty values Φ_{ijk} for $\in I,\ j \in J, k \in K$ and is referred to as f_4.

Cost penalty equity is therefore defined as

$$\text{Minimise } f_4 = \frac{\sum_{i\in I, j\in J, k\in K}(\Phi_{ijk} - \overline{\Phi_{ijk}})^2}{|\Phi_{ijk}|} \tag{5.21}$$

5.3 Constraints

5.3.1 Non-overlapping cluster constraints

Context-specific constraints may be added to the LECP according to the DM's preference. While this formulation does not include constraint definitions to cover all imaginable scenarios, the scenario outlined in section 1.2.1 requires a non-overlapping constraint.

It is supposed that, in most contexts, the solutions to the LECP would be implemented for operational use. For example, in the business scenario outlined in section 1.1, the origins and destinations defined for the carrier rates submitted during transport procurement events would likely be used subsequently to define the final selected rates that are put into operation. It is expected that the ongoing deployment and maintenance of this final rate card for the selection and payment of carriers is contingent on their reference to defined mutually exclusive geographic polygons. The effective use of clusters in this operational setting[16] would therefore be infeasible should clusters overlap. It is likely that this would also be the case in the other contexts in which the LECP must be solved. The LECP is therefore constrained such that an SNC node that lies geographically within the polygon defined by a cluster must also be assigned to that cluster.

A simple method of determining whether a two-dimensional point lies within a bounded polygon in the same plane is described by Bourke (1997) and shown by Figure 75. Should a horizontal ray be constructed in a chosen direction from the point of interest, the number of intersections with the polygon line segments can be used as an indication of the position of the point in relation to the polygon. If the number of intersections is odd, the point lies within the polygon. If this number is even, the point is outside the polygon. It follows that for the LECP, for all clusters, the horizontal rays constructed from all nodes that are not assigned to the specific cluster should produce an odd number of interactions with the cluster's enclosing polygon.

[16]This requirement is discussed in further detail in chapter 9.

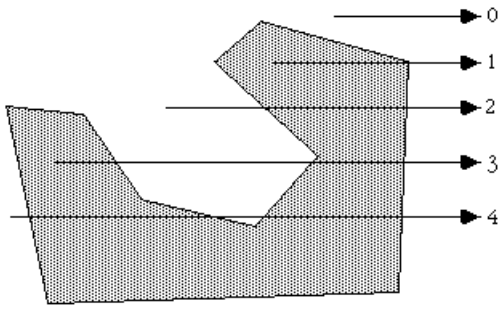


Figure 75: Illustration of the horizontal ray method of determining if a point lies within a polygon (Bourke, 1997).

To achieve this logical check, a function $h^x(G_i, G_{X_{sp}})$ is defined that gives the number of intersections of the horizontal ray emanating from a node i with the concave[17] polygon line segments defined by origin cluster X_{sp}. Function $h^y(G_j, G_{C_{Yq}})$ performs similarly, but for destination point kj and destination cluster Y_{sq}. The nodes and polygons exist in two-dimensional space and are defined according to the longitudes and latitudes (or *geocodes*) of the nodes and the node vertices of the polygons. The inputs to functions h^x and h^y are expressed in terms of the geocodes associated with the nodes, G_i and G_j, and the geocodes of the cluster polygon vertices, $G_{X_{sp}}$ and $G_{Y_{sq}}$, not the descriptions of the nodes.

The overlapping constraints for the LECP are therefore defined as

$$h^x(G_i, G_{X_{sp}}) \in 2Z + 1 \qquad \forall s \in \{A, B\}, \forall i \in I, \ \forall p \in \{x_{si}\} \tag{5.22}$$

$$h^y(G_j, G_{Y_{sq}}) \in 2Z + 1 \qquad \forall s \in \{C, D\}, \ \forall j \in J, \ \forall q \in \{y_{sj}\} \tag{5.23}$$

$$h^x(G_i, G_{X_{sp}}) \in 2Z \qquad \forall s \in \{A, B\}, \ \forall i \in I \wedge j \notin X_{sp}, \ \forall p \in \{x_{si}\} \tag{5.24}$$

$$h^y(G_j, G_{Y_{sq}}) \in 2Z \qquad \forall s \in \{C, D\}, \ \forall j \in J \wedge k \notin Y_{sq}, \ \forall q \in \{y_{sj}\} \tag{5.25}$$

These constraints represent the only aspect of the LECP that directly[18] references the spatial attributes of the SCN nodes. The LECP objective functions are concerned only with the linkage of nodes (i.e. lanes described in the rate card).

[17]The degree of concavity of a polygon constructed from a set of coordinates is an adjustable input parameter. The allowable concavity of the cluster regions generated for an LECP solution is therefore also configurable and would be defined according to the desirable geometry of the output clusters that would be implementable in a real-life context. Therefore, no concavity parameter value is prescribed to ensure a context-neutral formulation.

[18]The variable d_{ij} represents the distance from node *j* to node *k*. This formulation is not prescriptive in terms of whether this is calculated as a Euclidean distance or determined using a road network. Nonetheless, either method would use the longitude and latitudes of the nodes to determine this value.

The overlapping constraints also simultaneously enforce spatial contiguity. Should a polygon enclose all data points assigned to a single cluster without containing points assigned to other clusters, the enclosed cluster must be spatially continuous as there are no foreign data points within the polygon to divide the cluster. This is illustrated by Figure 76. Cluster A is fragmented by the nodes of cluster B. Figure 76(c) shows that four nodes of cluster B are bound by the convex polygon defined by the nodes of cluster A (i.e. the horizontal rays emanating from those nodes would intersect the polygon an odd number of times). This clustering scheme is therefore in violation of constraint (5.22) or (5.23), depending on whether these are origin or destination clusters. These constraints thereby ensure spatial contiguity in a similar manner to the constraints used for regionalization-based geographic clustering methods as described by (Assunção, *et al.*, 2006).

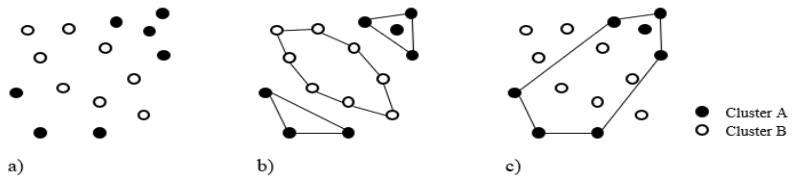


Figure 76: An illustration of how a spatially discontinuous cluster of nodes results in a polygon that contains nodes assigned to another cluster and thereby violates a non-overlapping constraint (author's own illustration).

These constraints limit the feasible solution space considerably by restricting the incorporation of nodes to a cluster to those directly adjacent to the cluster. This results in a highly fragmented decision space.

5.3.2 Additional constraints

Additional constraints necessary for the generation of sensible solutions to the LECP and are defined as

$$D \geq 0 \tag{5.26}$$

$$x_{sj} \in Z \tag{5.27}$$

$$y_{sk} \in Z \tag{5.28}$$

Constraint (5.26) ensures that only a non-negative distance break value may be selected. Constraints (5.27) and (5.28) restrict cluster labels to integer values. Although not a strict requirement for the LECP, modelling the LECP according to this requirement is recommended as these constraints facilitate easier management of solutions when the problem is solved.

5.4 Concluding remarks

Chapter 5 presented and discussed a formulation for the new LECP. This provided an unambiguous definition of the multi-objective problem as described in the problem statement of this thesis as a basis for the development of optimization methods to assist the decision-maker. Chapter 6 describes how this formulation was mapped to the algorithm domain as a single general evolutionary approach.

Chapter 6

An evolutionary approach of the LECP

In this chapter, the EA design considerations and decisions that are common to all the implemented solution approaches to the LECP are presented. This includes most of the fundamental EA elements defined during the mapping of the problem domain, defined in the LECP formulation in chapter 4, to the algorithm domain. This chapter thereby addresses research objectives 2.1 to 2.3 of this thesis. Selection operators, or combinations thereof, were materially different for specific applied solution approaches and are therefore presented with the approaches in chapter 7.

6.1 Chromosomal representation

The chromosome encoding scheme selected for all the LECP MOEA approaches closely aligns to the method by which the decision variables are defined in the problem formulation (see chapter 5). The scheme can therefore be described as a point-based encoding scheme.

For a clustering solution where the shipper's rate card includes J unique origin nodes and K unique destination nodes, the chromosomal representation of the solution's genotype consists of $2 \times (J + K) + 1$ alleles (or *bits*). The resultant solution representation method can therefore be considered a close match to a purely cluster label-based encoding scheme, as described by Mukhopadhyay *et al*. (2015). This allows for clusters of arbitrary shape to be encoded and for the number of clusters to be determined by the GA, which are both requirements of the LECP.

Each of the first J loci of a chromosome represent the origin nodes for defining clusters for local lanes. The second J loci allow for the specification of origin clusters for long-distance lanes. Similarly, the first and second sets of K loci are assigned to the local and long-distance cluster sets from a destination perspective. Each of these first $2 \times (J + K)$ loci are labelled according to a standard naming convention that is a concatenation of the following:

- Node type of the cluster set (i.e. origins and destinations denoted by "P"[19] and "D" respectively).
- Distance category of the cluster set (i.e. local and long-distance denoted by "LOC" and "LON" respectively).

[19] "P" is an abbreviation of "pick-up". Although the term "origin" is used more generally in this thesis to describe the starting location of a lane, as a symbol, "P" is preferred to "O" for ease of reading and analysis.

- The name of the node as it appears in the shipper rate card. For the solution approach implementations described for this research, these comprised of Southern African city names[20].

While many algorithms seek out and construct "good" building blocks during algorithm execution, such as that presented by Gero (1995), the cluster set building blocks are presupposed as logical subsets of the overall solution as the evolutionary process of an MOEA should identify and retain good grouping structures per set. Van Veldhuizen and Lamont (2000) state that the *successful* use of building blocks requires the ability to decompose the entire problem into subproblems. The LECP cannot be considered a truly linearly separable problem, as the four cluster sets must be simultaneously evaluated according to the distance bracket value. True building block MOEAs, such as MOMGA, should therefore be avoided as solution approaches to the LECP. However, the building block concept does assist in conceptualising and managing the makeup of the typically large chromosome required for the evolutionary approach to the LECP.

The resultant chromosomes consist of four principle building blocks of genes, each corresponding to a cluster set. Common allele values within the same block indicate common cluster assignments of nodes. These values should be restricted to integer values, not only to align with the formulation presented in chapter 3, but also to facilitate easier implementation with code.

The final chromosome allele indicates the specific distance to be used when determining when local and long-distance clusters should be referenced during the fitness evaluation process. This value can be defined using any distance unit of measurement, although consistency with the lane distance values referenced by the fitness functions is desirable from an implementation perspective. Kilometres was selected as the distance unit of measurement for the implementations described in this research. Although this allele can assume any real number from a theoretical standpoint, it is recommended that the value is restricted to an integer for easier management of the variable in code. This represents the fifth building block of an LECP solution, albeit an elementary building block composed of a single allele.

[20] The approach for this research, which was influenced by case-specific factors such as the subject companies' rate design and datasets, included a process of de-regionalization of the raw rate cards that converted all rate nodes to a city-level (see chapter 4). This need not be the case for all LECP implementations as nodes may be defined on other geographic levels, such as suburbs, towns, provinces or states, either as a standard or in combination. This implementation design decision is discussed in chapter 9.

Figure 77 shows the chromosomal representation of an example clustering solution for an LECP MOEA. The following information can be derived from this example encoding:

- The shipper rate card describes an SCN with two distinct origins (i.e. A and B) and four destinations (i.e. A, B, C and D). Cargo is both collected from and delivered to nodes A and B, while freight is only delivered to nodes C and D.
- The origin nodes for local lanes are entirely un-clustered (i.e. each node is assigned to its own cluster), while these nodes are combined into a single cluster for long-distance lanes.
- The destination nodes are grouped into three clusters for local lanes (B and C are clustered) and two clusters for long-distance lanes (B, C and D are clustered).
- During fitness evaluation, lanes of distances less than or equal to 150 kilometres should be considered local and grouped according to the cluster structures described by building blocks *i* and *iii*. The remainder of the lanes are categorised as long-distance and clustered using the structures described by blocks *ii* and *iv*.

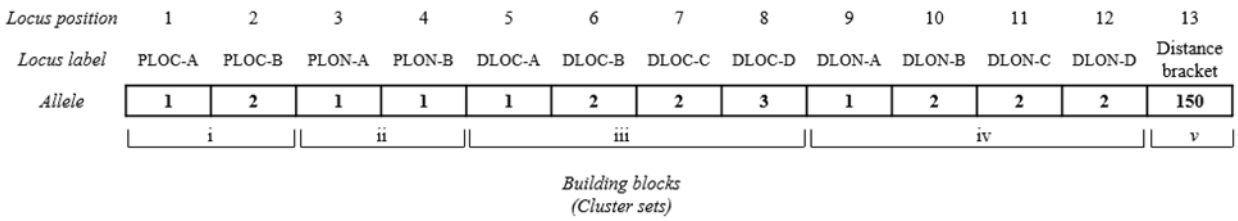


Figure 77: Chromosomal representation of an example LECP solution.

Each representation also includes auxiliary alleles that enable the evolutionary process, but are not used for recombination or crossover of individuals. These include genes for specifying the solution phenotype structure, such as fitness function and Pareto-ranking values associated with the chromosome. This approach is similar to that used by Knarr *et al.* (2004) and removes the requirement to maintain a separate data structure and robust cross-referencing scheme. An additional benefit is a saving in the computational overhead associated with continuously looking up these values as the EA works on the population.

Large chromosomes are characteristic of clustering GAs. The requirement to generate four distinct cluster sets as a full solution to the LECP results in exceptionally lengthy chromosomes when real-life datasets are used. A typical clustering solution of I nodes could be encoded using I bits, while the LECP requires a solution encoding of $2 \times (J + K) + 1$ alleles as explained above. The largest individuals modelled for the empirical tests for this research comprised of 1 285 bits.

Table 7 shows that alleles exist for all origins and destinations contained within the shippers' rate cards for both the local and long-distance cluster sets. A full solution can therefore possibly include cluster assignments of nodes for distance categories that, upon initial consideration, are irrelevant from an

existing rate card perspective and are therefore decision variables that may never be referenced during the lane elimination process simulated by the fitness functions of the LECP. For example, some of the 25 origin nodes shown in shipper A's rate card could be located in remote areas and, as such, no rates currently exist for short-distance deliveries from those nodes. The cluster assignment of these nodes for the local origin cluster set can therefore be easily assumed inconsequential from a fitness perspective. These nodes must still be represented in both cluster sets, however, as the distance bracket value and, by extension, the definition of "local" is a decision variable of the LECP. It is therefore not possible to determine which bits to exclude for each distance category before the execution of the applied algorithms and thereby trim the chromosome length.

Table 7: Shipper chromosome characteristics.

Shipper	Number of local origin bits	Number long-distance origin bits	Number of local destination bits	Number of long-distance destination bits	Distance bracket bit	Number of individual attribute descriptors[21]	Total chromosome length (bits)
A	25	25	613	613	1	8	1 285
B	193	193	317	317	1	8	1 029
C	132	132	337	337	1	8	947

Note that the integer values may simultaneously exist in multiple cluster sets. Figure 77 shows that each cluster set has a cluster labelled "1", although these are entirely independent. This must be considered when determining the total number of lanes for the clustered SCN, as a count of the unique origin-destination pairs across the entire chromosome may result in two entirely different local and long-distance lanes being considered a single lane. Although this contradicts the LECP formulation, which prohibits the repeated use of the same integer value between local and long-distance cluster sets, it was found to be simpler to count lanes separately per cluster set during fitness evaluation than modify the initial chromosome cluster sets that are constructed from a single cluster structure during population initialization.

The number of decision variables is dependent on the SCN reflected by the shipper rate card. The example datasets used for this study, however, show that a real-world LECP can easily exceed 100 decision variables and therefore can be considered a large-scale MOP (Cheng *et al.*, 2017).

[21] These include fitness function values, Pareto-ranking values and indices. The number of descriptors depends on the specific scale and algorithm applied to the problem. The values indicated in the table represent the maximum number of descriptors that could be reasonably required based on the definition of the LECP presented within the body of this research.

6.2 Fitness evaluation

The LECP formulation provided in chapter 5 includes two primary and two secondary fitness functions. These can be modified and expanded according to the modeller's discretion and the context without fundamentally changing how the remainder of the optimisation problem is solved. Nonetheless, the primary functions are mapped to the algorithm domain in this section according to how they are defined in the formulation.

There is an important practical consideration to how some input data is typically received that was relevant for the empirical analysis of the solution approaches conducted for this thesis. The method by which this was methodically addressed is included in this section as this dataset affects only fitness function evaluation. The existing rate card can define origins and/or destinations using pre-existing regions. This does not negate the case for an algorithmic approach to clustering nodes of the shipper's SCN according to the LECP, as these regions may not be well-defined by any measure. In the unlikely even that the pre-existing cluster scheme is optimally defined with respect to some metric, as a single solution it cannot represent a nondominated Pareto front of solutions for DM selection.

Both primary fitness functions require the rate card in deregionalized form as an input dataset. This is a static input and is not affected by any process executed by the MOEA algorithm. The process for reversing the previous clustering process is also deterministic. Deregionalization therefore need only be performed once and saved for repeated use. The definitions of the regions are an additional required input to this process. This dataset comprises the mapping of region descriptions to node names. The pseudocode for performing this process is given as Figure 78. It should be noted that deregionalization may not be required if the shipper rate card is already defined according to nodes without prior clustering.

```
Read rate card dataset
Read region definitions dataset
// Replicating rates with origins that are defined by regions
For i = 1 to original number of rates in rate card
    If (Origin exists as a region definition)
        Then for j = 1 to number of region definitions where origin matches
        rate i origin
            Copy and append rate i to rate card dataset with node associated with
            the region as the origin
        Remove rate i from rate card dataset
// Replicating rates with destinations that are defined by regions
For i = 1 to current number of rates in rate card
    If (Destination exists as a region definition)
        Then for j = 1 to number of region definitions where destination
        matches rate i destination
            Copy and append rate i to rate card dataset with node associated with
            the region as the destination
        Remove rate i from rate card dataset
```

Figure 78: Rate card deregionalization pseudocode.

6.2.1 Lane elimination fitness

The fitness of an LECP solution with respect to the degree of lane elimination achieved by clustering is defined as the ratio of the number of lanes after clustering to the number of lanes before clustering. This relationship is described by equation (5.11). In terms of the evolutionary approach to the LECP, this objective function is defined as *lane elimination fitness*.

This calculation requires a simulation of the transition from the original deregionalized rate card to a set of rates defined using the cluster set structures encoded in the solution chromosome. Counts of the distinct origin-destination pairs contained within the original and clustered rate cards then provide the denominator and numerator values for the lane elimination fitness equation respectively. This process must be performed for each evaluation and is described by the pseudocode in Figure 79.

```
Read rate card dataset
Create copy of rate card dataset
Read individual
// Applying origin clusters to rate card
For s = 1 to 2
    For i = to number of loci in cluster set s
        For j = to number of rates in rate card dataset
            If (node description in label of loci i equals origin of rate j)
                If (distance of lane if rate j less than distance bracket allele value
                and s = 1)
                    Then replace origin of rate j in copy rate card dataset with
                    allele value of locus i
                Else if (distance of lane if rate j greater than distance bracket
                allele value and s = 2)
                    Then replace origin of rate j in copy rate card dataset with
                    allele value of locus i
// Applying destination clusters to rate card
For s = 3 to 4
    For i = to number of loci in cluster set s
        For j = to number of rates in rate card dataset
            If (node description in label of loci i equals destination of rate j)
                If (distance of lane if rate j less than distance bracket allele value
                and s = 3)
                    Then replace destination of rate j in copy rate card dataset
                    with allele value of locus i
                Else if (distance of lane if rate j greater than distance bracket
                allele value and s = 4)
                    Then replace destination of rate j in copy rate card dataset
                    with allele value of locus i
// Calculating lane elimination fitness value
Concatenate origin and destination values for local and long-distance lanes
separately in copy rate card dataset and original rate card dataset
Divide number of unique origin-destination-lane type concatenation values in
copy rate card dataset by number of unique origin-destination concatenation
values in the original rate card dataset
```

Figure 79: Lane elimination fitness evaluation pseudocode.

Table 8 illustrates how the genotype shown in Figure 77 would be applied to an example deregionalized rate card. For this scenario, all the origin-destination pairs described for the original rate card are assumed to represent lanes with distances less than 150 kilometres and are therefore considered local. The application of the local cluster sets to the rate card resulted in the elimination of one lane, giving a lane elimination fitness value of 0.67 for the solution.

Table 8: An example of the simulated application of an LECP solution's cluster structures to a rate card for lane elimination fitness evaluation.

Original Rate Card			**Clustered Rate Card**		
Carrier	**Origin**	**Destination**	**Carrier**	**Origin**	**Destination**
a	A	A	a	1	1
a	B	B	a	1	2
b	A	C	b	1	2
b	A	A	b	1	2
Number of lanes		**3**	**Number of lanes**		**2**

Note that this process should be performed for local and long-distance lanes separately, with their numbers of lanes combined for the final determination, should rates for both categories exist.

6.2.2 Cost penalty fitness

The objective function described as the cost penalty ratio in the formulation is referred as *cost penalty fitness* in the evolutionary approach to the LECP. The determination of this cost impact incurred by lane elimination requires the simulation of the transformation of the rate card using the cluster scheme encoded in the individual's genotype. This process is identical to that employed for lane elimination fitness calculation and, as such, it is recommended to perform these fitness function evaluations as part of a single sequence of steps to prevent computational duplication. There is, however, an additional task required to determine the clustering replacement rate for each row of the clustered rate card. This value must equal the highest original rate value for each duplicate on the clustered rate card, thereby simulating an "alignment to the top" price adjustment by carriers when required to submit a single rate where multiple rates were previously allowed.

Equation (5.14) shows the difference between the original and clustering replacement rate calculated for each rate individually. A simpler equivalent approach to implement this in code is to sum the rates for each rate card and use the difference as the numerator for equation (5.15), which provides the cost penalty fitness value for the individual.

Table 9 is an expansion of Table 8, with the derivation of the clustering replacement rates illustrated. This example shows that a total cost penalty of 1000 is incurred by using the clusters encoded in the individual's chromosome, resulting in a cost penalty fitness of 0.1.

Table 9: An example of the simulated application of an LECP solution's cluster structures to a rate card for cost penalty fitness evaluation.

Original Rate Card				**Clustered Rate Card**			
Carrier	**Origin**	**Destination**	**Original Rate**	**Carrier**	**Origin**	**Destination**	**Replacement Rate**
a	A	A	1 000	a	1	1	1 000
a	B	B	2 000	a	1	2	2 000
b	A	C	3 000	b	1	2	4 000
b	A	A	4 000	b	1	2	4 000
Total			**10 000**	**Total**			**11 000**

6.2.3 Extreme points

The extreme points and, by extension, the utopia and nadir points of a particular LECP problem instance, are calculated using two artificial chromosomes. An entirely clustered chromosome is used to determine the best achievable lane elimination fitness value for a given shipper rate card, as well as the worst possible cost penalty fitness. An entirely unclustered chromosome is used to determine the best cost penalty fitness and the worst lane elimination. Figure 80 and Figure 81 show basic examples of these artificial individuals.

An arbitrarily high distance bracket value is used for the entirely clustered solution, as this maximizes lane elimination potential. This behaviour is discussed in section 5.2.1.3. Although the distance bracket value does not influence fitness values under these conditions, the same arbitrary value is used for the unclustered solution to provide the algorithm with a variable value with which to work.

Locus position	1	2	3	4	5	6	7	8	9	10	11	12	13
Locus label	PLOC-A	PLOC-B	PLON-A	PLON-B	DLOC-A	DLOC-B	DLOC-C	DLOC-D	DLON-A	DLON-B	DLON-C	DLON-D	Distance bracket
Allele	1	1	1	1	1	1	1	1	1	1	1	1	M

Figure 80: Chromosomal representation of an entirely clustered SCN.

Locus position	1	2	3	4	5	6	7	8	9	10	11	12	13
Locus label	PLOC-A	PLOC-B	PLON-A	PLON-B	DLOC-A	DLOC-B	DLOC-C	DLOC-D	DLON-A	DLON-B	DLON-C	DLON-D	Distance bracket
Allele	1	2	1	2	1	2	3	4	1	2	3	4	M

Figure 81: Chromosomal representation of an entirely unclustered SCN.

Figure 82 shows the position of example extreme individuals in phenotype space, as well as the ideal and nadir points that may be inferred from these artificial extreme points.

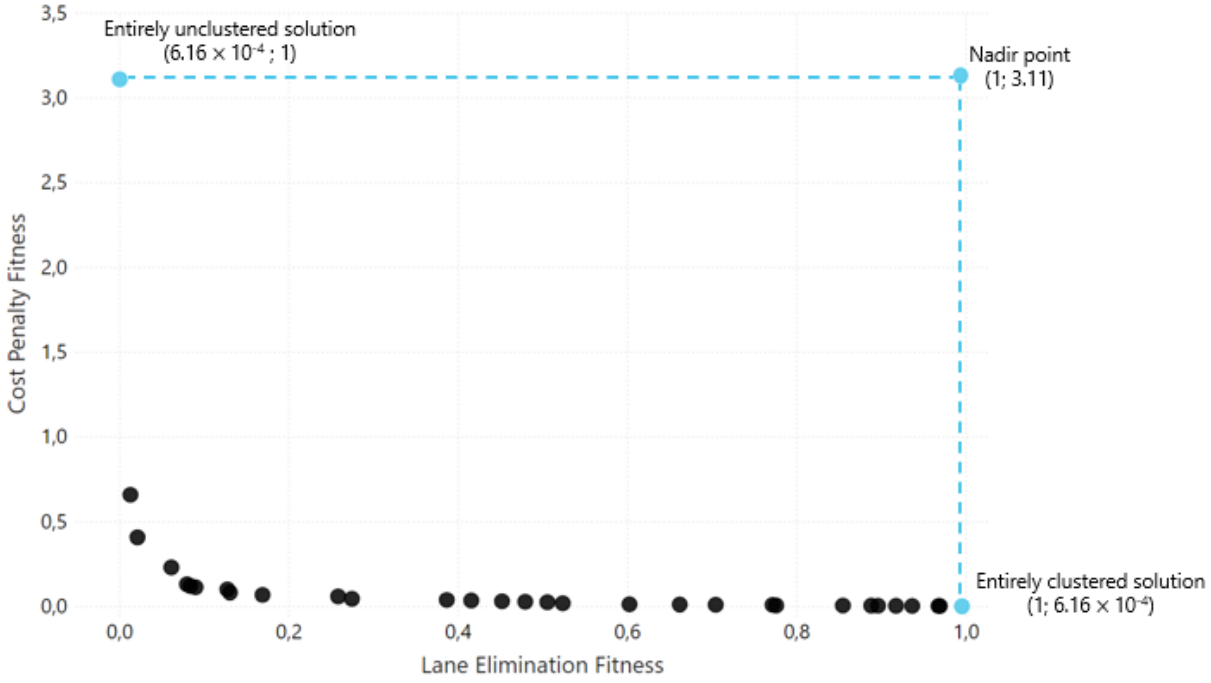


Figure 82: Scatter plot showing lane elimination fitness and cost penalty fitness values of a sample of nondominated solutions to the shipper A problem instance, including artificial individuals showing the extreme points of the instance.

6.3 Population initialization

Any MOEA approach to solving the LECP requires an initial population of solutions on which to begin performing selection, recombination and mutation operators. This population should contain only feasible solutions (Hassanat *et al.*, 2018). Several approaches are available for generating this population as a starting point.

A popular approach is to generate solutions entirely at random, or pseudo-randomly, as Maaranen, *et al.* (2006) correctly points out, due to the inherent genetic diversity generated and the computational ease with which they can be generated (Hassanat *et al.*, 2018). Random generation of clusters is the most popular method of generating initial populations for MOC problems (Mukhopadhyay *et al.*, 2015). The latter of these benefits, however, cannot be truly enjoyed if implemented for the LECP due to the number of infeasibilities this would produce. Random assignment of nodes to clusters would result in severe violations of the non-overlapping constraint, which would require computationally costly remediation to be performed on each individual.

An alternative is to incorporate problem domain knowledge into the seeding strategy. Research has shown that this can be beneficial from an algorithm performance perspective, as fitter individuals are generated and better building blocks are available to the algorithm from the beginning (Li *et al.*, 2006). Hassanat *et al.* (2018) cited k-means as a candidate technique for generating these enhanced initial populations, albeit for addressing the travelling salesman problem. Mukhopadhyay *et al*. (2015), however, confirmed this is a viable method of initiation for MOC. This technique's relevance to clustering problems and its classification as a sequential agglomerative hierarchical non-overlapping method (Bührmann, 2016) resulted in its selection as the seeding method for the LECP for all the solution approaches tested in this thesis.

The use of k-means to generate starting populations for the LECP from a shipper's rate card ensures all solutions are feasible. The specific procedure developed for its implementation for this research also empirically demonstrates the ability to produce genotypically and phenotypically diverse starting populations.

The k-means algorithm creates a master set of clusters per individual using the combined set of origin and destination nodes. An example combined set of nodes is shown in Figure 83(a), with a master clustering scheme for this set represented by Figure 83(b). This output is then used to create the four cluster sets for the solution according to the origin and destination node profiles of the shipper. As such, the k-means algorithm is run only once per solution. It also results in the local and long-distance sets being identical for each node type for each individual of the initial population. This does not represent a lack of genetic diversity, however, as these sets are represented in entirely different regions of a single chromosome and are not repeated in other individuals with which recombination could take place. The origin and destination cluster sets, per distance category, differ due to their different node profiles. This is shown by the differences in Figure 83(c) and Figure 83(d), which were generated using the same master scheme.

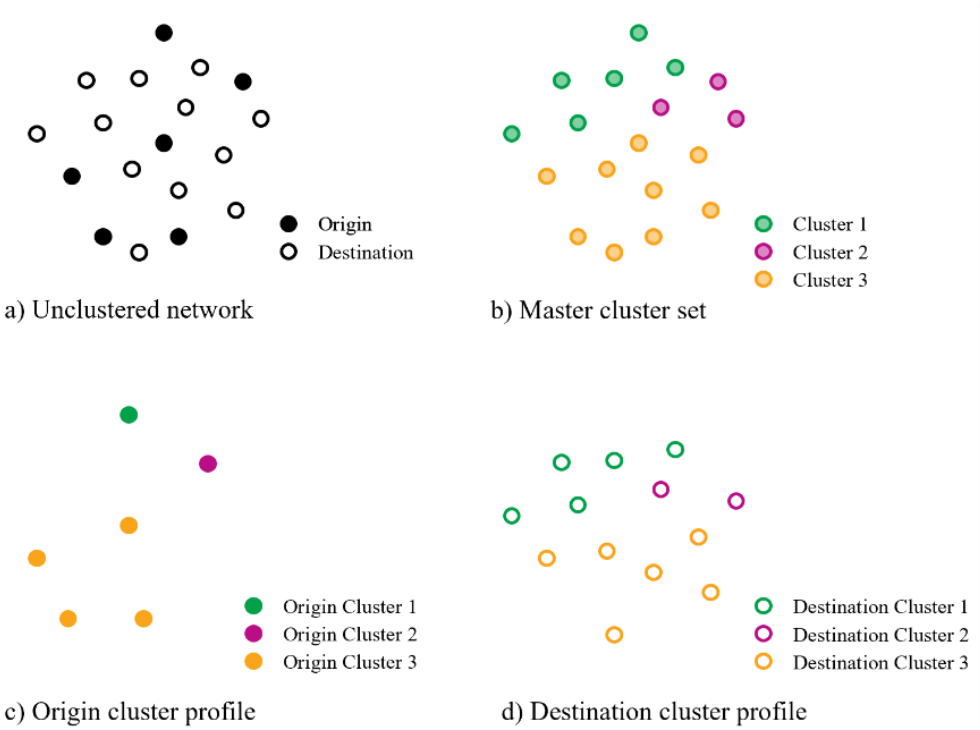


Figure 83: An illustration of the use of a combined master cluster scheme to define the origin and destination cluster sets of an SCN (author's own illustration).

The number of clusters generated by the k-means method is an input parameter for the creation of each individual. This is commonly referred to as the *k* value. For the LECP seeding process, the *k* value is set to a randomly generated integer value between 1 and the number of nodes in the combined set, with an equal probability assigned to each value in this range. This is shown by the pseudocode of Figure 84. A new random value is generated per individual. This results in the initial population solutions exhibiting an approximately even spread of degrees of clustering and therefore a high degree of diversity.

For each individual, a sample of *k* nodes is selected from the combined node set. The nodes are assigned sequentially incrementing integer values ranging from 1 to *k*. These values represent unique clusters. This node sample therefore forms the basis of the solution's master cluster scheme. The standard k-means algorithm is then applied, by which each of the remainder of the nodes inherit the cluster value assigned to its nearest cluster. The nearest node is determined according to geographic distance, which requires a distance matrix as an input dataset for the algorithm. This is a static dataset and may be populated once per shipper rate card using road network or straight-line distances. The latter was used for the empirical testing for this research[22].

[22] No adjustment was made to the straight-line distances to account for the curvature of the earth, as distances are required by the k-mean algorithm to determine only relative proximity of nodes.

Once all nodes in the combined set are assigned to clusters, the algorithm iteratively calculates the centroid of each cluster and reallocates the nodes according to their distances from these centroids. K-means is a hill-climbing algorithm and therefore continuously searches for improving modifications without considering non-improving moves to better explore the solution space (Dutta *et al.*, 2019). As such, the process terminates when a cycle is carried out with no reallocation of nodes performed as this signifies convergence to the optimum.

The final element to an LECP solution's genotype is the distance bracket value. This is generated by drawing a random number from a Gaussian distribution and stochastically adding it to or subtracting it from a base value. For the population initialization process used for this research's experimentation, the Gaussian distribution was described using a mean of 150 kilometres and a standard deviation of 100. The base value was set to 150 kilometres. This was based on norms within the South African full truckload industry in terms of what travelled distance commonly defines a load as local or long-distance.

```
Read shipper rate card r, distance matrix d and population size n
While (population size less than n)
    // Creating master cluster structure
    Generate random number of clusters k
    Generate full node set from distance matrix d
    Randomly select k cluster seed nodes
    Assign non-seed nodes to clusters
    While (change count is not null)
        For i = to number of nodes in full node set
            Determine closest nodes to node i per cluster
            If (node i not assigned to closest cluster)
                Then assign i to closest cluster and increment change count
    // Creating population member
    For s = 1 to 4
        Create empty cluster set s_c
        For k = to number of loci in cluster set s_c
            Assign cluster value of node i that corresponds to locus k as the allele value
    Randomly generate distance bracket value according to statistical distribution
    Concatenate s_c for s = 1 to 4 and distance bracket allele value in child chromosome
```

Figure 84: Population initiation pseudocode.

Figure 85 shows the distribution of individuals in the phenotype space produced using the described seeding approach for one of the test shippers. The data point labelled *i* corresponds to the clusters shown in Figure 86, while Figure 87 represents the clusters for the solution labelled *ii*. Clusters are presented

on these diagrams using different colours. Note that the geographic representation of the individual with the lower lane elimination fitness value (i.e. data point *i*) has fewer colours for both its origin and destination cluster sets, indicating a solution generated with a lower *k* value.

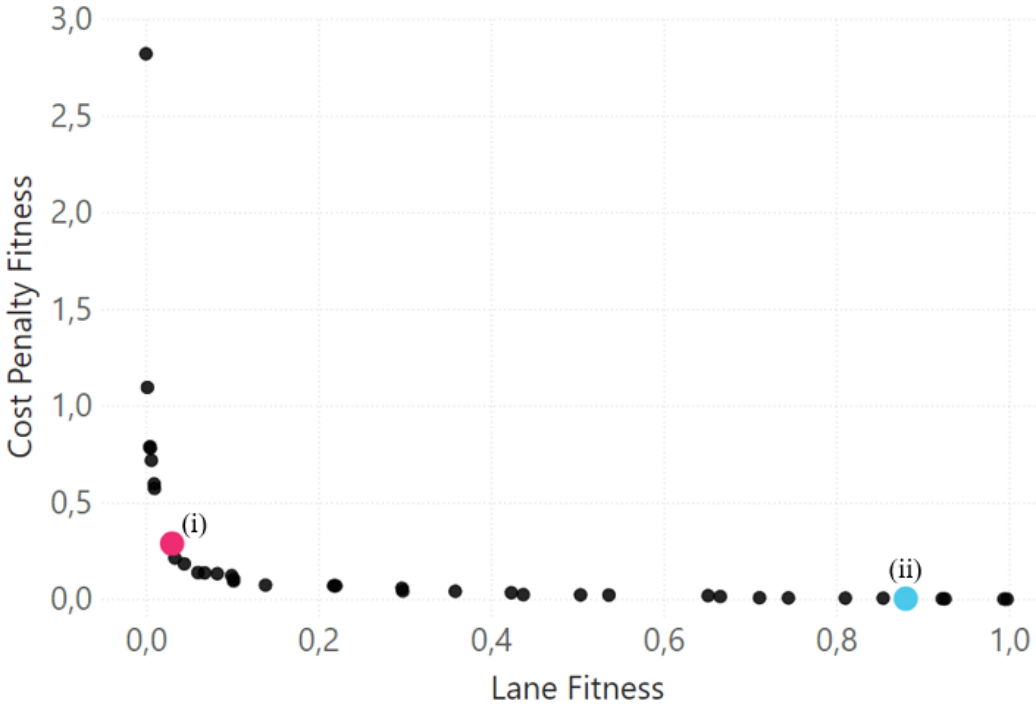
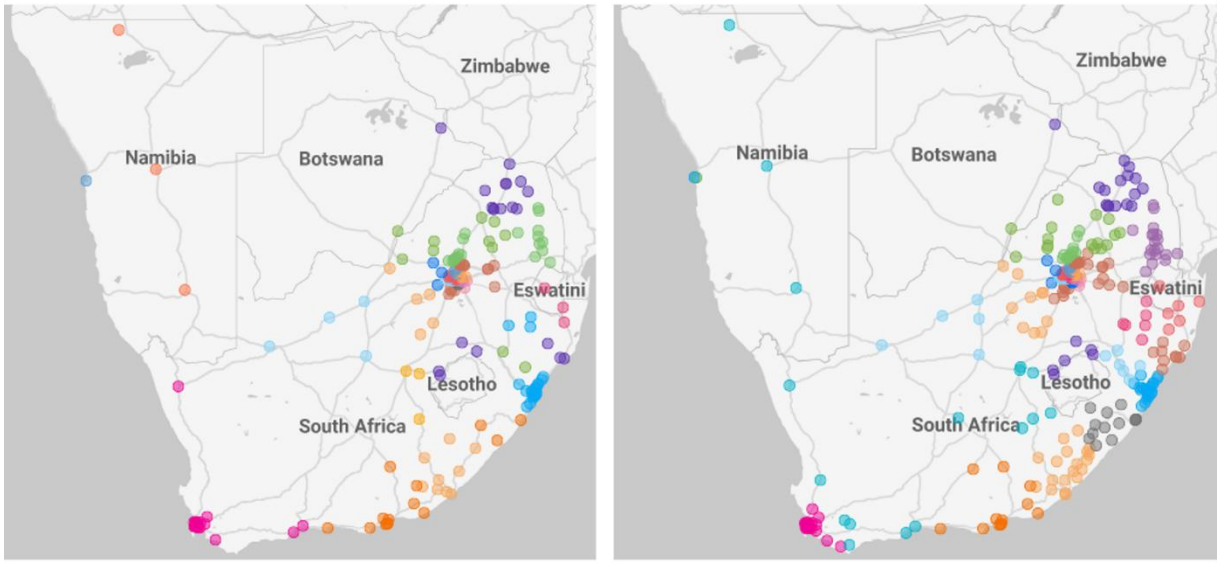
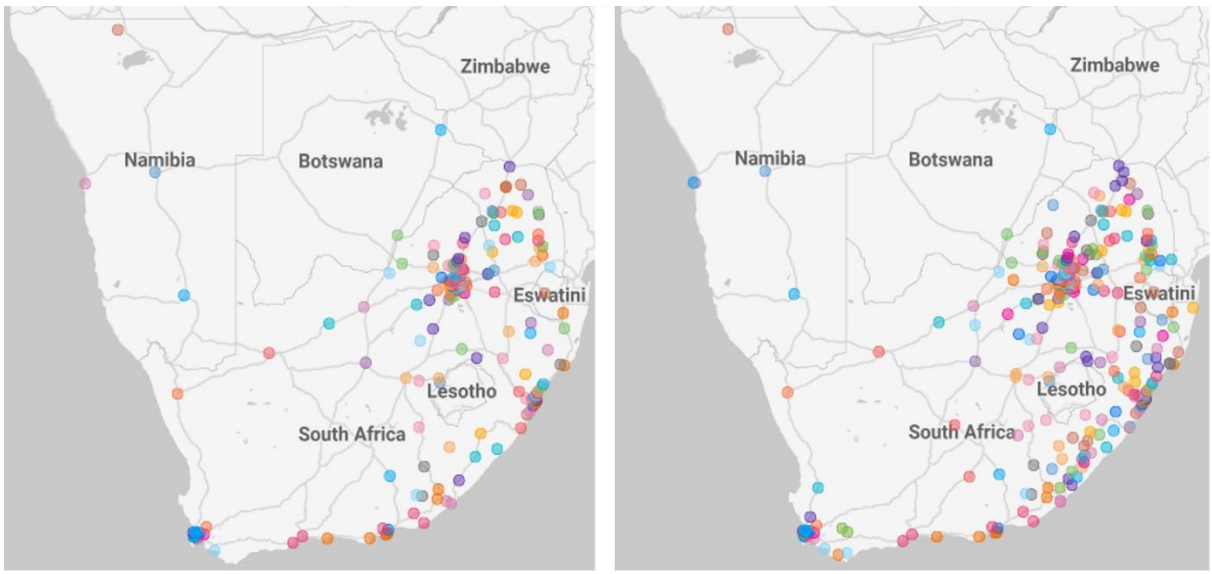


Figure 85: Scatter plot showing lane elimination fitness and cost penalty fitness values of an initial population of 100 solutions to the shipper B problem instance.



a) Origins b) Destinations

Figure 86: An example of a highly clustered solution generated during population initialization using shipper B's rate card.



a) Origins b) Destinations

Figure 87: An example of a solution with a low degree of clustering generated during population initialization using shipper B's rate card.

6.4 Recombination

The chromosomal representation scheme selected for the LECP MOEA is relatively simple and easier to implement than many alternatives found in literature, such as the locus-based adjacency method used by Handl and Knowles (2006). This is largely due to the ease with which cluster structures can be derived from the allele values. Solutions can be evaluated with minimal computational overhead associated with decoding the chromosome. In contrast, Handl and Knowles' use of a minimal spanning tree approach requires decoding of subgraphs and derivation of "interesting solutions" from the specified set of arcs, which represents computational cost that is avoided through the use of the chromosome and fitness evaluation function designs for the LECP presented in sections 5.1 and 5.2.

However, unlike Handl and Knowles' encoding scheme, the selected genotype encoding method for this research's implementation of an MOEA does not allow for standard crossover operators such as uniform, one-point or two-point recombination, without severe disruption to building blocks. Bespoke recombination operators are therefore necessary for the effective implementation of MOEA solution approaches to the LECP under the chosen encoding scheme. Two distinct operators were developed, each with the goal of preserving a different level building block. These can be described as the cluster set-level and the cluster-level building blocks. The recombination operators are named in accordance with the type of building block it uses to produce offspring solutions.

Each of these operators has associated advantages and disadvantages. While each could be solely implemented for solving the LECP, all empirical experimentation described in this research used these

operators in combination such that the advantages of each could be leveraged. This agrees with Li *et al.* (2014), who stated that due to the differing performance of certain operators in different scenarios, operators are often used in combination. During the recombination phase of each MOEA iteration, one of these two operators was selected according to a parameterized probability distribution.

6.4.1 Cluster set-level recombination

A somewhat distinctive feature of the LECP is its solution hierarchy. While most clustering problems require an individual clustering structure to be generated for a single set of data points, a LECP solution comprises four different cluster sets that interact with each other and, as such, must be evaluated simultaneously as a single solution. Figure 88 shows how each of these cluster sets can be unitized as building blocks.

A significant benefit of this pre-defined building block categorization is the ease with which recombination of individuals can be achieved during the MOEA execution process. During cluster set-level recombination, a child solution is progressively constructed by sequentially inheriting entire corresponding cluster set structures from its parents. Once a parent is selected for a specific building block, the full set of allele values for that specific cluster set is copied from the parent to the child. The selection of the parent from which each cluster set is inherited is independent and random. The same technique is used to select the final building block of the child solution, which is the single-allele distance bracket value.

Figure 88 illustrates cluster set-level recombination by showing two simple example LECP solutions, each with its own assigned colour, acting as parents to generate a single child solution. The colour of the loci of the child chromosome corresponds to the colour of the parent from which the allele values were copied. It can be seen that the example child's first and third building blocks were inherited from parent 1 (i.e. the local origin and destination cluster sets), while the remainder of the cluster sets and the distance bracket value were derived from parent 2.

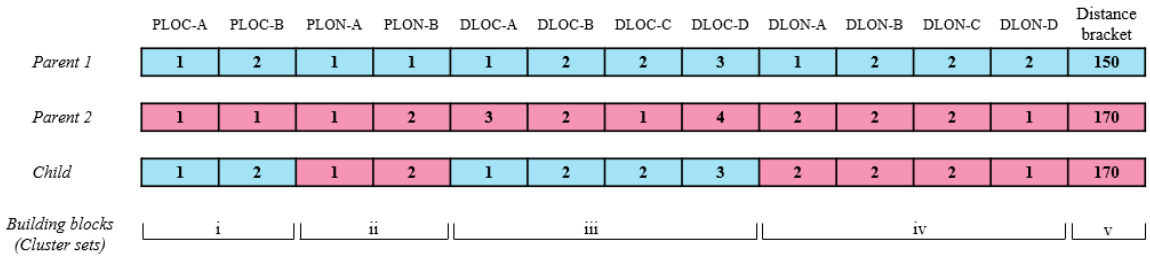


Figure 88: Chromosomal representations of an example pair of LECP parents recombining of a cluster set-level to produce a child solution.

Cluster set-level recombination provides a computationally light method of producing offspring solutions by adding substrings with fixed positions in the chromosome together without performing any logic that uses these strings. This copying process is no more onerous than that used for a five-point crossover operator, with the exception that the locus positions for each point on the chromosome remains fixed. Furthermore, the constraints of the LECP all apply within individual cluster sets. The retention of the integrity of these building blocks achieved by this type of recombination therefore ensures that child solutions cannot introduce new constraint violations, thus negating the requirement to check individuals for feasibility or remedy them.

However, this operator has some limitations. This presupposed, static definition of the chromosomal building blocks means that any operator that recombines individuals at this level cannot modify the genetic composition within a cluster set. Overuse of this operator can therefore result in a lack of genetic diversity and associated premature convergence of algorithms to a significantly sub-optimal Pareto front approximation. Sole use of his recombination operator would result in a complete reliance on mutation to produce new clusters within a cluster set.

This approach also bears a high inherent risk of clones and twins being produced. Considering that a child solution comprises only five building blocks under this definition and the probability of inheriting a specific building block from a certain parent is 0.5, it follows that there is a 0.5^5 probability that this operator produces a child identical to a specific parent. There is a probability of 0.03125 that a child generated by cluster set-level recombination is an exact copy of one of the parents. Such a clone represents computational waste, as this child provides no genetic value to the population to which it is added. There is an additional 0.5^5 probability that two children, produced using this operator and the same set of parents, are identical to each other. It follows that there is a 0.09085 probability that a child is a duplicate, either of a parent or its sibling. These twins also signify computational inefficiency, as effort is expended to produce a duplicate child that adds no diversity to the population. The probability of cloning and twinning increases as the MOEA progresses as favourable building blocks are more likely to be repeated in randomly selected mating pairs.

6.4.2 Cluster-level recombination

The recombination of individuals by blending cluster structures within parent cluster sets allows for natural dynamic identification and retention of building blocks at a level lower than the cluster set during the execution of an MOEA. This generates genetic variation within the cluster set and allows for favourable clusters from both parents to be incorporated into the same child solution, thereby overcoming significant shortcomings associated with cluster set-level recombination.

Incorporation of individual clusters from both parents into a child cluster set, however, is significantly more computationally expensive. The recombination operator is required to iterate through each cluster set itself instead of only through five predefined high-level building blocks.

Although the GraSC heuristic crossover operator developed by Demir *et al.* (2010) is applied when parent solutions are encoded using locus-based adjacency graph encoding, the operator used for solving the LECP for this research uses a similar principle. It builds a child chromosome by iterating randomly through the uncovered nodes and stochastically selects a parent cluster. The child's node inherits the selected parent's cluster label of the corresponding node. All child nodes corresponding to those assigned to this cluster in the parent's cluster structure are similarly allocated to the child's new cluster except for nodes previously clustered, or *covered*, during the loop. This process is repeated until every node in the child's cluster set is assigned a cluster value. Each of the four cluster sets is generated in this manner, with the distance bracket randomly inherited from a parent in a manner similar to that used during cluster set-level recombination.

The process of producing a child cluster set by blending the clustering structures of two parents is demonstrated in Figure 89. Each circle represents a node in a given cluster set, while the pattern fill of each node indicates the cluster to which it is assigned. For simplicity, the cluster nodes are arranged in a 5x5 square grid with an accompanying coordinate system, although nodes in a real-life SCN are likely to be significantly more numerous and unevenly spatially distributed.

For each iteration of the child cluster set generation process, an unallocated node is selected at random. For iteration 1, the node in position D4 is shown as the selected node using a number corresponding to the iteration. Another stochastic process immediately follows with a random parent selected and its cluster structure for the corresponding position copied to the child. It can be inferred from the figure that the parent for this iteration is parent 1. For iteration 2, position B2 is randomly selected. Parent 2 is then stochastically chosen. Cluster 5 is found at this position for parent's cluster set 2, which is then copied to the child. However, those child nodes already assigned at iteration 1 are unaffected. Cluster 4 is then selected at random to be copied to the child at iteration 3, which completes the new cluster set.

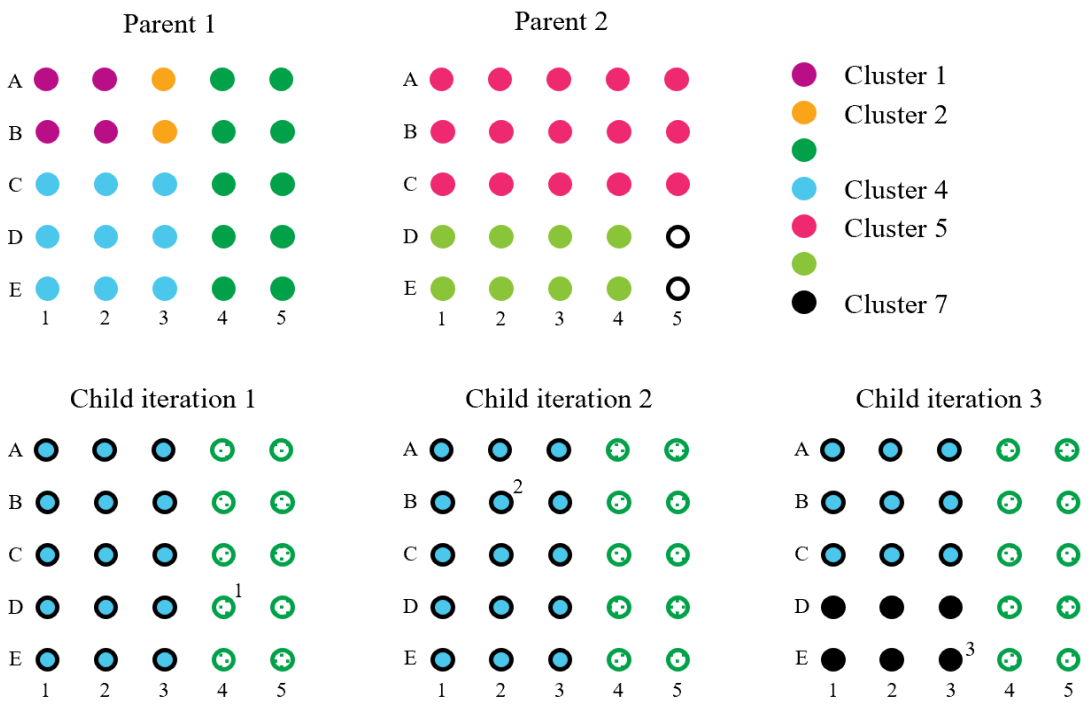


Figure 89: An illustration of the cluster-level recombination process for the LECP (author's own illustration).

Figure 89 shows that this process preserves the full integrity of only some parent clusters. Other clusters may be severed as they are incorporated into the child's cluster set, with these subsets cut from the parent cluster along the boundary of another cluster introduced during a previous iteration. This results in new clusters being generated through the recombination process and improved genetic diversity without significant disruption to building blocks at this level of the solution.

The described cluster blending procedure does, however, introduce constraint violations under certain conditions. Consider an alternative scenario shown in Figure 90. The relatively small cluster 5 is copied from parent 2 to the child cluster set at iteration 1. When the much larger cluster 1, which occupies a region of the grid that overlaps cluster 5, is copied from parent 2, a constraint violation results in the child cluster set. Cluster 1 lies within the polygon that encloses cluster 5, which is a violation of the non-overlapping constraint of the LECP.

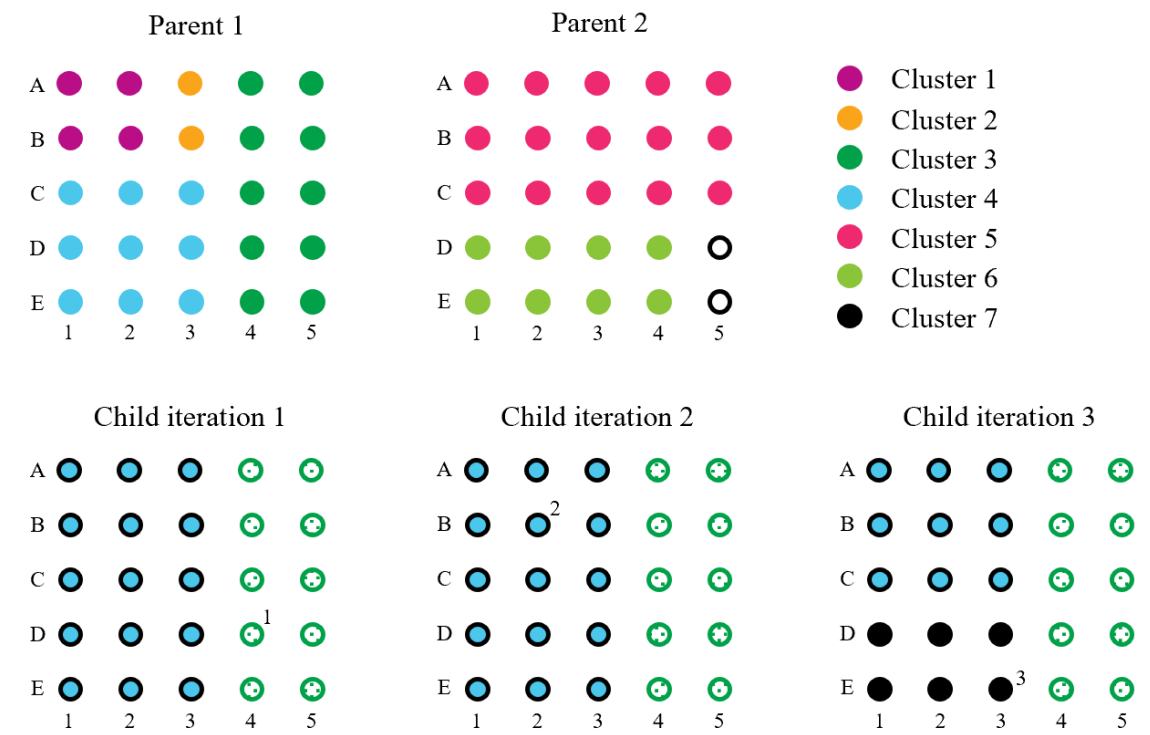


Figure 90: An illustration of how the cluster-level recombination process for the LECP can produce child cluster sets with overlapping clusters (author's own illustration).

Figure 90 presents a favourable cluster labelling situation in which no cluster label is common to both parents. If this were not the case, the integrity of the child's cluster partitions could be compromised. The cluster-level recombination operator for the LECP therefore assigns previously unused integer labels to clusters that are transferred to child. This new cluster label is found by incrementing the highest previously used allele value integer by one. This results in a continuous replacement of cluster labels in the population gene pool with larger integer values. This mitigates the risk of redundancy and removes the requirement of a renumbering procedure.

The overview of known and implemented constraint treatment methods for GAs given by Ponsich *et al.* (2008) included the elimination of infeasible solutions from the population, the penalization of the solution's objective function value according to the extent of violations, the use of dominance concepts during the selection phase of a GA, the preservation of feasibility during recombination, the reparation of solutions and hybrid techniques. The selection of the constraint handling methods for the LECP requires an appreciation that, regardless of the implemented technique, significant computational effort would be expended to detect the infeasibility. The code must iterate through each node and each cluster to determine if each node lies within the clusters to which it is not currently assigned. This is additional to the computational cost already committed to generating the child by combining parents at a cluster-level. As such, the elimination of a violating individual means that the considerable cost of creating and

evaluating the child yields no further genetic material and that alternative strategies should perhaps be employed to retain it.

It is also important to consider an important advantage of this feasibility checking method under the chosen chromosomal encoding scheme, which is that it is possible to apply a correction to the solution to make it feasible during the procedure as the violations are identified with minimal effort relative to the cost of detecting the infeasibility. Coello Coello (2022) stated that the use of repair algorithms is a good strategy if an infeasible solution can be transformed into a feasible solution at low computation cost. A reparation process is therefore incorporated into the cluster-level recombination operator itself, as shown by the pseudocode of Figure 91. This represents a hybrid approach by which feasibility is preserved during recombination by checking for and remedying constraint violations after children are produced. This is achieved by overwriting the cluster assignment of the violating nodes with the cluster in whose enclosing polygon they are found to lie. The *shapely.geometry* Python library of functions was used to achieve this check for empirical experimentation conducted as part of this research. The Python function that tests for constructing the polygon receives a parameter to define the concavity of the generated polygon. Although the formulation of the LECP presented in chapter 5 indicates that a clustered node should not lie within the concave polygon defined by another cluster, the degree of concavity is subject to the discretion of the modeller. In accordance with the underlying philosophy of this thesis that the most neutral position is taken when *a priori* setting of parameters is unavoidable, the input parameter for the polygon function was set to zero for the empirical tests of this research. The code therefore constructed only fully convex polygons during the overlapping test process.

This reparation process, when applied to the child cluster set shown in iteration 3 of Figure 90 as part of the recombination operator, would result in the assignment of nodes B3 and C3 to cluster 1 to produce a final cluster set with no overlapping clusters. The shown example output child happens to be a clone of parent 1, which is an undesirable outcome. However, for a real-life pair of parent cluster sets consisting of orders of magnitude more nodes and clusters, clones and twins rarely occur.

```
Read parents p1 and p2
// Blending clusters
For s = 1 to 4
    Create empty child cluster set sc
    While (any allele value in child sc is empty)
        Randomly select empty locus i from sc
        Randomly select parent p from p1 and p2
        Determine parent p's allele value v at locus i
        Determine all loci j in parent p where allele value equals v
        For j = to number of loci assigned to cluster v in parent p
            If (allele value in sc at locus j is empty)
                Then assign v as allele value in sc at locus j
    // Repairing constraint violations
    For k = to number of loci in cluster set sc
        Determine allele value m at locus k in in cluster set sc
        For q = to number of distinct allele values in cluster set sc
            Determine the convex polygon r enclosing all loci where allele
            value equals q
            If (k lies in r)
                Then find all allele values m in sc and replace with q
// Selecting distance bracket value and compiling full child chromosome
Randomly select parent p from p1 and p2
Determine allele value t in distance bracket locus of parent p
Assign t as distance bracket allele value in child chromosome
Concatenate sc for s = 1 to 4 and distance bracket allele value in child
chromosome
```

Figure 91: Cluster-level recombination pseudocode.

6.5 Mutation

Although the LECP formulation presented in chapter 5 does not include many distinct definitions of problem constraints, the optimisation problem is, in fact, highly constrained. This generally complicates the task of generating high quality solutions for MOPs (Vas *et al.*, 2021). It is specifically the non-overlapping constraint of the LECP which greatly restricts the feasibility of assignment of nodes to clusters. This constraint thereby contributes to a high probability of infeasible solutions being produced by evolutionary operators during MOEA execution unless knowledge of this problem domain is incorporated into the operators. Section 6.4.2 discusses how this is achieved for a recombination operator developed for this research. This operator included a post-processing chromosome repair

component to address any constraint violations that are generated during the primary recombination stage. This is required as infeasible solutions are unavoidably produced via this type of recombination.

In contrast, the mutation operators for the LECP are designed for the in such a way that introduction of violations of the non-overlapping constraint is entirely avoided during the production of mutants. This is advantageous due to the computational expensiveness of generating large numbers of mutants and the prevention of this effort being expended producing violating individuals that must be subsequently discarded or remedied at further cost (Zhang *et al.*, 2010). Generic mutation operators that cannot prevent violations, such as bit-flip mutation, were not considered for incorporation to the solution approaches applied to the LECP for this thesis.

Three custom mutation operators were developed. While the recombination operators presented in section 6.4 can be applied independently, the LECP mutation operators should be implemented collectively to ensure a balanced genetic diversification process. Two of these operators, namely cluster-split and cluster-merge mutation, can be considered inverse operations and ensure cluster-level modification of the four cluster sets of a selected individual. The former performs the division of randomly selected clusters, while the latter is responsible for the random combination of adjacent clusters. To implement one of these operators without the other, or to use different mutation probabilities for each, would direct the MOEA toward entirely clustered or un-clustered solutions. They should therefore be implemented with equal mutation probability parameter values. The third mutation operator modifies the fifth high-level building block of the individual, namely the distance bracket value, without editing its clustering structures in any way.

6.5.1 Cluster-split mutation

The process of dividing a cluster within a cluster set receives a single cluster structure, consisting of the set of all nodes assigned a specific cluster value, as an input. However, the process used for the LECP generates an assignment of two new cluster values to these nodes as an output. The cluster-split mutation operator developed for this research achieves this in such a way that the division is randomised. It is therefore similar to the split mutation process described by Cole (1998), with the exception that entirely new labels are generated for both of the resultant clusters.

The design of this operator accommodates for the fact that, once a child solution is identified in accordance with the mutation rate parameter set for the algorithm run, it may be overly disruptive and computationally onerous to split every cluster within the individual's cluster sets. It therefore includes a secondary mutation rate input value to enable sampling of a child's clusters for splitting. This value represents the proportion of clusters contained in the cluster set that must be divided by the operator. It follows that, for a cluster set consisting of n clusters and a secondary mutation probability of p, pn

clusters[23] would be divided by the operator. This would give rise to *2pn* new clusters, while the remaining *(1-p)n* clusters remain unaffected. The process results in a mutant cluster set with a total of *(1+p)n* clusters. It can therefore be seen that cluster-splitting has a de-clustering effect on a solution. As such, this operator typically affects the phenotype representation of the individual, with the resulting mutant having a lower and more desirable cost penalty fitness value, and a higher, less desirable lane elimination fitness.

A cluster, when selected for division, is partitioned according to a straight line constructed to coincide with a randomly chosen pivot node of the cluster. The orientation of the line is also selected randomly. Once this line is constructed, each of the cluster's nodes are assigned one of two newly generated cluster values based on the side of the line on which they lie. The pivot node, which always lies directly on the partition line, is randomly assigned to one of the new clusters. This process is repeated for every cluster, followed by a consolidation of the modified cluster sets into a single mutant chromosome. The distance bracket is unchanged by the cluster-split mutation process.

The process of mutating a child cluster set by splitting clusters is demonstrated by Figure 92. For this example, the cluster proportion parameter *p* is chosen to be 0.4. As the cluster set consists of five distinct clusters in the cluster set, two clusters are chosen at random to undergo division. For the first iteration, cluster 4 is chosen randomly and node 1 is selected as the pivot as shown in Figure 92(b). A random number between 0 and 180 is generated as the angle of inclination of the partition line. A partition line, coinciding with the pivot node, is then constructed with this orientation. Figure 92(c) shows each node above this line being assigned to the new cluster 6, while each node below the partition is assigned to cluster 7. Node 1, being the pivot node, is randomly assigned to cluster 6. This process is repeated for cluster 2, which is the second cluster randomly selected for splitting. Figure 92(d) and (e) show this cluster give rise to clusters 8 and 9. Clusters 2 and 5 are no longer present in the final structure as they are discarded during the splitting process. It can therefore be seen that the mutant cluster set has *(1+p)n* clusters.

[23] This value must be an integer. *pn* would, in most cases, require rounding. It is recommended to round up this value to ensure that at least one cluster is split by this operator.

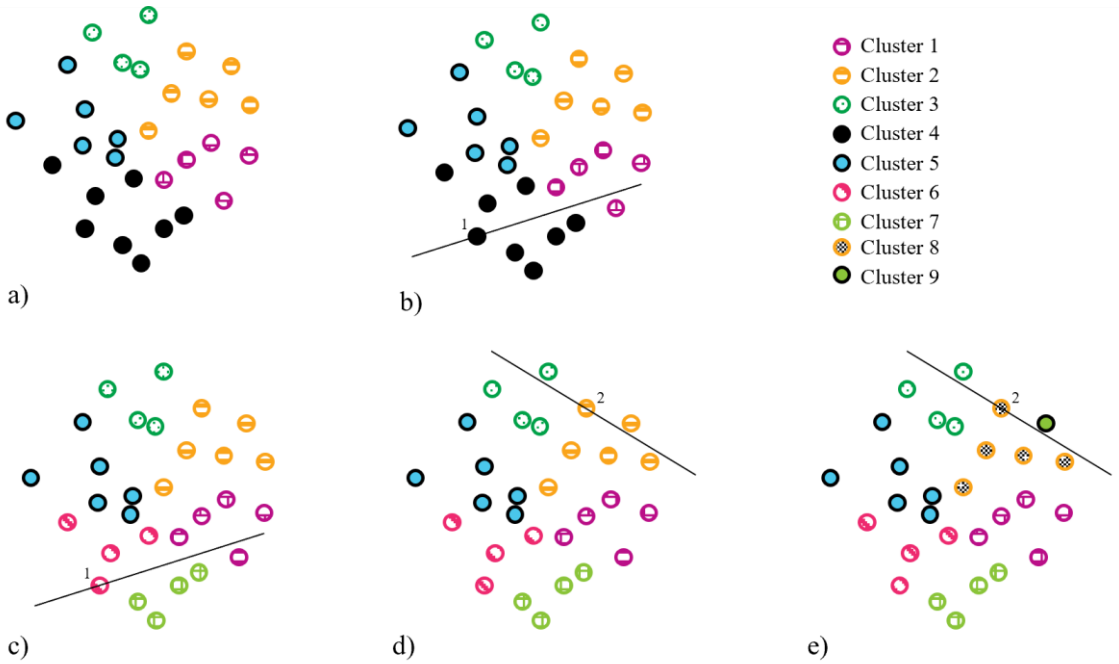


Figure 92: An illustration of a cluster set undergoing cluster splitting (author's own illustration).

In addition to the simplicity of its implementation, the use of a straight-line partition has the benefit of ensuring the resulting smaller clusters do not geographically overlap regardless of the degree of concavity considered when constructing the clusters' associated polygons, provided that the input cluster did not overlap another prior to division. No reparation of individuals is therefore required after cluster splitting is performed and, therefore, no reparation steps are included in the pseudocode shown by Figure 93.

```
Read child population c
// Selecting children for mutation
Read primary and secondary mutation probability parameters p1 and p2
For i = to number of children in child population c
    Generate random number between 0 and 1
    If (random number less than p1)
        Add child i to children for mutation cm
// Splitting clusters
For j = to number of mutation children population cm
    For s = 1 to 4
        For k = to number of clusters in cluster set s
            Generate random number between 0 and 1
            If (random number less than p2)
                Add cluster cl_k to clusters for mutation cl m
        For l = to number of clusters in mutation clusters cl m
            Determine allele value v for loci in mutation cluster
            Determine highest current cluster value h in cluster set
            Randomly select locus n assigned to cl m_l
            Randomly select angle of inclination α
            Construct straight partition line coinciding with locus n's
            geographic coordinates at angle α
                For (all loci in mutation cluster geographically positioned
                above partition line)
                    Replace all allele values v with h+1
                For (all loci in mutation cluster geographically positioned
                below partition line)
                    Replace all allele values v with h+2
            Replace allele value v for locus n with h+1 or h+2
// Compiling full mutant chromosome
    Concatenate s for s = 1 to 4 and distance bracket allele value from child
    mutation candidate to form mutant chromosome j
```

Figure 93: Cluster-split mutation pseudocode.

6.5.2 Cluster-merge mutation

The cluster-merge mutation operator fulfils the opposite function of the cluster-split operator. This operator consolidates two clusters into a single structure according to a stochastic process. This process

thereby builds larger clusters within the cluster sets of an individual chromosome, typically resulting in a higher, less desirable cost penalty fitness, but a lower lane elimination fitness value.

Similar to the cluster-split operator, the merging operator also uses a secondary mutation probability value for selecting the clusters of an individual that is undergoing this type of modification. It is recommended that this parameter, in addition to the primary mutation probability value, be set equal to the corresponding setting of the cluster-split operator. This ensures that, in general, clusters within the population are split and merged at approximately equal rates. It is therefore also comparable to the merge mutation process described by Cole (1998), with the exception that a new label is used to describe the resultant cluster.

In contrast to the cluster-split operator, it is not only the clusters that are originally selected using this secondary parameter that are modified by the cluster-merge process. A single node within a selected anchor cluster is identified randomly as a root node. A closest neighbour node is then identified by a simple randomised search of the geographic neighbourhood of the cluster. This is the closest node to this root, according to straight-line distance[24]that is not assigned to the root's cluster. The clusters to which the root and closest neighbour nodes are assigned are subsequently combined by assigning a newly generated cluster value to all the nodes within the cluster set that share the cluster value of those two nodes.

The cluster merging process is demonstrated in Figure 94. The cluster set originally contains five cluster structures. The illustrations show how this cluster set, when a secondary mutation probability of 0.4 is used, undergoes a cluster consolidation process that yields two fewer clusters. Figure 94(a) shows cluster 4 selected as the first anchor cluster. Node 1 is the randomly selected root node for this anchor, with node 2 identified as the closest neighbour. Nodes 1 and 2 are assigned to clusters 4 and 1 respectively and all nodes belonging to these clusters are now assigned to the new cluster 6, shown in Figure 94(b). The second iteration is shown by Figure 94(c) and (d), during which clusters 5 and 3 are merged into the new cluster 7.

[24]Similar to the distances used for the population initiation process, no adjustment must be made to these values to account for the curvature of the earth, as the distances are only required to determine only relative proximity of nodes.

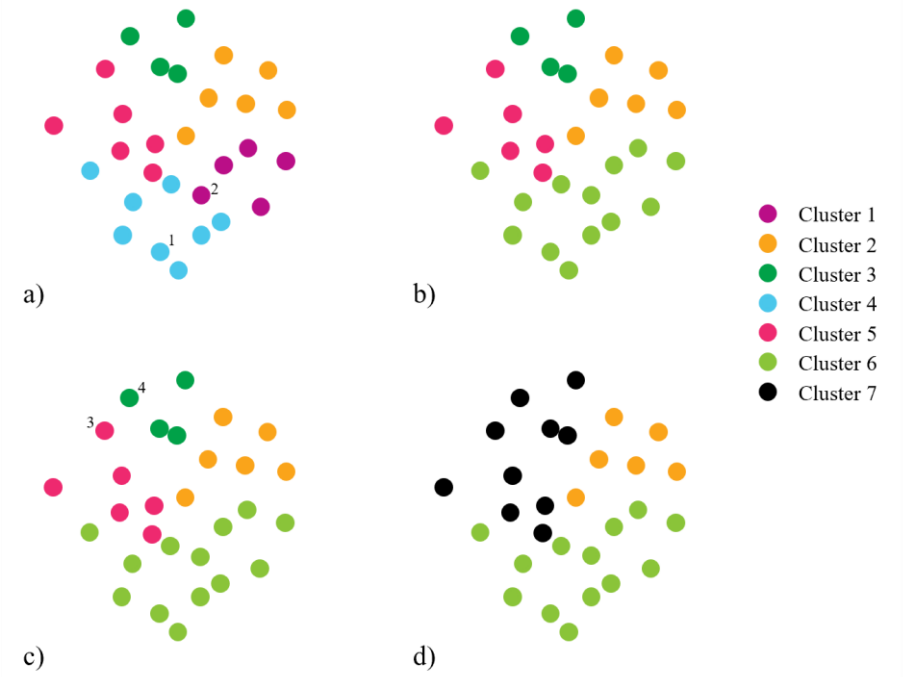


Figure 94: An illustration of a cluster set undergoing cluster merging (author's own illustration).

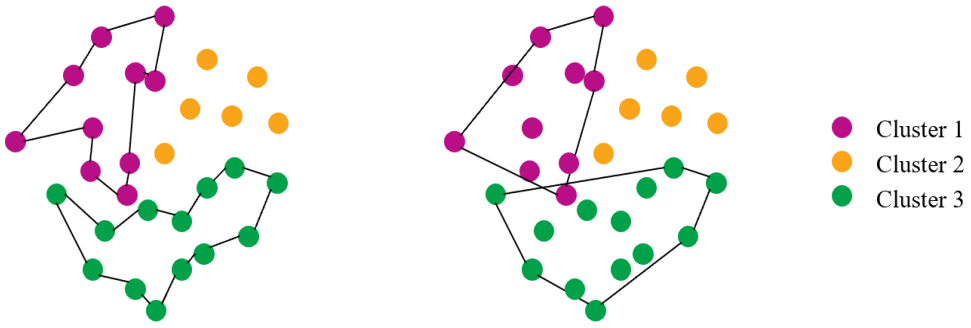
This process is easy to implement in code. Improvements to the operator are available, however, at the cost of additional computation. Some of this computation may be reusable and therefore represent a once-off investment. For example, road network distances can be used instead of straight-line distances, which is generally accepted as a more accurate means of clustering physical locations that are connected by road transport (Buczkowska *el al.*, 2017). These need only be calculated once for every origin-destination pair of nodes, with the results maintained as a distance matrix for subsequent and repeated reference by the algorithm.

Furthermore, the purely random selection of anchor clusters could be replaced with a mechanism that achieves a more even spatial spread of merging through the cluster set. This would therefore add significant computation for each execution of this mutation operator. This would reduce instances of a failed merging. This occurs when a selected anchor cluster is eliminated during a preceding iteration by consolidation with another, adjacent anchor during a single mutation event. For example, if the anchor cluster for iteration 2 shown in Figure 94(c) was pre-selected to be cluster 1, there would be no nodes available for selection as the root of this iteration. Merging, for this iteration, would fail, unless additional logic is added to the operator to select an alternative anchor, adding complexity and computational cost. This is an undesirable phenomenon, as it results in less predictable behaviour in terms of this process' input and output number of clusters.

Prevention by non-random sampling, however, would require additional calculation as it is dependent on each cluster set of the selected individual, which would typically be unique. Nonetheless, mutation failure is inherent to the cluster-split operator as well. No significant accumulated imbalance in the

population after mutation is performed on each generation's offspring is therefore expected provided the mutation probability parameters are equal for both operators.

The nearest neighbour method of selection of adjacent clusters for merging ensures that this operator does not introduce spatial discontinuity within the cluster sets. Provided the input cluster structures do not geographically overlap, the output structures typically do not violate the non-overlapping constraint of the LECP. This, however, is dependent on the degree of cluster polygon concavity specified for this constraint. Consider the scenario shown in Figure 95. Both cluster sets are identical representations of the output of the example cluster merge demonstrated in Figure 94. Figure 95(a) shows concave polygons constructed for the clusters produced by the cluster merge operator, while Figure 95(b) shows these clusters bounded by purely convex polygons. The example shows that, should it be forbidden for the convex polygons of clusters to overlap, cluster merging may introduce infeasibilities within the cluster sets. In such a case, a post-merging reparation process may be implemented similar to how introduced violations are addressed during cluster-level recombination. Alternatively, production of these violating mutants may be allowed under the assumption that reparation of any of these infeasibilities will be addressed during subsequent cluster-level recombination as a matter of course, or performed at the end of the algorithm run or every epoch. For this thesis, the former option was implemented, shown by the pseudocode of Figure 96, as it is the most computationally efficient and neutral approach, especially considering that the degree of concavity for the non-overlapping constraint of the LECP is not assumed.



a) b)

Figure 95: An illustration of a degree of polygon concavity and its effect on the constraint violation status of the output of cluster merging (author's own illustration).

Read child population **c**

Read distance matrix **d**

// Selecting children for mutation

Read primary and secondary mutation probability parameters $\mathbf{p_1}$ and $\mathbf{p_2}$

For **i** = to number of children in child population **c**

Generate random number between 0 and 1

If (random number less than $\mathbf{p_1}$)

Add child **i** to children for mutation **cm**

// Merging clusters

For **j** = to number of mutation children population cm

For **s** = 1 to 4

For **k** = to number of clusters in cluster set **s**

Generate random number between 0 and 1

If (random number less than $\mathbf{p_2}$)

Add cluster $\boldsymbol{cl_k}$ to clusters for mutation **c*l*m**

For ***l*** = to number of clusters in mutation clusters **c*l*m**

Determine allele value **v** for loci in mutation cluster

Determine highest current cluster value **h** in cluster set

Randomly select locus n assigned to $\mathbf{cl m_l}$

Instantiate closest neighbour locus variable **cn**

Instantiate closest neighbour distance variable **cd**

For (**q** = to number of loci in mutation cluster where allele value not **v**)

If distance nq from locus **q** to locus **n** less than **cd**

Then **cn** = **q** and **cd** = **nq**

Determine allele value **w** for locus **cn**

Replace all allele values **v** and **w** in $\mathbf{s_c}$ with **h+1**

// Compiling full mutant chromosome

Concatenate **s** for **s** = 1 to 4 and distance bracket allele value from child mutation candidate to form mutant chromosome j

Figure 96: Cluster-merge mutation pseudocode.

6.5.3 Distance bracket mutation

The operator responsible for providing variation of the fifth high-level building block of a chromosome, the distance bracket value, is considerably simpler than those that alter the structures of the four cluster sets. Distance bracket modification requires only one allele value to be modified and can be achieved without consideration of any of the other alleles of the solution or the population. The manner by which this modification is achieved can therefore be adjusted, or even redesigned, per implementation with

minimal impact on the design of the overall solution approach or the other operators therein. In accordance with the principle of maximum neutrality adopted for the design of the MOEA operators and solution approaches for this thesis, a simple method of adjusting the distance bracket of a chromosome was devised and used for the empirical testing of the solution approaches.

This mutation operator receives an individual chromosome and isolates the distance bracket allele value. A decimal number between 0 and 2 is generated and multiplied by the original distance bracket. The product is rounded down and used to replace the original distance bracket value before the resulting chromosome is returned as the output of the operator. An individual selected for this type of mutation can therefore experience its distance bracket reduced to any integer as far as zero, or increased to any whole number up to double its original value according to a linear scale of probability. Any feasible value of the distance bracket can therefore be reached via successive decreases or increases, although the selective pressure of the solution approach would assist in limiting the distance brackets contained in the population to reasonable values.

6.6 Population evaluation

A single set of metrics was selected for the evaluation of a given population of LECP solutions. These were used to measure the quality of the current nondominated set as found by any MOEA between generations or upon termination of the algorithm. The extent of this selection was chosen with the intention to provide a modeller a wieldy number of indicators that are sufficient to distinguish between MOEAs according to the goals of multi-objective optimisation as described by Zitzer *et al.* (2000). As such, metrics were chosen such that the convergence, uniformity and spread achieved by the algorithms would effectively quantified.

The known characteristics of the LECP were also considered. Quality indicators were therefore selected while recognizing that the true Pareto optimal front in phenotype space PF_{true} would almost certainly be unknown for any exercise that seeks to solve this MOP using real-world data either in an academic or industry context. The latter consideration eliminated metrics that use the proximity of the Pareto front approximation to the true front, such as the inverted generational distance (IGD) and hyperarea ratio, from consideration as candidate metrics for the LECP. The ideal and nadir points for a problem instance of the LECP, however, can be determined, as shown in section 6.

The hypervolume (HV) metric was selected as the primary population evaluation measure due to its simultaneous measurement of all three optimisation goals and it not requiring the specification of PF_{true} as a reference. Instead, it requires only the Pareto front approximation PF_{known} and the nadir point as a reference point, which can be determined for the LECP without PF_{true} being known. The selection of

HV was supported by evidence in literature of this metric being a widely used and trusted multi-objective optimisation method.

The HV_d value was also measured as a quality indicator. This provided a more focussed view of the uniformity and spread of a generated Pareto front approximation. As such, HV was measured and analysed as the primary measure of convergence, while HV_d was used as an indication of population diversity.

Both the HV and HV_d measures of a population's quality require a reference point in phenotype space. As the nadir point for each problem instance was known, this set of coordinates was used as the reference point for each evaluation process. Furthermore, as the nadir point for each shipper's dataset was also known, it was possible to calculate the theoretical maximum hypervolume for each instance. As the nadir points differ between problem instances, the maximum HV value also varies depending on the shipper rate card. The HV values were therefore normalized as percentages of their theoretical maxima for ease of comparison of results by shipper. These were used as the primary quality measures for the experimental component of this research and are represented by the symbol $HV_{(\%)}$. The process of projecting the data points onto the reference line to calculate the HV_d measure of a population already has a normalizing effect on the data and, as such, this value is not explicitly normalized.

Figure 97 and Figure 98 illustrates the usefulness of normalizing population HV values. By representing HV as a percentage of the total volume of the hypercube enclosed by the origin and nadir points of the phenotype space, the quality of each solution is easier to compare across problem instances. Trend analysis is also simplified by using $HV_{(\%)}$.

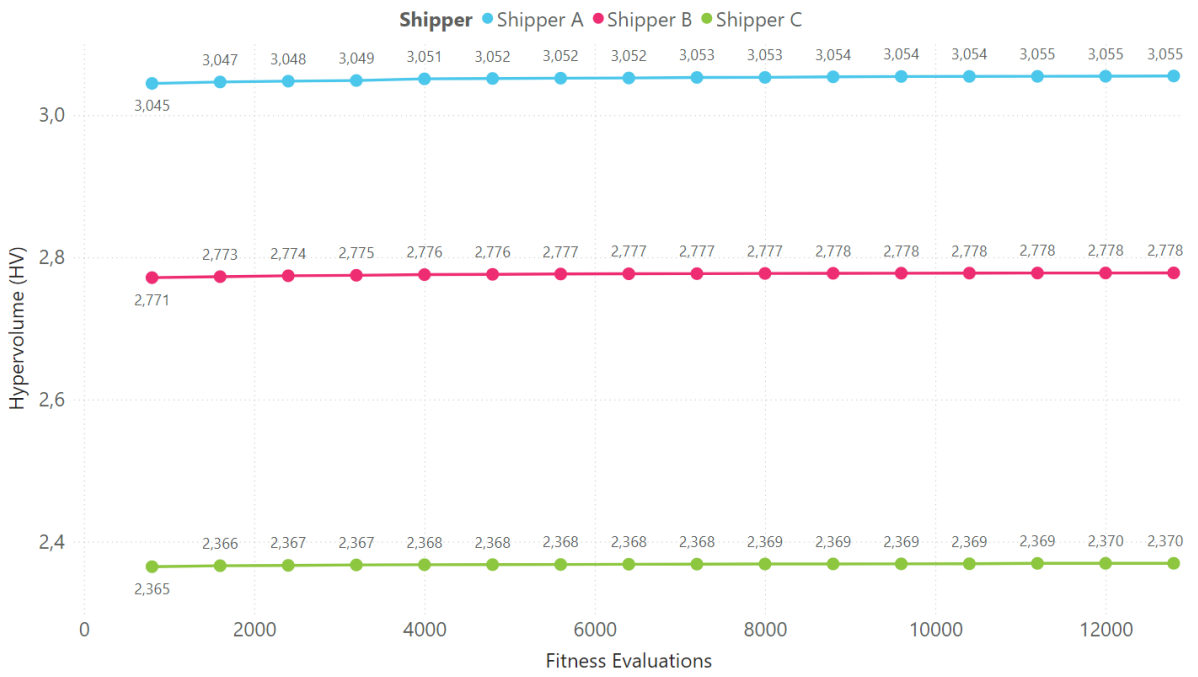


Figure 97: Line graph showing population hypervolume as a function of number of fitness evaluations per shipper problem instance.

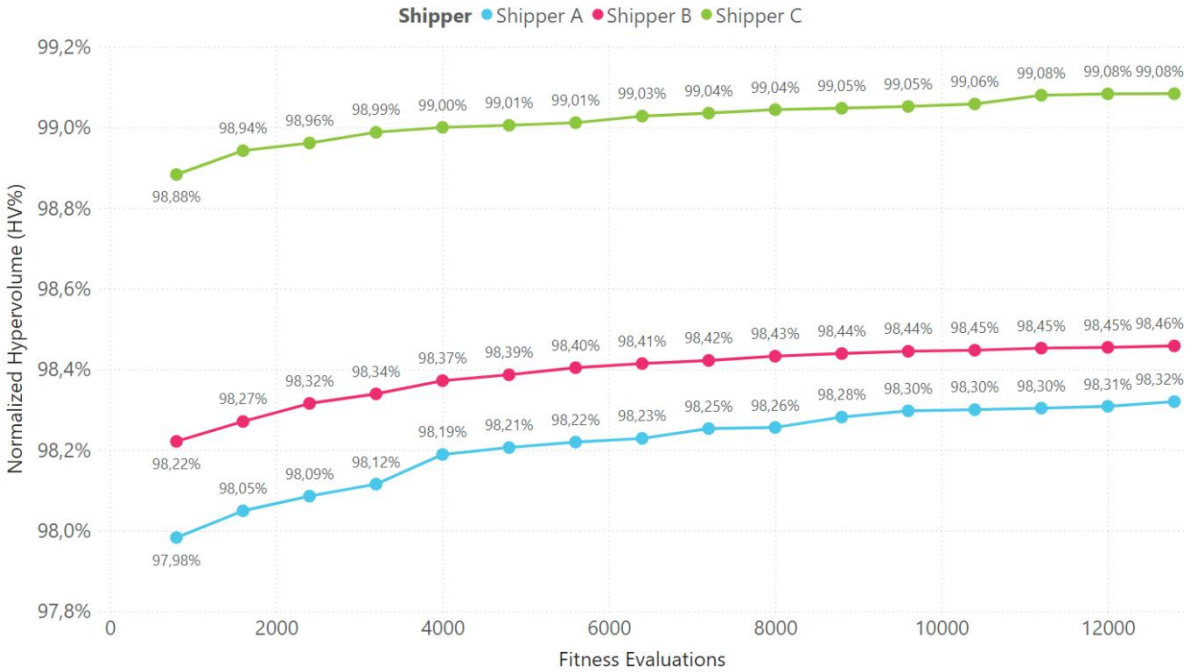


Figure 98: Line graph showing normalized population hypervolume as a function of number of fitness evaluations per shipper problem instance.

6.7 Stopping criteria

The selection of a termination criterion was informed by the circumstances of the optimisation runs rather than the characteristics of the LECP. A number of stopping criteria are available for incorporation to an MOEA to terminate optimisation, including direct, derived, cluster-based, operator-based, performance and progression indicator termination. These require a parameter to determine a proper ending of the EA (Ghoreishi *et al.*, 2017). Many of these parameters involve additional calculations that are not already executed as part of all solution approaches to the LECP, including those criteria that make use of the mutual domination rate (MDR) indicator as described by Marti *et al.* (2009). This additional overhead would hamper the testing process and have an unequal relative impact on the performance of each solution approach. These criteria were therefore disqualified from selection for the experimentation methodology.

Ghoreishi *et al.* (2017) described the dual considerations of performance and reliability in the selection of a termination criterion for an MOEA, while recognizing that trial-and-error is required for the determination of parameter values that allow for proper termination. They recommended the "hitting a bound" method for comparative studies. Although the empirical experimentation conducted for this research could indeed be described as a comparative study, they further qualified this recommendation by adding that the global optimum should be known *a priori*. This is not the case for the LECP. Even if a good, sub-optimal bound were to be found empirically and used for this method, this criterion introduces the risk that runs of some solution approaches may never terminate or only terminate after an unreasonable amount of time. The algorithms may converge to a population of solutions that, as a set, perform below the bound or simply do not progress with sufficient velocity toward the bound. This risk is considerable due to the number and variety of solution approaches tested for this research.

The "maximal time and number of generations" criterion is the only condition presented by Jain *et al.* (2001) that terminates the search independently of the progression of the search. This condition will not stop the evolutionary process of an MOEA upon detection of convergence. This means that the algorithm may waste time needlessly searching for better solutions that do not exist. However, the investigation of the selected solution approaches for this thesis is not highly time sensitive. Reliability and fairness therefore carried more weight in the criterion selection process, although any operational process that requires lane elimination may require a condition that prioritises timely termination. The "maximal time and number of generations" criterion also bears no risk of premature termination or failure to terminate due to poorly configured parameters. This behaviour is desirable for the comparative analysis of different solution approaches to the LECP, provided a sensible wall clock time, number of generations or objective function evaluations is used as the t_{max} value.

The “maximal time and number of generations” criterion was therefore selected as the stopping condition for the testing of the solution approaches to the LECP. Wall clock time was not considered as the basis of this criterion. This was due to the varying run times that may be achieved using the same equipment under varying conditions, such as CPU temperature and running background processes, and the anticipated difficulty in controlling these conditions. Furthermore, certain elements of the MOEAs may be inefficiently implemented in code for this research or other studies. Furthermore, different of programming languages may be used for this research and those undertaken by another practitioner[25]. These differences would likely produce different effects on clock time for the various solution approaches, thereby introducing unfairness to the testing.

Considering the modelling undertaken for this research is intended to provide recommendations to the practitioner attempting to solve the LECP, wall clock time is therefore an undesirable basis for comparison in terms of MOEA performance and would serve poorly as a fair stopping criterion for this comparative analysis. Nonetheless, absolute time may be effectively used when applying a solution approach to the LECP in practice for non-comparative processes outside the scope of this research. The alternative method of using a maximum number of generations as the t_{max} basis was also not considered due to the variable degree of computational effort required to complete a generation, as well as the strategic intent behind the design of a generation, for each tested solution approach. For example, a generation for the NGSA-II approach includes highly computationally costly recombination and reparation processes, while a generation of the PAES algorithm constitutes a single mutation with no reparation required. However, the PAES strategy is to iterate through quickly through many generations and could therefore be prematurely stopped should t_{max} be defined in terms of number of generations.

The number of objective function evaluations was therefore chosen as the basis of t_{max} as it provides the fairest measure of algorithm age progression across all the solution approaches applied to the LECP. In accordance with the stated scope of this research, which excludes parameter optimisation, the t_{max} value was not specifically optimised for the datasets through extensive experimentation. However, t_{max} was chosen such that each optimisation run provided sufficient opportunity to converge to a solution regardless of its quality.

[25]The code is developed to test the MOEAs is a prototype that be optimised further. The objective was to develop prototypes that can be used to compare and determine the best alternative rather than optimising the execution of all the alternative prototypes. This may penalise some solution approaches more than others from an absolute time perspective.

6.8 Concluding remarks

Chapter 6 covered the MOEA-agnostic aspects of the algorithms that were implemented and tested using the case studies of this thesis. These designs, including the solution encoding, decoding and evolutionary operators, take the formulation of the LECP, presented in Chapter 5, into account and ensure their implementation would address the problem statement.

The evolutionary approach to the LECP of Chapter 6 can therefore be used and expanded upon by practitioners looking to optimize their own lane definitions without necessarily replicating the exact procedures used to address the problem instances of this exercise, which are explained in Chapter 7. As the design of some finer details of the procedures relied on some empirical feedback during testing, the algorithm design and results are inseparably linked. The results of the exploratory testing are therefore also discussed in Chapter 7.

Chapter 7

Exploratory analysis of solution approaches to the LECP

This chapter represents the first of two that describe the experimental component of this research. It explains how the prepared real-world data was used to verify the encoded evolutionary operators used by the tested MOEA solution approaches. Focus is given, however, to how the general evolutionary approach to the LECP of chapter 6 was applied to the three shipper datasets according to the MOEAs scoped for this research. Pseudocode of these approaches, definitions and figures are provided to support explanations when the developed algorithms differed materially to those presented in section 3.2.4, or if the specific procedure being described is not outlined in the literature review. The behaviour of each algorithm was investigated with some course parameter tuning and procedural adaptations applied as necessary. This chapter thereby addresses research objectives 2.4 of this thesis, which is to implement existing MOEAs to solve the LECP.

This chapter also includes explanations and discussions that address research objective 3.2, which is to establish the evolutionary approach as a viable method to solve the LECP. Algorithm logic was also validated using modified datasets and parameters.

7.1 Exploratory analysis methodology

7.1.1 Empirical verification and validation of evolutionary operators

The evolutionary operators were verified prior to any solution approach testing. Although all the empirical verification entailed some form of comparison of the model output to a prediction, a different approach was selected for each type of operation taking into consideration the nature of the inputs and outputs.

Fitness functions

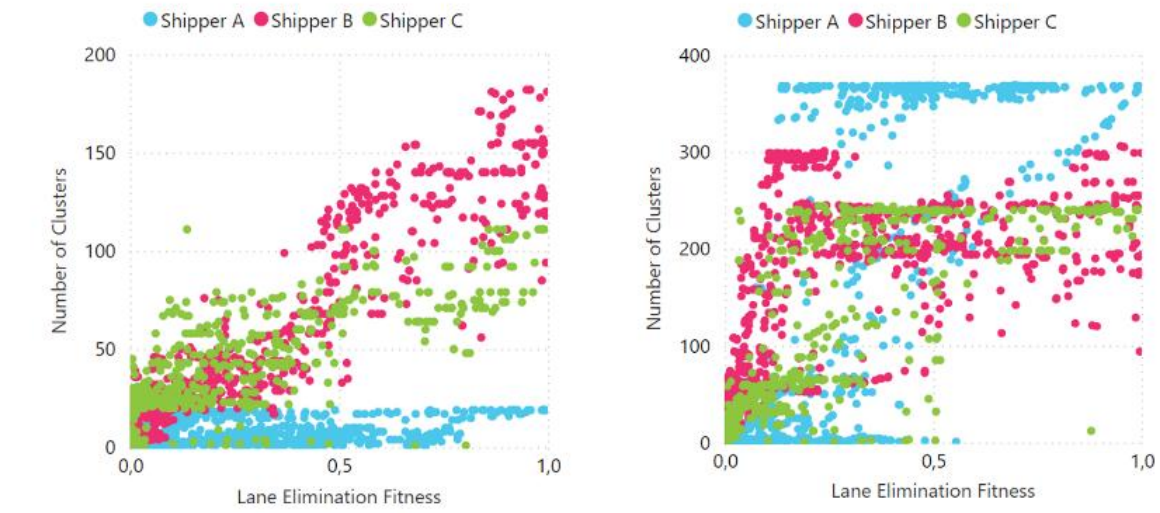
The fitness functions were verified by a spreadsheet exercise in Excel. Selected solutions were evaluated using the Python functions and their fitness values noted. They were then subject to an evaluation in Excel. The spreadsheet and Python fitness values were compared and found to be equal in all cases.

These functions were also validated by investigating the empirical relationship between the number of clusters in each cluster set and the calculated fitness values. The nondominated solutions to each problem instance, found by each MOEA after 12 800 fitness evaluations, were used for this analysis. For consistent measurement of the degree of associated clustering, the distance bracket value of each nondominated individual was replaced with a standard value of 150 kilometres.

Figure 99 shows the general positive correlation between the lane elimination fitness value and the number of clusters of each cluster set[26], which was expected. Figure 100 shows the expected general negative correlation between the cost penalty associated with a solution and the number of each type of cluster[27].

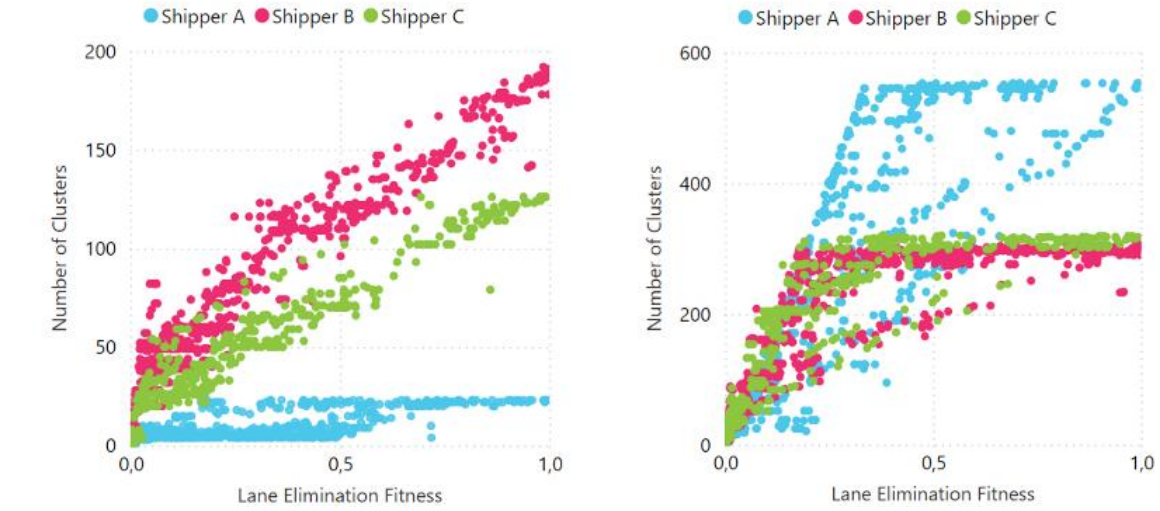
[26] Decreasing fitness values signify a higher degree of lane elimination, which is expected when fewer, larger clusters are used.

[27] Increasing cost penalty values signify a higher degree of lane elimination, which is expected when fewer, larger clusters are used.



a) Local origin clusters

b) Local destination clusters



c) Long-distance origin clusters

d) Long-distance destination clusters

Figure 99: Scatter plots showing the number clusters and lane elimination fitness values of the combined set of nondominated solutions generated by aggregation selection, NSGA-II, PAES, SPEA2 and PESA.

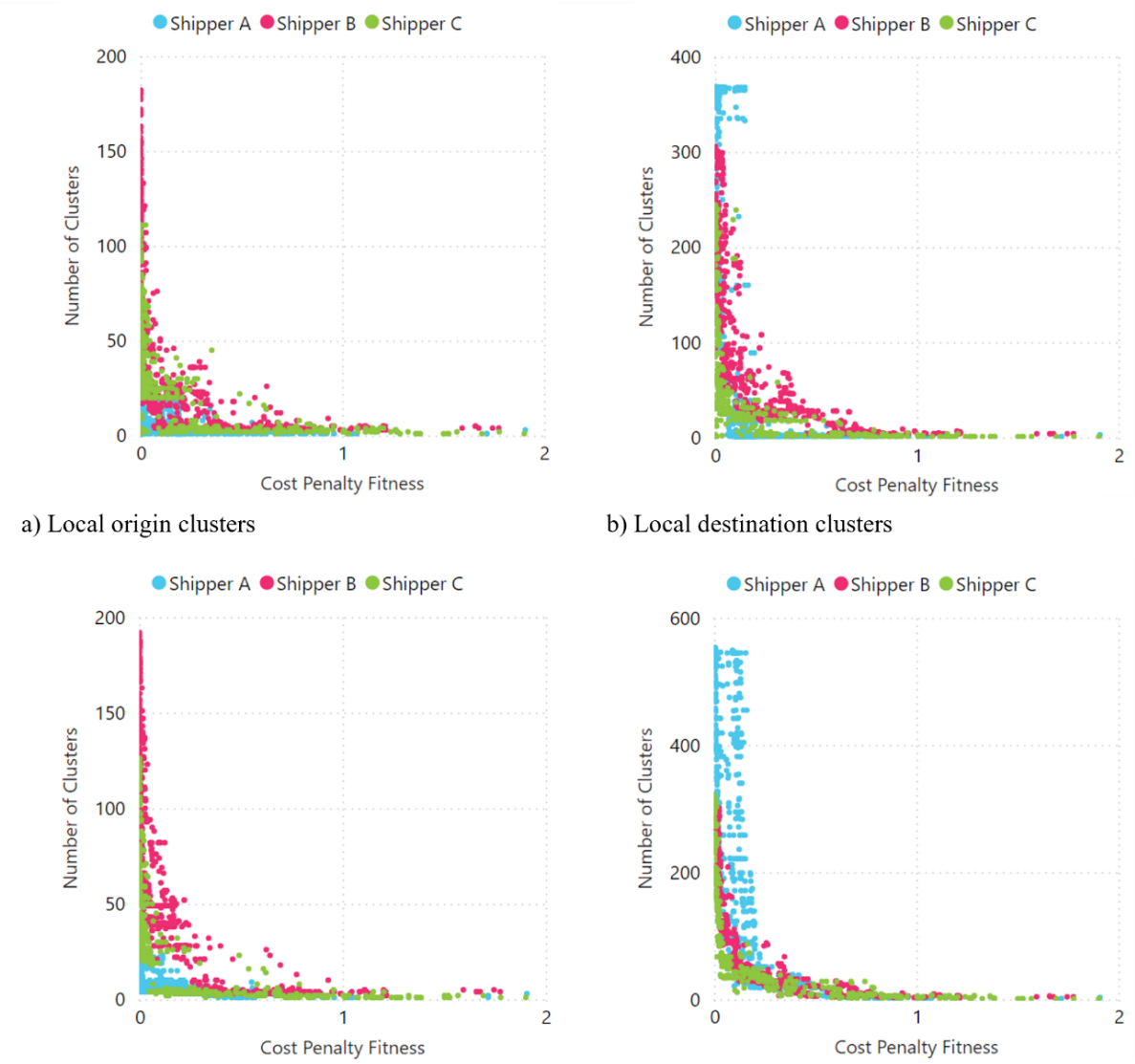


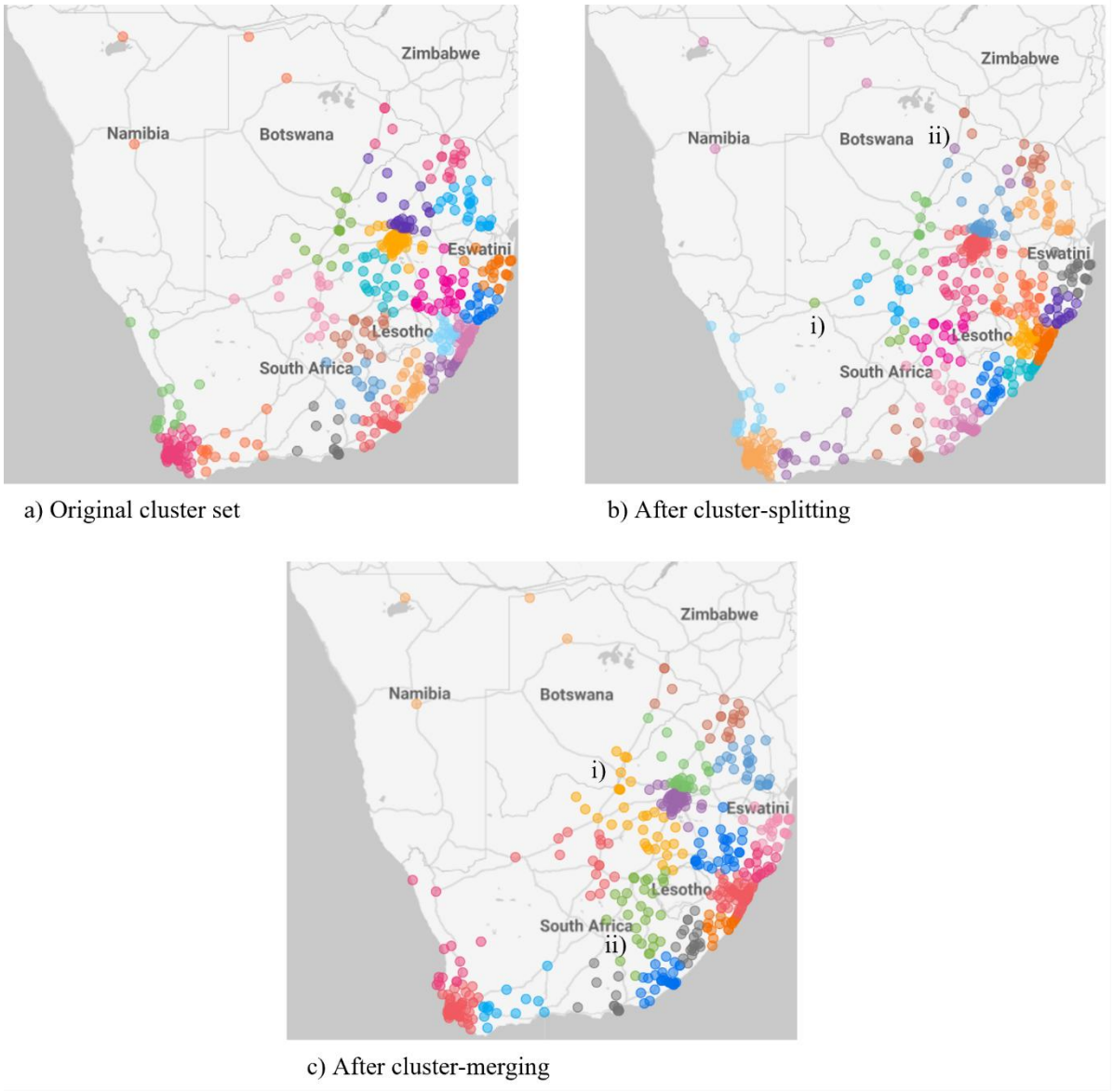
Figure 100: Scatter plots showing the number of clusters and cost penalty fitness values of the combined set of nondominated solutions generated by aggregation selection, NSGA-II, PAES, SPEA2 and PESA.

Mutation operators

The cluster-split and cluster-merge mutation operators were verified by visual analysis. Example solutions were processed by the Python functions. The geographic positioning of the nodes and associated clusters, shown by colour, were visualized. The extent and nature of the modification of the cluster scheme were identified and checked for issues.

Figure 101 provides an example of such a verification. Figure 101(a) shows an example combined local and long-distance cluster set for the shipper A problem instance of the LECP. Each colour represents a cluster. Figure 101(b) shows the same how two additional clusters were introduced, during verification, by the cluster splitting function with the new green cluster at i) and the new purple cluster at ii). These new clusters appear to have resulted from a linear splitting of a parent cluster. Figure 101(c) similarly

shows an example verification check of the cluster merging mutation operator. The clusters at i) and ii) appear to represent the combination of two previously separate adjacent clusters.



a) Original cluster set

b) After cluster-splitting

c) After cluster-merging

Figure 101: An example of a set of delivery cluster sets of a shipper A solution undergoing cluster-splitting and cluster-merging.

The distance modification function was verified by observing the genotypes change in this value during this process. These changes in the distance bracket value were confirmed to be working as designed.

7.1.2 Structure of MOEA threads

Each encoded test algorithm thread consisted of three components. These, to a large degree, shared parametrized functions encoded in Python, including the evolutionary functions developed in accordance with the evolutionary approach to the LECP and verified as the first part of the exploratory

analysis. These components were initiation, the evolutionary cycle, and the reproduction cycle. The descriptions of the MOEA procedures in this chapter are presented according to this framework.

Each run of an algorithm required the tester to select the problem instance, the solution approach and the initial population size. A set of standard starting populations, one for each tested initial population size, were generated by the k-means initialization method described in section 6.3 prior to any MOEA testing. Each experimental run of an algorithm therefore required only for the appropriate initial population to be imported, as opposed to generated. Although these banks of starting solutions had undergone fitness evaluation prior to testing, all individuals' cost penalty and lane elimination fitness values were recalculated during the initiation stage to simulate a true population initiation and to record the fitness evaluations in a consistent manner. Other specific processes executed during initiation are described separately for each MOEA in the remainder of this chapter.

Each iteration of the evolutionary cycle represented a population generation of the solution approach. An iteration usually began with an evaluation of the archive population of solutions and logging of population quality indicator values and other metrics of interest. The algorithms also checked against the termination condition at the beginning of each generation. The management of the archive and active populations, the selection of parents from these solutions and the integration of children into these populations usually represented the key distinctions between MOEAs and were implemented on the evolutionary cycle-level.

The mating cycle involved the iterative creation of children solutions. Each generation typically had one mating cycle, which continued until the requisite number of children were produced. This was usually a function of the target population size n, equal to the size of the initial population selected by the user. The mating cycle procedure was mostly consistent across the tested MOEAs, although it was sometimes a characteristic of the approach to employ a specific process during mating, which created some notable differences. These are described separately in the remainder of this chapter.

7.2 Baseline k-means

7.2.1 Procedure

Although it was previous recognized in literature as a business problem, this thesis is the first to introduce the LECP as a mathematical MOP. It follows that the problem instances for the experimental component of this research were newly sourced and prepared, and that benchmark data did not exist. However, the k-means clustering algorithm, as implemented for creating initial populations for the evolutionary approach to the LECP, provided an objective method to generate a non-evolutionary baseline against which the results of the five MOEA solution approaches could be compared.

The k-means method, described by section 6.3, was encoded in Python. A pool of 12 800 of such individuals was generated for each of the tested shippers from which solutions could be drawn during the progression of each problem instance's baseline run. An initial population of 800 individuals was used for a run, with 800 additional solutions introduced for each iteration.

An iteration consisted of the fitness evaluation of the new individuals, Pareto ranking of the new, expanded population, followed by evaluation of the nondominated section of this population using the metrics discussed in section 6.6. Each iteration was therefore tantamount to an MOEA generation, with no selection process followed and new individuals introduced by the population initiation method as opposed to crossover and mutation. This method thereby provided output data that indicated the quality of LECP solutions generated by a generic clustering approach, with minimal incorporated problem domain knowledge, as a function of computational effort in a manner consistent with the evaluation methodology used for the tested MOEAs.

Figure 102 describes a generic form of this process using pseudocode. It follows that variables *n* and *e* could be more specifically represented as 12 800 and 800 for the exercise completed for the experimental component of this research.

```
Read shipper rate card r, distance matrix d, population size n and evaluation
increment size e
Generate initial population of size n using k-means clustering
Create empty subpopulation
While (subpopulation size less than or equal to n)
    Add e new population members to subpopulation
    Evaluation new subpopulation member fitness
    Pareto-rank subpopulation (by dominance count)
    Evaluate subpopulation
```

Figure 102: Baseline k-means method for the LECP pseudocode.

7.2.2 Data analysis

Figure 103 shows the expected positive relationship between the number of fitness evaluations and the percentage of the theoretical maximum hypercube occupied by the population generated by the k-means method. It also shows that the population hypervolumes generated for shipper C are substantially higher than those for shippers A and B. This was expected, as this shipper's rate underwent a substantially

larger degree of deregionalization[28]. The original regions used to define these align closely with existing government-defined geographic boundaries. As such, there is a greater degree of underlying correlation between transport rates and the geographic positions of nodes and greater opportunity to eliminate lanes via clustering without significant cost penalty impact. This is also shown by the low-riding set of data points associated with shipper C shown in Figure 105.

Figure 104 shows a less uniform relationship between the number of fitness evaluations and population diversity, measured using HV_d. As the spread and uniformity of a population is not necessarily correlated with its associated convergence to the true Pareto front PF_{true}, and due to the absence of any kind of diversity preservation mechanism in the baseline k-means algorithm, this behaviour was unsurprising. This figure not only established the baseline HV_d values per shipper and number of fitness evaluations, but also established shipper B as the likely problem instance with the easiest achievable population diversity. A summary of the results shown in Figure 104 is provided in Table 10.

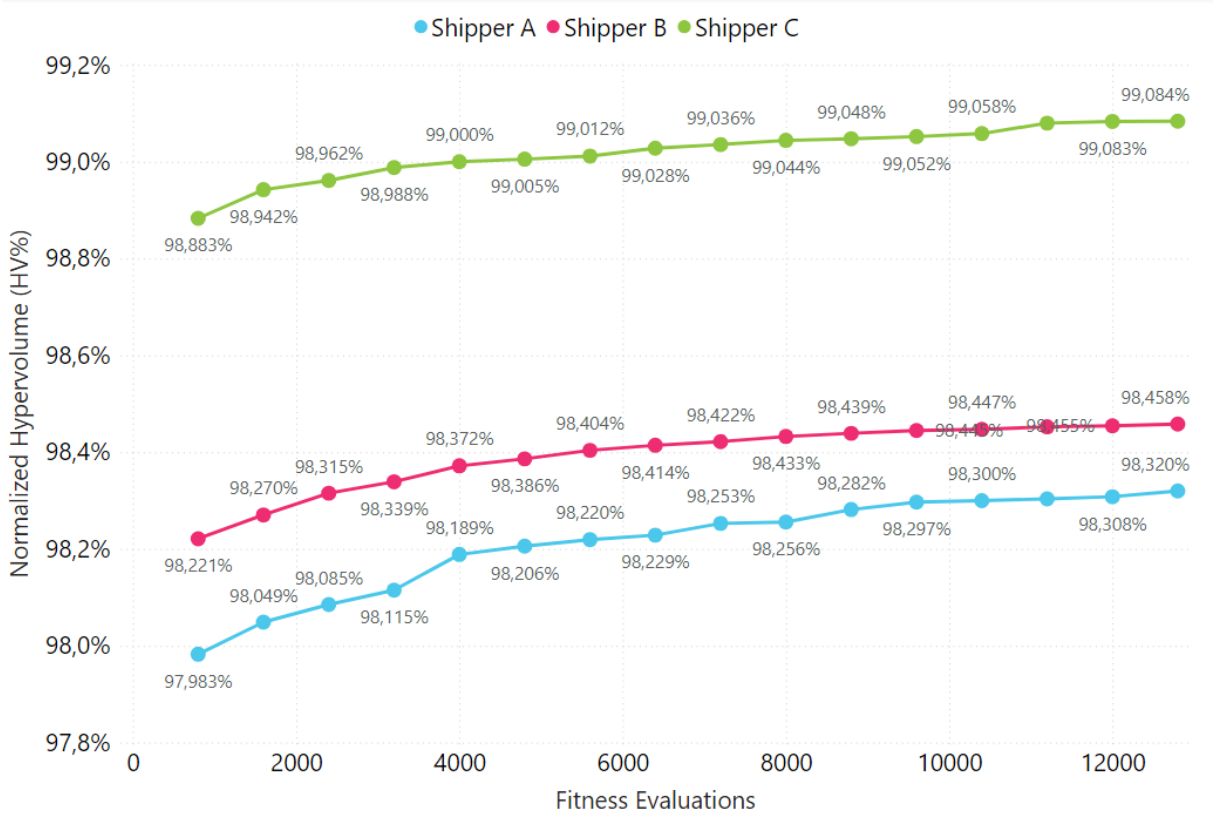


Figure 103: Line graphs showing the normalized population hypervolume of populations generated by the k-means baseline method as a function of number of fitness evaluations.

[28] See section 4.3.1 for an explanation regarding the deregionalization of the shipper rate cards.

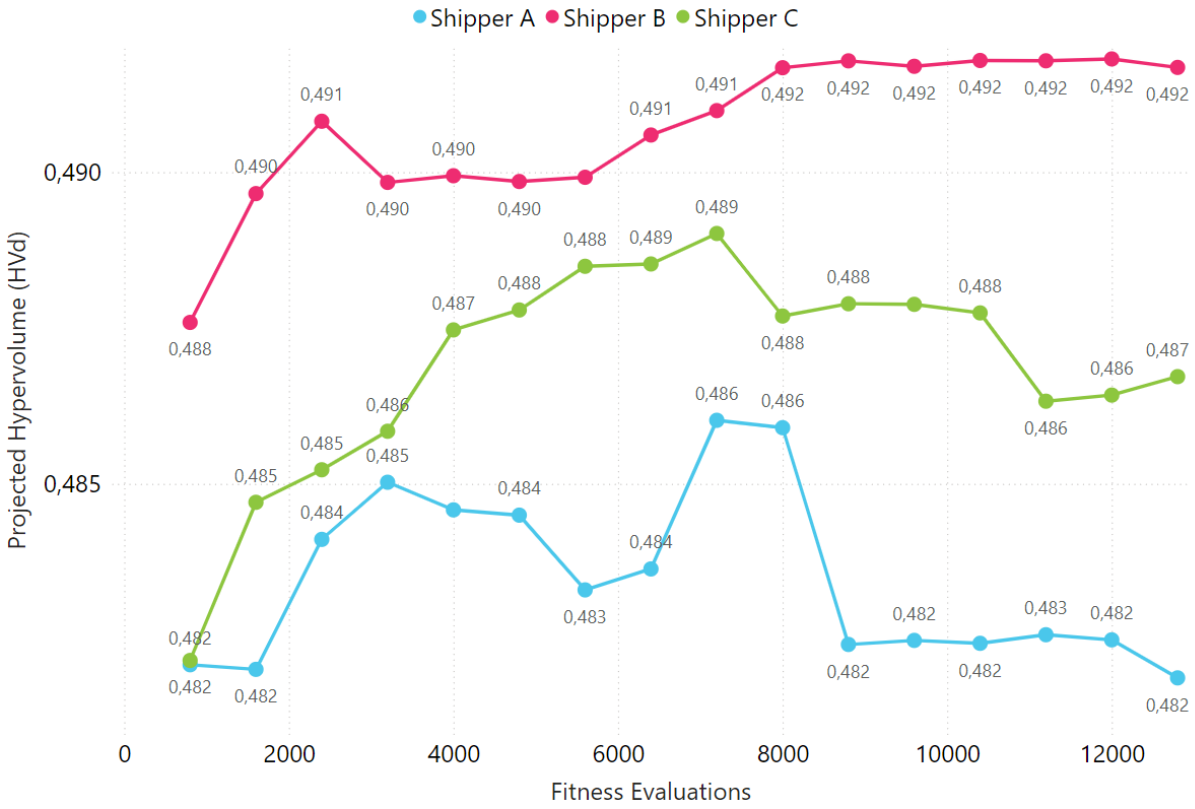


Figure 104: Line graphs showing the projected hypervolume of populations generated by the k-means baseline method as a function of number of fitness evaluations per shipper problem instance.

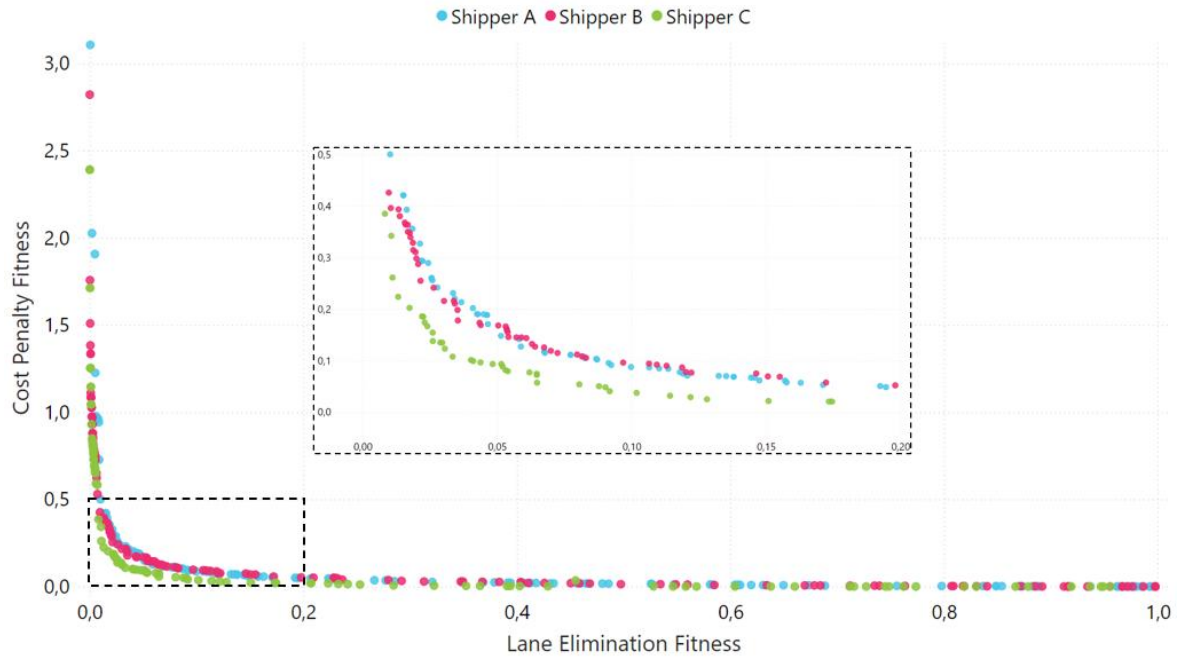


Figure 105: Scatter plot showing lane elimination fitness and cost penalty fitness values of the k-means baseline populations of 12 800 individuals for each test shipper problem instance.

Table 10: Evaluation scores of populations generated by the baseline k-means methods for each shipper consisting of 12 800 individuals.

Shipper	Hypervolume (HV)	Normalized Hypervolume (HV%)	Projected Normalized Hypervolume (HV_d)
A	3.055059	98.320	0.481887
B	2.778043	98.4582	0.491681
C	2.369517	99.0838	0.48672

The average number of clusters used for each cluster set generated by the baseline k-means method was also noted for each k-means baseline population of 12 800 members. Due to the distinction between local and long-distance clusters, these values could only be calculated once applied to the lanes represented in the rate card. This resulted in this measure being highly sensitive to the distance bracket value of the solution. A distance bracket of 150 kilometres was therefore used to ensure a fair comparison with populations generated by the MOEAs to be subsequently tested. This comparison would later serve as a baseline for validating the logical direction of each algorithm, as it would be expected that the ratios of local clusters to long-distance clusters would increase as the Pareto front approximation converges to the true Pareto optimal set.

Table 11: Average number of clusters per shipper cluster set when a standard distance of 150 kilometres is applied to the k-means baseline populations of 12 800 individuals.

	Origin Clusters		Destination Clusters	
Shipper	Average Number of Local Clusters	Average Number of Long-Distance Clusters	Average Number of Local Clusters	Average Number of Long-Distance Clusters
A	16.24	19.35	191.50	284.27
B	79.48	108.57	112.07	154.55
C	74.09	83.10	128.52	164.08

7.3 Aggregation selection with linear combination of weights

7.3.1 Procedure

The aggregation selection process described by Van Veldhuizen and Lamont (1998) was applied to the three problem instances using the operators designed for the evolutionary approach to the LECP presented in chapter 6.

The following thread was encoded in Python to replicate the aggregation selection algorithm as described in section 3.2.4. A starting population of solutions of a specified size was read. This parameter

was retained as the target running population size *n*. Lane elimination and cost penalty fitness values for the starting population were determined, thereby generating the active population for the first generation of the EA.

For each generation, a weighted average of these fitness scores was calculated for each solution using a randomly generated set of weighting coefficients. The active population then underwent a roulette selection process that eliminated half of the individuals according to their aggregated fitness values. The remaining solutions provided the parents for subsequent recombination and mutation.

Preliminary tests of the execution of the evolutionary operator functions revealed a significant difference in wall clock time associated with the cluster blending and cluster set swapping recombination operations. Each of these operators, as well as other key functions, were executed 1 000 times for each problem instance. Table 12 shows that the average duration to complete a cluster blending procedure is between 1 093 and 3 165 times the duration of a cluster set-level recombination.

Table 12: Average wall clock completion times of LECP fitness and crossover operators per tested problem instance.

	CPU time (s)		
Shipper	**Fitness evaluation**	**Cluster-level recombination**	**Cluster set-level recombination**
A	0.0714	19.9422	0.0063
B	0.6566	5.4632	0.0050
C	1.4345	10.1162	0.0051

The relative frequency of each type of recombination was therefore selected to provide a reasonably expedient solving process from a wall clock perspective while leveraging the benefits associated with cluster blending described in section 6.4.2. Each pair of parents reproduced by cluster blending with a probability of 0.1, while a probability of 0.9 was assigned to the use of cluster set-level recombination for generating child solutions. A pair of parents produced two children.

A mutation rate of 0.05 was used and each type of mutation was assigned an equal probability of being applied. As such, approximately a third of all mutants were produced by cluster merging, another third by cluster splitting, while the balance was generated via distance bracket modification. The children and parents entered the active population for subsequent selection, thus ensuring a stable active population size. This could therefore be described as a (2 + 2) evolutionary strategy.

A logging operation was performed at the end of each generation. The active population was added to a secondary archive population, which was then dominance ranked and reduced to only nondominated solutions. The archive and active population were also evaluated according to the measures defined for the evolutionary approach, with these metrics saved to a performance log.

7.3.2 Data analysis

The initial results, generated using only one shipper rate card (i.e. shipper A) and 2 500 fitness evaluations, showed that aggregation selection was prone to premature convergence. This is illustrated by the imperceptibly small rise of the blue lines in Figure 106 before they assumed almost perfectly horizontal orientations for the remainder of the runs. The low selection pressure of the method resulted in a rapid, significant decline in the quality of the active population from which the algorithm did not recover. This is shown by the sharp decline of the blue lines of Figure 107, which stabilize between 20% and 70% of their original quality of convergence. Thereafter, introduction of new solutions to the Pareto front approximation PF_{known} dwindled. This is shown by the blue lines of Figure 108.

Inspection of the generated populations revealed that an underlying cause was the destruction of genetic diversity due to the high-level cluster-swap crossover method. While computationally efficient, it works with relatively large building blocks[29]. These building blocks, namely the cluster sets and the distance bracket value, when associated with highly fit individuals, were often repeated in child solutions and continuously dominated others in the selection process. Dominant building blocks therefore quickly propagated through the population as the EA progressed. It was observed that the active population would often rapidly converge to a set of identical individuals, especially when small populations were used.

An increase in the proportion of cluster-blend operations or the mutation rate might mitigate this loss of genetic material. This, however, would be achieved at the cost of significant increases in wall clock time.

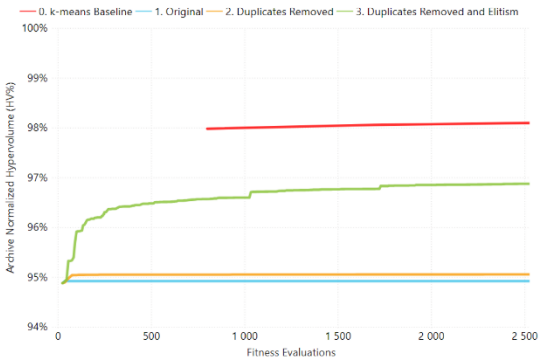
An adaptation was therefore introduced to the algorithm itself an as alternative method to address the premature convergence issue. Each child, upon creation, was checked for duplicate solutions in the active population of parents and children. If found to be a duplicate, the child was discarded and a replacement child created. To ensure genetic diversity was reintroduced under these circumstances, the crossover method was automatically set to cluster-level recombination to produce the replacement child. As a further measure, all replacement children also underwent mutation.

[29] See section 6.4.1 for a detailed explanation regarding the probability of two parents producing duplicate children when this type of recombination is applied.

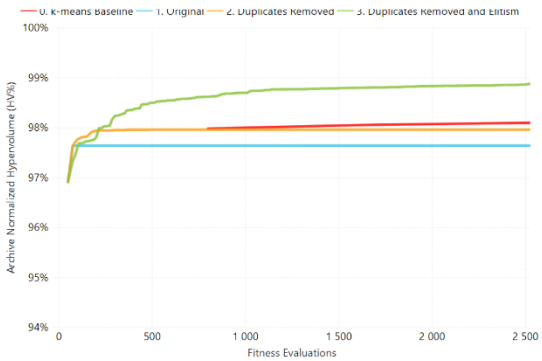
The reproduction loop of the algorithm had to be adapted to ensure that parents mated to fill up the population until the active set reached n even when duplicates were detected. This meant parents that had already mated often had to reproduce again. This was achieved by restarting the usual reproduction process at a random position of the parent set when all parents had mated.

This strategy of continuously filtering duplicate child solutions achieved the desired effect of significantly delaying the convergence of the algorithm. This was shown by the orange lines of Figure 106, Figure 107 and Figure 108. However, even with duplicate removal, the HV% of the archive population only exceeded that of the baseline method in two of the six trial runs.

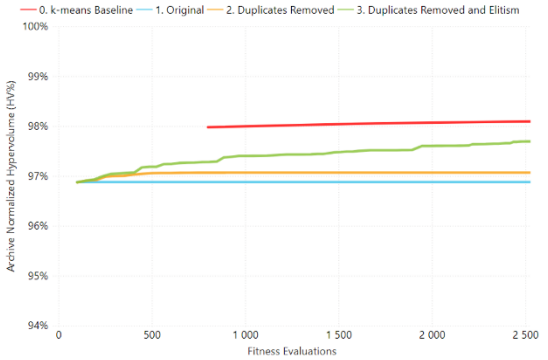
Elitism was therefore subsequently introduced to this solution approach to further mitigate its tendency to prematurely converge. The archive population was appended to the active population prior to roulette selection. This larger, higher quality pool of candidate solutions stymied the deterioration of the active population. It did, however, introduce a variable active population size, which was inconsistent with population management policy of the previous versions of this evolutionary algorithm. This was addressed by implementing an iterative selection process by which the roulette selection operator was repeatedly applied until the number of selected parents fell below n. The green lines of Figure 106 and Figure 107 show the marked improvement in the performance of the algorithm with the implementation of elitism. The green lines of Figure 108 specifically highlight that duplicate removal and elitism mitigated premature convergence.



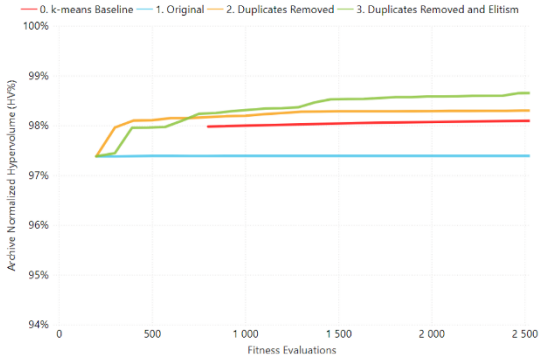
a) Initial population of 25 individuals



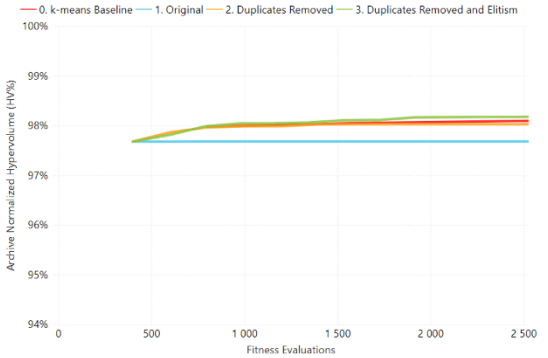
b) Initial population of 50 individuals



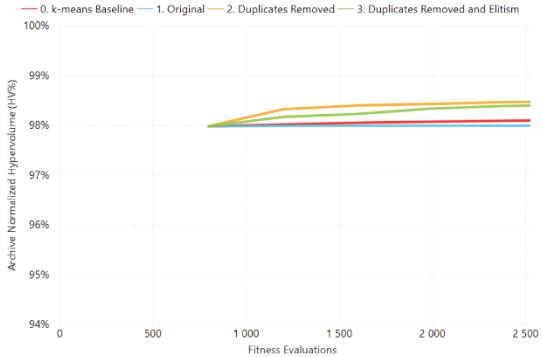
c) Initial population of 100 individuals



d) Initial population of 200 individuals



e) Initial population of 400 individuals



f) Initial population of 800 individuals

Figure 106: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per run of the aggregated selection method.

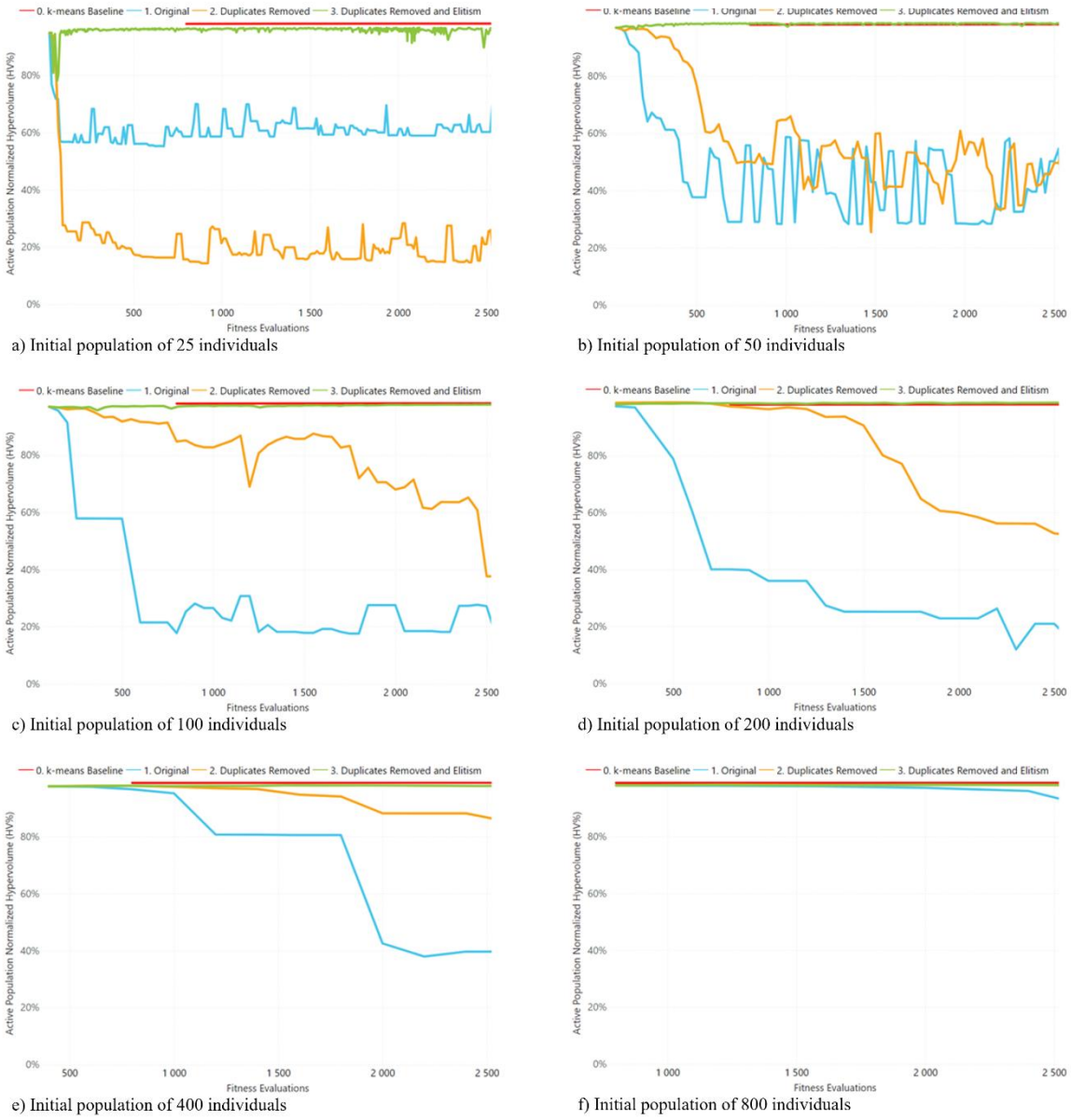
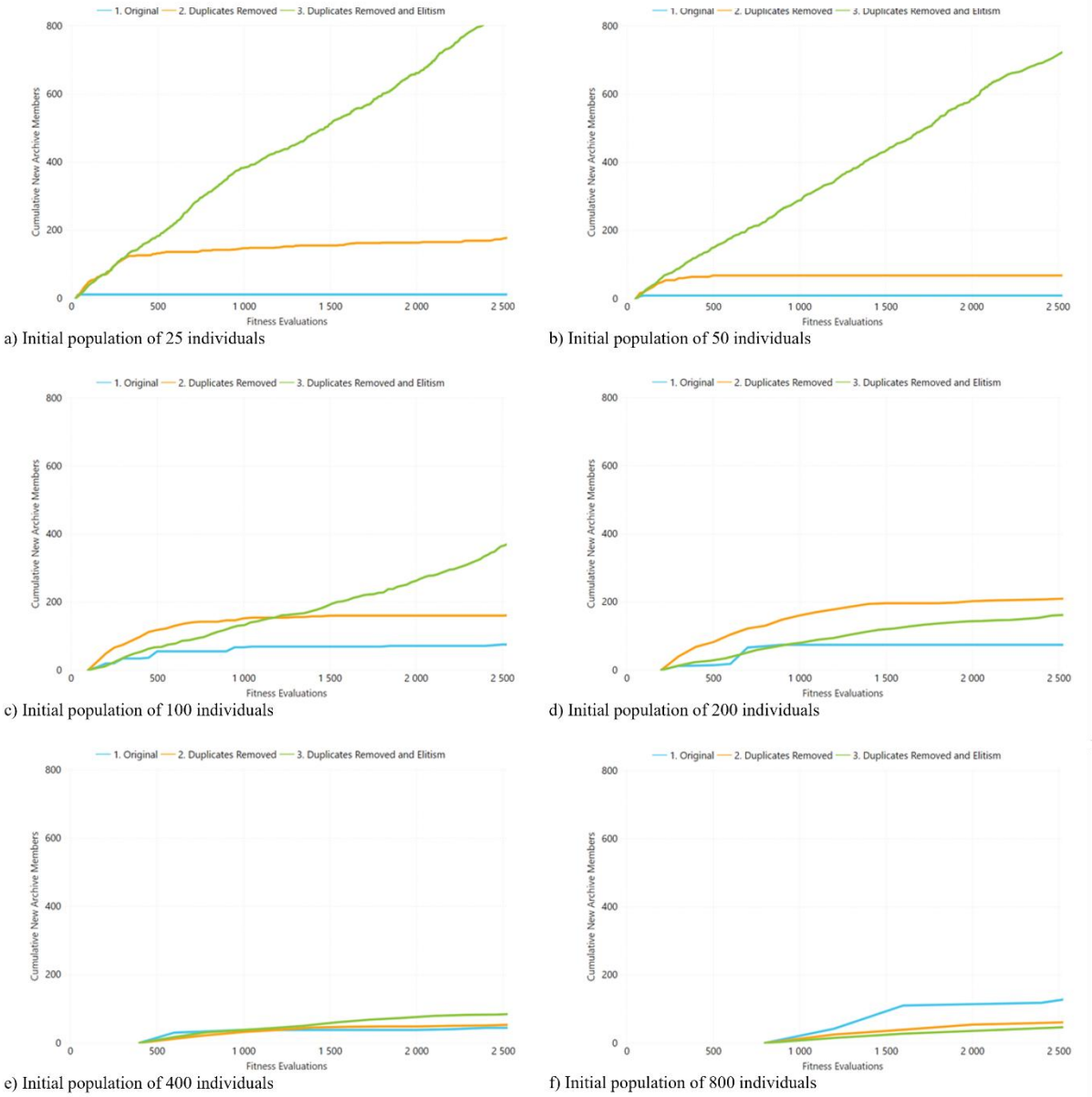


Figure 107: Line graphs showing normalized hypervolume of the active population as a function of number of fitness evaluations per run of the aggregated selection method.



a) Initial population of 25 individuals

b) Initial population of 50 individuals

c) Initial population of 100 individuals

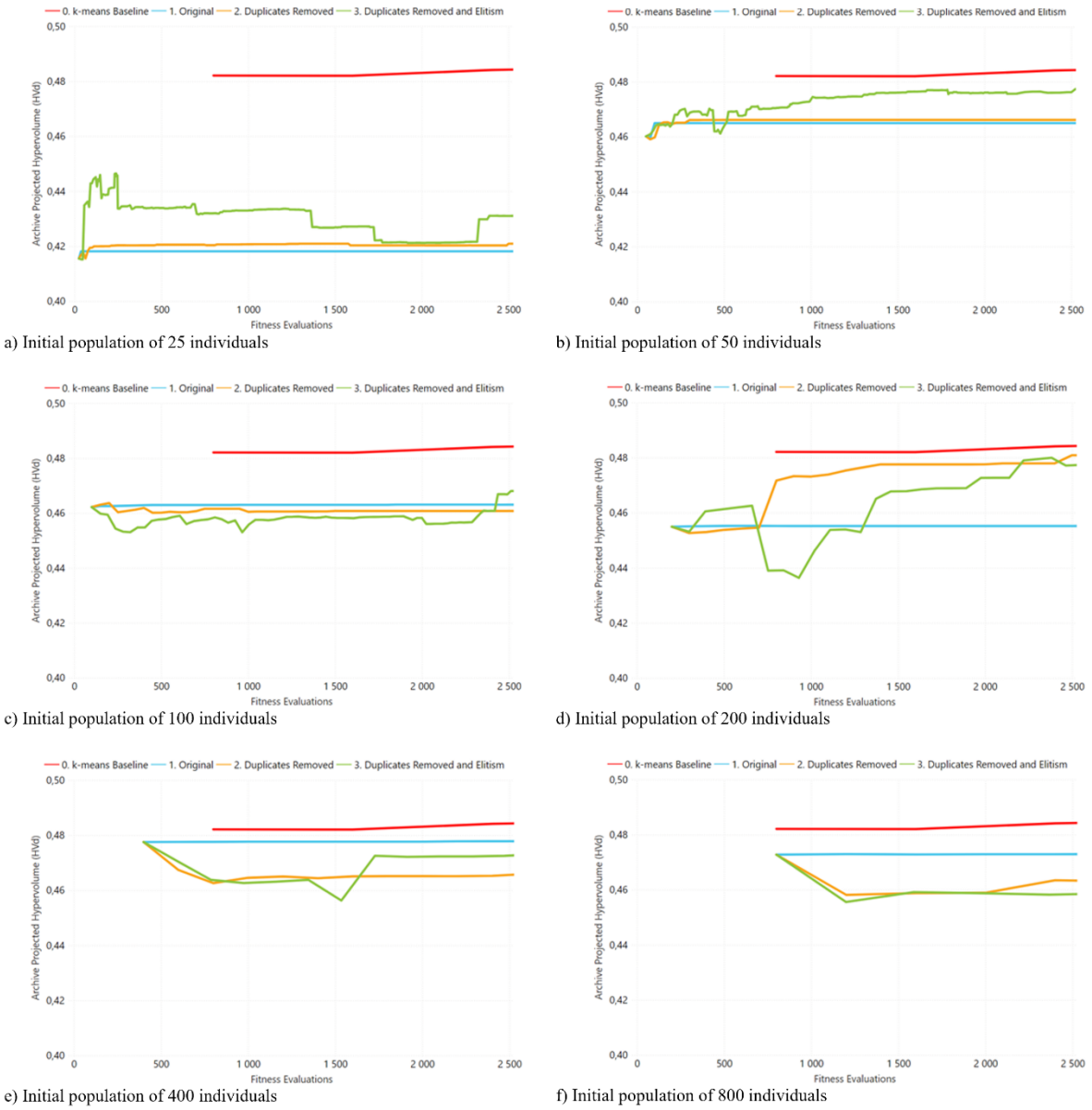
d) Initial population of 200 individuals

e) Initial population of 400 individuals

f) Initial population of 800 individuals

Figure 108: Line graphs showing the number of solutions generated and added to the archive population as a function of number of fitness evaluations per run of the aggregated selection method.

It is apparent that aggregation selection with linear combination of weights, with elitism, is a viable method of solving the LECP from the perspective of convergence to the true Pareto front PF_{true}. However, Figure 109 shows that, from a diversity standpoint, the k-means baseline method was significantly superior to any version of the aggregation selection method for all the tested initial populations. This can be attributed to the even distribution of numbers of clusters used to seed the k-means algorithm when populations are generated by this technique and the drift of the genetic algorithm away from this distribution as individuals are iteratively selected with no consideration of diversity.



a) Initial population of 25 individuals

b) Initial population of 50 individuals

c) Initial population of 100 individuals

d) Initial population of 200 individuals

e) Initial population of 400 individuals

f) Initial population of 800 individuals

Figure 109: Line graphs showing projected hypervolume of the archive population as a function of number of fitness evaluations per run of the aggregated selection method.

The same set of tests were performed using the other shipper rate cards to determine whether this behaviour was specific to the problem instance. The results, collected only for an initial population size of 200, showed that the effects of duplicate removal and elitism were generally consistent with those observed for first tested shipper. However, aggregation selection, for these shippers, achieved a population diversity that was significantly closer to that achieved by the k-means baseline method. The poor performance of this evolutionary method in terms of diversity, observed for the first tests, might therefore be attributed to characteristics of the specific shipper rate card. It is also apparent that the introduction of elitism contributed to this desirable behaviour, which was not always the case for the first shipper instance.

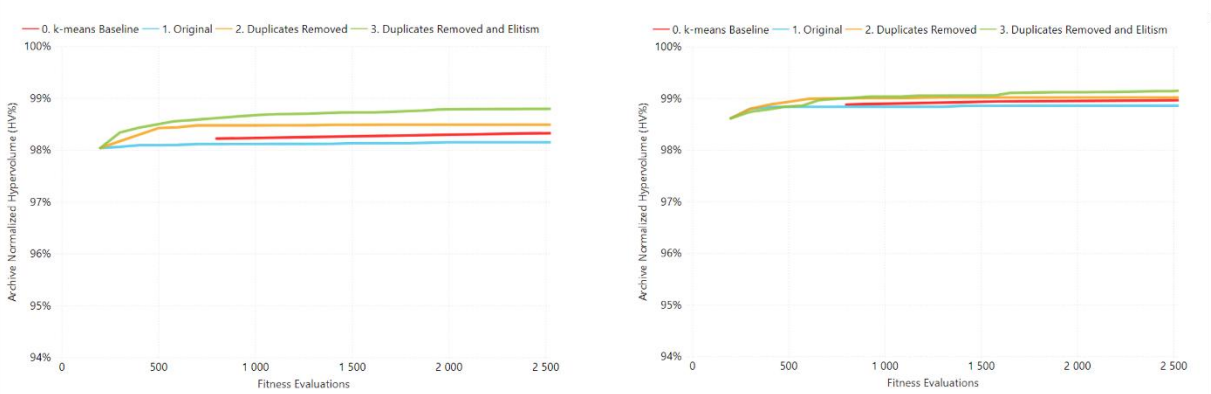


Figure 110: Line graphs showing HV% of the archive as a function of number of fitness evaluations per adaptation run of the aggregated selection method for an initial population size of 200.

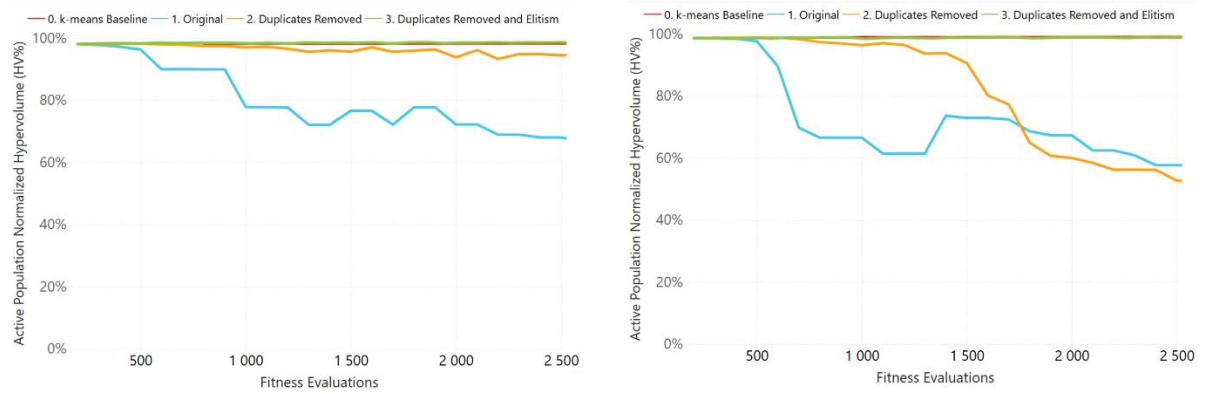


Figure 111: Line graphs showing HV_d of the active population as a function of number of fitness evaluations per adaptation run of the aggregated selection method for an initial population size of 200.

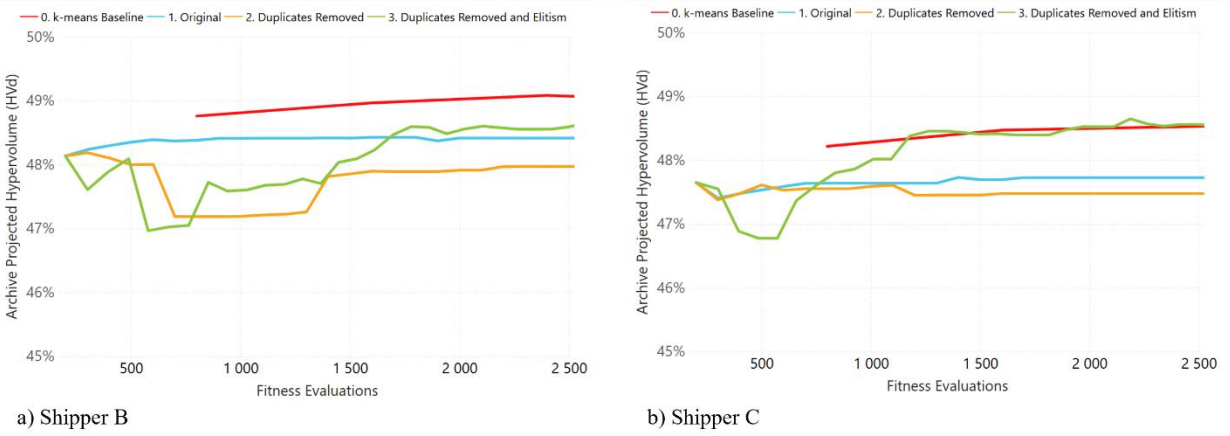


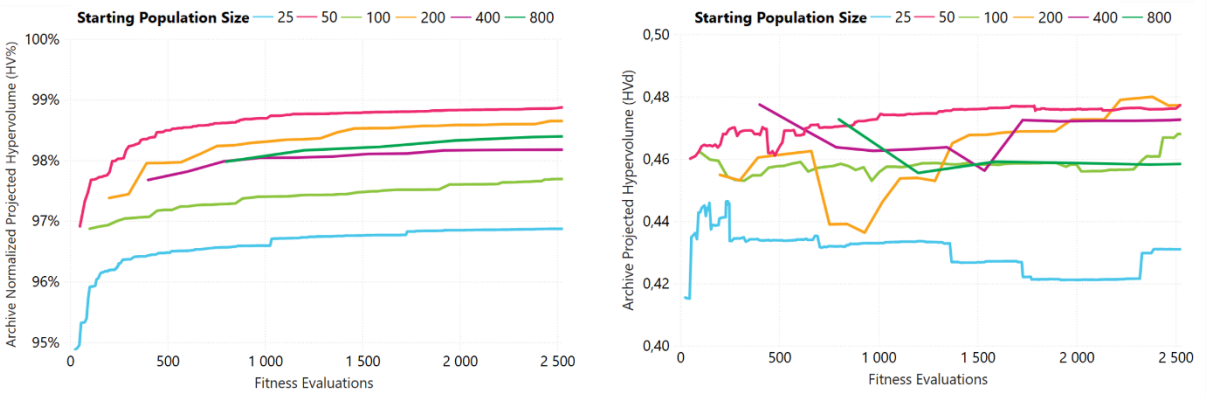
Figure 112: Line graphs showing HV_d of the archive as a function of number of fitness evaluations per adaptation run of the aggregated selection method for an initial population size of 200.

The final algorithm, adapted for the LECP, of the aggregation selection with linear combination of weights solution approach is represented by the pseudocode of Figure 113.

```
Read shipper rate card r, distance matrix d, population size n and maximum fitness evaluations
size f
Read initial population of size n as active population
Create empty archive
Initialize fitness counter variable = 0 and duplicate counter variable = 0
While (fitness counter less than f)
    Generate random fitness weight coefficients
    Calculate aggregate fitness for each solution in active population
    Perform roulette selection to generate parent set
    Copy active population as parent population
    While (active population size less than n)
      If (number of unmated parents in parent set less than 2)
        Select random parent and mark all parents below it as unmated
      If (duplicate counter is greater than 0)
        Generate two children using first two unmated parents in parent set by cluster blending
        recombination
      If (duplicate counter is equal to 0)
        Generate two children using first two unmated parents in parent set either by cluster
        blending or cluster set-level recombination according to assigned probabilities
      Mark parents as "mated"
      Mutate children according to mutation rate and random mutation type
      // Apply duplicate removal
      Check children for identical solutions in active population
        If (child matches an active population member)
          Elimate child
          Increment duplicate counter
      Evaluation children fitness and add to active population
    // Apply elitism
    Add archive to active population to create new active population
    Remove duplicate solutions from active population
    Pareto-rank active population (by dominance count)
    Replace archive with nondominated solutions from active population
    Evaluate active population and log results
```

Figure 113: Aggregation selection with linear combination of weights adapted for the LECP pseudocode.

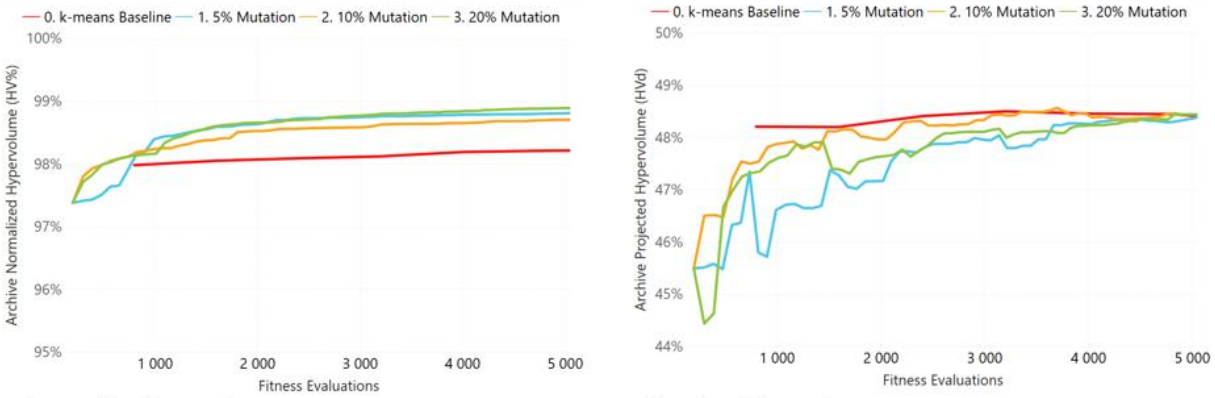
The convergence to PF_{true} and diversity achieved by the aggregation selection algorithm with different starting population sizes was compared. The visualization is shown in Figure 114. The best convergences to PF_{true} were achieved with initial population sizes of 50, 200 and 800. The 50 and 200 runs also achieved the best diversity, while the 800-run fared relatively poorly in this aspect. The run using 400 starting population members, however, produced a more consistent performance across these two metrics with yield good convergence to PF_{true} and diversity.



a) Normalized hypervolume b) Projected hypervolume

Figure 114: Line graphs showing a) normalized and b) projected hypervolume of the archive as a function of number of fitness evaluations when the aggregated selection method.

A final test was performed to determine if the adapted aggregation selection method's convergence to PF_{true} beyond 2 500 fitness evaluations would be enhanced by an increased mutation rate. The algorithm was rerun for 5 000 evaluations using shipper A's rate card and an initial population of 200 using the original mutation rate of 0.05, as well as rates of 0.1 and 0.2. The output of this test, shown by Figure 115, suggests that this MOEA does not benefit from an increase in mutation.



a) Normalized hypervolume b) Projected hypervolume

Figure 115: Line graphs showing a) normalized hypervolume and b) projected hypervolume of the archive as a function of number of fitness evaluations per mutation rate used with aggregation selection.

7.4 Nondominated sorting genetic algorithm II (NSGA-II)

7.4.1 Procedure

Deb's (2002) improved version of the nondominated sorting genetic algorithm (NSGA-II) was the first true MOEA that was used for the experimental component of this research. The results of the aggregation selection method testing provided an input to the testing approach to the NSGA-II. The premature convergence due to the duplication of solutions by cluster set-level recombination, empirically shown when aggregation selection was tested, was assumed to be solution approach-agnostic. As such, the duplicate removal strategy that proved an effective mitigation against this undesirable behaviour was carried over from the adapted aggregation selection method to the NSGA-II, as well as the other solution approaches subsequently tested.

A Python code thread was used to implement the NSGA-II MOEA, including the duplicate removal strategy, to solve the LECP according to the following procedure. The starting population was imported and its members evaluated in terms of lane elimination and cost penalty fitness. Like the aggregation section process, the specified initial population size was retained as a target active population size n. Each member of this population was then evaluated. After an empty archive population was created, the algorithm entered the iterative process of evolving this active population, generation upon generation, until the specified number of fitness evaluations was reached.

The first stage of an iteration updated the archive. The active population was copied to the archive, which was then dominance-depth ranked. The archive was subsequently pruned of all dominated solutions.

The second stage used the active population as parents to generate new individuals according to a (2 + 2) evolutionary strategy. Children were generated by the crossover and mutation operators using the same procedure and parameters used for aggregation selection testing. The duplicate removal step was also executed in an identical manner. However, for NSGA-II, children were produced until the combined set of parents and offspring reached 150% of the target population size, in accordance with the approach outlined by Deb (2002).

The third stage of a generation performed selection in a manner like that presented by Deb (2002). Upon completion of all mating, the children and parents were combined as the new active population. The archive was also added to this new set of solutions. This new merged population was then dominance-ranked and the crowding distance for each member was calculated. As such, both the active and archive solutions were considered when the proximity of neighbours, belonging to same Pareto front, in phenotype space were quantified. A value of 50 was assigned to the extreme points of each front to

ensure these were assigned an arbitrarily large score[30]. The algorithm for calculating crowding distance is shown by the pseudocode of Figure 116.

Read the dominance ranked active set of solutions P

Determine the number g of dominance ranks in P

for i=1 to g **do**

Selection of individuals assigned to Pareto front i $P \rightarrow P^i$

Sort P^i by lane elimination fitness value

for j=1 to number of individuals in P^i **do**

If (individual j is first or last of P^i)

Then set lane elimination fitness crowding value to 50

Else

Then set lane elimination fitness crowding to

$$Lane\ Elimination\ Fitness_j + \frac{Lane\ Elimination\ Fitness_{j+1} - Lane\ Elimination\ Fitness_{-1}}{Lane\ Elimination\ Fitness_{max} - Lane\ Elimination\ Fitness_{min}}$$

end for

Sort P^i by cost penalty fitness value

for j=1 to number of individuals in P^i **do**

If (individual j is first or last of P^i)

Then set cost penalty fitness crowding value to 50

Else

Then set cost penalty fitness crowding to

$$Cost\ Penalty\ Fitness_j + \frac{Cost\ Penalty\ Fitness_{j+1} - Cost\ Penalty\ Fitness_{-1}}{Cost\ Penalty_{max} - Cost\ Penalty\ Fitness_{min}}$$

end for

for j=1 to number of individuals in P^i **do**

Set crowding distance to $Lane\ Elimination\ Fitness_j + Cost\ Penalty\ Fitness_j$

end for

end for

Figure 116: Crowd distance calculation for the LECP pseudocode.

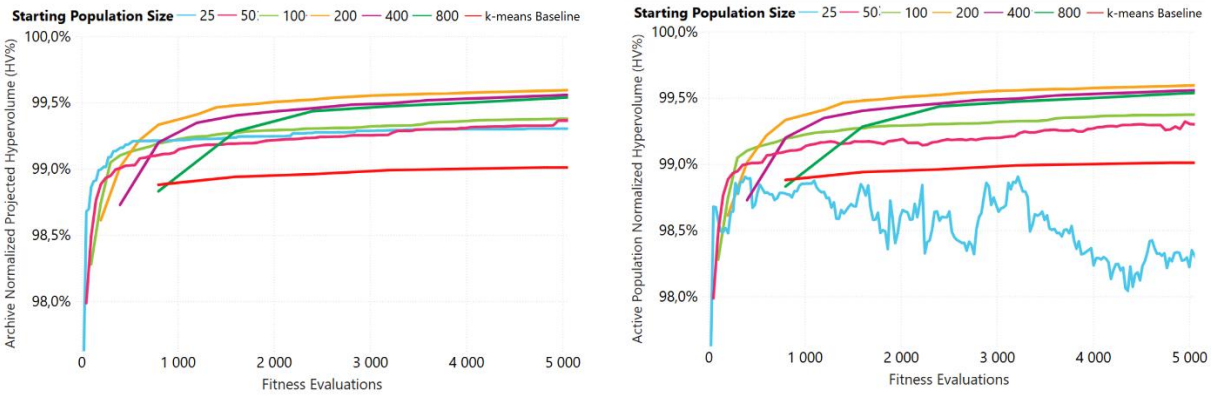
The merged population could now be reduced to n. The primary selection criterion was dominance depth, while crowding distance was the secondary consideration. As such, should the number of solutions in the nondominated front of the combined population have exceeded n, only the n best individuals, according to crowding distance, were retained as the active population for the next generation. If this was not the case, the full nondominated set was retained, with the remainder drawn

[30] A large crowding distance is favourable in the context of an NSGA-II and its associated selection procedure.

from the second, third, fourth front, and so on, until the target population size was met. The crowding distance always provided the tie-breaker criterion in the likely event that the final considered Pareto front required trimming to complete the active population to the exact target size n.

7.4.2 Data analysis

The behaviour of the adapted NSGA-II was initially explored using shipper C's rate card. This MOEA demonstrated rapidly convergence toward the true Pareto front PF_{true} with all tested target population sizes n, as shown by Figure 117(a). This can be attributed to the elitism of the selection method, which protects that the quality of the active population as the evolutionary algorithm progresses, which is shown by Figure 117(b).



a) Archive population b) Active population

Figure 117: Line graphs showing normalized hypervolume as a function of number of fitness evaluations per initial population size when NSGA-II was tested using the shipper C problem instance.

The only run that showed any significant decline in active population convergence to PF_{true} during the first 5 000 fitness evaluations used a starting population of 25 members. Such a decline can be attributed to the crowding distance consideration as a tiebreaker when selection between members of the same Pareto front took place. This could result in solutions occupying sparsely populated regions of phenotype space being preferred over individuals that were more desirable from a population hypervolume perspective. Runs that use relatively small starting populations are especially susceptible to this behaviour, as the archives rapidly swell and exceed n. This is shown in Figure 118. Selection, in these cases, only considers the first Pareto front on the merged population, thereby placing more emphasis on the crowding measure as a selection criterion. Nonetheless, despite the initial quality degradation of the active population, this population for the 25-run showed resilience, with an improvement in convergence to PF_{true} after the decline.

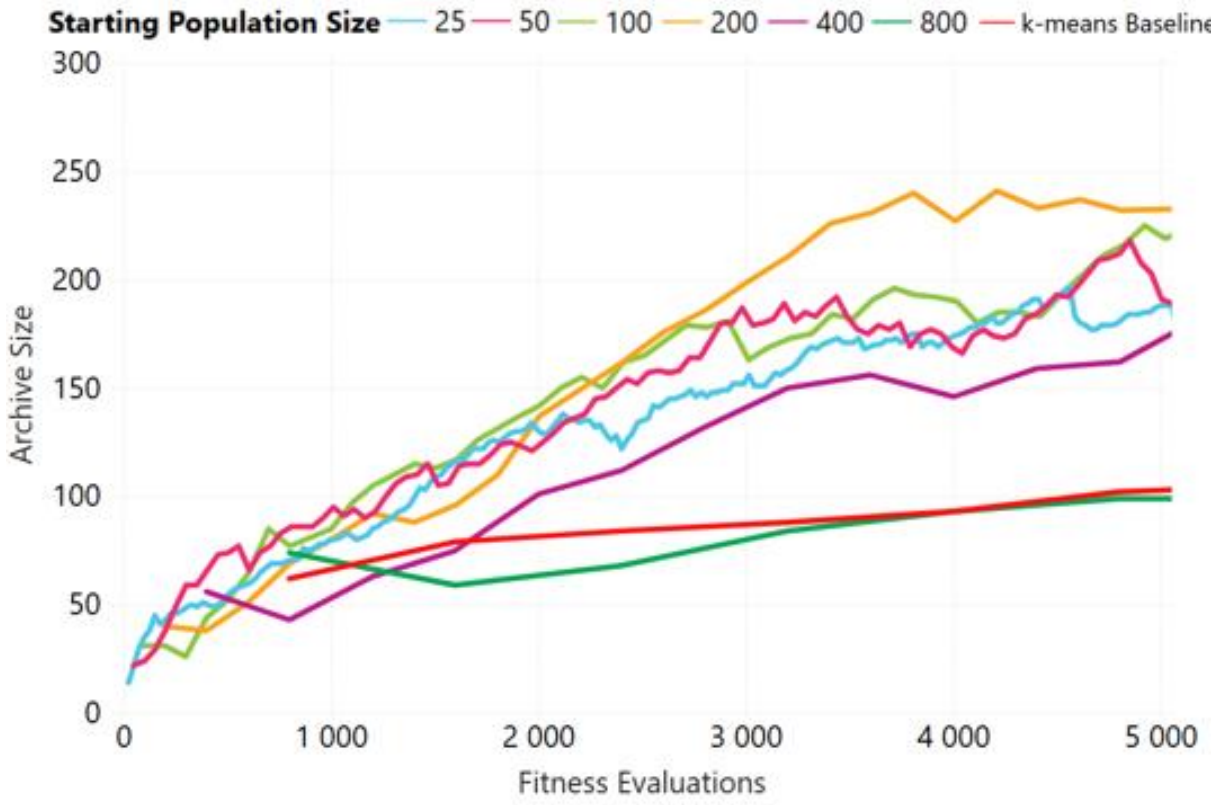


Figure 118: Line graphs showing the number of individuals of the archive as a function of number of fitness evaluations per initial population size when NSGA-II was tested.

The population diversity achieved by the NGAS-II was, in most cases, comparable with that achieved by the k-means baseline[31]. This can be attributed to the crowding distance consideration of its selection process. Furthermore, NSGA-II produced superior diversity, according to the projected hypervolume measure, in two of the six initial runs. These results are shown by Figure 119. Like the temporary loss in active population convergence to PF_{true}, the relatively rapid diversity improvement seen when a starting population of 25 solutions was also due to archive swell.

[31]It was explained in section 7.2 that the k-means baseline method was well-suited to achieve good population diversity.

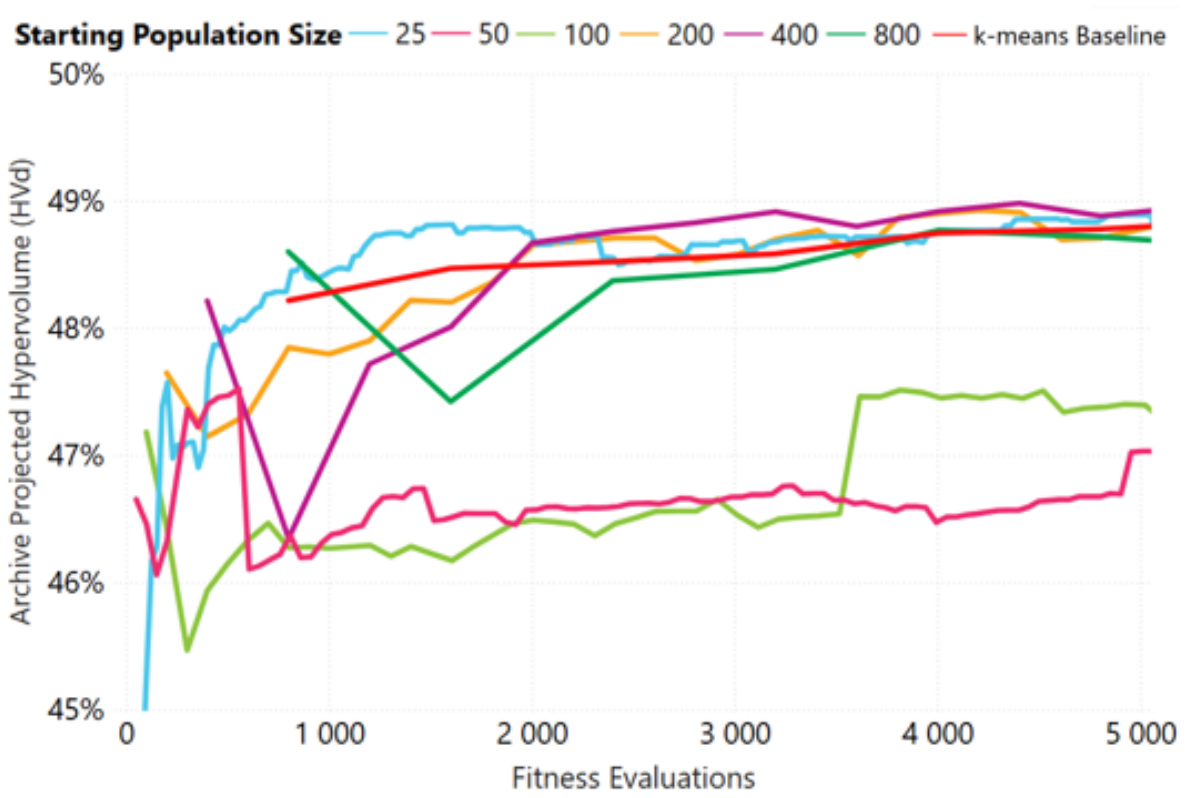
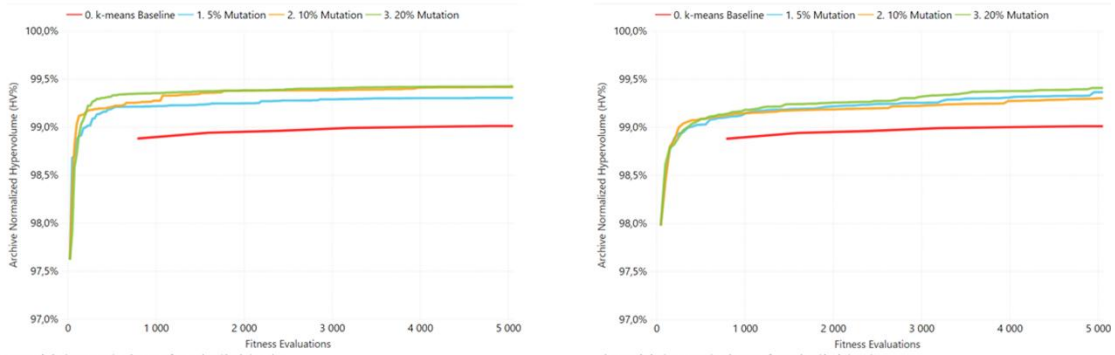


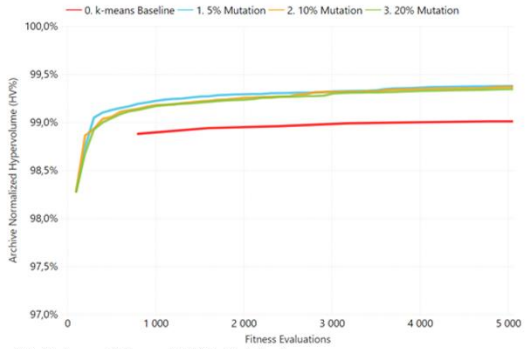
Figure 119: Line graph showing projected hypervolume of the archive as a function of number of fitness evaluations per initial population size when NSGA-II was tested using the shipper C problem instance.

Analysis of the normalized hypervolumes achieved by NSGA-II indicated that little improvement could be achieved by this MOEA for this problem instance after approximately 2 000 fitness evaluations had been completed. While diminishing returns were expected, the mutation rate used for this solution approach was adjusted for each tested initial population size to determine if more aggressive exploration of the solution space might yield improved performance in this regard. Figure 120 shows that this strategy did not significantly improve NSGA-II's convergence to PF_{true}. Furthermore, Figure 121 shows that, in some cases, the increase in mutants resulted in fewer introductions of new solutions to the archive over the course of the run.

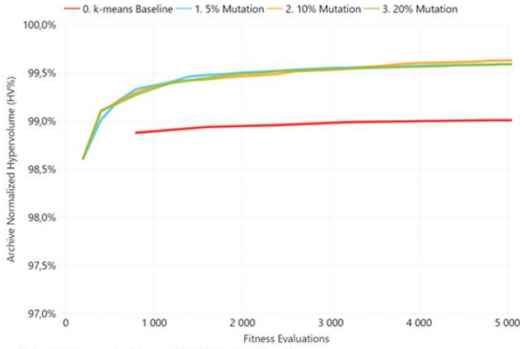


a) Initial population of 25 individuals

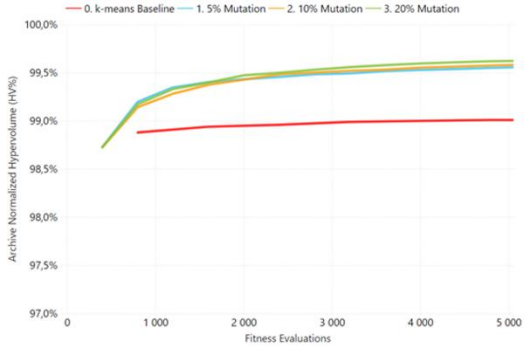
b) Initial population of 50 individuals



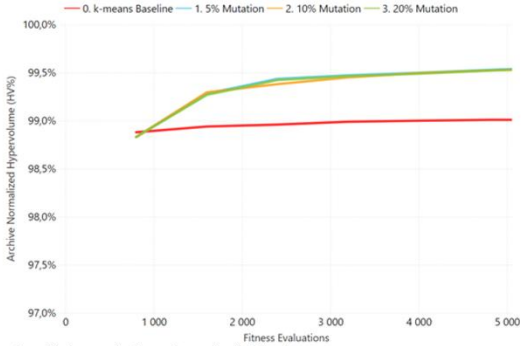
c) Initial population of 100 individuals



d) Initial population of 200 individuals



e) Initial population of 400 individuals



f) Initial population of 800 individuals

Figure 120: Line graphs showing normalized hypervolume of the archive as a function of number of fitness evaluations per mutation rate used with NSGA-II for the shipper C problem instance.

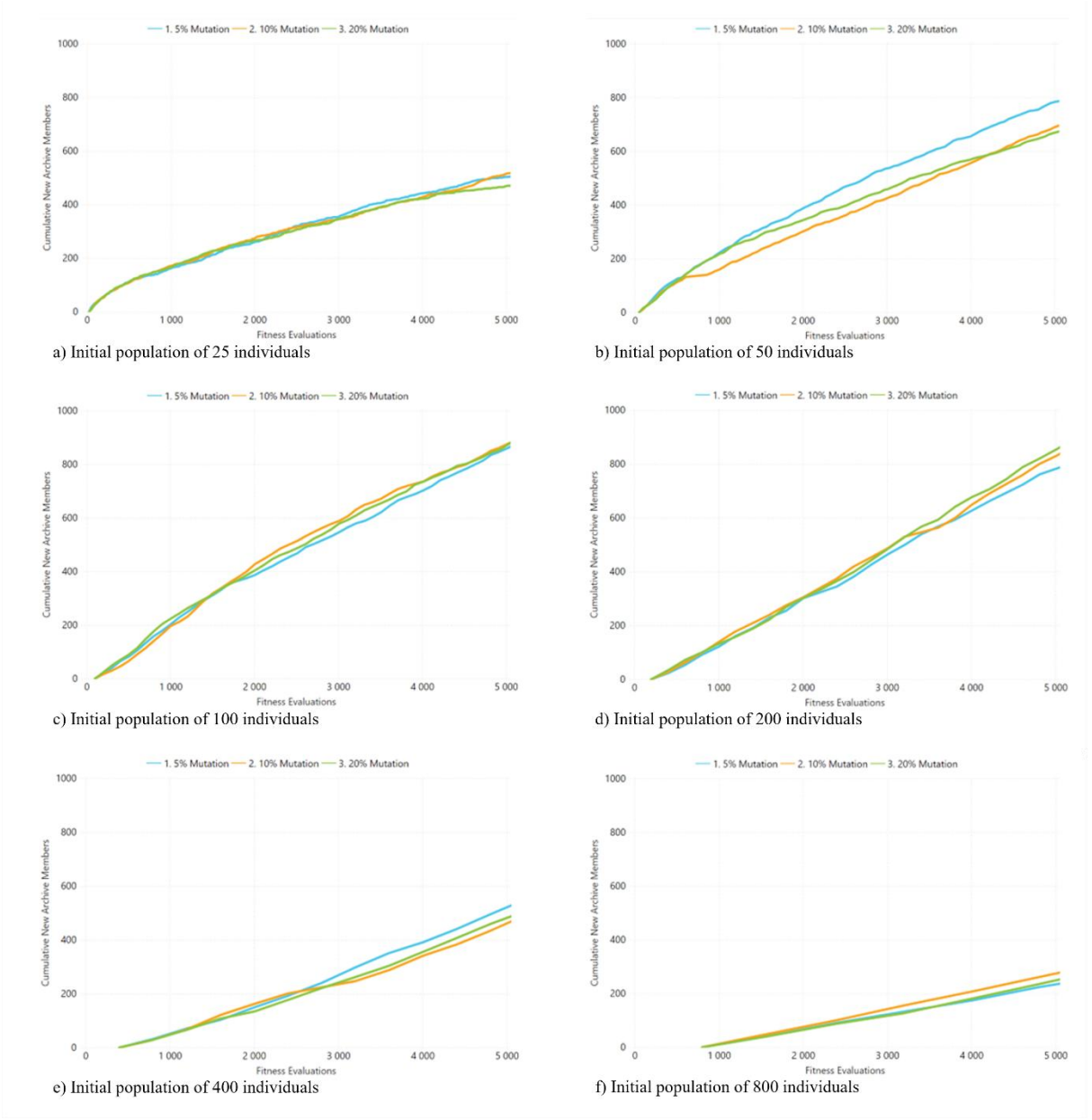


Figure 121: Line graphs showing the solutions added to the archive as a function of number of fitness evaluations per mutation rate used with NSGA-II for the shipper C problem instance.

Considering that mutation, as designed for the evolutionary approach to the LECP, is a computationally costly operator, the apparent indifference of the performance of the algorithm to this parameter favours the implementation of lower mutation rates in practice. The similar test performed using the aggregation selection method also supported this assertion. The original rate of 0.05 was therefore retained for the remainder of this research's exploratory analysis.

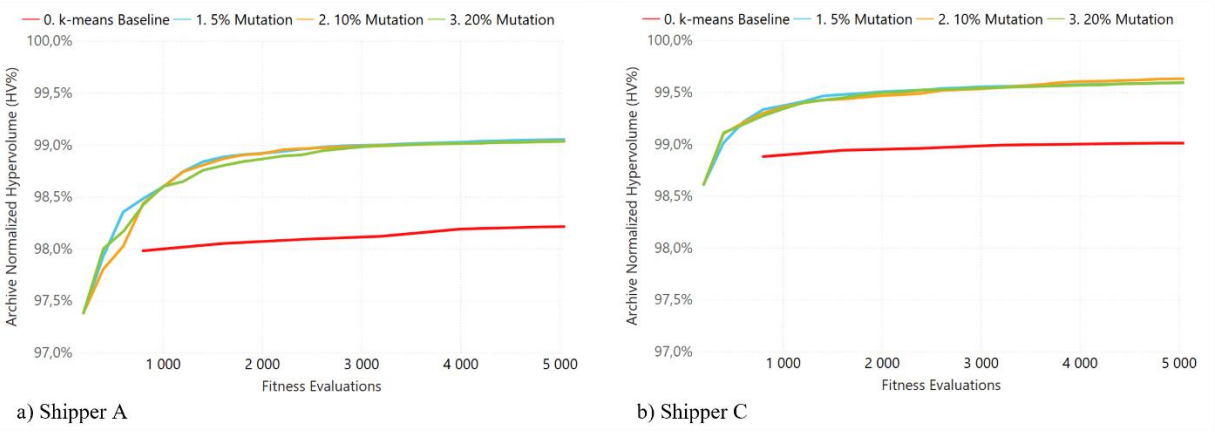


Figure 122: Line graphs showing normalized hypervolume of the archive as a function of number of fitness evaluations per adaptation run of the NSGA-II MOEA.

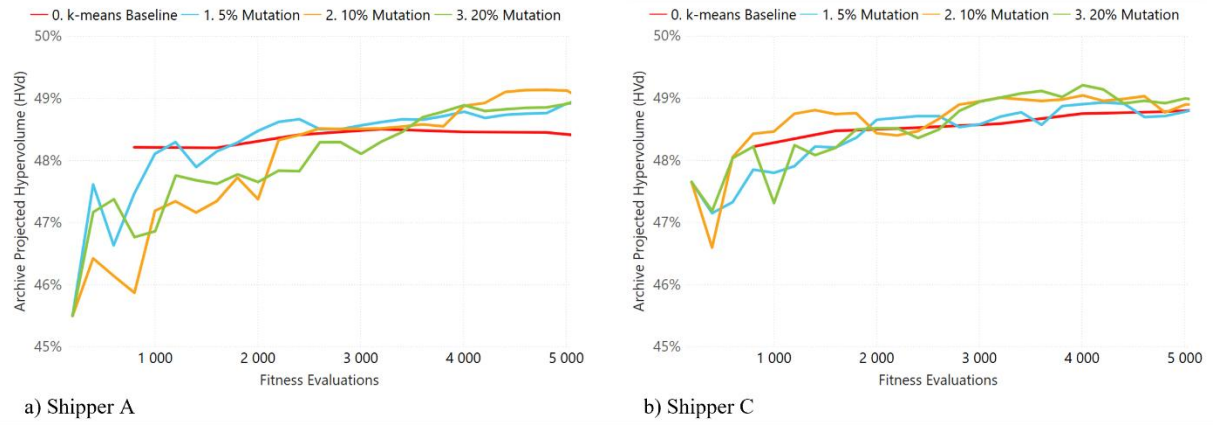


Figure 123: Line graphs showing projected hypervolume of the archive as a function of number of fitness evaluations per adaptation run of the NSGA-II MOEA.

7.5 Strength Pareto evolutionary algorithm 2 (SPEA2)

7.5.1 Procedure

The improved, second version of the strength Pareto evolutionary algorithm (SPEA), presented by Zitzler *et al.* (2001) as SPEA2, was implemented to solve the LECP. Like the application of NSGA-II to this problem, the duplicate removal strategy developed for aggregate selection was immediately incorporated to the SPEA2 algorithm without prior testing under the assumption that this MOEA would also be susceptible to premature convergence due to twins and clones.

The adapted SPEA2 algorithm thread was encoded in Python. Each run was initiated according to the aggregation selection and NSGA-II methods described in sections 7.2.1 and 7.3.1. The raw fitness for the initial population was then determined. This calculation involvds an initial count of the number of

solutions that dominated a population member[32]. This dominance-depth rank value is referred to as the *strength* of the solution in an SPEA2 context and is generally defined by Zitzler *et al.* (2001) as follows:

For every solution (($i \in Q(t)$), the strength *S(i)*, is

$$S(i) = |\{j | j \in Q(t) \wedge j \succ i\}| \quad (7.1)$$

The raw fitness was then determined. The raw fitness *R(i)* therefore defined by Zitzler *et al.* (2001) as:

$$R(i) = \sum_{j \in Q(t), i \succ j} S(j) \quad (7.2)$$

The strengths of all the solutions that dominated an individual were therefore summed, such that a population member dominated by a solution that was, in turn, dominated other solutions was given a higher fitness score. This score was subsequently converted to a fitness value by adding the calculated *density* of the solution. This value was derived from the distance, in phenotype space, of the *kth*-nearest neighbour to the solution[33].

The solution density is defined by Zitzler *et al.* (2001) as:

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (7.3)$$

, where σ_i^k represents the straight-line distance, in objective space, of the *kth*-nearest individual to solution *i*.

The initial population, with fitness scores assigned, was copied to populate the archive and became the starting active population of the MOEA. The algorithm then entered its iterative stage during which new generations of solutions were produced until the stopping criterion was met.

Each generation began with a tournament selection procedure that referred only to the individuals' assigned fitness scores. Two contestants were drawn at random from the active population, with the individual with the lower associated fitness added to the generation's specific mating pool population. Such contests were simulated until the mating pool reached the target population size *n*.

[32]For the LECP, a solution A dominates another solution B if both solution A's associated lane elimination and cost penalty fitness scores are lower to those associated with the solution B.

[33]The value of *k* is dynamically calculated as a function of the combined size of the active and archive populations. As the empty archive population has not been generated at the time of fitness values assignment to the initial population, *k* is initially determined using only the size of the initial population as an input.

The mating pool served as the source of parents for subsequent child production, as opposed to the active population, which is often the case with other MOEAs. The pool was first randomly sorted. The first pair of unmated individuals in the pool were selected as parents, who crossed over according to cluster set-level or cluster-level recombination. The type of recombination was selected using a randomly generated number and probabilities assigned to each crossover operator. Two children were created, which could subsequently mutate. The same mutation procedure used for testing the aggregation selection and NSGA-II MOEAs was used for SPEA2.

Duplicate checking and removal were performed on the two finalized children. Checking was done only against the children already created during the iteration. This strategy differed from the duplicate removal procedure employed by the other adapted MOEAs, such as NSGA-II. This was mainly attributed to SPEA2's use of a secondary mating pool as the parent source during the mating cycle. Children were not added to the parent or active population during the mating cycle, which added some computational complexity to the duplicate checking process. Furthermore, this parent selection mechanism required a greater number of children to be produced during an SPEA2 generation than the other tested MOEAs, allowing for a more robust duplicate removal process when children were only compared with other children. Figure 124 shows that the number duplicates found by SPEA2 was comparable to those found by NSGA-II with this lighter checking process. As such, the assumption was made that the procedure, as designed, was sufficiently adept at detecting duplicates using only the children of the mating cycle iteration and that SPEA2 was not unduly disadvantaged.

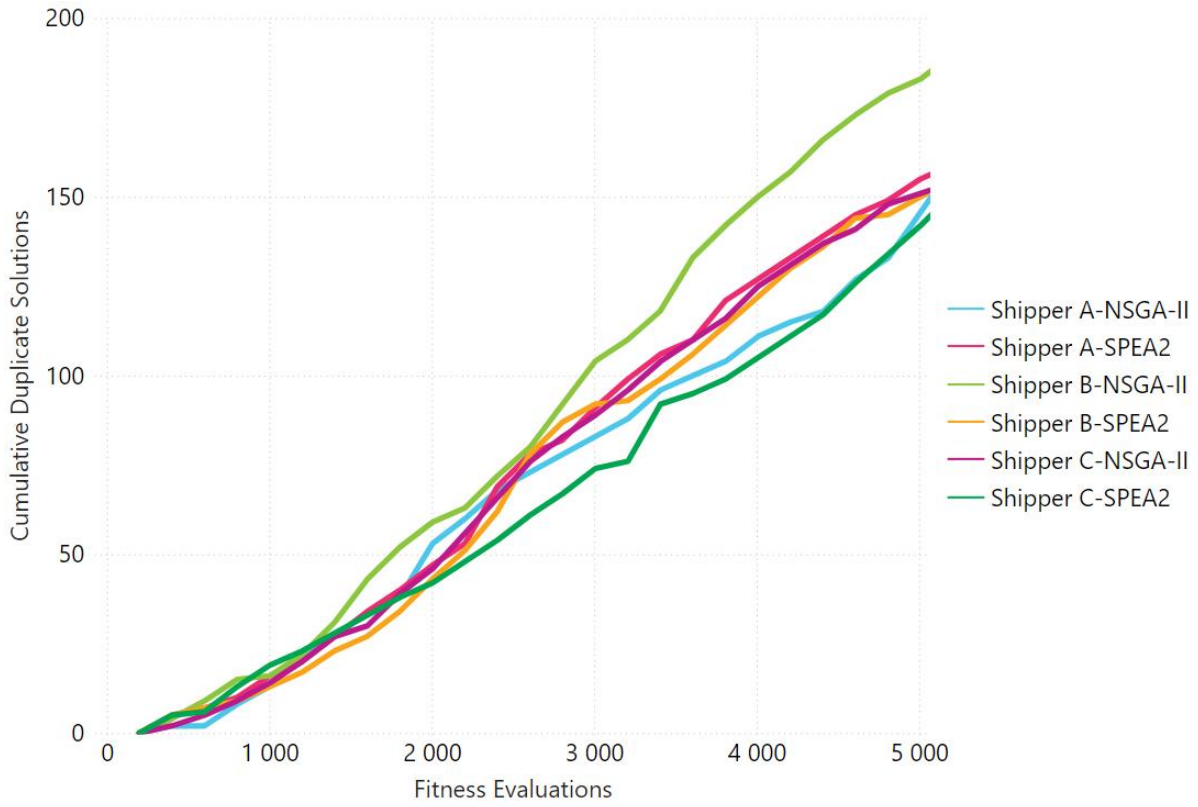


Figure 124: Line graphs showing the number of duplicate solutions eliminated from the active population as a function of fitness evaluations per problem instance for the NSGA-II and SPEA2.

Parents mated until the number of children reached n. The elimination of duplicate children necessitated a restart mechanism to draw further parents from the pool at a new position once all members of the pool have mated. The reset method used for the previously tested MOEAs was used for SPEA2.

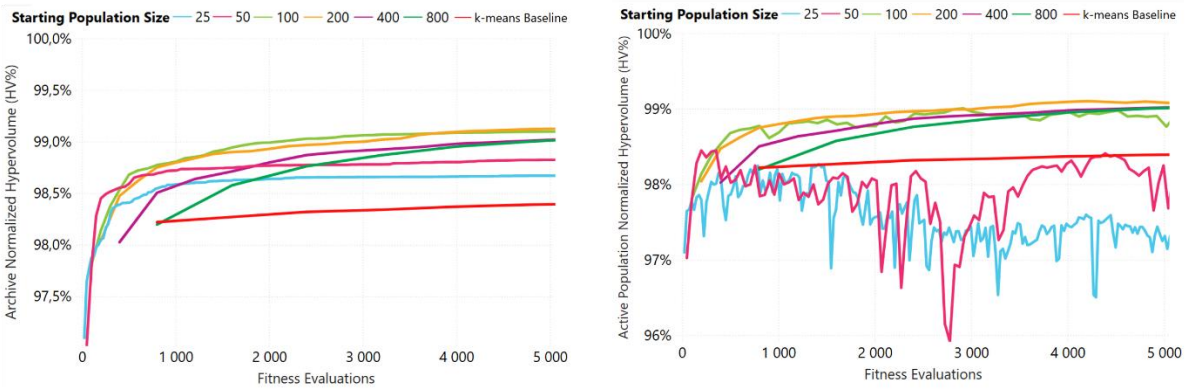
The children were added to the active population. This new active population, comprising $2n$ individuals, underwent raw fitness and fitness assignment. A temporary active population, consisting of only nondominated individuals[34] was created. The size of this temporary population was compared to n. The outcome of the comparison determined whether the temporary population was copied, filled or truncated to provide the active population for the next generation. These three procedures were executed implemented as described by Zitzler *et al.* (2001).

7.5.2 Data analysis

The adapted SPEA2 was tested using shipper B's rate card. This algorithm demonstrated good convergence to the true Pareto front PF_{true} over 5 000 fitness evaluations. The absence of explicit elitism did, however, result in a loss in active population quality, especially when smaller initial populations were used. This prevented significant improvement in the archive's hypervolume value after

[34] These are identified using existing raw fitness scores. A nondominated solution has an associated raw fitness value of zero.

the first 1 000 fitness evaluations were completed, shown by Figure 125. This undesirable effect was also seen in the reducing rate of new individual introduction to the archive as the algorithm progressed, as shown by Figure 126.



a) Archive population b) Active population

Figure 125: Line graphs showing normalized hypervolume as a function of number of fitness evaluations per initial population size when SPEA2 was tested using the shipper B problem instance.

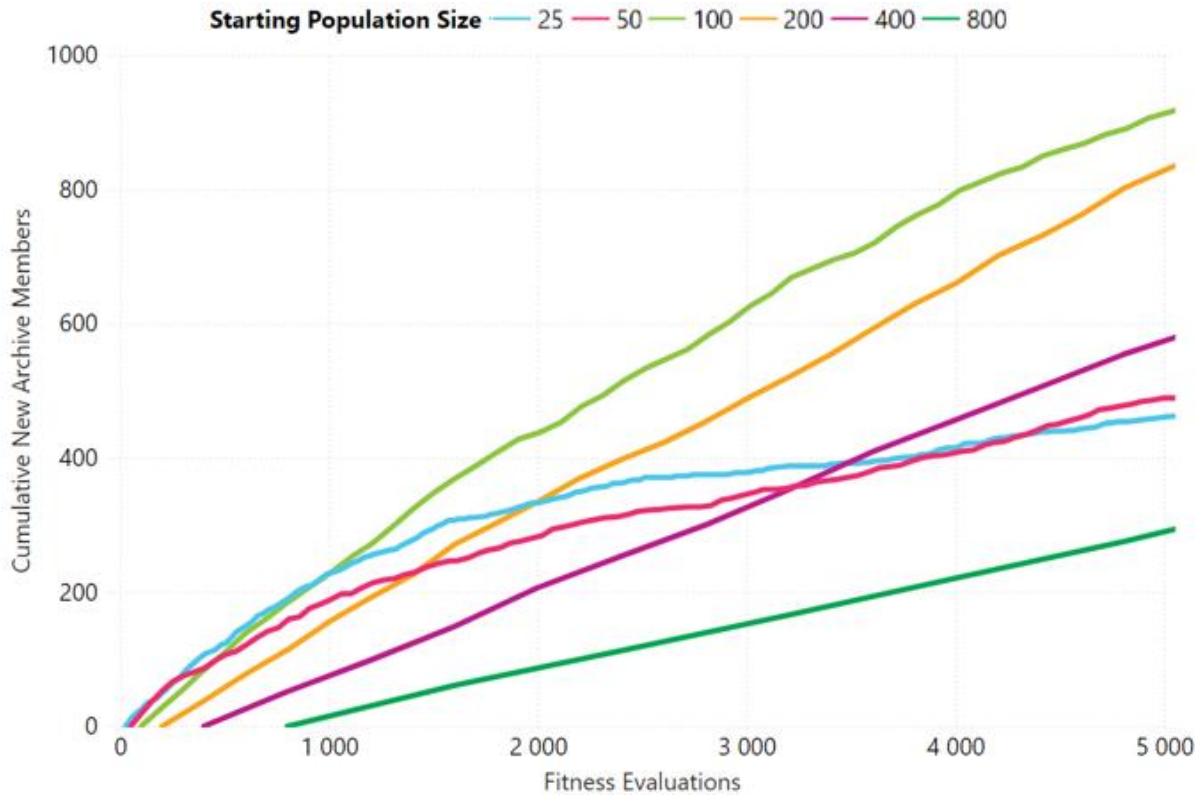


Figure 126: Line graphs showing the cumulative solutions introduced to the archive population as a function of number of fitness evaluations per initial population size when SPEA2 was tested.

The *kth*-nearest neighbour component of the fitness calculation appeared to have a significantly positive effect on the diversity of the archives found by SPEA2. The projected HV values of the various runs of this MOEA, shown by Figure 127, were all comparable to that of the k-means baseline results. Every

HV_d value was within 1% of the baseline value after 5 000 fitness evaluations. The good diversity achieved when initial sets of 25 and 50 individuals were used suggest the SPEA2 is particularly adept at finding good spreads of solutions to the LECP in phenotype space, even when handling small populations.

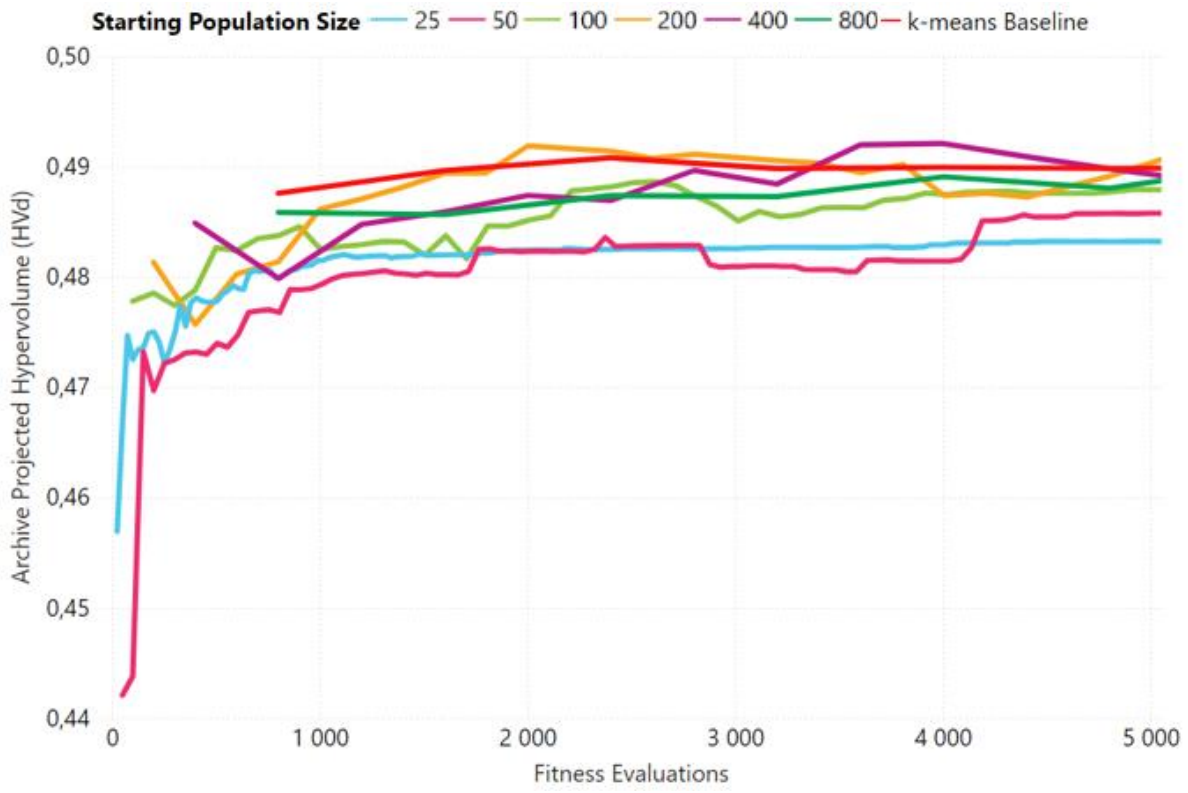


Figure 127: Line graph showing projected hypervolume of the archive as a function of number of fitness evaluations per initial population size when SPEA2 was tested using the shipper B problem instance.

Previously completed testing indicated that the additional genetic material introduced by a higher mutation rate is likely ineffective in maintaining the improvement stage of the MOEA runs for the LECP. The mutation rate was therefore kept at 5% for the SPEA2 tests.

For SPEA2, however, an alternative source of genetic diversity was explored. The original SPEA2 runs were executed using a cluster-blend recombination probability of 0.1. By increasing the proportion of children generated through cluster-level recombination, more offspring were created by merging the cluster sets of parents, as opposed to swapping cluster sets. This increased the rate of new building block creation. As such, this parameter was adjusted to investigate the effect of this type of crossover on the convergence to PF_{true} achieved by SPEA2.

The MOEA was executed on shipper B's rate card using the same populations, but using cluster-blend probability values of 0.05 and 0.2. Figure 128 shows that there was no discernible, consistent

improvement in performance across all initial populations. However, the increased amount of cluster-level recombination did appear to improve convergence to PF_{true} when smaller populations were used.

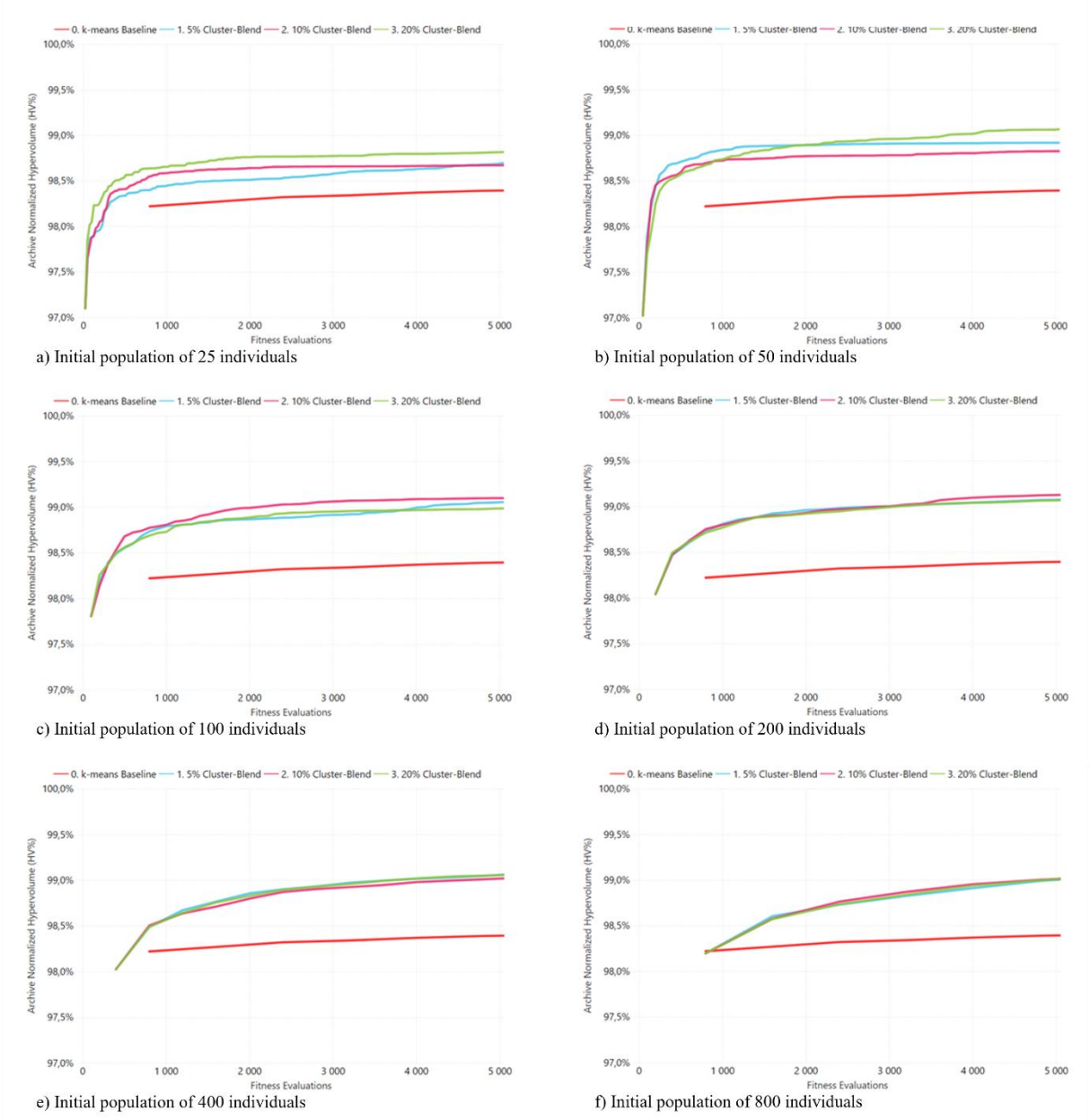


Figure 128: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per cluster-level recombination rate used with SPEA2.

Figure 129 provides some insight into this behaviour. There was a significant reduction in duplicate solutions generated when cluster blending was increased. The duplicate management strategy, as designed for aggregate selection and reused for subsequently tested MOEAs, forced cluster blending for the following mating procedure. As such, the duplicate removal technique effectively dynamically tuned the cluster-blend probability, without modifying the variable itself. The higher duplication rates associated with lower cluster-level recombination rates therefore had an equalizing effect on the number

of children produced by this type of crossover, negating the benefit of higher probabilities of cluster blending.

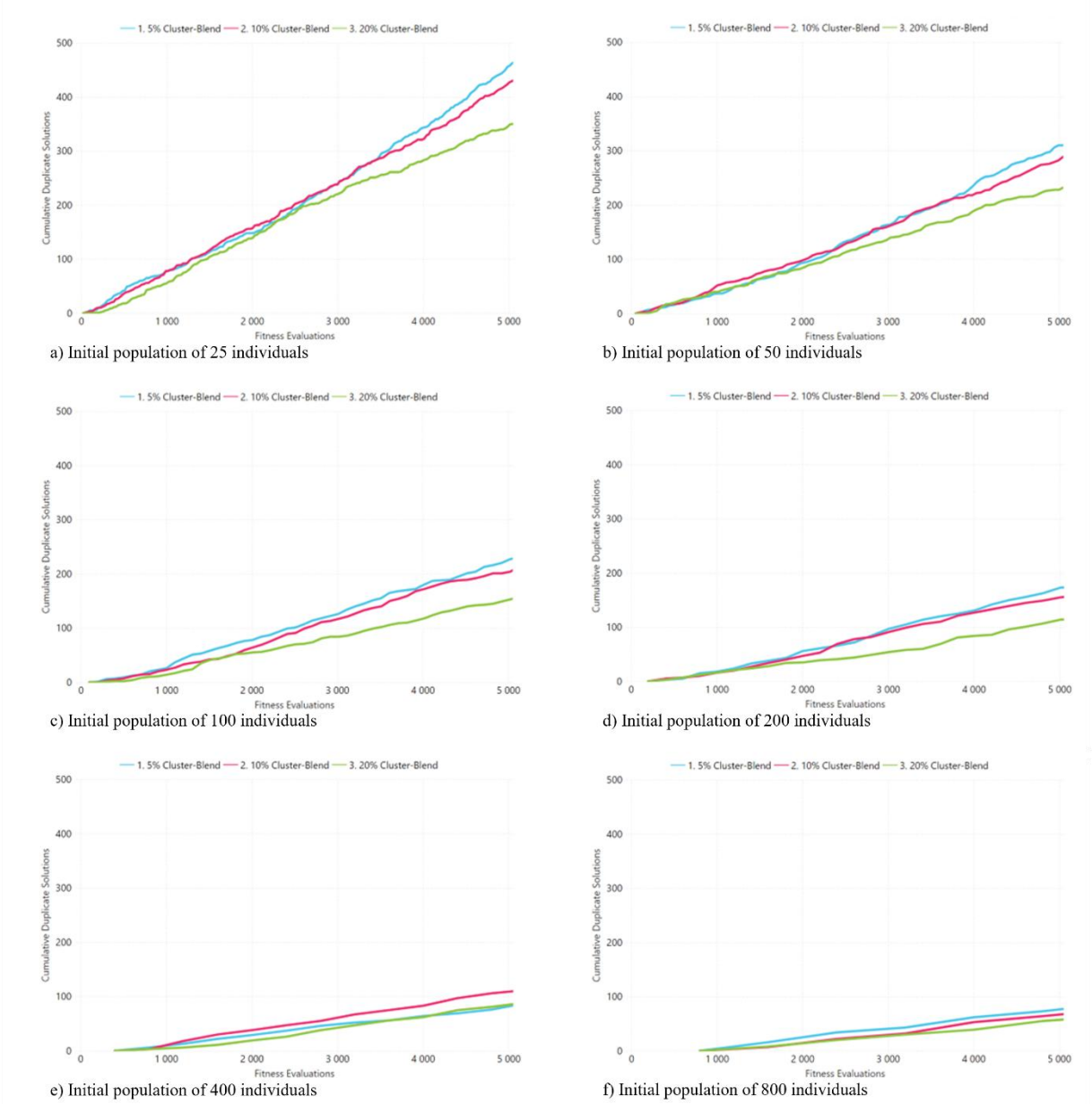
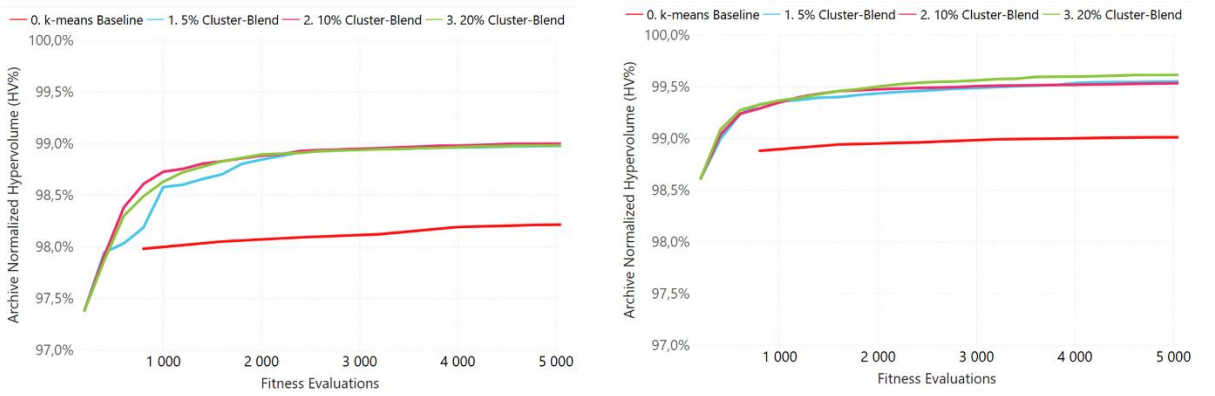


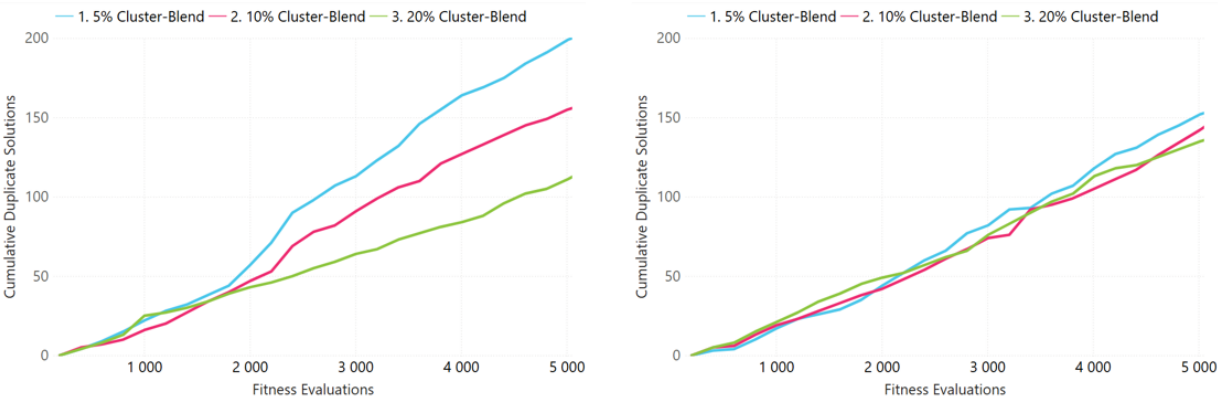
Figure 129: Line graphs showing the duplicate solutions eliminated from the active population as a function of number of fitness evaluations per cluster-level recombination rate for SPEA2.

This relationship was also observed when SPEA2 was tested using the other problem instances. No significant deviation from the previously observed behaviour was identified when this MOEA was applied to shipper A and C's rate cards with initial populations of 200 individuals. This is shown by Figure 130 and Figure 131.



a) Shipper A b) Shipper C

Figure 130: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per run of the SPEA2 MOEA.



a) Shipper A b) Shipper C

Figure 131: Line graphs showing the duplicate solutions eliminated from the active population as a function of number of fitness evaluations per adaptation run of the SPEA2 MOEA.

Although the dynamic self-tuning of the MOEA resulted in algorithms with different values for the cluster-blend parameter performing similarly, there are practical implications for consideration when this parameter is set. Cluster-level recombination is significantly more time-consuming than cluster set-level crossover to perform. Duplication is, however, undesirable, as it represents an expenditure of computational effort to produce a child that is immediately discarded. It is not possible to determine the optimal parameter value without extensive CPU time experimentation, which is beyond the scope of this research. This aspect of the algorithm setup was therefore also explored with subsequently tested MOEAs to investigate if a preferable setting became more apparent with another solution approach.

7.6 Pareto envelope-based selection algorithm (PESA)

7.6.1 Procedure

The Pareto envelope-based selection algorithm (PESA) developed by Corne *et al.* (2000) was the fourth solution approach to be applied to the LECP. It was adapted, prior to any instigative testing, to include the same duplicate removal strategy incorporated to the previously tested MOEAs.

PESA, with duplicate removal, was encoded in Python to solve the LECP according to the evolutionary approach developed for this research. Like the threads developed for testing the preceding solution approaches, a run of this MOEA began with reading and evaluating the fitness of the initial population of the same size as the specified target population size n. This population was used as the active population, or as Corne *et al.* (2000) described it, the *internal population*, for the first iteration of the evolutionary cycle without any further processing.

An archive dataframe was also created, which, in the context of PESA, is referred to as the *external population*. Corne *et al.* (2000) recommended that the external population is used as the representation of the Pareto front approximation found by the MOEA. This set of solutions, however, was subject to truncation with each generation, which may have eliminated desirable solutions from the approximation. The external population therefore did not truly represent PF_{known} and, as such, a second archive was initiated as a repository for all found nondominated solutions to the LECP.

Each generation began with dominance rank ranking of the internal population, with the nondominated solutions appended to the external and archive populations. The archive population was subsequently updated to removed newly dominated solutions and evaluated in terms of convergence to the true Pareto front PF_{true} and diversity.

The size of the external population was compared to n. If it exceeded n, the external population was truncated using an adaptive grid crowding measure called the *squeeze factor* by Corne *et al.* (2000). This value represented the number of solutions in the external population that occupied the same hyperbox in phenotype space as the individual in question. The hyperboxes were defined by dividing both the lane elimination and cost penalty fitness dimensions of the objective space into equally sized partitions starting from the origin to extreme solution on that particular axis. The number of partitions was dynamically calculated as one quarter of the prevailing size of the external population. The truncation operator removed solutions with the highest squeeze factors until the external population size equalled n.

The squeeze factor calculation is demonstrated in Figure 132. An example external archive of 31 solutions is shown in phenotype space. The grid was generated to consist of eight-by-eight equally-sized cells, as one quarter of 31 equals 7.75, which was then rounded to eight. The cell dimensions were determined using the distances between extreme points A and B such that each axis was partitioned into eight sections between these points, making this an adaptive grid. The squeeze factors for each individual can be determined by counting the number of phenotypes located in the cell in which it is located. For example, individuals with phenotypes that lie in cell *i* would be assigned squeeze factors of two, while those located in cell *ii* would receive factors of six.

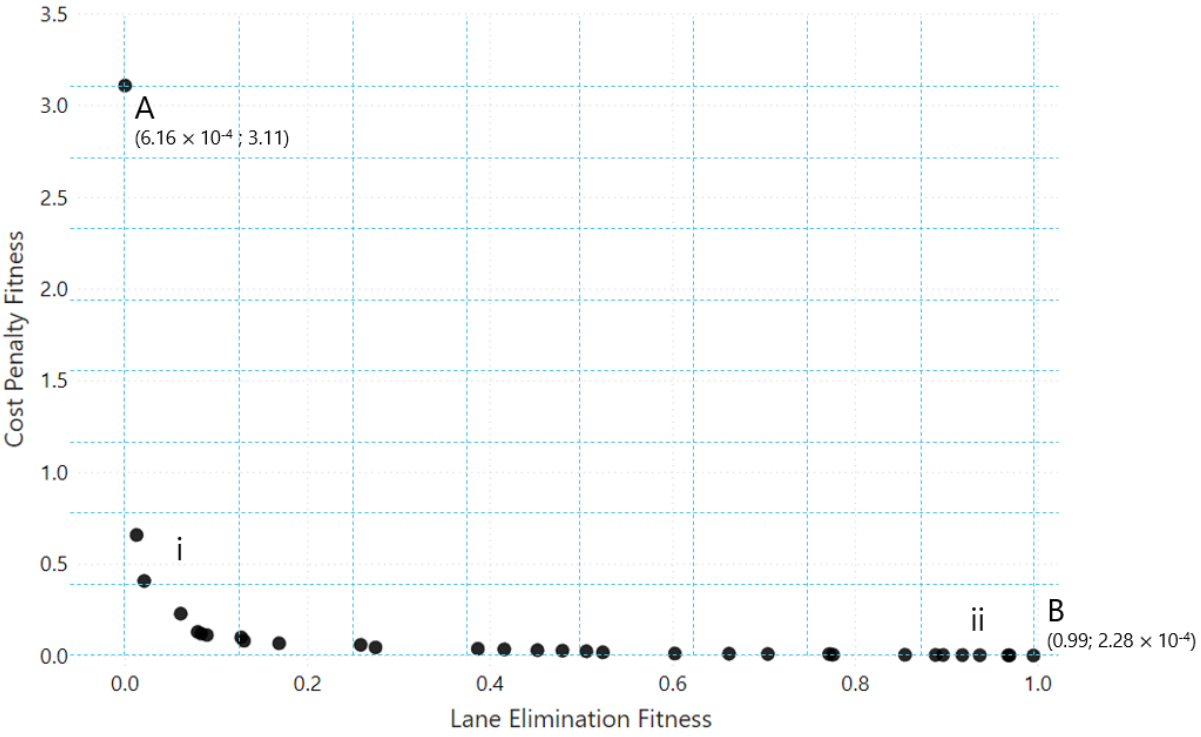


Figure 132: Scatter plot showing lane elimination and cost penalty fitness values of an example PESA external population for the shipper A problem instance with the adaptive grid shown.

Similar to SPEA2, the parents were drawn from a specially formed mating pool, not the active population itself. For PESA, however, this pool was selected by a tournament selection process that considers only the squeeze factor of the contestants. The squeeze factors of the external population members were recalculated and then underwent this tournament process to give rise to a mating pool half the size of the external population.

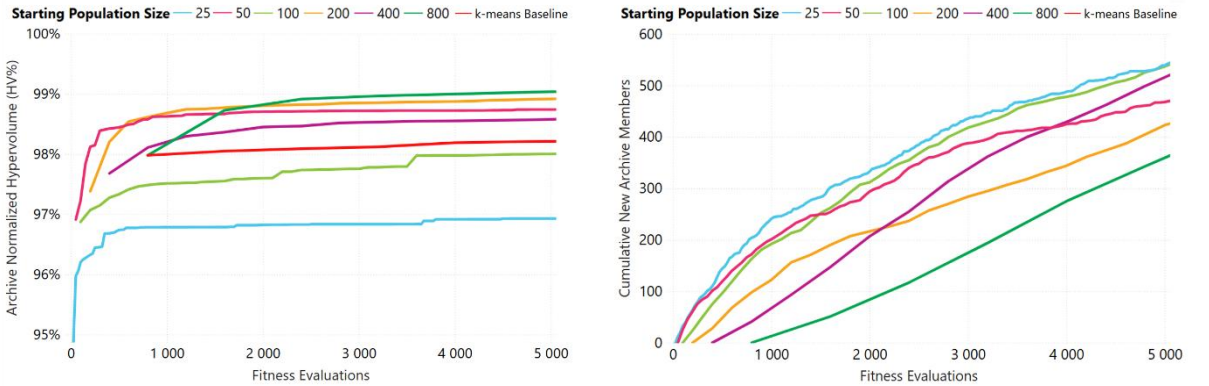
Parents mated in a similar fashion to the that simulated for testing SPEA2. The possibility of the size of the external population assuming any value from one to *n* meant that there was no definite relationship between the mating pool size and *n*. The mating process therefore had to cycle through the pool until the *n* children were generated. Duplicate management was also performed with each pair of children produced, which further increased the likelihood of a member of the mating pool parenting multiple sets

of children. The n children created with each generation entirely replaced the internal population for the next iteration.

7.6.2 Data analysis

The PESA procedure encoded in Python was tested using the shipper A problem instance over 5 000 fitness evaluations. This MOEA was, for two of the six initial trials, not able to outperform the k-means baseline method in terms of convergence to the true Pareto front PF_{true}. This is shown by Figure 133(a).

However, as shown by Figure 133(b), even in cases when PESA failed to reach the baseline hypervolume value, new individuals continued to be added to the archive at a significant rate for the full duration of the runs. Considering the run that used an initial population size of 100 was within 0.25% of the k-means hypervolume achieved at 5 000 fitness evaluations, it is reasonable to suppose that PESA might have surpassed the baseline in most trials if provided a sufficient duration to develop the external population.



a) Normalized hypervolume of archive population

b) Cumulative new members added to the archive

Figure 133: Line graphs showing a) normalized hypervolume of the archive and b) cumulative number of new solutions introduced to the archive as a function of number of fitness evaluations for PESA.

Despite the strong emphasis placed on crowding in its selection mechanism, PESA did not appear to achieve the same continuous improvement during the runs in terms of population diversity as other tested MOEAs. Upon reaching 5 000 fitness evaluation, the HV_d values were, in most cases, in decline. The trials using initial populations of 25 and 100 members performed particularly poorly, as shown by Figure 134.

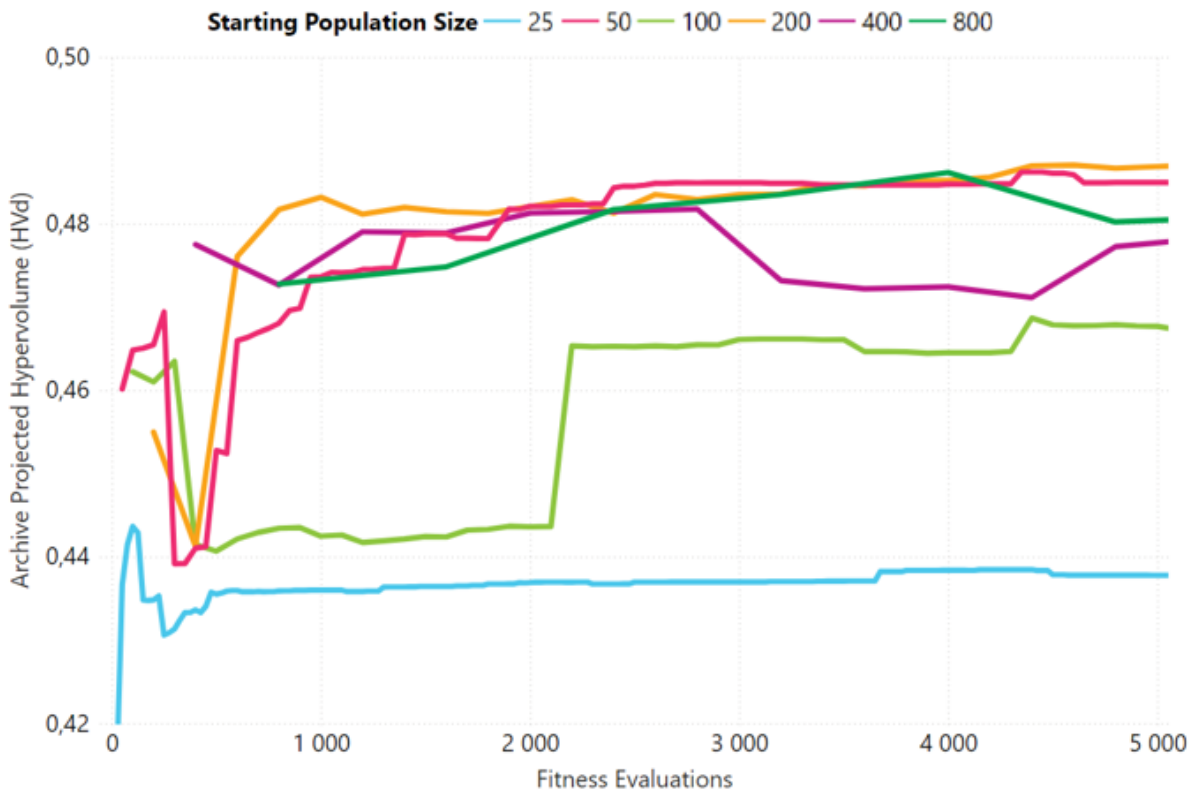


Figure 134: Line graph showing projected hypervolume of the archive population as a function of number of fitness evaluations per initial population size when PESA.

Figure 133(a) and Figure 134 show that PESA has a propensity to experience sudden significant improvements in terms of convergence and diversity. The performance of this MOEA may therefore be sensitive to the arbitrary stopping condition.

Testing the effect of the cluster-level recombination on the performance of the SPEA2 MOEA was somewhat inconclusive. PESA was used to determine if there is value in a practitioner tuning this parameter and whether the previously tested solution approaches might be well-served with further experimentation regarding recombination types.

The probability assigned to cluster blending as a recombination method was adjusted like the approach used to investigate the sensitivity of SPEA2's performance. The results, shown by Figure 135, indicate that adjustment of this parameter does not significantly affect convergence of the Pareto front approximation to PF_{true}. The cluster-blend probability was therefore kept at 0.1 for all MOEAs for the remainder of the exploratory and comparative testing.

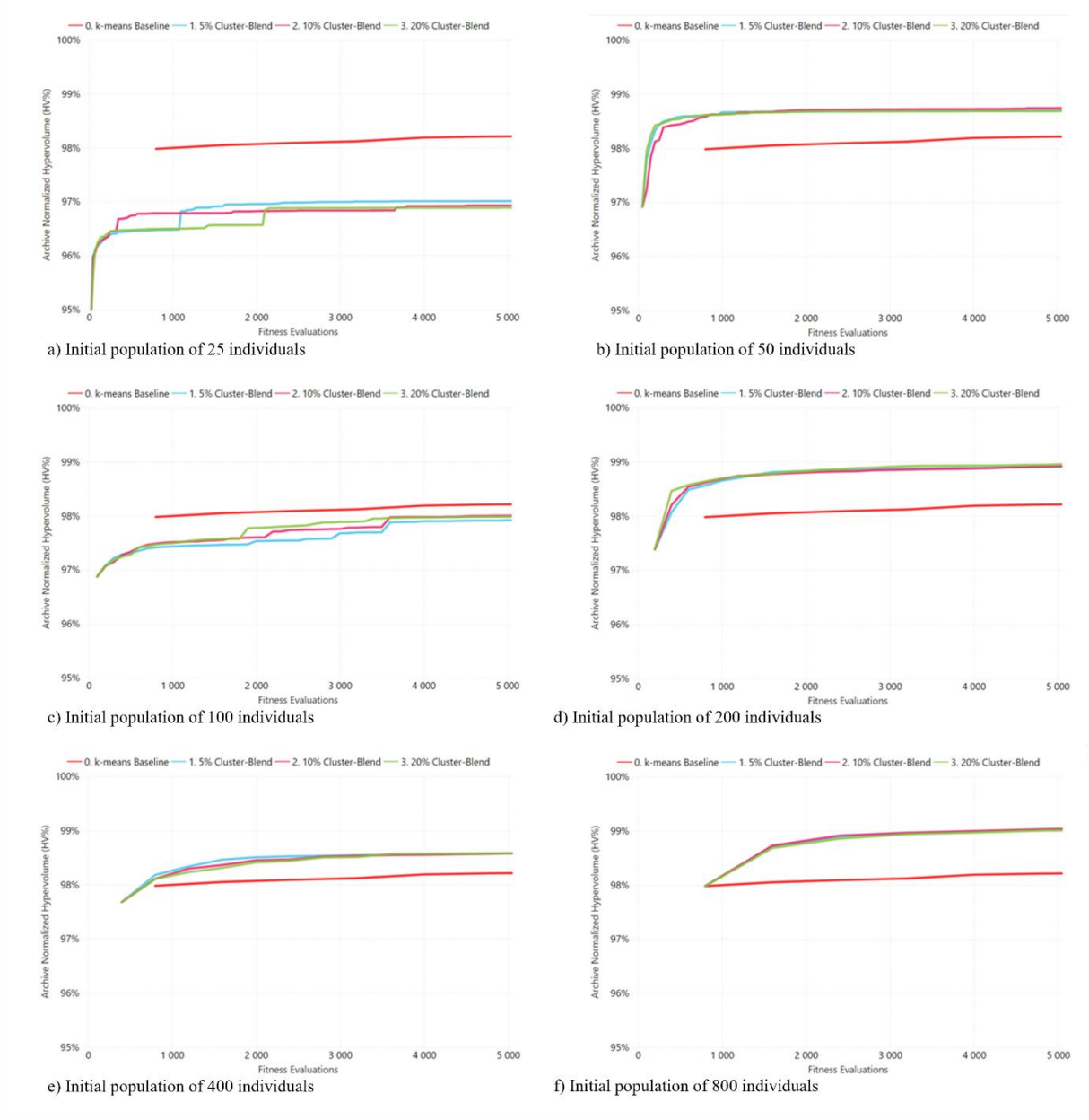


Figure 135: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per cluster-level recombination rate used with PESA.

7.7 Pareto archived evolutionary strategy (PAES)

7.7.1 Procedure

Knowles and Corne's (2000) Pareto archived evolutionary strategy was the final solution approach to be tested on the LECP. It was the only purely hill-climbing MOEA to be implemented for this research. The strategy therefore included no crossover of parent solutions and carried low risk of duplicate solution production. As such, no duplicate management was implemented. The PAES thread was therefore encoded in Python without inheriting any improvements implemented during the previously concluded exploratory testing.

The initiation of each run of the algorithm followed the same process used for the previously tested MOEAs. However, as PAES does not use the concept of an active population, the initial population was saved as the archive. A single solution was then selected from this population as the starting parent, or *current* solution, of the evolutionary cycle.

The (1 + 1) evolution strategy of PAES resulted in a simple evolutionary and mating cycle that is best described as a single-layered iterative process. Each iteration began with a Pareto rank ranking of the archive and purging of dominated solutions from this population. The current solution was then mutated to produce a single mutant, or *candidate*, using either cluster-split, cluster-merge or distance bracket mutation according to equal probabilities. For cluster-split and cluster-merge mutation, a static secondary mutation parameter of 0.1 was used[35]. This determines how many clusters in each cluster set were selected for splitting or merging by these operations.

The candidate's cost penalty and lane elimination fitness values were immediately determined and compared to those of the current solution. If the current individual dominated the candidate, the latter was discarded, and the current was retained as the single solution for mutation the next iteration.

If the candidate dominated the current, both the solutions were added to the archive. Either the current or candidate solution could then be selected as the next generation's current solution. This was randomly determined each iteration, with equal probabilities assigned to each of these individuals.

If neither the current or the candidate dominated the other, the candidate underwent a similar acceptance testing process as presented by Knowles and Corne (2000), which is shown in Figure 136. Acceptance, in this context, meant the candidate could be selected as the new current solution for the next generation with a probability of 0.5.

A simplification, however, was applied to the Knowles and Corne (2000) process for the LECP. The steps in which the archive was checked for completeness and diversity improvement were bypassed. Instead, if the candidate was found to not dominate any of the archive individuals, the crowding density check was immediately performed. Both solutions were added to the archive, giving rise to a temporary population that underwent a cell-based density procedure identical to the squeeze factor calculation used for PESA. If the candidate was found to reside in a less crowded region of phenotype space than the

[35] This parameter value was used for the four other MOEA procedures. It is included only for discussion in section 7.7.1 due to the PAES' significantly greater reliance on mutation to explore the solution space, which resulted in a higher probability of the secondary mutation rate impacting the behaviour of this algorithm than the others.

current solution, it was accepted for possible selection as the next current solution and both the candidate and current solutions were added to the archive.

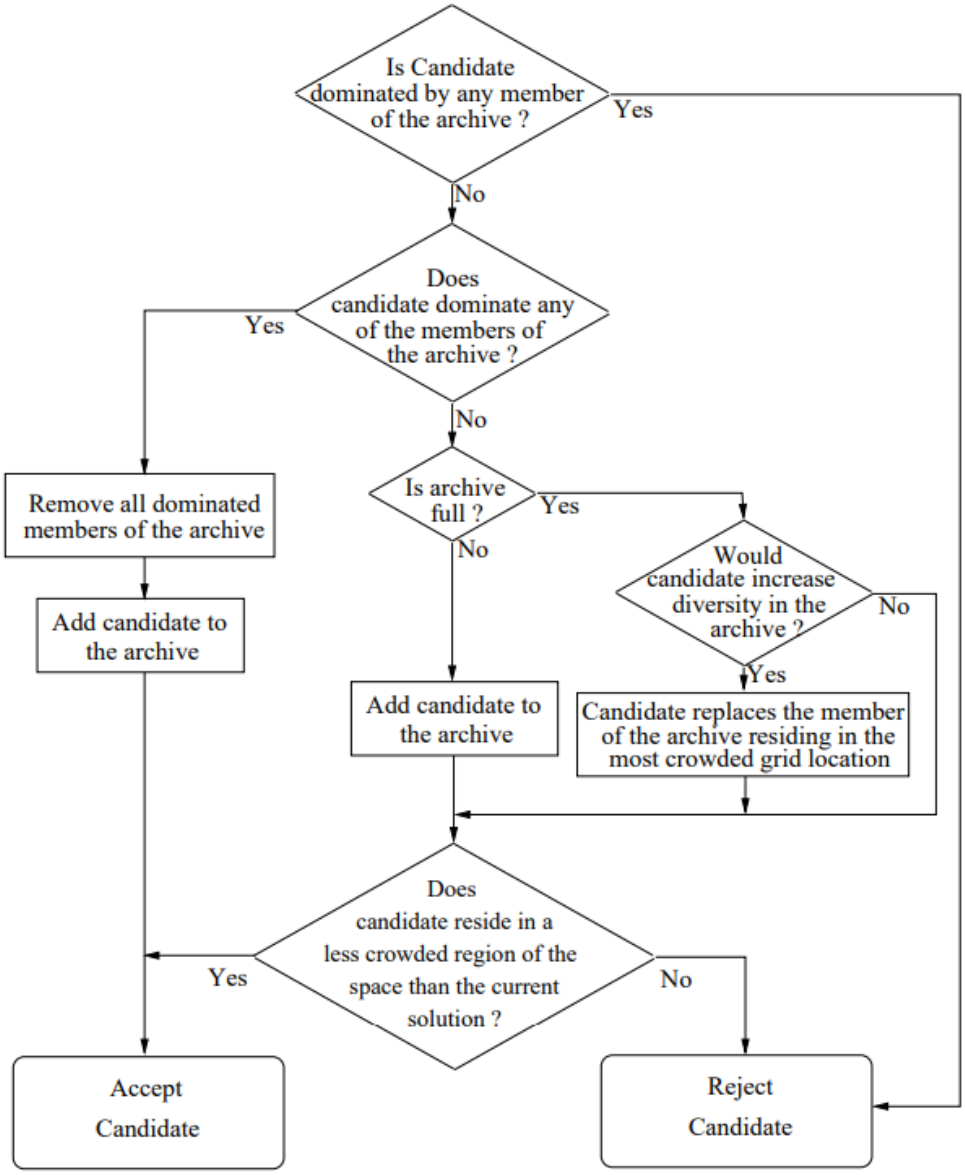


Figure 136: Candidate acceptance logic of PAES (Knowles and Corne, 2000).

In the absence of any crossover operators, PAES did not require any solution remediation step. While the cluster-split and distance bracket modification operators did not introduce any risk of producing overlapping clusters[36], the progressive merging of adjacent clusters during a PAES run could result in cluster-sets that violate the overlapping constraint.

[36] This depends on the degree on concavity used when determining overlapping. See section 6.5.2 for the discussion regarding this relationship.

The computational cost of remediation was considered when determining whether to introduce the remedy function to the PAES thread. Table 13 shows the average remediation wall clock time as measured over 1 000 executions of the operator. This metric is shown alongside the average time for the mutation functions. Assimilation of the remediation function would expectedly increase the CPU time of the cluster-merge operator approximately by between 625% and 857%. Considering the high computational cost associated with the remedy function, the subjectivity associated with the definition of overlapping, as well as the remedying effect of cluster-splitting inherent to the PAES process, the risk of overlapping was accepted for the exploratory analysis exercise. The designed procedure therefore excluded remediation. Example solution cluster sets were checked as part of the empirical validation of this MOEA.

Table 13: Average wall clock completion times of LECP remedy and mutation operators per tested problem instance.

	CPU time (s)			
Shipper	**Remediation**	**Cluster-split mutation**	**Cluster-merge mutation**	**Distance bracket mutation**
A	16.5367	0.1146	2.4925	0.0004
B	3.5840	0.1191	0.5736	0.0004
C	7.6611	0.0100	0.8943	0.0006

7.7.2 Data analysis

The first set of results, generated using only shipper A's rate card, showed that PAES' progress towards the true Pareto Front PF_{true} was prohibitively slow. Figure 137 shows that this solution approach failed to generate Pareto front approximations that surpassed the k-means baseline method in terms of convergence to PF_{true} or diversity over 2 500 fitness evaluations. The hill-climbing strategy of PAES failed to yield any significant improvement to the initial population when larger populations were used, and rates of improvement when smaller populations were tested did not suggest that this MOEA would achieve parity with the baseline method over a longer run.

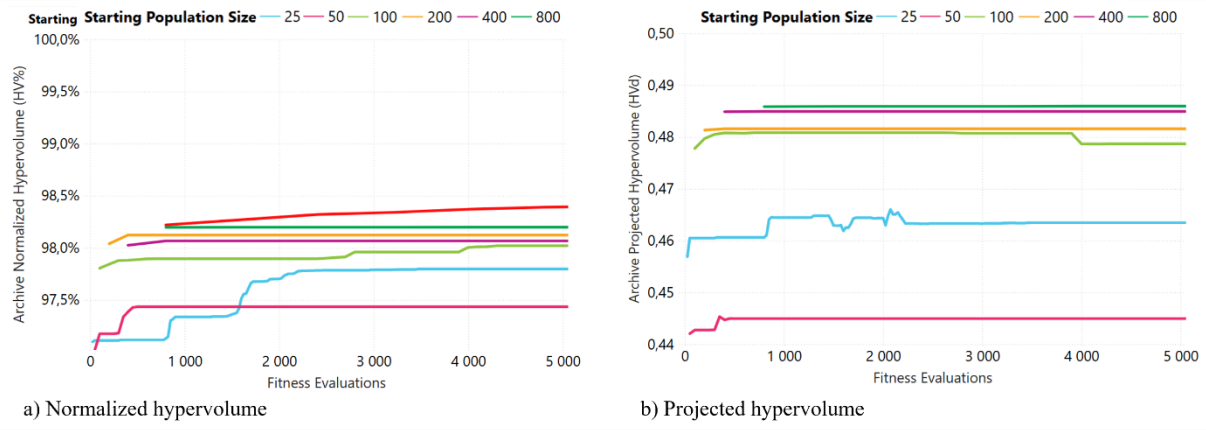


Figure 137: Line graphs showing a) normalized and b) projected hypervolume of the archive population as a function of number of fitness evaluations per initial population size for PAES.

The relatively poor performance of PAES could be attributed to the absence of an elitism strategy and insufficient exploration due to the lack of crossover recombination. Table 14 shows that a relatively large number of nondominated solutions in the final test archives originated from the initial population, indicating that the hill-climbing strategy addressed only a section of the Pareto front. This shortcoming, however, could not be repaired without fundamentally altering the characteristics of the MOEA. The absence of recombination also limited the extent of parameters that could be tuned to overcome barriers to PAES' exploration of the LECP solution space.

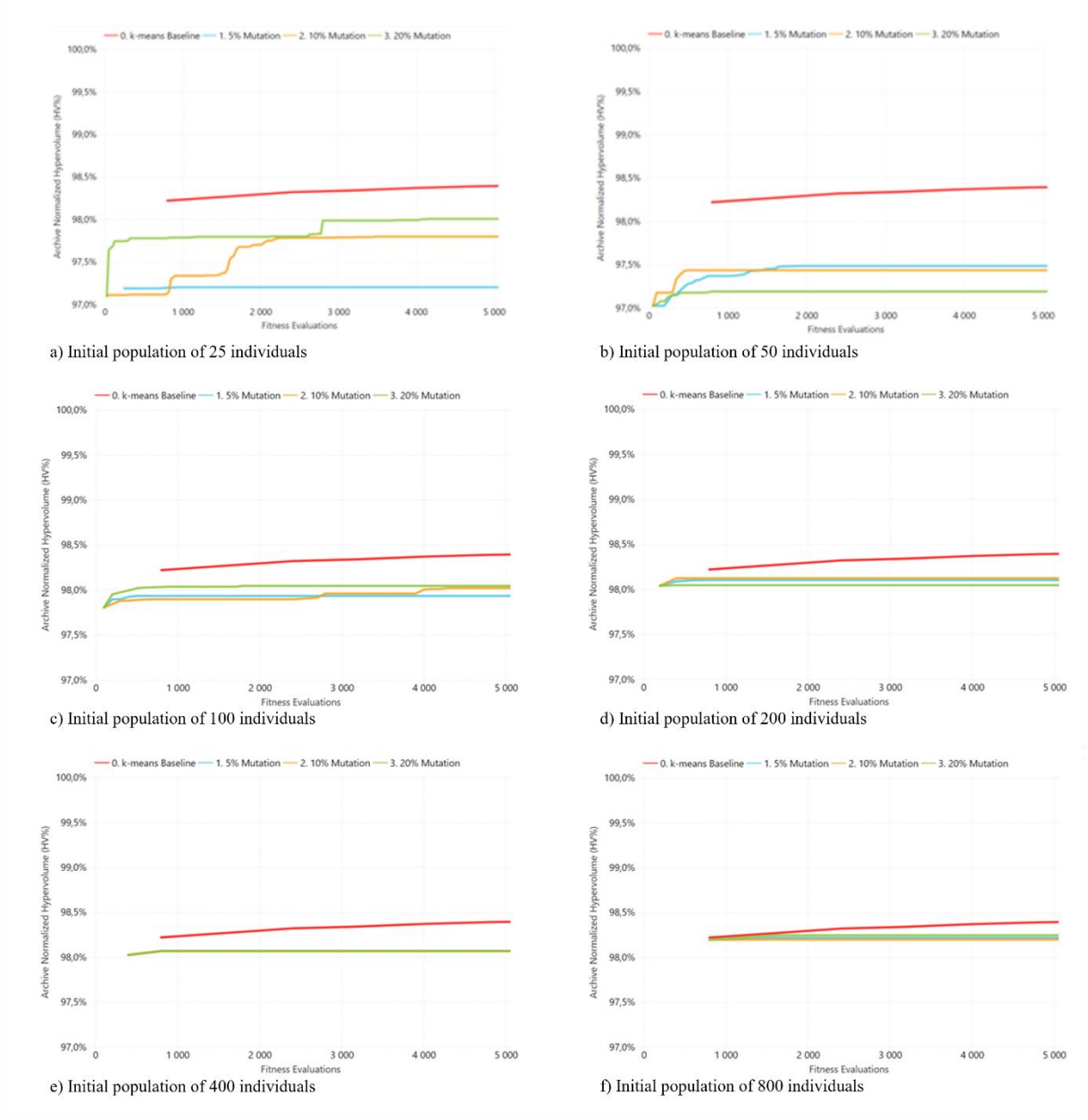
Table 14: The percentage individuals of the archive that originate from the initial population upon PAES reaching 5 000 fitness evaluations.

Initial Population Size	Percentage of Archive Retained from Initial Population
25	13.89
50	46.51
100	24.69
200	92.31
400	88.89
800	75.79

The 100% mutation rate associated with PEAS, however, brings a set of parameters, previously overlooked for course-turning, into sharper focus. Considering the 0.05 mutation probability decided for the other solution approaches, parameters used by the operators themselves are arguably twenty times more impactful on the overall behaviour of the PAES algorithm. The secondary mutation rates used by the cluster-split and cluster-merge operators are therefore more relevant when the MOEA relies solely on mutation for exploration and PAES could be sensitive to their chosen values.

PAES was rerun using the same six initial populations with both decreased and increased secondary mutation rates, specifically 0.05 and 0.2. These rates were kept equal for both the cluster splitting and merging functions to prevent the algorithm leaning toward either highly clustered or unclustered cluster sets.

Figure 138 shows that the effect of these adjustment on the algorithm's performance was varied. This highlights the sensitivity of PAES to the initial direction of the hill-climbing strategy, which was selected randomly. An onerously large number of tests would therefore be required to determine which secondary mutation rate would be most beneficial. However, the absence of a consistently prevailing rate over the six tests conducted and the failure of PAES to achieve the k-means baseline convergence to PF_{true} was deemed sufficient empirical evidence that this parameter needed no further course-tuning. Furthermore, Figure 139 shows that the performance of PAES was similarly ambivalent to this parameter when applied to the other problem instances. The secondary mutation rate of 0.1, which was utilized for all initial runs of the five tested solution approaches, was therefore retained for the comparative analysis.



a) Initial population of 25 individuals

b) Initial population of 50 individuals

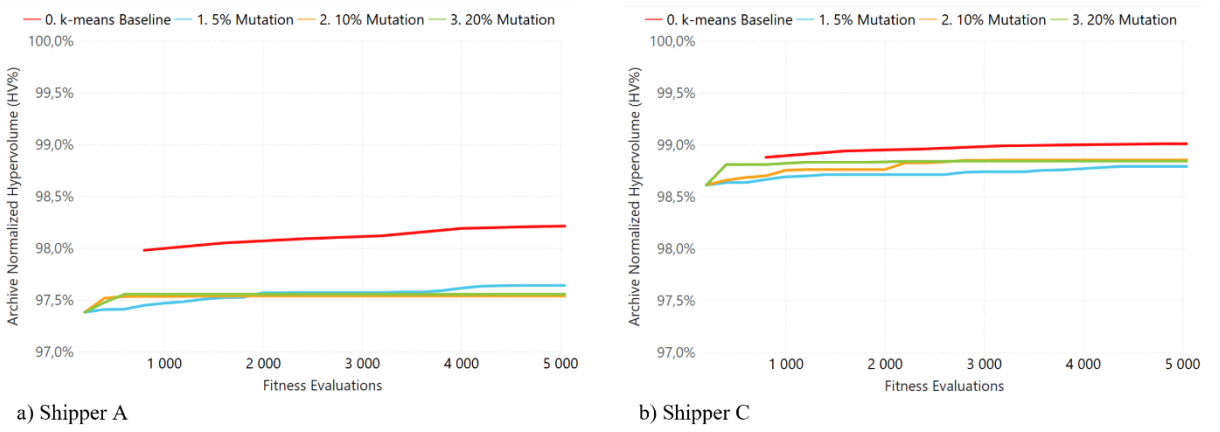
c) Initial population of 100 individuals

d) Initial population of 200 individuals

e) Initial population of 400 individuals

f) Initial population of 800 individuals

Figure 138: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per secondary mutation rate used with PAES.



a) Shipper A

b) Shipper C

Figure 139: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations per secondary mutation of the PAES MOEA.

7.8 Empirical validation of MOEAs

Although the evolutionary operators had been verified and validated prior to this research's exploratory phase of testing, a further exercise was required to validate the evolutionary approach to the LECP and each adapted MOEA as genuine solution approaches to the problem. This was performed using a single set of compromise parameter values derived from the preceding experimentation. These values, including the termination criterion parameter that controlled the length of the runs, are shown in Table 15. The parameter settings shown also form part of this research's overall recommendation for soling the LECP.

Table 15: Compromise parameter values used for the MOEA empirical validation and comparative analyses.

Parameter name	**Compromise value**
Initial population size	200
Cluster-blend crossover probability	0.1
Cluster-swap crossover probability	0.9
Primary mutation rate	0.05
Secondary mutation rate	0.1
Cluster-split mutation probability	0.33
Cluster-merge mutation probability	0.33
Distance bracket mutation probability	0.33
Maximum number of fitness evaluations (t_{max})	12 800

Three specialized tests were designed and executed to validate the solution approaches. sections 7.8.1 to 7.8.3 explain the testing procedures and provide analysis of the individual outputs. Section 7.8.4 collates the results of the validation exercises and provides an overall conclusion to the testing.

7.8.1 Extreme value test

Methodology

Shipper A was arbitrarily chosen as the problem instance for the extreme value validation exercise. Limited data points of this shipper's original rate card were modified for this test. The geographic spread of the destination nodes of shipper A's distribution network was analysed, as well as the carrier rates associated with transport to these nodes. Two nodes, namely Krugersdorp and Krugersdorp West, were identified as being in relatively close spatial proximity. They are also, according to the rate card, indistinguishable from a transport pricing perspective[37]. These nodes therefore carry a high probability of being assigned to the same destination grouping in a high-quality LECP solution cluster set.

Furthermore, these nodes, according to the rate card, are linked to the remainder of the SCN only over relatively short straight-line distances[38]. This is another desirable quality of this pair of nodes for the introduction of extreme associated rates as their assignment to clusters is, in most cases, only be relevant to the local destination cluster-set. This simplifies the analysis for this exercise.

The rates of a single randomly selected carrier associated with the Krugersdorp and Krugersdorp West destination nodes were adjusted to an artificial arbitrarily large value. This pair of nodes therefore served as the *extreme nodes* of the test. Each adapted MOEA was then run for 12 800 fitness evaluations using both the original and modified shipper A rate cards. The ability of each MOEA to successfully identify the substantial benefit of separating the Krugersdorp and Krugersdorp West nodes in the local destination cluster sets was then evaluated.

The percentage of final archive solutions in which the extreme nodes were grouped together in the local destination cluster set, when both the original and modified rate cards were used, were compared. A significant increase in the percentage of nondominated individuals in which Krugersdorp and Krugersdorp West remained ungrouped when the modified rate card is used can be interpreted as the success of the adapted LECP solution approach to find an obviously beneficial SCN partition.

A certain region of the Pareto front cannot, however, be explored without clustering the extreme nodes together[39]. This was shown by elevated cost fitness values associated with the PF_{known} of Figure 140

[37] All carrier rates associated with lanes to these nodes were equal.

[38] Krugersdorp and Krugersdorp West we, according to problem instance A, the destination nodes of rates for transport from the Isando and Roodepoort nodes. The straight-line distances of these lanes, according to the distance matrix used by the MOEAs, range from nine to 51 kilometres.

[39] Regions with low lane elimination ratios require solutions with relatively high degrees of clustering. As the exploration of each MOEA pushes further into this area of the solution space, grouping of the extreme nodes, despite the extreme associated cost penalty fitness, becomes unavoidable. The larger range of cost penalty fitness

when the lane elimination fitness approaches zero. Nonetheless, this test expected an effective MOEA to generate a Pareto front approximation that, for much of the range of the front, is still populated by solutions that assign the extreme nodes to different clusters. As such, the percentages were calculated only for a large subspace of the phenotype space for the comparison. Specifically, only archive members with a lane elimination ratio exceeding 0.005 (i.e. 99.5% of the objective space) were considered for the analysis.

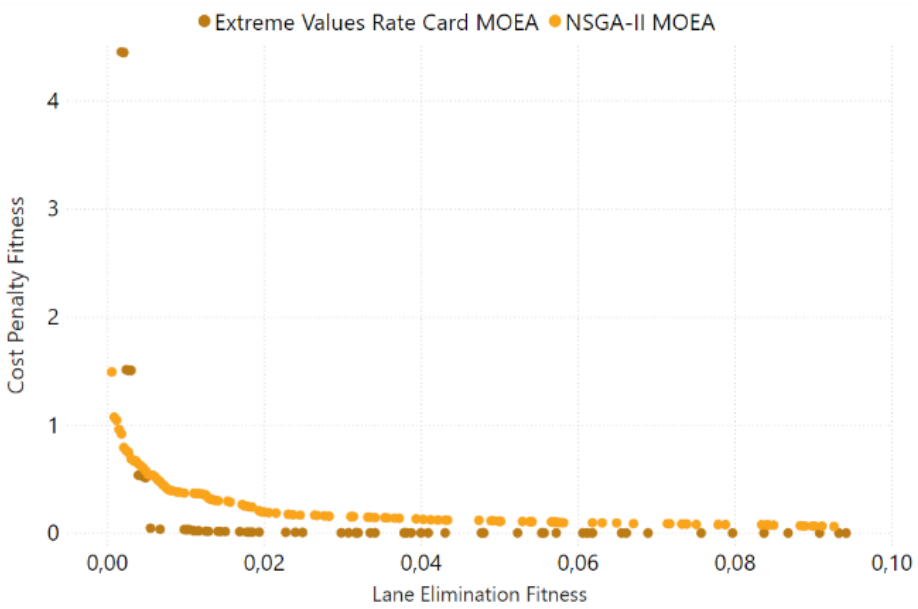


Figure 140: Scatter plot showing lane elimination and cost penalty fitness values of the archives found by the NSGA-II MOEA using the original and modified shipper A rate cards.

Results and Discussion

Figure 141 shows that, when the extreme value rate card was applied, every adapted MOEA returned a significantly higher percentage of nondominated solutions in which the extreme nodes were unclustered in the local destination cluster set. As such, all five of the developed solutions displayed the desired level of sensitivity to the input data.

The largest observed contrast is shown by the results of the NSGA-II test. Only 11.7% of the analysed cluster sets generated by this adapted solution approach grouped Krugersdorp and Krugersdorp West when original local rates were applied. This demonstrates the ability to reliably find the opportunity to eliminate the Isando and Roodepoort lane duplications with no cost impact. When the extreme data points were introduced to the rate card, however, this MOEA did not group these nodes in any of the archive solutions' local destination cluster sets. This reflected the algorithm's responsivity to significant values in the dataset.

---

values result in lower achievable cost penalty ratios. This behaviour is demonstrated by the Pareto fronts shown by Figure 140.

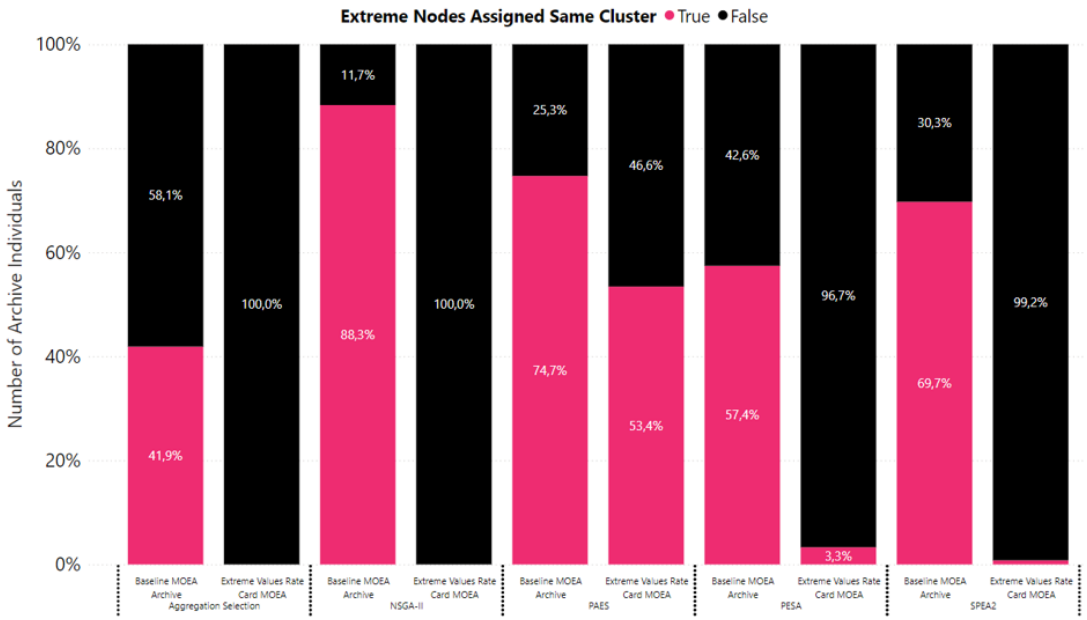


Figure 141: Stacked bar chart showing the percentage of each adapted MOEA's archive solutions in which the extreme nodes were assigned to the same cluster in the local destination cluster set.

Table 16

Table 16 provides a sharper view of the extreme value test. It can be seen that all adapted MOEAs displayed the ability to identify artificially obvious partitions. NSGA-II demonstrated the greatest sensitivity to the extreme data points, with PAES achieving the least.

Table 16: Summary of number of archive solutions[40] found by each adapted MOEA over 12 800 fitness evaluations in which the extreme nodes were assigned to the same cluster.

Adapted MOEA	**Original rate card used**		**Rate card with extreme data points used**		**Increase in proportion of archive with ungrouped extreme nodes**
	Number of solutions with extreme nodes grouped	**Number of solutions with extreme nodes ungrouped**	**Number of solutions with extreme nodes grouped**	**Number of solutions with extreme nodes ungrouped**	
Aggregation Selection	204	283	0	33	0.41
NSGA-II	**325**	**43**	**0**	**285**	**0.88**
SPEA2	205	89	2	246	0.69
PESA	147	109	63	175	0.31
PAES	62	21	39	34	0.21

[40]With a lane elimination fitness value exceeding 0.005

7.8.2 Government boundary comparison test

Methodology

LECP solutions for each problem instance were developed using standard government geographical boundaries. These were derived from the subplace spatial data layer provided by StatsSA (2011). Four levels of geographical classification, namely suburb, municipality, city and province, were used to generate four different definitions of each of the four cluster set building blocks of an LECP solution. This process was repeated for each of the three shipper rate cards. The cluster sets were then combined in all possible combinations, with a standard distance bracket value of 150 kilometres providing the final building block, to give rise to sixteen *government solutions* to each problem instance solely using pre-existing geographical constructs.

The government solutions were subsequently Pareto-ranked and the nondominated solutions analysed with reference to the Pareto front approximation PF_{known} found by each of the adapted MOEAs after 12 800 fitness evaluations. This test expects the government solutions to be dominated by at least one solution found by an effective LECP solution approach. This translates to the government individuals being visually positioned higher in phenotype space than the found archives[41].

Results and Discussion

The baseline k-means method underwent the government boundary comparison test as an initial validation of the exercise itself. Figure 142 indicates that the k-means clustering process for generating MOEA starting solutions can produce populations of 12 800 individuals that generally consist of better solution alternatives to the use of government boundaries within their respective regions of phenotype space. This somewhat discredits the notion that transport rates should be defined according to existing partitions such as municipal and provincial boundaries.

The notable exception, however, was the government boundary solution to problem instance B with a lane elimination fitness of 0.05 and cost penalty fitness of 0.13. This individual was not dominated by any solution generated by the k-means baseline method for this trial.

[41] The visuals provided in this section show only a subsection of the phenotype space to allow easier discernment of the relative qualities of the archives in the region of the front where this is most apparent. The full phenotype is shown in visuals included in Appendix A.

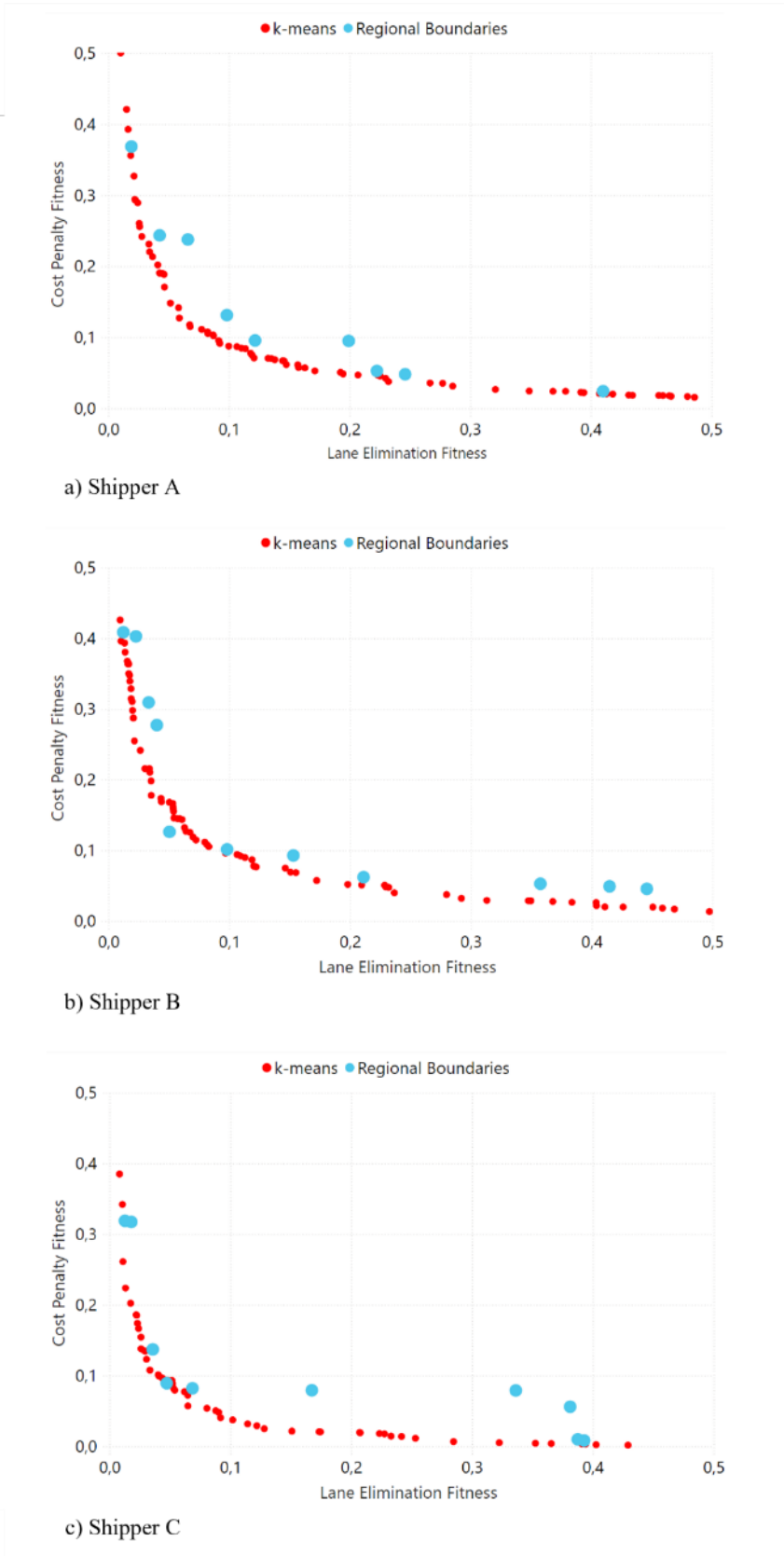
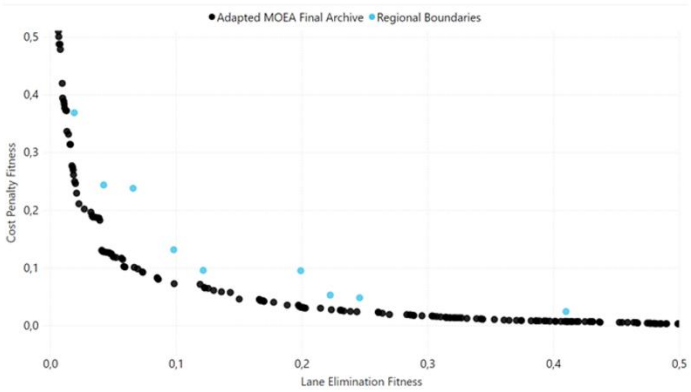
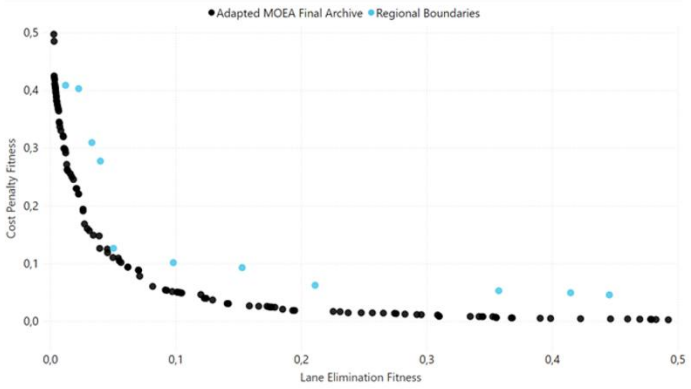


Figure 142: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the k-means baseline method.

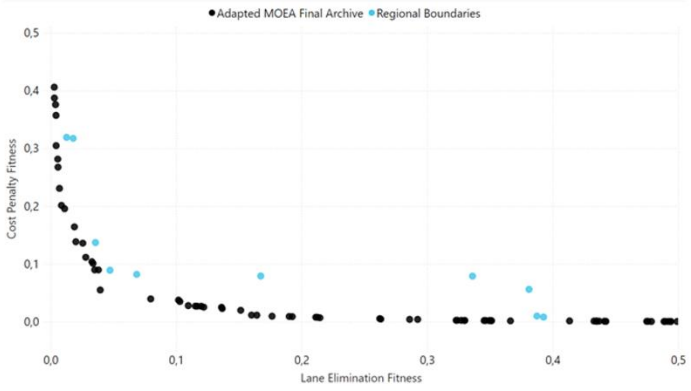
Figure 143, Figure 144, Figure 145 and Figure 146 show that aggregation selection with linear combination of weights, when used with duplicate management and elitism, as well as the adapted NSGA-II, SPEA2 and PESA MOEAs, are able to find more desirable LECP solutions for all three problem instances than the use of government partitions. The validities of these MOEAs, as effective solution approaches to the LECP, were therefore supported by this test.



a) Shipper A

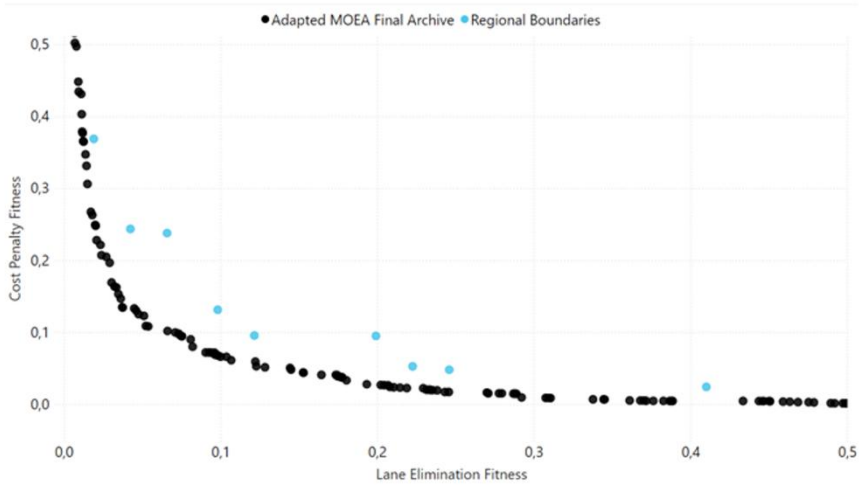


b) Shipper B

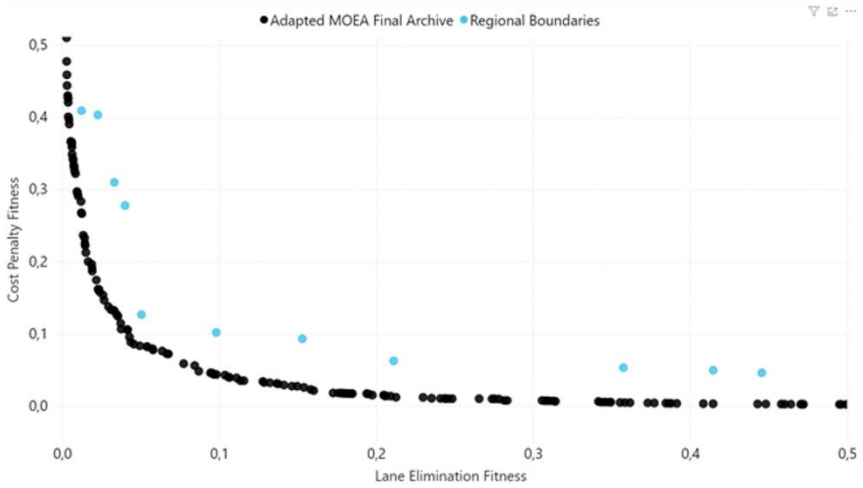


c) Shipper C

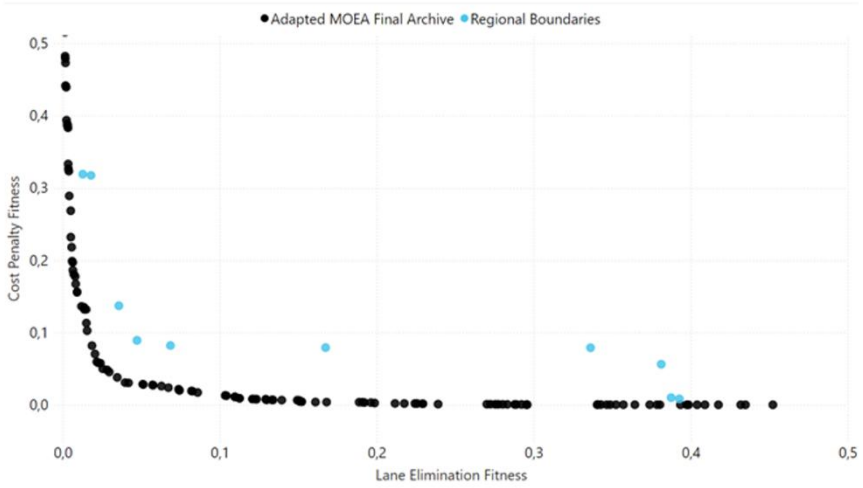
Figure 143: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the adapted aggregation selection MOEA.



a) Shipper A

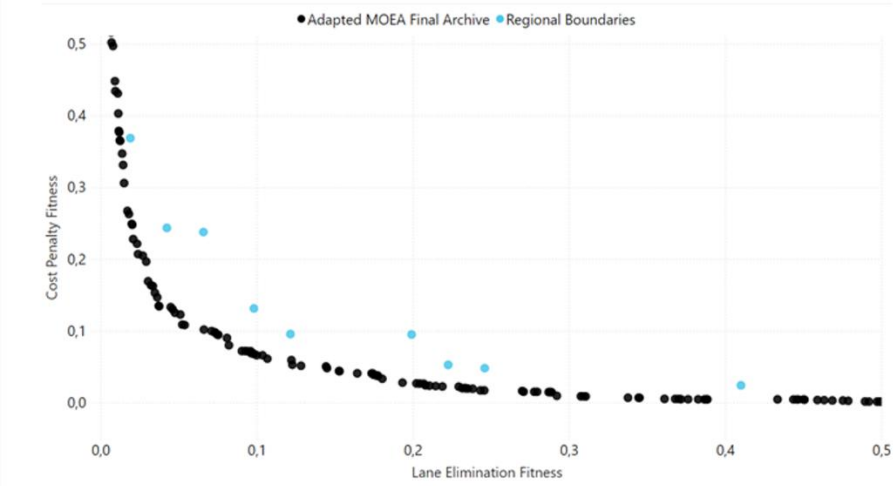


b) Shipper B

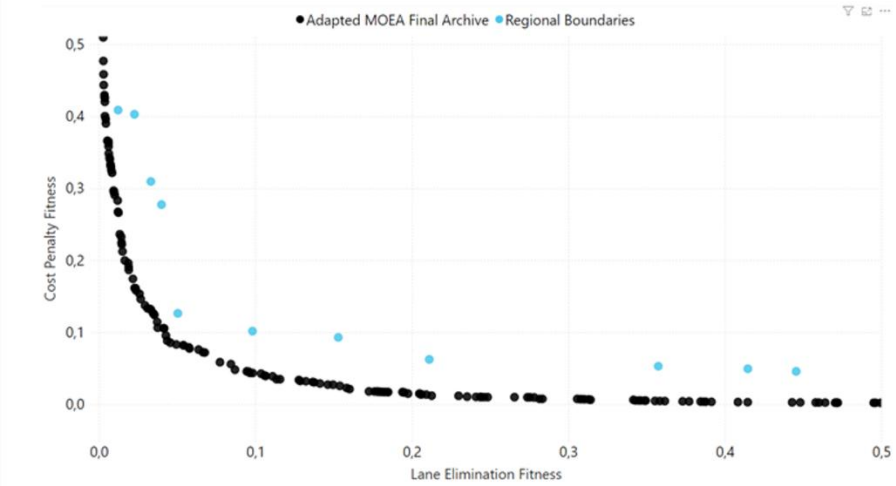


c) Shipper C

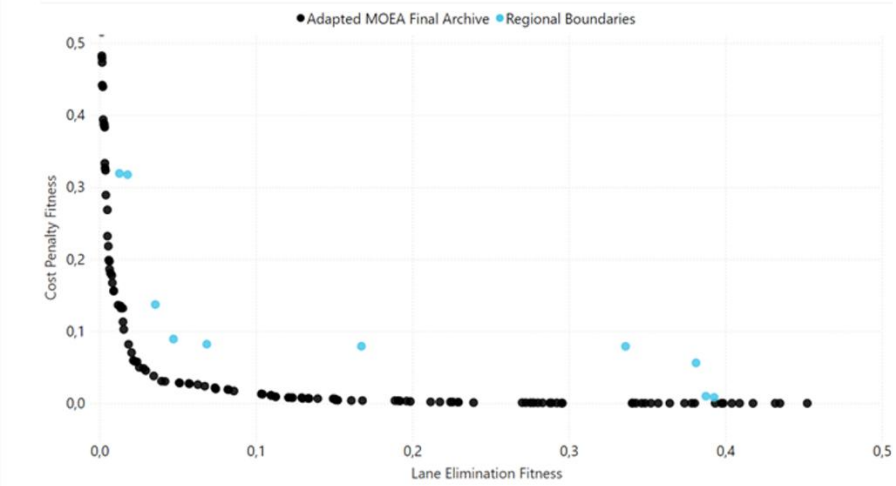
Figure 144: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the adapted NSGA-II MOEA.



a) Shipper A

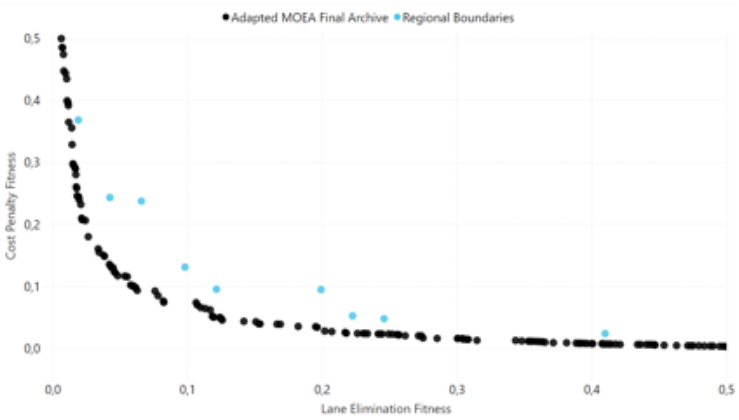


b) Shipper B

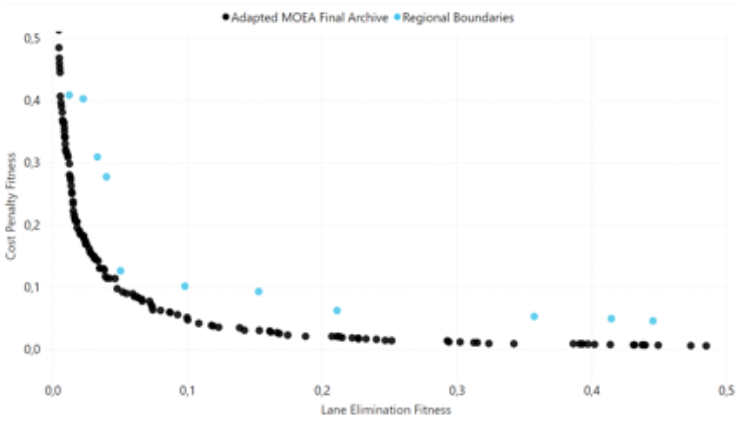


c) Shipper C

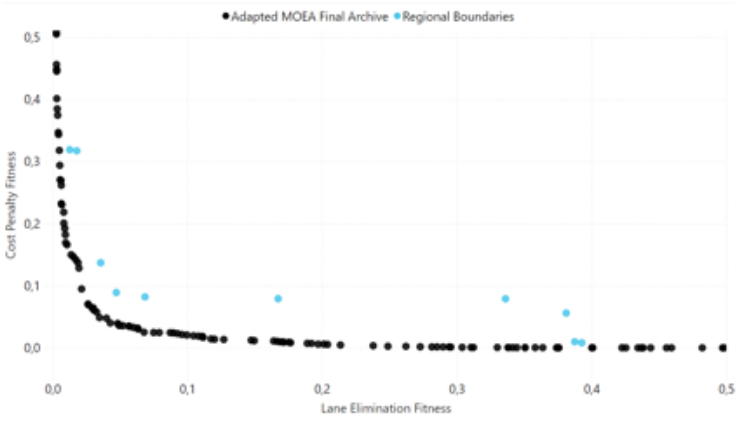
Figure 145: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the adapted SPEA2 MOEA.



a) Shipper A



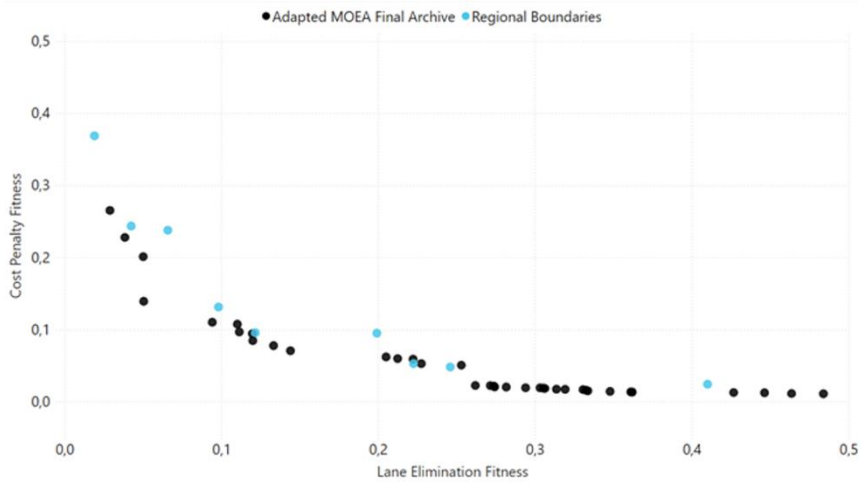
b) Shipper B



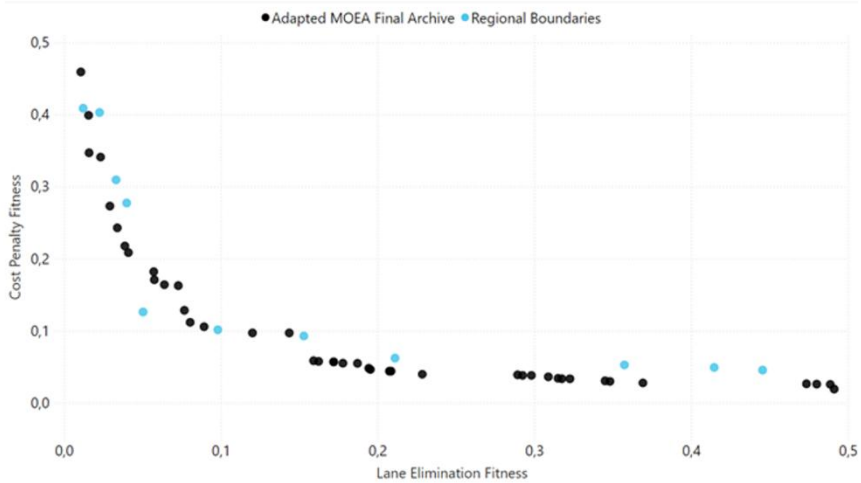
c) Shipper C

Figure 146: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the adapted PESA MOEA.

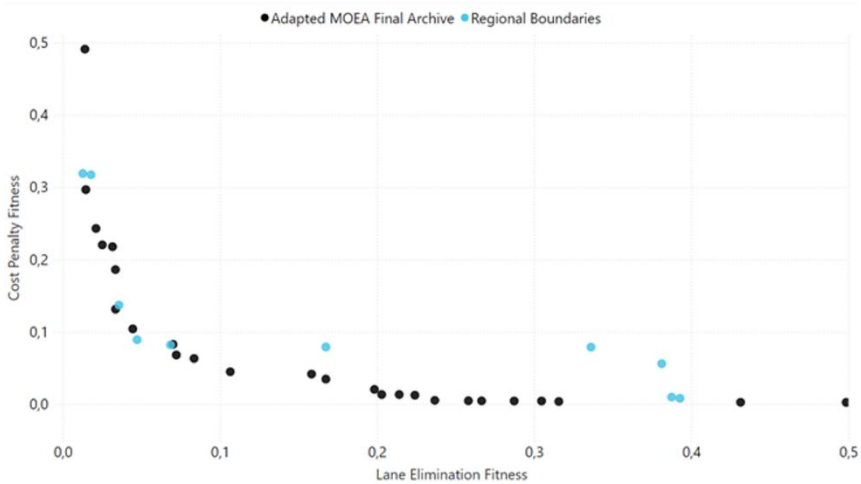
Figure 147 shows that the output of the PAES solution approach for this test was less favourable than those of the previously tested adapted MOEAs. The use of government boundaries to define cluster sets produced at least one visibly identifiable solution per shipper that was not dominated by any PAES solution. This provided an argument against this MOEA as a valid technique for solving the LECP.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 147: Scatter plot showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and nondominated solutions generated by the adapted PAES MOEA.

The positions of the nondominated sets generated by each MOEA for each problem instance relative to the regional boundary baseline solutions are summarized in Figure 148.

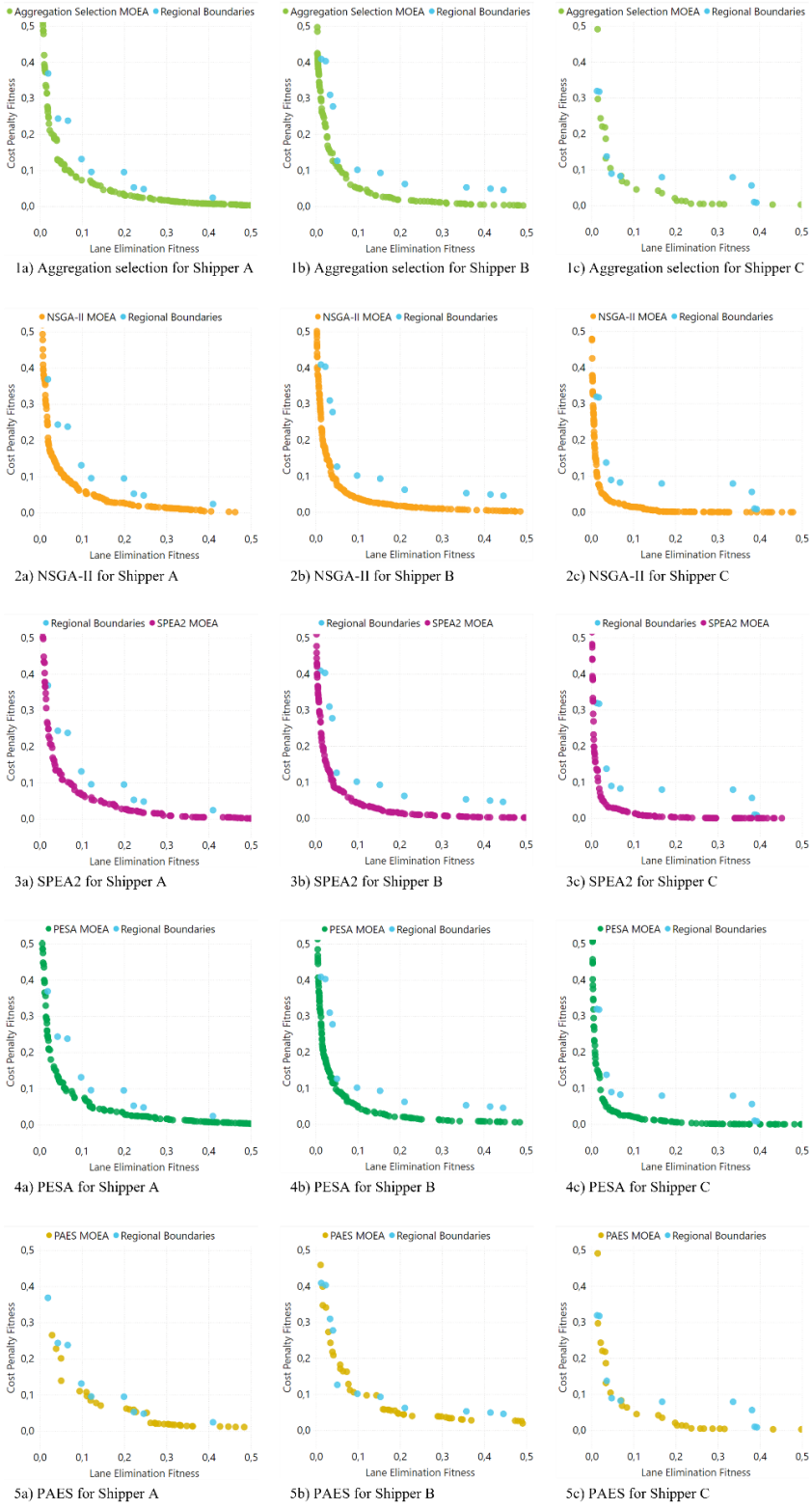


Figure 148: Scatter plots showing lane elimination and cost penalty fitness values of individuals that use government-defined boundaries and the nondominated solutions generated by the adapted MOEAs.

Table 17 shows that the government boundary solutions are entirely dominated by the archives generated by all tested adapted MOEAs, except PAES. Most notably, PAES' PF_{known} dominated only three of the ten nondominated government solutions for shipper A's rate card.

Table 17: Summary of number of government boundary solutions not dominated by any individual of the archive generated by each MOEA for each problem instance after 12 800 fitness evaluation.

Adapted MOEA	**Nondominated government boundary solutions**					
	Shipper A		**Shipper B**		**Shipper C**	
	Number	%	**Number**	%	**Number**	%
Aggregation Selection	0	0	0	0	0	0
NSGA-II	0	0	0	0	0	0
SPEA2	0	0	0	0	0	0
PESA	0	0	0	0	0	0
PAES	5	50.00	4	25.00	4	30.77

7.8.3 Baseline k-means comparison test

Methodology

Pareto front approximations PF_{known} generated by each adapted MOEA over 12 800 fitness evaluations were compared to the nondominated members of 12 800 sample solutions created via the baseline k-means method. The extent to which the MOEA archive dominates the k-means solutions was interpreted as a proxy measure for the ability of the solution approach to find a genuinely enhanced set of alternatives for a DM according to the definition of the LECP.

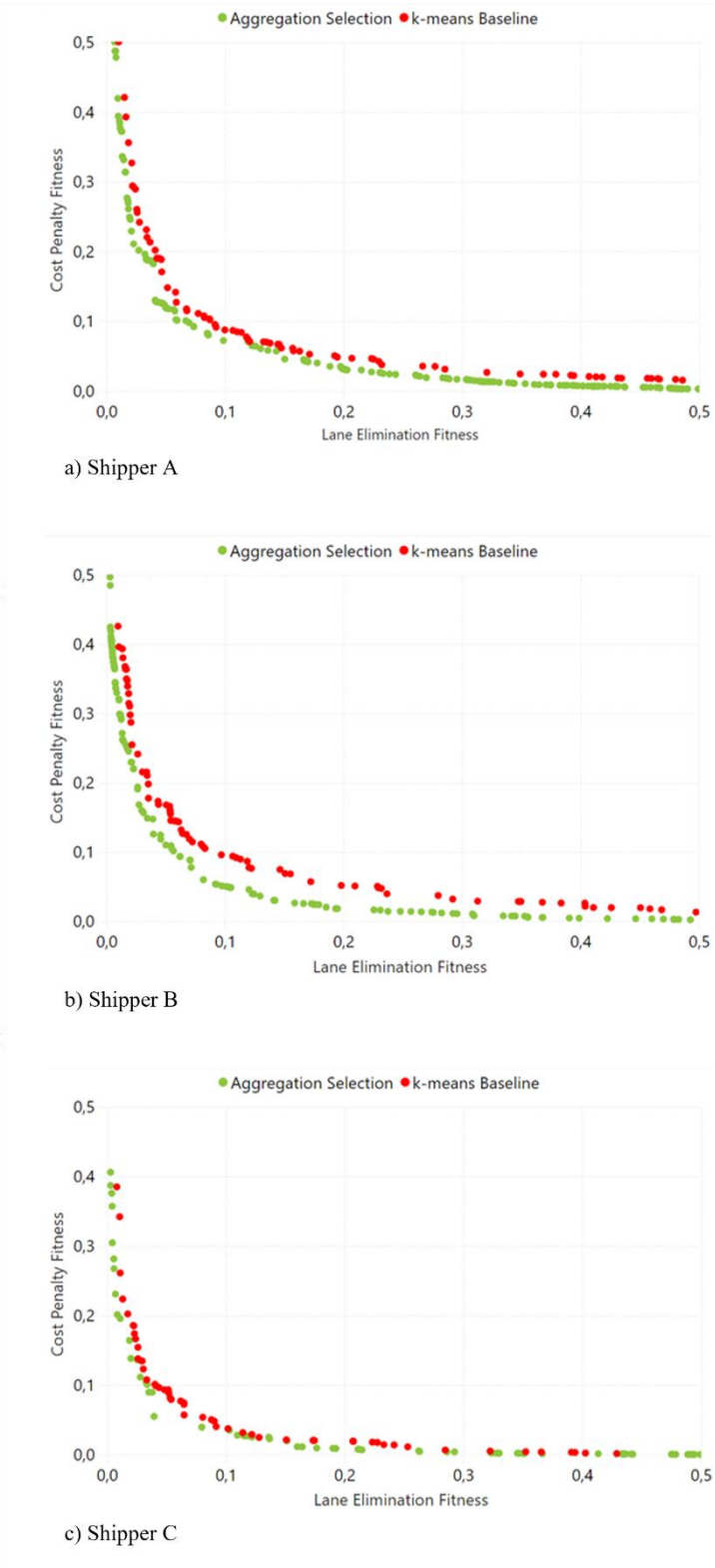
Should the MOEA archive not dominate every member of the baseline k-means population, it could be argued that the solution approach was not conclusively effective. This would be represented visually[42] as an archive with members sometimes positioned higher in phenotype space than the front that represents the nondominated members of k-means population.

Results and Discussion

Figure 149, Figure 150, Figure 151 and Figure 152 visually indicate that aggregation selection with linear combination of weights, when used with duplicate management and elitism, as well as the adapted NSGA-II, SPEA2 and PESA MOEAs, provided better Pareto front approximations after 12 8000 fitness evaluations than a baseline k-means population of 12 800 individuals. These figures also show that

[42] The visuals provided in this section show only a subsection of the phenotype space to allow easier discernment of the relative qualities of the archives in the region of the front where this is most apparent.

NSGA-II and SPEA2 generated archives that were the most conclusively superior to the k-means method.

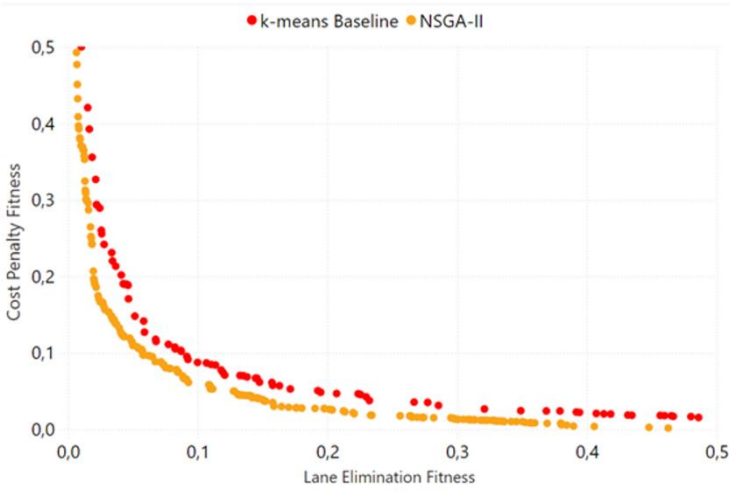


a) Shipper A

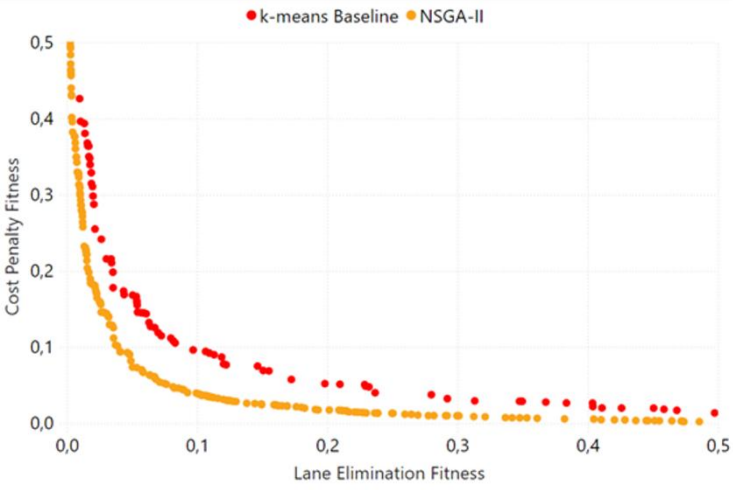
b) Shipper B

c) Shipper C

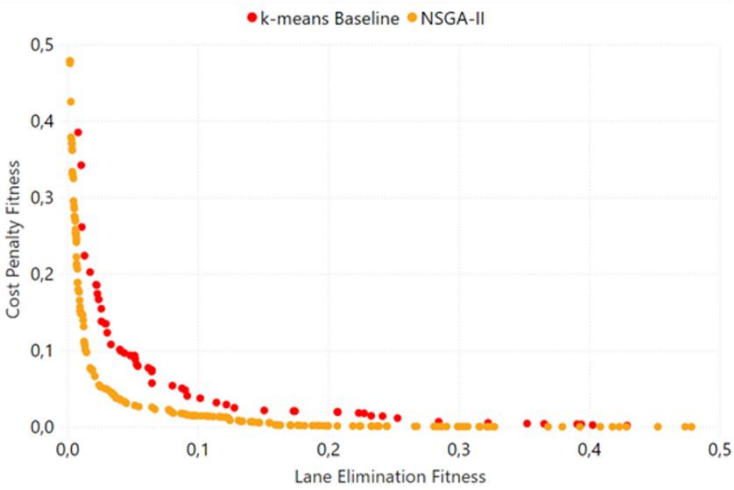
Figure 149: Scatter plot showing lane elimination and cost penalty fitness values of nondominated individuals generated by the k-means baseline method and the adapted aggregation selection MOEA.



a) Shipper A

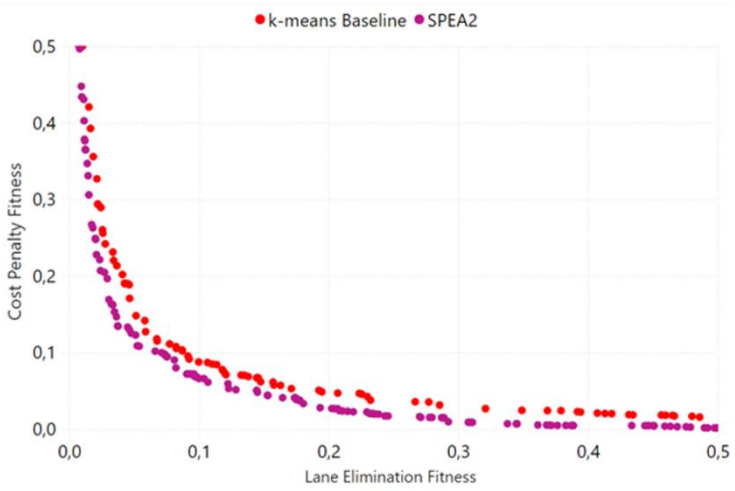


b) Shipper B

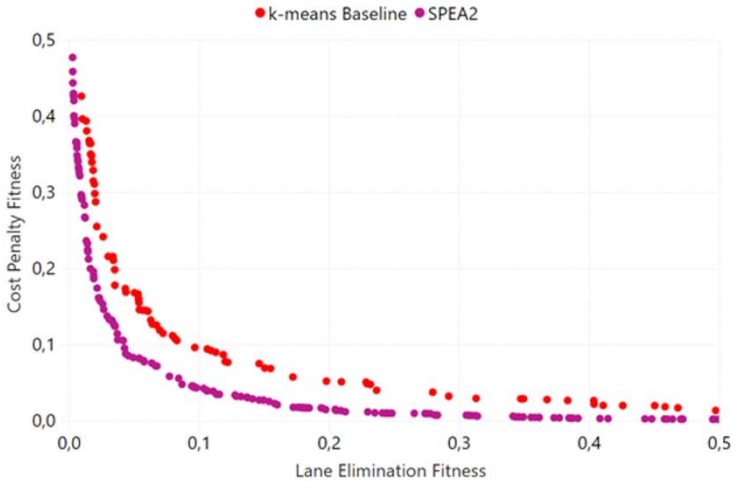


c) Shipper C

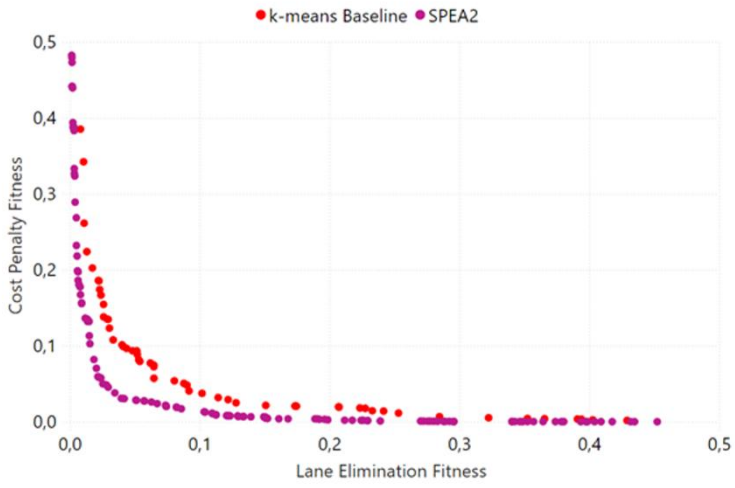
Figure 150: Scatter plot showing lane elimination and cost penalty fitness values of nondominated individuals generated by the k-means baseline method and the adapted NSGA-II MOEA.



a) Shipper A

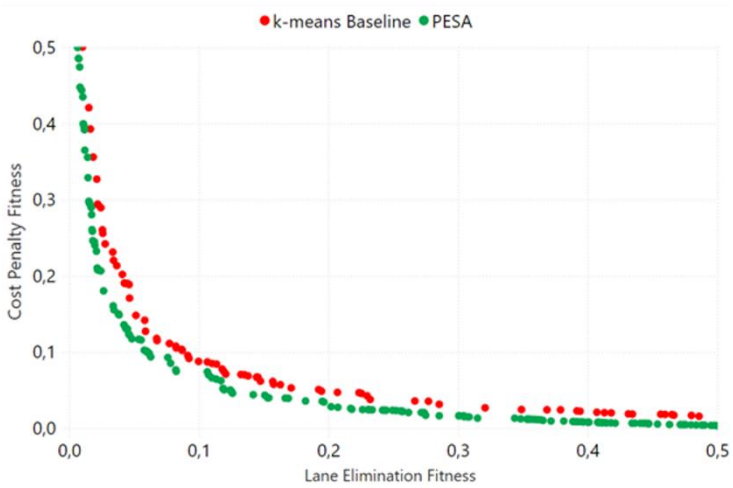


b) Shipper B

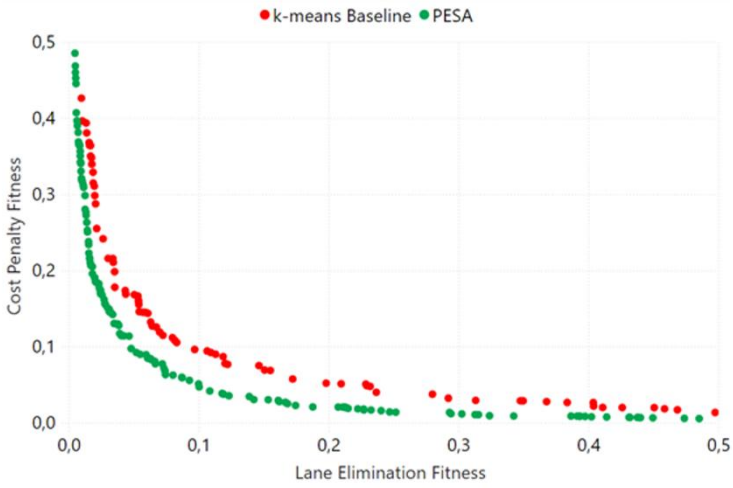


c) Shipper C

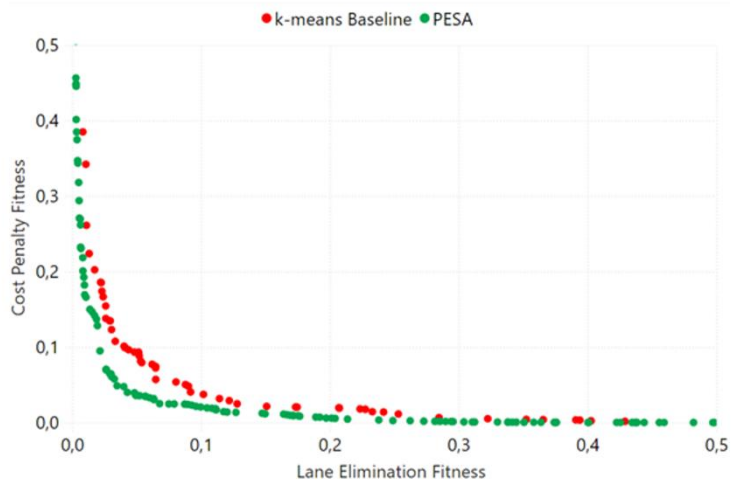
Figure 151: Scatter plot showing lane elimination and cost penalty fitness values of nondominated individuals generated by the k-means baseline method and the adapted SPEA2 MOEA.



a) Shipper A



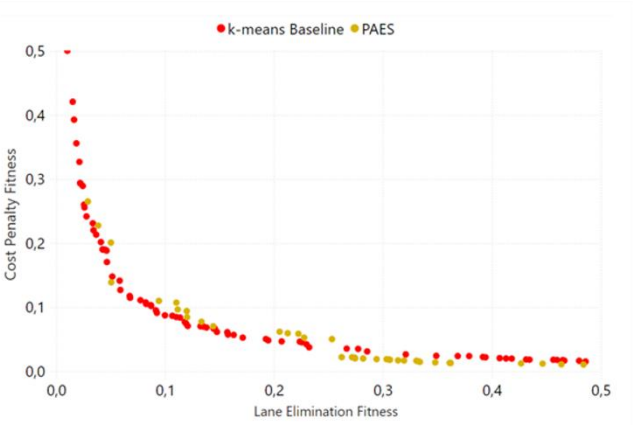
b) Shipper B



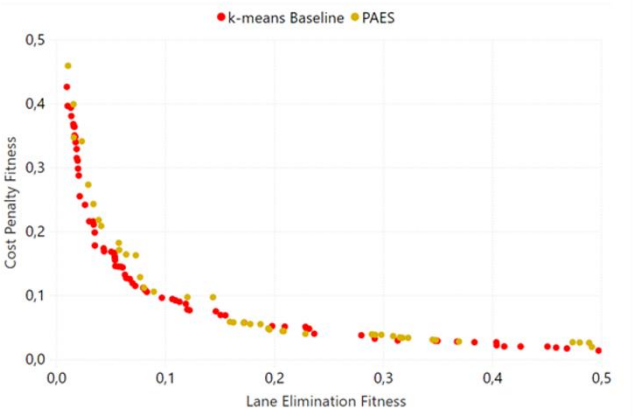
c) Shipper C

Figure 152: Scatter plot showing lane elimination and cost penalty fitness values of nondominated individuals generated by the k-means baseline method and the adapted PESA MOEA.

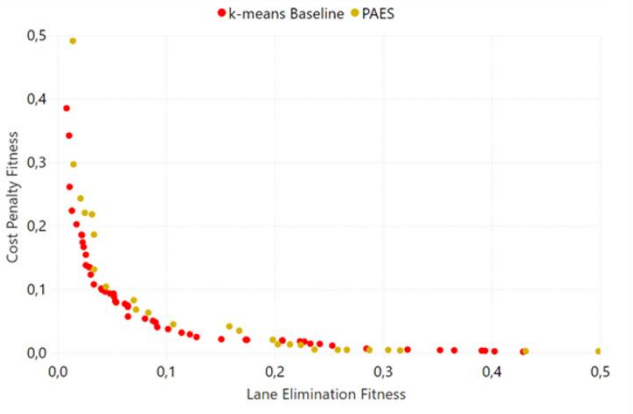
Figure 153, however, indicates that the adapted PAES MOEA was unable to sufficient improve upon a starting population of 200 solutions and provide a good selection of alternatives for a DM. That the k-means baseline method, with the same number of fitness evaluations, was able to achieve a conclusively superior PF_{known} strongly suggested that the PAES algorithm is not a valid solution approach to the LECP.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 153: Scatter plot showing lane elimination and cost penalty fitness values of nondominated individuals generated by the k-means baseline method and the adapted PAES MOEA.

Table 18 shows that, in fact, only the archive generated by the adapted SPEA2 MOEA fully dominates the baseline k-means population of 12 800 individuals for all three problem instances. This provides a strong argument in favour of SPEA2 as a genuine solution approach to the LECP. However, aggregation selection with linear combination of weights, NSGA-II and PESA all generated Pareto front approximations that dominate over 90% of the k-means populations after 12 800 fitness evaluations. This represented an ability to consistently produce better alternatives to a DM across a significant span of the Pareto front range.

NSGA-II's archive only failed to dominate two k-means starting solutions over the three tests and therefore was shown to be a valid approach to the problem as well. Aggregation selection, which fails to dominate 8.25% of the k-means population for the shipper C problem instance, and PESA, which dominated only 137 of 147 of the nondominated set baseline population, provided less conclusive evidence of validity according to this test.

PAES' archive fared poorly in this comparison to the baseline k-means method. This adapted MOEA's best performance, that being for problem instance B, was inferior to the worst performance across all other MOEAs across all shipper rate cards. The inability of PAES to substantially dominate the k-means population for two of the three shippers was interpreted as a potential disqualifier as a genuine solution approach to the LECP.

Table 18: Summary of number of baseline k-means solutions not dominated by any individual of the archive generated by each MOEA for each problem instance after 12 800 fitness evaluation.

Adapted MOEA	**Nondominated baseline k-means solutions**					
	Shipper A		**Shipper B**		**Shipper C**	
	Number	%	**Number**	%	**Number**	%
Aggregation Selection	3	2.33	0	0	8	8.25
NSGA-II	0	0	0	0	2	1.72
SPEA2	**0**	**0**	**0**	**0**	**0**	**0**
PESA	2	1.55	10	6.80	6	5.17
PAES	84	65.12	18	12.24	97	83.62

7.8.4 Conclusion

There is an inherent element of subjectivity to the evaluation of an MOEA as a valid solution approach. Table 19 summarizes the outcomes of a conservative interpretation of the validation test results described in sections 7.8.1 to 7.8.3.

Table 19: Summary of empirical MOEA validation results.

Adapted MOEA	Extreme value test	Government boundary comparison test	Baseline k-means comparison test
Aggregation Selection	Pass	Pass	Pass
NSGA-II	Pass	Pass	Pass
SPEA2	Pass	Pass	Pass
PESA	Pass	Pass	Pass
PAES	Pass	Fail	Fail

Aggregation selection with linear combination of weights, NSGA-II, SPEA2 and PESA passed all validation tests. As such, the evolutionary approach to the LECP, when applied using the adapted versions of these MOEAs and the compromise parameters, were deemed valid MOEA solution approaches to the problem by this research. As such, research objective 3.2. was achieved. In contrast, on account of failing two validation procedures, the adapted PAES solution approach was considered an unsuitable method of applying the developed approach.

7.9 Concluding remarks

Chapter 7 described how the three real-world problem instances were solved using MOEAs and the general evolutionary approach presented in Chapter 6. The individual elements of the evolutionary algorithm were successfully validated and then integrated with five solution approaches from the literature reviewed in Chapter 3. The behaviour of the evolutionary algorithms' search of each decision space was monitored and some corrective steps, such as duplicate solution management, were implemented to address problem domain-specific peculiarities. It was notable that the behaviour and performance of each algorithm were not noticeably sensitive to the problem instance, which added validity to the approach as a general method.

The only MOEA that failed to outperform the baseline k-means and government boundary methods was PAES, which was the only approach that did not include any combinative processes. The solution approaches that did include recombination, however, produced Pareto front approximations that were far superior to those generated by baseline techniques. They therefore graduated to the comparative testing phase of testing, which is discussed in Chapter 8.

Chapter 8

Comparative analysis of solution approaches to the LECP

The exploratory component of this research was followed by a comparative analysis of the performances of the MOEAs using a single set of compromise parameters to allow the selection of a recommended MOEA, which is covered in this chapter. It achieves 3.3 of this thesis, which is to provide a recommendation of a known solution approach for solving the LECP. This included analyses of the performances of each tuned and adapted MOEA. These data were then compared to baseline data specifically generated for the experiment to establish each MOEA's ability to generate superior Pareto front approximations of each problem instance than alternative, non-algorithmic methods.

8.1 MOEA comparisons

The comparative analysis exercise represents the key experimentation conducted for this research. This analysis specifically addresses research objective 3.3. by providing recommendations in terms of the solution approach to be applied to the LECP.

The four MOEAs found to be valid solution approaches to the LECP were retained for the comparative exercise. The PAES algorithm was therefore not used for collecting data. The parameters used for the empirical validation exercise, shown in Table 15, were reused for the comparative analysis as they had proven effective in allowing the valid adapted MOEAs to generate high-quality Pareto front approximations. This included the number of fitness evaluations per run, which had provided the algorithms sufficient opportunity to explore and exploit the solution spaces of the three problem instances. Each of the four valid MOEAs were therefore allowed 12 800 fitness evaluations per test run.

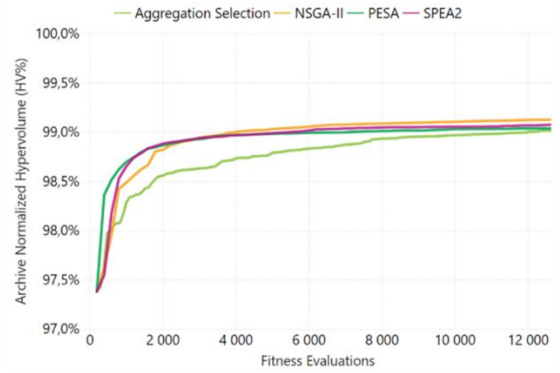
The adapted aggregation selection with linear combination of weights, NSGA-II, SPEA2 and PESA solution approaches were run five times for each shipper rate card. The performance metrics, as well as the current and archive populations, were logged upon completion of each generation of the respective evolutionary algorithms. The worst, best and median runs of each MOEA were compared, per problem instance, and ranked. These rankings were analysed to establish which of the valid LECP solution approaches would be the recommended MOEA for addressing this clustering problem.

8.1.1 Data analysis

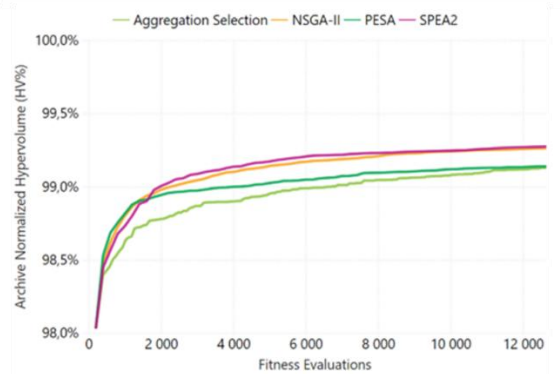
The results of the comparative analysis are summarized in Table 20 and Table 21. The adapted NSGA-II solution approach demonstrated superior performance for all three problem instances in terms of

convergence to the true Pareto optimal front PF_{true}. This MOEA's worst, median and best achieved hypervolume values surpassed those of the remaining three algorithms for shipper A and C's rate cards.

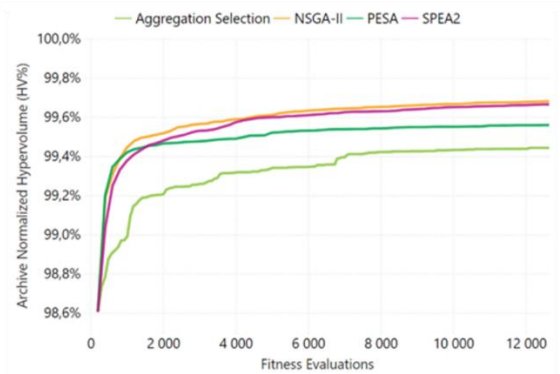
For the shipper B problem instance, however, the SPEA2 generated the solution archive with the highest degree of convergence to PF_{true}. This is reflected by Figure 154. The three graphs shown in this figure also show that the solution approach that demonstrates the greatest acceleration toward PF_{true} does not necessary reach 12 800 fitness evaluations with the better archive. NSGA-II also did not achieve the most consistent performance, as measured by the range of HV values associated with its archives, for any of the shipper rates cards.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 154: Line graphs showing normalized hypervolume of the archive population as a function of number of fitness evaluations for each MOEA's best performing run for the tested problem instances.

Nonetheless, on account of NSGA-II achieving the superior convergence for eight of the nine comparisons, this MOEA is deemed the best performing solution approach to the MOEA in this regard.

The assessment of the diversity achievement of the valid MOEAs was less straightforward. According to the projected hypervolumes (HV_d) of the archives generated after 12 800 fitness evaluations, the adapted PESA dominated the other solutions when tested using shipper rate card A. The adapted NSGA-II achieved the superior median and best HV_d for problem instance B, while SPEA2's worst and best archives exhibited the most diversity of their categories. NSAG-II and SPEA2 each achieved a best consistency in this regard.

Table 20: Summary of the normalized hypervolume associated with the archives found by each adapted MOEA over 12 800 fitness evaluations.

Adapted MOEA	**Normalized hypervolume (HV)**											
	Shipper A				**Shipper B**				**Shipper C**			
	Worst	**Median**	**Best**	**Range**	**Worst**	**Median**	**Best**	**Range**	**Worst**	**Median**	**Best**	**Range**
Aggregation Selection	98.97	98.99	99.02	**0.05**	98.95	99.04	99.13	0.18	99.34	99.38	99.44	0.10
NSGA-II	**99.06**	**99.12**	**99.13**	0.07	**99.21**	**99.24**	99.26	0.05	**99.65**	**99.67**	**99.68**	0.03
SPEA2	99.02	99.04	99.08	0.06	99.19	99.21	**99.28**	0.09	99.59	99.62	99.67	0.08
PESA	98.99	99.01	99.04	**0.05**	99.10	99.13	99.14	**0.04**	99.55	99.56	99.56	**0.01**

Table 21: Summary of the projected hypervolume associated with the archives found by each adapted MOEA over 12 800 fitness evaluations.

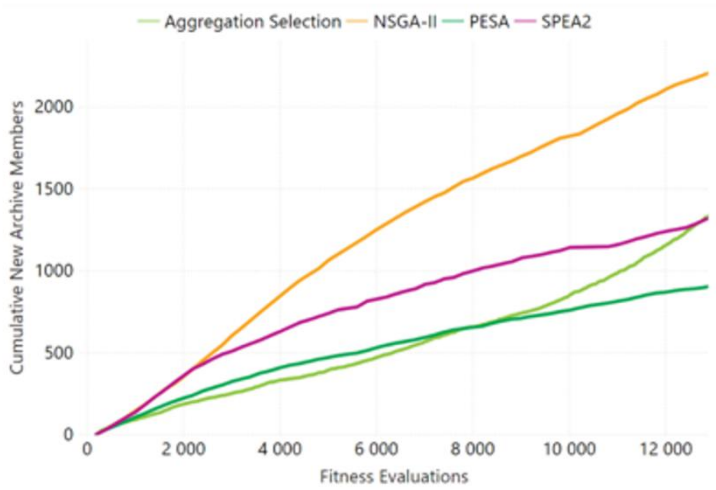
Adapted MOEA	**Normalized projected hypervolume (HV_d)**											
	Shipper A				**Shipper B**				**Shipper C**			
	Worst	**Median**	**Best**	**Range**	**Worst**	**Median**	**Best**	**Range**	**Worst**	**Median**	**Best**	**Range**
Aggregation Selection	48.95	48.97	49.15	**0.20**	48.70	48.93	49.26	0.56	48.80	48.83	49.02	0.22
NSGA-II	48.85	49.16	49.27	0.42	49.25	**49.43**	**49.51**	0.26	49.16	**49.26**	49.29	**0.13**
SPEA2	48.82	49.03	49.12	0.30	49.21	49.28	49.36	**0.15**	**49.17**	49.20	**49.41**	0.24
PESA	**48.99**	**49.20**	**49.33**	0.34	**49.29**	49.39	49.49	0.2	49.15	49.20	49.36	0.21

8.1.2 Discussion

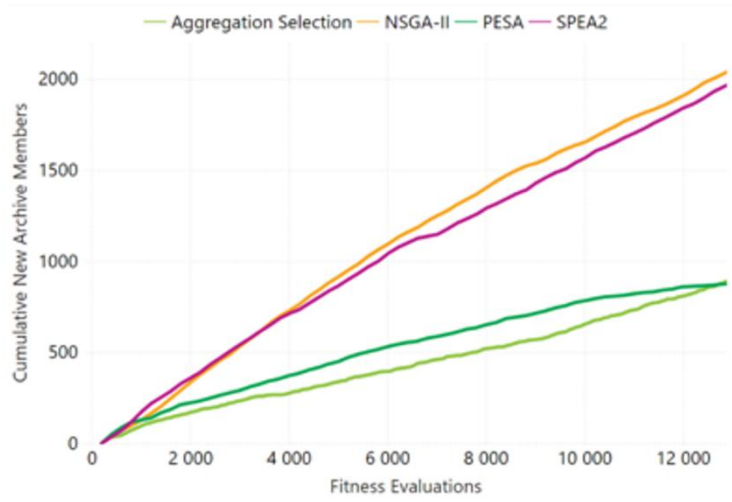
The clear superiority of the adapted NSGA-II in terms of convergence to the true Pareto front PF_{true} strongly favours this MOEA as the solution approach of choice. Furthermore, while its dominance with respect to archive projected hypervolume could not be established, its apparent diversity parity with the other valid adapted MOEAs positions NSGA-II as the favoured LECP solution approach.

However, it was apparent that tests may not have fully explored each MOEAs potential. While visualizations such as Figure 154 suggest that each algorithm converged within 12 800 fitness evaluations, measurement of the number of archive introductions suggests otherwise. Figure 155 shows that all the valid adapted MOEAs showed little indication of converging to a single archive population by the end of their test runs. Each algorithm continued to generate solutions that dominated archive individuals throughout their execution. Notably, aggregation selections rate of archive introductions increased near the end of its best runs for two of the three shippers.

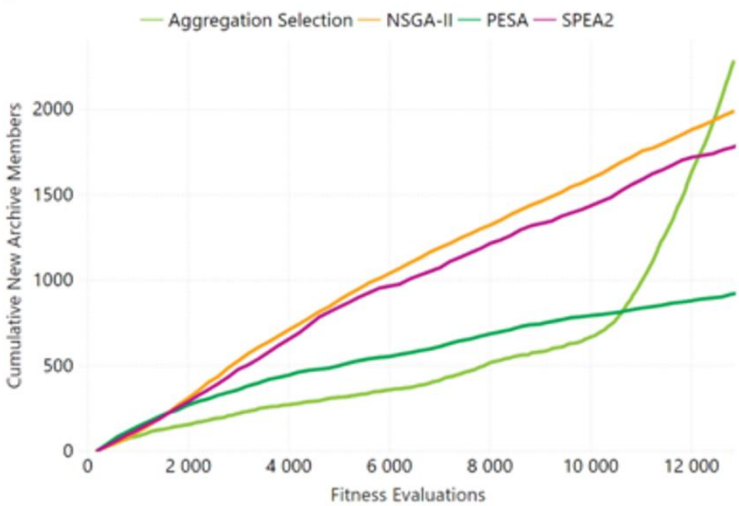
This continued evolution of the collective genotype suggests that although the improvement in the phenotype space slows considerably, each algorithm has the potential to improve the quality for their Pareto front approximation PF_{known}. The real possibility of this affecting the outcome of comparisons between the MOEAs along these dimensions was implied. Nonetheless, the assertion that the adapted NSGA-II was the best-performing MOEA for the three tested problem instances of the LECP can still be argued, as the solving of this problem in practice requires a termination criterion regardless and this solution approach generated superior archives within the scope of the experiment.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 155: Line graphs showing the number of solutions generated and added to the archive population as a function of number of fitness evaluations for each MOEA's best performing run.

8.2 Parallelization comparisons

The scope of this thesis does not explicitly prescribe parallel processing as a technique to address the LECP. However, the potential of these strategies, indicated by literature reviewed for this research, to enhance an MOP solution approach warranted investigation of its effectiveness in the LECP context. The reviewed parallelism methods were evaluated in terms of their applicability to this problem. An island parallelism model was selected, incorporated into the recommended MOEA for the LECP and then tested. The findings of this experimentation served as an input to the overall solution approach recommendation and provides additional points for consideration by the practitioner looking to solve the LECP in her context. This section thereby supports research objective 3.3.

For the empirical testing of this pMOEA variation, the compromise parameters previously used for the empirical validation and comparison of the adapted MOEAs were adjusted for each thread as described in Table 22. These were selected such that threads 1 and 2 were configured to perform a greater degree of exploration than previously NSGA-II MOEA runs, while the parameters chosen for threads 3 and 4 encouraged more exploitation.

Table 22: Parameter values used for each thread of the heterogenous pMOEA.

Parameter name	**Original compromise value**	**Thread 1**	**Thread 2**	**Thread 3**	**Thread 4**
Initial population size	200	200	200	200	200
Cluster-blend crossover probability	0.1	0.25	0.15	0.05	0.01
Cluster-swap crossover probability	0.9	0.75	0.85	0.95	0.99
Primary mutation rate	0.05	0.2	0.1	0.02	0.01
Secondary mutation rate	0.1	0.25	0.15	0.05	0.025
Cluster-split mutation probability	0.33	0.33	0.33	0.33	0.33
Cluster-merge mutation probability	0.33	0.33	0.33	0.33	0.33
Distance bracket mutation probability	0.33	0.33	0.33	0.33	0.33
Maximum number of fitness evaluations (t_{max})	12 800	12 800	12 800	12 800	12 800

8.2.1 Data analysis

The use of parallel processing is shown by Figure 156 to significantly enhance the adapted NSGA-II MOEA's effectiveness and efficiency in solving the LECP for all three problem instances. Figure 156(b) shows a particularly large improvement. Figure 156 also shows the best- and worst-performing run from the comparative analysis exercise presented in section 7.9, which demonstrated the significance of the improvement achieved with the incorporation of parallelism.

Table 23 shows that simple threading results in quicker and greater convergence to the true Pareto front PF_{true}. It also shows that, for shipper rate cards with greater alignment between transport rates and geography, shown by higher starting HV values, parallelism produced a greater benefit in terms of efficiency. However, parallelism yielded less improvement in terms of the quality of the archive found with the same computational expense that when it was used with more disordered rate cards. This was expected, as the k-means algorithm was more suited to find financially homogenous clusters in such cases. The exploitative advantages of the pMOEA were therefore more relevant and the opportunity for a parallelism strategy that does not enhance the explorative ability of the overall algorithm to improve the quality of the final archive was reduced.

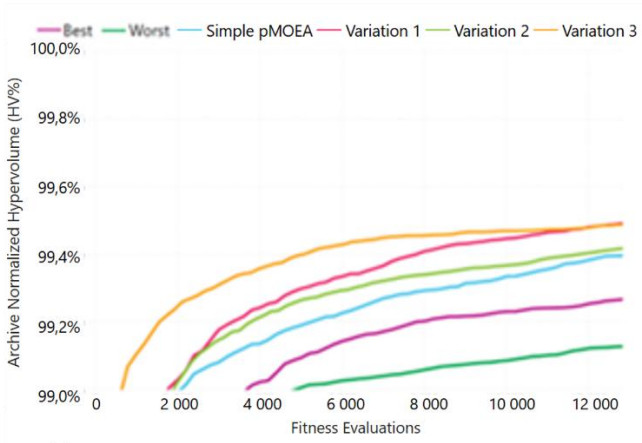
The results achieved by variation 1 of the pMOEA showed that the use of different individuals for each island subpopulation greatly enhanced the efficiency of the adapted NSGA-II in all tested problem instances. The improvement with respect to the final achieved convergence to PF_{true}, however, was less pronounced. This suggested that the simple pMOEA strategy did not require further genetic diversity from the outset to reach a good Pareto front approximation.

The introduction of migration generally enhanced the parallel processing of the LECP, especially when tested on shipper B's rate card. The decline in both pMOEA efficiency and effectiveness for problem instance A may have been a consequence of genetic drift. Although the use of heterogenous islands for variation 3 did yield an improvement in the final convergence achieved for this shipper, it recovered only to the same level of performance observed for variation 1. It is therefore possible that, for some problem instances of the LECP, migration was generally detrimental to the behaviour of the NSGA-II algorithm.

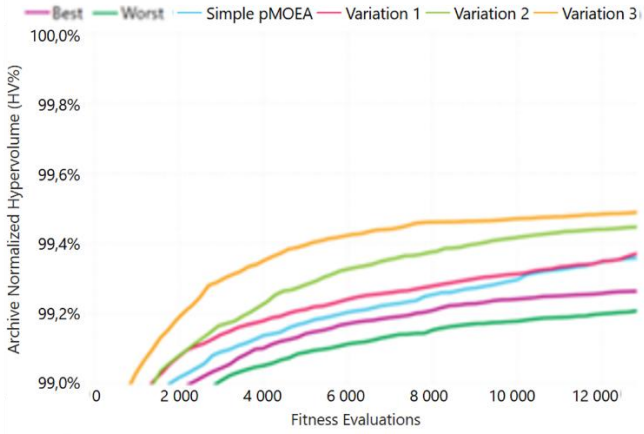
Heterogeneity improved the performance of the migratory pMOEA for all tested problem instances. Variation 3 was not evaluated in terms of efficiency due the expected and observed differences in rate of algorithm progression between the exploratory and exploitative threads[43]. The graphs for this variation's recorded normalized HV_d represent the highest performing threads as a function of fitness evaluations for each problem instance. These were invariably the exploitative threads, which benefitted from a united pool of individuals to which the further-advanced exploitative threads had contributed. The stopping criterion of 12 800 fitness evaluations was typically met by the latter threads materially sooner than the former threads, providing an extended exploitation phase of the pMOEA that yielded significantly greater final convergence to PF_{true} than that achieved by variation 2.

[43] The exploitative threads ran with a significantly higher computational load primarily due to the greater proportion of cluster-blending recombination shown in Table 12.

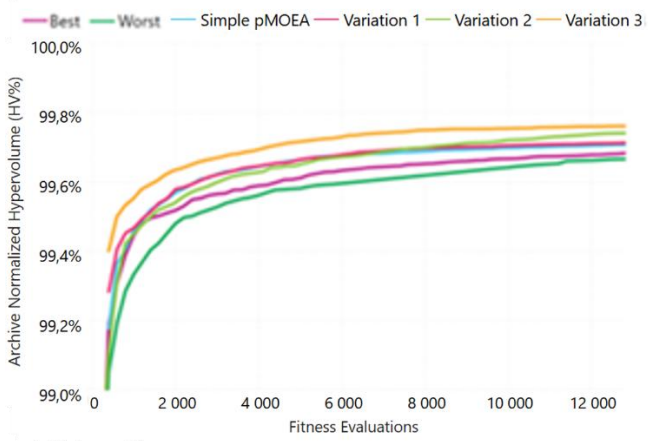
Table 24 shows that simple parallelization of the NSGA-II algorithm was detrimental to the diversity achieved by this adapted MOEA-II. However, when migration and heterogeneity were introduced, this aspect of the combined archive was much enhanced compared to that found by a serial NSGA-II run.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 156: Line graphs showing normalized hypervolume of the combined archive population of all threads running for the tested pMOEA variations, as well as the best and the worst of five serial runs of the adapted NSGA-II MOEA as a function of number of fitness evaluations for the tested problem instances.

Table 23: Summary of the performance of the pMOEA variations in terms of the number of fitness evaluations required to achieve an equal or greater archive normalized hypervolume than the serial adapted NSGA-II MOEA after 12 800, as well as their own final normalized hypervolume values achieved over 12 800 fitness evaluations compared to the serial MOEA.

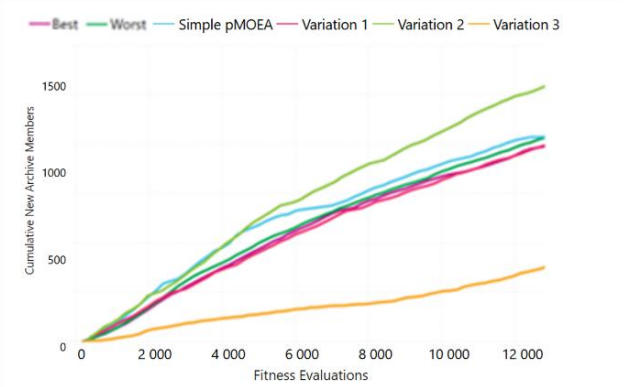
pMOEA variation	Shipper A			Shipper B			Shipper C		
	Evaluations to Parity	Final HV	Final Improvement	Evaluations to Parity	Final HV	Final Improvement	Evaluations to Parity	Final HV	Final Improvement
Simple pMOEA	7 014	99.20	0.07	8 016	99.36	0.1	5 813	99.71	0.03
Variation 1	4 009	99.24	0.11	6 620	99.36	0.1	5 612	99.71	0.03
Variation 2	4 841	99.21	0.08	4 289	99.45	0.19	6 391	99.74	0.06
Variation 3	-	99.24	0.11	-	99.49	0.23	-	99.76	0.08

Table 24: Summary of the performance of the pMOEA variations in terms of final projected hypervolume values achieved over 12 800 fitness evaluations compared to the serial MOEA.

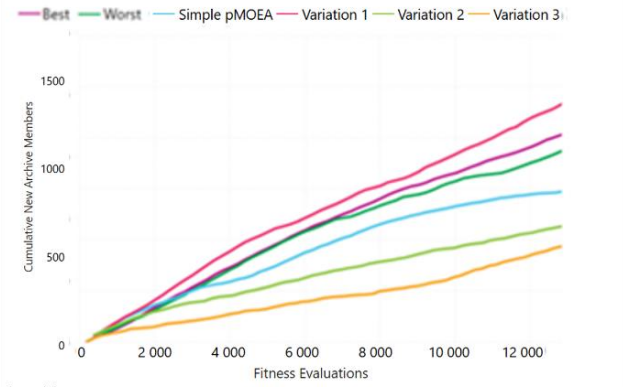
pMOEA variation	Shipper A		Shipper B		Shipper C	
	Final HV_d	Final Improvement	Final HV_d	Final Improvement	Final HV_d	Final Improvement
Simple pMOEA	48.94	-0.11	49.14	-0.06	49.38	0.11
Variation 1	49.03	-0.02	49.08	-0.12	49.19	-0.08
Variation 2	49.15	0.1	49.37	0.17	49.32	0.05
Variation 3	49.27	0.22	49.51	0.31	49.42	0.15

8.2.2 Discussion

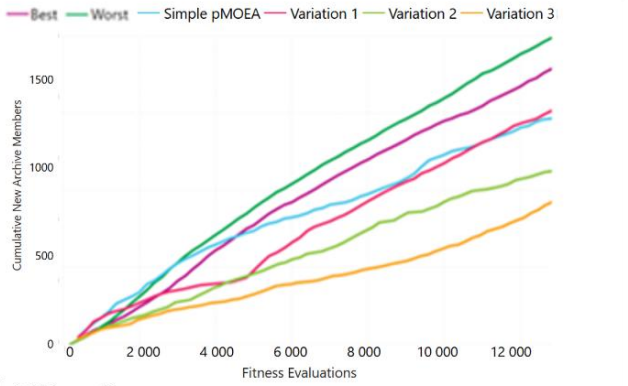
Figure 157 shows that neither the serial nor parallel processing tests were run to a state of equilibrium. Few runs that are shown appeared to be slowing their refinement of their Pareto front approximations after 12 800 fitness evaluations. It therefore cannot be stated with certainty that the pMOEA variations would have achieved superior PF_{true} than the serial adapted NSGA-II if allowed to converge to final archives.



a) Shipper A



b) Shipper B



c) Shipper C

Figure 157: Line graphs showing the number of solutions generated and added to the archive population as a function of number of fitness evaluations for the tested pMOEA variations, as well as the best and the worst of five serial runs of the adapted NSGA-II MOEA for each tested problem instance.

Nonetheless, the successive implementation of parallelization sophistication generally produced compounding gains in pMOEA performance within the chosen number of fitness evaluations. These were observed both in respect to convergence to true Pareto front PF_{true} and the diversity of the final archive. The overall result of these incremental improvements position parallel processing of the LECP as a highly recommended strategy for the practitioner looking to solve this problem within a limited timeframe.

8.3 Concluding remarks

Chapter 8 compared the performance of the four MOEAs that qualified by outperforming the baseline methods. It showed that the NSGA-II generated the best Pareto front approximations from a convergence perspective across all three problem instances. As NSGA-II, SPEA2 and PESA performed similarly in terms of the diversity of their Pareto front approximations, NSGA-II was deemed the best overall solution approach to address the LECP based on this research's experimentation.

The introduction of parallelization to the NSGA-II algorithm greatly enhanced its efficacy. The final recommendation of this thesis is therefore to utilize a heterogenous island model of the adapted NSGA-II MOEA to solve instances of the LECP, if limited to a single approach. Chapter 9 covers some use cases and implementation notes for practitioners looking to implement the recommended approach, or others like it, to the solve the LECP.

Chapter 9

Implementation considerations

Chapter 1 of this thesis introduces the optimisation goals of the LECP in the context of a transport procurement event. These are managed by a shipper requesting the transport market for rates for specific lanes. This, however, is not the only transport management scenario in which the elimination of lanes, with simultaneous consideration of rate similarities, may be beneficial.

The first section of this chapter unpacks these further scenarios and discusses some procedural requirements for implementing an MOEA solution approach to eliminate lanes in each case. These include the procurement event process. The second section of this chapter is a real-world case study in which the recommended solution approach, the adapted NSGA-II heterogenous pMOEA with migration, was implemented to assist a company prepare for a transport RFP.

9.1 Transport RFP

Hedvall *et al.*, (2017) described the transport procurement process as strategically critical to companies that use external carriers. A shipper's rate collection is often referred to as a *request for quotations* (RFQ) or a *request for proposals* (RFP), depending on the degree of prescription the shipper exerts when requesting bids from the market (Suominem, 2018). Suominem (2018) stated that a bid sheet is used in both the RFQ and RFP process. In both cases, this sheet describes the transport lanes of the bid, which must be defined. He also indicated that RFPs are more common in transport procurement. As such, for ease of reference, this thesis refers to either bid collection process as an RFP.

A request for proposals (RFP) represents the foundational business activity for which the LECP is formulated and solved in this thesis. Caplice (2021) stated that the frequency of procurement events has increased since 2016 and that efforts must be made to make these processes more dynamic and responsive to the transport market. Lanes are also often continuously introduced to a price dataset as the shipper's network adapts (Yang *et al.*, 2019), which often requires ongoing analysis (Parsa, 2019). Given these dynamics and the importance of transport to supply chains, optimization of transport procurement should be explored (Yadati *et al*., 2007).

Griffis and Goldsby (2007) identified "lane design" as a primary motive for shippers adopting transportation management systems (TMS). TMS are technologies facilitating management activities such as freight flow optimization, shipment tracking, and carrier settlement (Coyle, Bardi, and Langley, 2003). However, the design of lanes remains a relatively unstudied aspect of transport management, and no suitable algorithmic optimization of lanes that could be applied in a transport procurement context

could be found in the literature. The literature reviews of this thesis establish the need for the mathematical treatment of lane definition in preparation for an RFP and the experimental component of this thesis validates the developed MOEA approach to optimize this activity. However, there are some important real-world considerations that should be noted by the practitioner that wishes to implement this approach for such a purpose.

Arguably the most significant limitation of the MOEA approach is that the rate card is simultaneously a primary data input requirement of the algorithm and the key output of the subsequent business process, namely the RFP. According to the problem statement of this research, both the input and output rate cards must be defined according to some lane representation of the SCN. However, the value of solving the LECP in this context is predicated on the existence of opportunities to improve upon the input lane structure for the RFP based on rates that are yet to be collected. It follows that some estimation of the future rate card must be sourced or generated such that the solution approach can generate a set of lanes that aggregates lanes for a relevant representation of the end-state transport pricing landscape. To do this, the context and intention of the RFP must be understood.

A shipper may be conducting an RFP as part of a fundamental change in its operating model. These include changes from single strategic transport provider strategies to multi-carrier models, or an intention move toward greater usage of outsourced fleets with the intention of selling or deprecating its owned assets. Groenendijk (2016) recommended that the shipper use its own historical transport data for defining the RFP routes, but also suggested that intentions regarding market expansion should be incorporated into the SCN definition. However, in such a case, the shipper's historical data would yield little insight due to the limited visibility of cost attributes of origins, destinations and lanes at this stage. The data available to the shipper would typically be restricted to the incumbent carrier's rate card or a record of transport invoices. There are, however, approaches available to fill this data gap.

The first approach is to acquire and use transport benchmarking data for the input rate card. This can be procured from transport brokers and market specialists, as mentioned by Caplice (2021). The shipper should take care when defining the parameters of the data to ensure that the benchmark costs, specified per lane, are representative of the transport requirements and operational constraints that will be communicated to the participating carriers during the RFP. These may include vehicle types and payload capacities, goods-in-transit insurance requirements, site turnaround times and minimum service levels. The benchmark rates can then be used as the basis of the input rate card for the LECP MOEA.

Benchmark rates often aggregate rates at the lane-level (Caplice, 2021). In such a case, each rate on the MOEA input rate card would be attributed to a single dummy carrier and thereby still support the fitness function evaluations executed by the MOEA. This may, however, result in a misrepresentation of the

cost penalty impact of lane elimination by the fitness values and can thereby be expected to fundamentally influence the ability of the algorithm to find meaningful economic dissimilarities between SCN nodes. This is due to variance between difference origins and destinations on a carrier-level possibly being substantially cancelled out when prices are averaged on a lane-level. It is therefore preferrable to use data for which the carrier of each rate is specified[44].

Benchmark rates can be based on carrier contract rates or actual transport costs. The former would include all sourced rates that might have been available for transport on a particular lane on a particular date. The latter would represent the single rate that was paid for an actual load that was transported. As such, the former provides a more extensive view of transport rates both in terms of breadth of lanes and depth of carriers, while the latter provides is a more accurate reflection of a realistic rate that would secure transport capacity. Although the practitioner must consider the advantages and disadvantages of each before selecting the benchmark dataset to use for the input rate card, if both alternatives are available, the use of an extensive set of contract rates is recommended when the LECP is used in preparation for an RFP.

The second approach is to collect this data from a broad pool of carriers invited to participate in the first of two rounds of the RFP. The initial bidding parties would be asked to submit rates according to the most granular view of the SCN the shipper is able to provide. The submitted rates, which are typically used to narrow the set of participants for a second round of bids, can then also be used as an input to the MOEA to define a set of aggregated lanes to provide to the final group of bidding carriers.

Alternatively, a shipper that already uses multiple external carriers may wish to only receive rates from a wider pool of carriers that may or may not include some or all its existing carriers. This is typically intended to provide the shipper with cheaper rates and cost saving opportunities and is a less disruptive process than a fundamental operating model change. These would often have been defined according to the shipper's SCN at the time of the previous RFP.

In such cases, the shipper can use its existing multi-carrier rate card as the input to the LECP MOEA. This supports Groenendijk's (2016) suggestion that that all RFP data should be archived as it may be needed in the future. However, if this rate card is already defined according to origin and destination clusters, the algorithm would be biased toward redefining the SCN lanes in accordance with the existing structure[45]. This somewhat defeats the purpose of the LECP exercise. As such, if the shipper's existing

[44] The carrier descriptions would certainly be obfuscated by the data provider, but would still provide the MOEA with the ability to calculate cost penalties on the individual carrier-level.

[45] This was observed in the empirical testing covered in chapters 7 and 8. Shipper C's input rate card was already relatively highly clustered and required a large degree of deregionalization. The tested MOEAs were therefore

rates are defined according to relatively large origin and destination areas, it is recommended to supplement the rate card with benchmarking data. This practice was supported by Caplice (2021), who stated that the addition of external rate data provides better predictive rate models. Alternatively, this data can be supplemented with bids collected during a first round of the RFP.

In each scenario, the shipper should define the input rate card according to the highest resolution representation of the SCN that can be practically and accurately compiled. Figure 158 illustrates this concept. Figure 158(a) graphically represents a simple SCN as described by a shipper's rate card. This rate card uses towns as well as zones to define lane origins and destinations. Figure 158(b) shows how the deconstruction of this SCN to the more granular town-level results in the number of lanes increasing from eight to 14 and provides the LECP MOEA with a consistently defined set of lanes as the input for lane elimination. Where the rate card includes a carrier rate for a parent lane that is disaggregated, the same price value for the disaggregated lane is used for all resultant child rates. Should a hypothetical carrier have a rate of R10 000 from regions *X* to *Y*, deregionalization replaces it with R10 000 rates from x_1 to y_1, x_1 to y_2, x_2 to y_1 and x_2 to y_2.

The option often exists to deregionalize to a highly specific level of detail, such as a street address, suburb- or town-level. However, if few existing rates are specified at these levels of specificity, this approach can add many orders of magnitude of computational complexity to the MOEA with little benefit from a clustering perspective. Practitioners must use their discretion based on the nature and extent of the shipper's network, as well as its current and desired carrier pool, to select the node definition level of the input rate card. Regardless of this selection, however, it is recommended that the LECP, as formulated for this research, is solved using consistently defined origins and destination.

It is also worth noting that the procedure developed for this thesis requires a single longitude and longitude pair to describe the location of a node. This is used for distance determination, which is required for the separate treatment of long-distance and local lanes, as well as enforcement of the non-overlapping constraint. As such, the use of more specific nodes on the input rate card is preferable, as it is easier to determine a representative geocode for a smaller node area that supports these two MOEA processes, rather than a large geographic area.

---

able to achieve a significantly higher HV with this dataset than with those of the other shippers, as they were able to seek out cost relationships that were inherently more aligned to geography through the previous manual lane elimination exercise. Should C use the LECP MOEA as part of its next procurement event, it would be recommended that it complement its historical input rate card with data sourced from other shippers to provide a less pre-supposed view of the economically significant zones of its SCN.

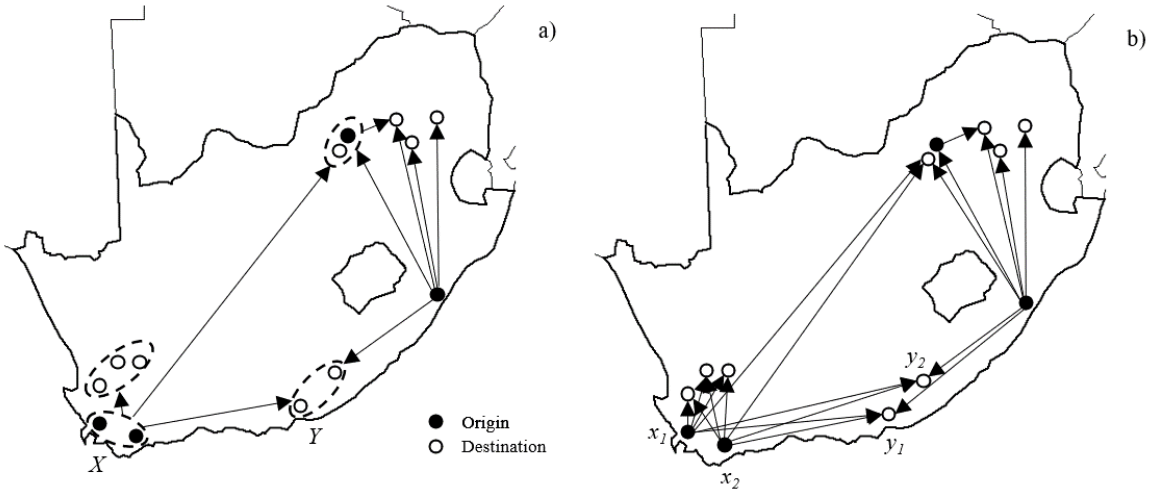


Figure 158: An illustration of the deregionalization process such that granular flows are consistently defined as an input to the LECP MOEA (author's own illustration).

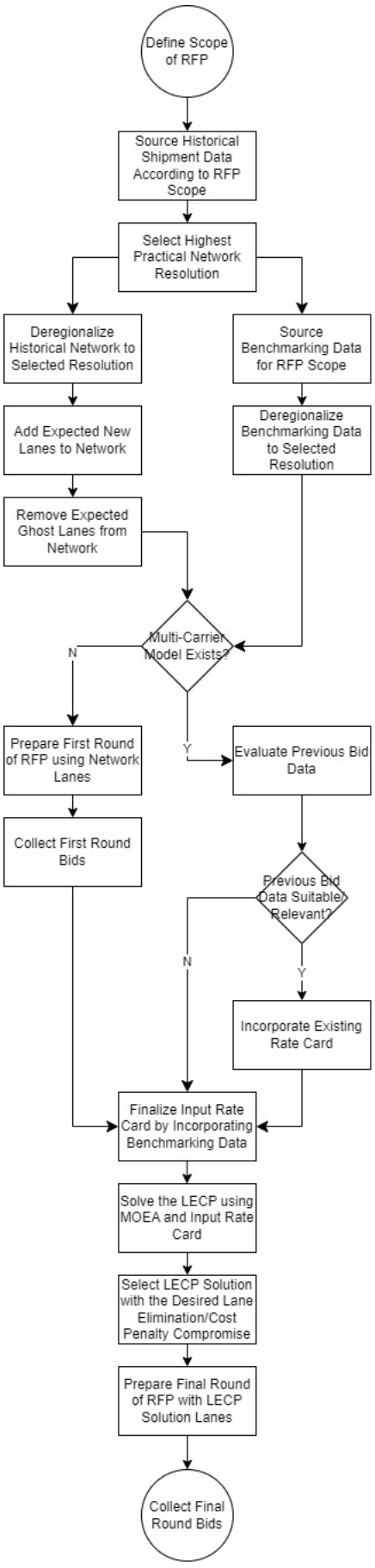


Figure 159: Lane definition process for an RFP (author's own illustration).

9.2 Transport strategy and carrier award

Section 2.1 of this thesis discusses the effect of lane definition on transportation optimisation as presented in literature. Several of the referenced cases addressed supply chain optimisation on a strategic level and used the concept of a lane to describe an input or output of the techniques. These studies, however, assumed that lane definition and carrier rates would be defined outside the scope of the optimisation exercise. Section 1.1 explains why this can be problematic.

Defining lanes as a preparatory step of transport strategy optimisation by addressing the SCN as an LECP can bridge this gap. This can be done immediately after the conclusion of the transport RFP and use the newly established rate card as an input to the MOEA. However, review of a shipper's logistics strategy may, and arguably should, be a less formal and more frequent undertaking than a procurement event. It is also important to note that even the most well-considered and extensive RFP process is likely to produce a rate card that is subject to some evolution before the next procurement event is planned and executed. New transport origins and destinations are likely to be continuously introduced to the SCN after the RFP as the shipper's stockholding footprint adapts to changing market conditions, as well as when customers in new and unexpected geographies are serviced (Caplice, 2021). Additional carrier rates may also be introduced to existing lanes as *ad hoc* negotiations are concluded with new or existing service providers that wish to remain competitive on those routes. Figure 160 supports this concept by showing the actual number of rates introduced to an actual shipper over time. The peaks represent annual RFPs run by this shipper.

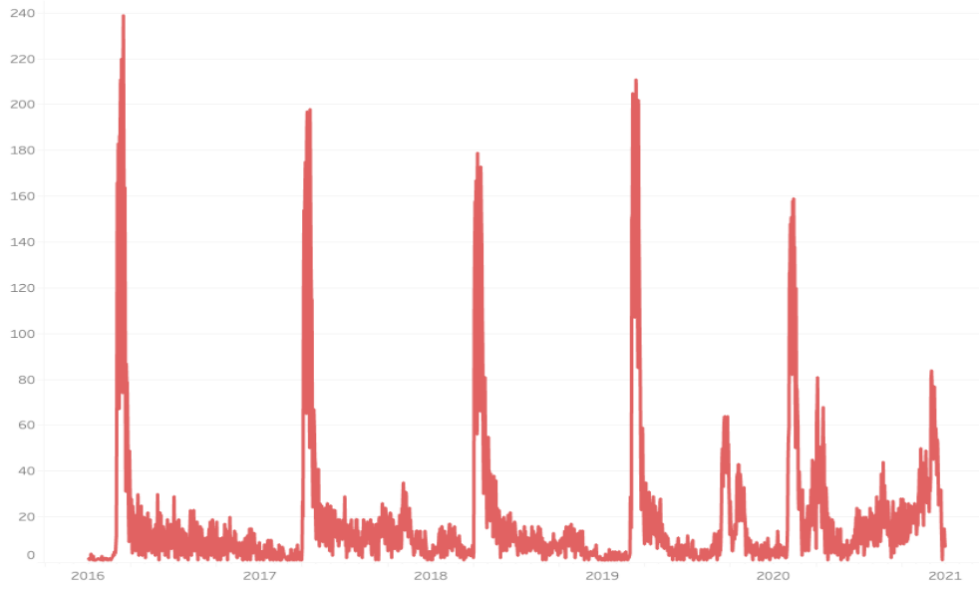


Figure 160: The number of contract rates added to a shipper's rate card by day (Caplice, 2021).

The previous RFP's rates are therefore increasingly more likely to represent an outdated view of the shipper's existing network and cost relationships as time between RFPs progresses. As such, a transport

strategy design that does not immediately follow an RFP should use the existing rate card, as opposed to the latest RFP output, as the input to the LECP MOEA.

Regardless of its source, the underlying lane structure of the rate card may inherit some previous lane aggregation. If the transport strategy follows an RFP executed according to the recommendations of section 9.1.1, the rate card would inherit the MOEA-generated lane structure chosen during the bid preparation stage. If the strategy review is independent of a procurement event, the rate card would inherit whichever lane structure used by the shipper during the ongoing rate management process. Caplice (2021) stated that these rate submissions typically follow a shorter cycle time and, as such, it can be assumed that lane structure used by the shipper for these "mini-bids" would be less structured and scientifically defined that when an RFP is prepared. As such, the origins and destinations of some or all the rates may be articulated on the rate card as suburbs, towns, cities, zones, regions or MOEA-generated polygons. As such, and similar to the RFP process, the rate card should first be deregionalized to lowest sensible single resolution.

The deregionalized rate card should subsequently be used as the input to the LECP MOEA. The generated Pareto-front approximation can then be used by the shipper analyst to select a single lane structure for the transport strategy formulation exercise.

Section 5.2 highlights that additional objectives can be introduced to the LECP according to the problem context. In a transport strategy formulation scenario, there are several additional considerations that may be introduced, although the analyst is advised to avoid excessive proliferation of fitness functions. The shipper may wish to implement the secondary fitness functions formulated in section 5.2.2, which were excluded from the experimental component of this research. Alternatively, the shipper may also desire a balanced network of lanes and, as such, introduce a measure of lane volume equality.

The selected LECP solution may describe lanes with relatively low associated freight volumes or geographically large origins and destinations. These are candidates for exclusion from the principal transport strategy exercise, as the shipper may address these lanes with a simple ad hoc price-first or spot market approach. This mitigates the risk of ghost lanes. In such a case, the shipper may run the LECP MOEA iteratively as key lanes are identified, retained, and re-defined with each run of the algorithm.

It is important to note that the transport strategy may be formulated at a significantly lower SCN resolution than that used for or by the rate card. This would occur when a shipper wishes to strategize with lanes of large associated volumes of freight while still providing carriers with a fine-grained representation of the network when submitting rates.

Figure 161 illustrates how different SCN representations can be used simultaneously for different purposes. Suppose the shipper wished to define its rates according to Figure 161(a) and maintain a rate card of 10 lanes, which would provide carriers with sufficient detail to adequately differentiate prices. For example, a carrier with a depot located nearby node *2* may be able to provide a significantly lower rate to the shipper for lane *z*, to both the carrier and shipper's benefit.

The shipper, however, is not limited to the use of these 10 lanes for a subsequent volume allocation exercise. It may be preferable to further cluster the network to five lanes using another lane structure, as shown in Figure 161(b), such that the amount of transport on each lane is increased and awards are more meaningful. In the example, the special rate for lane *z* would be an input to the optimisation of volume allocation to the discussed carrier. The percentage of assigned available work, however, would be defined for the aggregated *wxyz* lane. The carrier would therefore likely perform less transport on lane *z* than what would be expected if the allocation was performed on the same level as the rate card. In this way, the lower resolution strategic view of the SCN dilutes the saving on lane *z*. However, the saving is still achieved when that carrier does receive work on lane *z* due to the more granular rate card.

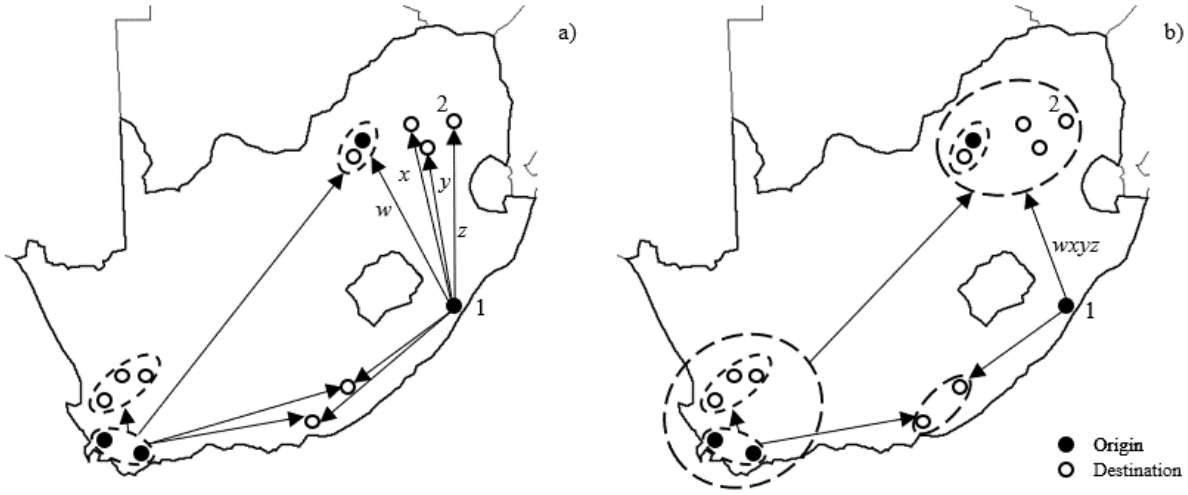


Figure 161: An illustration of the simultaneous use of a) high SCN resolution rates and b) low SCN resolution lanes for transport strategy and carrier awards (author's own illustration).

The expected volume should be forecasted at a resolution higher or equal to that used for the transport strategy exercise. This allows the forecasted amount of freight to allocated to the final selected lane structure generated by for the transport strategy. The strategy and, if relevant, the carrier award, is then completed using lanes that are aggregated on the basis of pricing as opposed to arbitrary boundaries that do not segment the landscape according to transport economics.

The award of work to carriers can be framed and solved as the winner determination problem or optimized using other techniques presented in literature (Caplice, 2021). Alternatively, more manual

and intuitive methods can be used to determine the strategy per lane. Key strategic decisions include whether an owned fleet should provide the transport on a specific lane. Alternatively, the strategy for the lane should specify whether a dedicated fleet of outsourced vehicles should be used or if the rates sourced through RFP should be used to source transport as an ad hoc carrier shopping event per shipment. A further option is to implement a reverse-billing process, often referred to as a spot market, to secure services from a pool of carriers that bid once-off rates per shipment. These cascading tendering models are required as the commitment of transport supply is rarely enforced by shippers (Caplice, 2021). This decision-making process is shown by Figure 162.

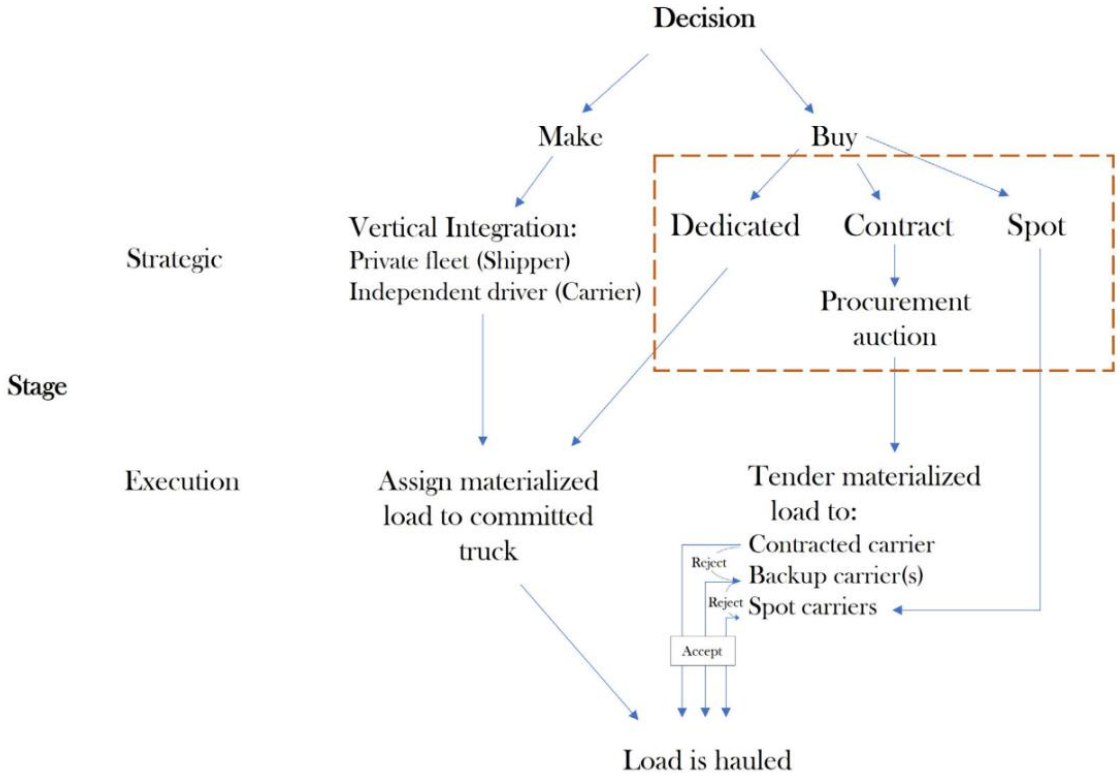


Figure 162: The lane strategy decision-making process (Acocella and Caplice, 2023).

A lane strategy may include any combination of these approaches depending on the volume, variability and consistency of demand. For example, a shipper may choose to first assign shipments to its own fleet of vehicles to ensure their complete utilization. The remainder of the volume is subsequently assigned to a dedicated fleet of outsourced vehicles. Any remaining work is then offered to the contracted RFP carriers. The remaining unaccepted shipments are finally placed on the spot market (Caplice, 2021). These lane strategies can also be specific to certain times of the year to respond to seasonal market changes in supply and demand of transport.

Figure 163 shows the framework developed by Zheng and Oliver (2023) to guide transport strategy select by lane category. Their method classifies lanes based on the annual number of loads on a market-

and shipper-level. The use of the LECP MOEA to define the lanes can potentially result in better lane placement within this framework.

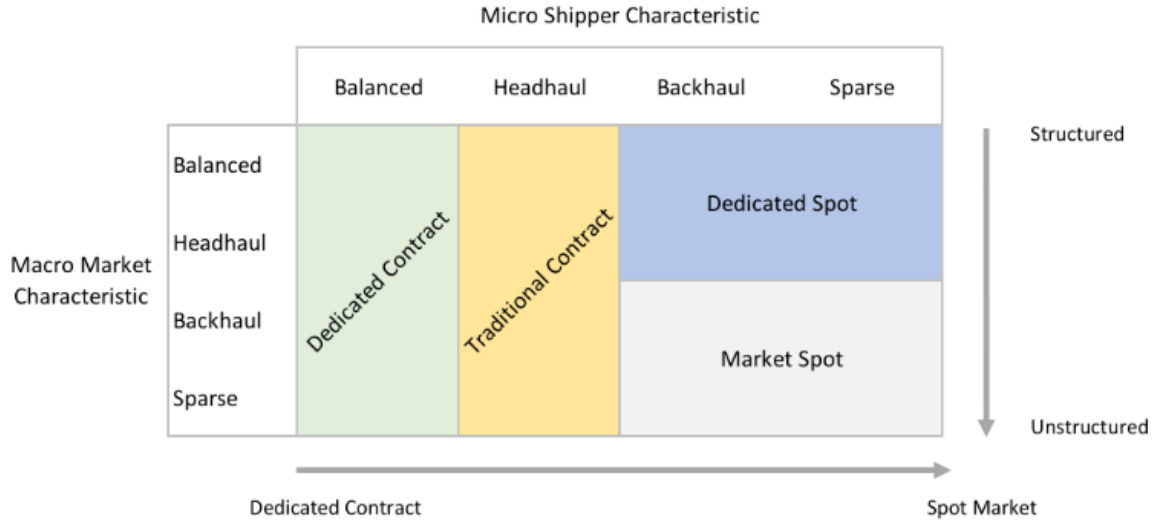


Figure 163: Recommended procurement framework (Zheng and Oliver, 2023).

The consolidation of volume by aggregating lanes using the LECP MOEA can enhance the executability lane strategies. For example, the consolidation of the four northbound lanes from node *1*, shown by Figure 161(a), into a single lane *wxyz*, shown in Figure 161(b), allows the assignment of the shipper's entire owned fleet to the unified lane according to a single strategy. The execution of the strategy then involves the allocation of the individual assets to materialized loads across all four routes as needed. The lane's outsourced dedicated fleet can also be more easily sized according to the combined volume.

The allocation of freight to carriers using predefined percentage is a common practice to ensure that carriers continuously provide capacity. This strategy can prove meaningless if granular lane definitions are used for implementation. The use of low-resolution lanes mitigates this risk, as shown by the example described by Table 25. The lanes correspond with Figure 161. It shows that the application of the same transport strategy yields a different result when applied to four individual routes as opposed to a single unified lane. When lanes *w*, *x*, *y* and *z* are combined as lane *wxyz*, the allocation of shipments to carriers *A*, *B* and *C* correspond closer to the pre-defined allocations. Carrier *C*, particularly, benefits from this model by receiving meaningful work from the shipper over the course of the week. This consolidation of volume also provides more predictability in terms of the use of spot markets to source ad hoc capacity from pre-approved bidders assigned to the lane. The configuration of lanes can also directly affect the rates secured via spot market processes (Collins and Quinlan, 2010).

Table 25: An illustration of the effect of lane aggregation on the outcome of a lane strategy.

	Lane *w*	**Lane *x***	**Lane *y***	**Lane *z***	**Lane *wxyz***
Weekly volume (shipments)	6	8	10	12	36
Owned fleet allocation (8 trucks)	2	2	2	2	8
Dedicated fleet allocation (4 trucks)	1	1	1	1	4
Contracted carrier A (50% allocation)	2	3	4	5	12
Contracted carrier B (30% allocation)	1	2	3	3	8
Contracted carrier C (20% allocation)	0	0	0	1	4

It is important to note that, in reality, the pricing provided by the various carriers for each of these four lanes is unlikely to be proportional and, as such, the optimal allocations would not be the same for each lane. Combining the four lanes into a single lane therefore forces the implementation of a compromise strategy. It would therefore be recommended that the shipper aggregate lanes according to transport costs to minimize the extent of this compromise. This can be achieved by solving this problem as an LECP using an MOEA.

The lane strategy exercise may be repeated for multiple lane structures selected from the MOEA's Pareto front approximation until the desired compromise between the number of lanes, volume per lane and lane strategy specificity is reached. The shipper's TMS is subsequently configured, and its standard operating procedures (SOPs) are defined such that the strategy for each lane is executed on an operational basis.

A routing guide is a heuristic that determines the transport routing decision for an individual shipment. This decision can consist of multiple facets, including the selection of the carrier, mode, equipment and level of service associated with the transport of the freight. A routing guide is often an output of a logistics strategy, which may include a procurement event or carrier allocation optimisation exercise, and defines the routing rules for each type of shipment in accordance with the strategy (Caplice, 2016). Although, in its most basic form, a routing guide can be a simple instruction to use a certain carrier for a particular lane, it is often configured on the shipper's TMS to assist a planner or entirely automate the routing step of the transport planning process when scale or complexity puts the execution of the strategy at risk.

Due to their support role in executing a lane transport strategy, routing guides are often lane specific (Caplice, 2016). The routing guide should inherit the lane definition used to define the transport strategy for the lane. This alignment ensures that the operational reality assigns as closely as possible to the theoretical strategy.

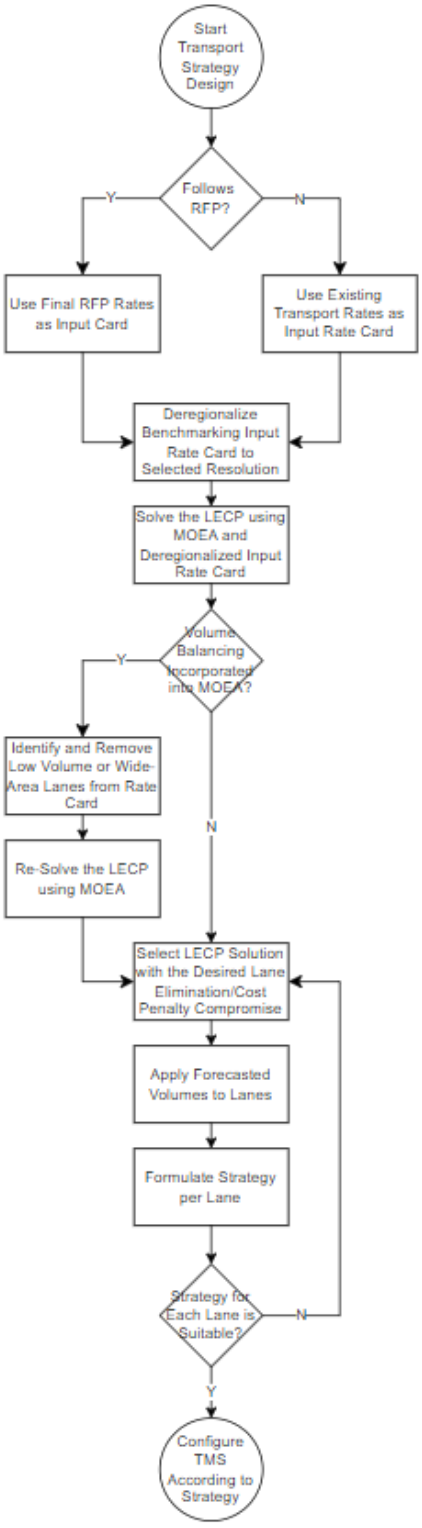


Figure 164: Lane definition process for a transport strategy (author's own illustration).

9.3 Transport budgeting

Shippers often budget for future transport cost. The budgeting process typically begins with the sourcing and preparation of historical transport data for analysis. This can include the normalization of the recorded cost to account for inflation and fuel price changes. An indicative cost of transport is then calculated[46], which may be expressed as any representative statistic of payable amount per unit weight or volume, selling or handling unit, or truckload. It cost can also be described as a percentage of invoice value. The representative cost is often segmented by equipment type, shipment size or time-of-year to facilitate a granular calculation that accounts for expected changes to the shipper's known cost levers.

Regardless of its method of calculation, basis or classification, the representative cost is almost always further segmented by lane and subsequently applied individually to each lane's forecasted volume to determine an expected cost per lane for the forecasted period. Like transport rate cards and strategies, the lanes used this budgeting segmentation require concrete definitions.

The use of the data source system's origins and destinations at a street-address or city level typically results in an unnecessarily large and unwieldly number of lanes[47]. Bandaru and Dolci's (2020) decision to exclude the "long-tail" of low-volume lanes, shown by Figure 165, from their budget reconstruction exercise, provides an example of an analysis that was over-complicated by a granular definition of the historical SCN. Lane aggregation could have provided an alternative remedy by simplifying the view of the network while allowing all historical information to be assimilated into the model without increasing its complexity.

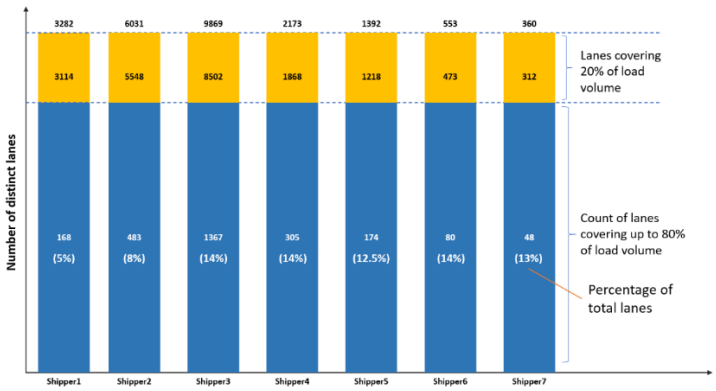


Figure 165: Number of distinct lanes per shipper SCN (Bandaru and Dolci, 2020).

[46] While some shippers may use the carrier rate card for the budgeted period to determine this representative cost, this is generally discouraged due to the uncertainty regarding the eventual proportion of freight that will be transported by each carrier on each lane.

[47] This can often be attributed to the address master data quality issues, such as the use of alternative names for the same town or city, or inconsistent use of the address name, street, suburb, town and/or province fields.

It follows that prior segmentation of lanes can benefit the segmentation of costs when transport budgets are developed. Bandaru and Dolci's (2020) study of budget failures, shown by Figure 166, provides a further example scenario in which the aggregation of lanes may benefit the budgeting process. Segmentation of lanes may have increased the percentage of lanes for which budget data was available for their analysis, which represents a similar dilemma faced by shippers when attempting to calculate historical costs for their budgeting exercise.

Shipper	Count of Load Transactions	# of Load Transactions with Budget data	# of Load Transactions without Budget data	Percentage of Lane Budget Availability
Shipper 19	58083	44126	13957	75.97%
Shipper 7	197652	140335	57317	71.00%
Shipper 1	445737	314977	130760	70.66%
Shipper 6	210518	126060	84458	59.88%
Shipper 33	16441	8666	7775	52.71%
Shipper 4	382926	201197	181729	52.54%
Shipper 3	401304	206830	194474	51.54%
Shipper 8	166590	54996	111594	33.01%
Shipper 41	5999	1854	4145	30.91%
Shipper 28	22452	6211	16241	27.66%
Shipper 9	140422	34694	105728	24.71%
Shipper 15	76678	14347	62331	18.71%
Shipper 21	34007	4658	29349	13.70%
Shipper 27	27093	3510	23583	12.96%
Shipper 23	32048	3780	28268	11.79%

Figure 166: Shipper lane budget availability (Bandaru and Dolci, 2020).

The use of fewer lanes aggregates forecasted volumes and gives rise to less noisy estimations and extrapolations of historical costs. This mitigates the risk of overfitting the budget model. Over-simplification of the SCN, however, can result in a desensitized forecasting model that fails to predict overall spend based on foreseen changes in the network.

The articulation of the shipper's future-state SCN can be optimized according to this trade-off by addressing it as an LECP and using the evolutionary approach to this problem, as well as the recommended solution approach given for this research. It is recommended, however, that the historical costs, as opposed to the prevailing shipper rate card, should be used as the input to the MOEA, as they are typically a superior representation of expected future costs. For example, a rate card may indicate a significant disparity in transport cost to two adjacent towns due to a large pricing difference associated with a single carrier. The historical data, however, may not reflect this disparity due to this carrier seldomly being used to service either town. As such, the rate card used as an input to the LECP MOEA should, in this context, be define the historical representative costs all assigned to a single dummy carrier.

Previously travelled routes might not be included in the shipper's definition of the future SCN. The nodes associated with these flows should be excluded from the MOEA run to ensure maximum alignment of the results to the expected future network. Should a budget cost be required for new routes, benchmarking data can be procured to complete the MOEA input rate card. Alternatively, the prevailing rate card can be used in conjunction with some basic assumptions regarding future carrier usage to estimate the future unit cost of transport for these previously unused lanes.

The output of this optimisation exercise would be a selection of LECP solutions, from which a single lane structure can be selected by the analyst for calculating representative costs. These costs can then be applied to the forecasted freight to provide a budget.

9.4 Concluding remarks

Chapter 9 described some practical considerations for the implementation of an algorithmic approach to lane definition in a transport management context. It explains that a starting dataset is not necessarily available to modellers looking to optimize SCNs, especially before executing a freight rate procurement event. However, it discussed strategies that could be employed to provide the initial granular rate card as an input to the LECP MOEA, each with associated benefits and disadvantages.

Central to the effective use of any attempt to solve the LECP is the appreciation that there is no single optimal lane structure for a given distribution network. A high-quality solution is one that provides alternatives to existing cities, towns, districts, municipalities, provinces and states as origins and destinations that better capture the pricing considerations of carriers. The overarching objective is to simplify transportation management and analysis with minimal rate card disruption, which can be defined and quantified differently depending on the decision-maker's context and preferences. Implementation of the evolutionary approach to the LECP must, therefore, incorporate significant stakeholder engagement to sensitize parties to the nature of this optimization problem and refine elements of the model based on user feedback where needed.

Chapter 10
Conclusions

10.1 Lane Definition as an Optimization Problem

The literature reviewed in Chapters 1, 2 and 3 of this thesis showed the topic of lane definition received surprisingly little attention considering the concept's prevalence in transportation management and optimization. Due to the multitudinous methods by which a supply chain node can be specified, the prevailing definition of a lane as an "origin-destination pair" (Basu *et al.*, 2010) does not provide sufficient direction for precise transport cost analysis, objective optimization or unambiguous commerce engagements between shippers and carriers.

Faced with overwhelming amounts of transactional and freight rate data, as well as a general ambivalence toward lane definitions in their fields, analysts often default to defining lanes at the highest available specificity. This research highlighted multiple cases in which the lanes of a mathematical model were defined on a city-to-city basis without the appropriateness of this method being questioned. Other literature, however, pointed to negative consequences of defining a supply chain network at an unnecessarily high resolution. The literature review also showed that a prevalent alternative method of using administrative boundaries to identify similar origin and destination nodes also may not adequately demarcate the shipper's landscape from a transport cost perspective.

This research purported that a previously unexplored opportunity existed to improve upon these simplistic approaches and standardized lane definitions by customizing the composition of origin and destination nodes as clusters according to third-party carrier rate similarities. This exercise could be treated as a multi-objective optimization problem, which would yield several tangible and intangible benefits to a shipper's transport management. Chapter 2 showed there was no formal definition of any optimization problem that actively sought to cluster networks to achieve this goal. It also explained that existing clustering methods were limited in their ability to cluster in a suitable manner to solve this problem and that a new technique would be required.

Chapter 5 of this thesis provided a first formulation of lane definition as a multi-objective optimization problem that balances network lane coverage and effectiveness, as described by Caplice (2007), as supporting objective 1. This was newly termed the "lane elimination problem" (LECP). The decision variables for an LECP solution were defined assuming that different cluster structures could be defined for origins and destinations for lanes of two different distance categories. The two objective functions presented provided a general view of the goals of the clustering problem that reflected the fundamental

trade-offs of the problem. These functions, however, could be adapted or expanded upon without invalidating the remaining research methodology. Similarly, the set of constraints of the formulation represented a minimalistic take on the necessary LECP constraints required to generate useful solutions in the transport management context that could be extended to cater for situation-specific requirements.

The formulation was used as the basis for an evolutionary approach designed to solve the LECP, which was presented as a generic design in Chapter 6 and addressed supporting objective 2. This included a chromosomal representation scheme, fitness evaluation procedures, algorithm seeding process, evolutionary operator procedures, population evaluation methods and a stopping criterion.

The overall evolutionary approach was then tested for supporting objective 3. The experimental procedure integrated the approach with five established solutions referred to as multi-objective evolutionary algorithms (MOEA). Four of these MOEAs proved effective in solving the three real-world problem instances of the LECP. Further testing established NSGA-II as the recommended solution approach to this optimization problem based on the three case studies. NSGA-II was subsequently implemented as an island-model parallel MOEA (pMOEA). The data generated by these final experiments indicated parallelization is a worthwhile strategy for consideration by analysts looking to solve the LECP.

10.2 Evolutionary Methods for Effective Lane Optimization

The observations of the three real-world case studies (Shipper A – C), presented in Chapter 7, strongly suggest that the method by which a transport lane is defined is not trivial. As such, the decision space associated with the clustering of nodes for multiple distance categories should be explored to better define shippers' networks. The results also suggested that using a generic evolutionary approach was an effective method by which to do this.

Without an algorithmic approach that considers rates to define the origins and destinations of lanes, analysts looking to optimize transport costs often default to simplistic solutions. Observations of this investigation, however, suggest these approaches are suboptimal. By testing how different levels of regional boundaries fare compared to the performance of algorithms, it was observed that there is much opportunity for improvement of lane origin and destination definitions. Figure 142 shows that even a basic k-means clustering method achieved parity with the use of regional boundaries in terms of convergence to the true optimal Pareto front while attaining an enhanced spread.

The developed evolutionary approach excelled at identifying lane elimination opportunities where multi-carrier rates were similar. Only one of the five tested MOEAs, the PAES algorithm, failed to completely dominate the regional base solutions for all case studies. Figure 148 shows that the developed

evolutionary approach, when integrated with the remaining four MOEAs, generated a significantly superior diversity of alternative clusters by which to aggregate lanes across all test cases. Notably, it shows that effective lane elimination can be achieved by clustering origins and destinations with negligible impact on rates.

Furthermore, results showed that the suggested generic evolutionary approach with MOEA can automate the otherwise onerous exercise of manually identifying lanes that can be eliminated without significantly impacting rates. The empirical evidence also shows that by defining the LECP and developing an effective evolutionary approach, this research addressed Caplice's (2021) call for more in-depth analyses to better segment networks for transport contracting.

Chapter 8 describes a more detailed analysis of the four validated MOEAs. Table 20 and 21 specifically show that NSGA-II was, overall, the best-performing of the solution approaches. Table 23 and 24 summarize the significant improvement that parallelization introduced to the NSGA-II approach to the LECP. This research thereby provided additional evidential support of NSGA-II as an MOEA of choice, as well as parallel processing for solving combinatorial multi-objective optimization problems.

10.3 Procurement's Requirement to Minimize Lanes

Although the transport industry, in general, should consider the research observations when defining lanes, particular focus should also be placed on the contribution to lane count optimization for procurement event design. Collins and Quinlan (2010) found that bundling proximal lanes can achieve cost savings due to the reduced probability of reverting to ad hoc rates, and this process can be optimized using the suggested evolutionary approach. Not only may over- or under-simplification of a shipper's network significantly impact the outcome of a procurement event, but other management activities can also be affected by the rate card inherited from the exercise. For example, maintaining a multi-carrier price dataset often complicates ongoing modeling of freight rate costs (Seiler, 2012). Our observations show that significant simplifications of these datasets are achievable with minimal loss of pertinent pricing details. Chapter 9.1 addresses how the evolutionary approach to the LECP can be implemented in a transport procurement event setting.

These observations should be considered especially by those inclined to solve primary distribution management problems by leveraging functionality of incumbent procure-to-pay systems. While these systems support strong general transactional processes, they typically do not allow for custom lane definitions and transport rate design. In this case, a common solution is the over-simplified view of a truckload as a procured item maintained on a two-dimensional material master, which is often the genesis of rate proliferation. Implementing a fit-for-purpose TMS can overcome this issue and sustainably support lane optimization.

Collins and Quinlan (2010) added that shippers seek representations with fewer lanes to reduce rate management costs. These include configuring TMSs for rate-based carrier allocation (Caplice, 2021). Chapter 9.2 describes specifically how the LECP can be solved as part of strategy formulation such that a TMS can be configured to sustainably support sustainable optimal carrier allocation.

Acocella and Caplice (2022) showed that highly specific routes are more likely to become undesirable ghost lanes. Bandaru and Dolci's (2020) description of the long-tail of transport routes provided a further example of a management issue arising from highly granular network representations. The incorporation of an LECP solving exercise, discussed in Chapter 9.3, would mitigate these issues while retaining important pricing resolution.

Network simplification also presents optimization benefits, as shown by Kuyzu (2007) and Ergun et al. (2007), while Basu et al. (2010) recognized that, for this type of optimization, the magnitude of difficulty decreases with lane count. It follows that MOEAs have the potential to greatly enhance such optimization exercises by pre-processing the input data to provide a better fit to the model.

10.4 Areas for future research

The multi-objective aspect of the evolutionary approach presented in this thesis lends itself to the incorporation of nuanced desired outcomes. These could easily be modelled and added to the formulation and approach as alternative or additional fitness functions. For example, an existing rate card may include numerous carrier prices for a single lane, many of which are speculative or have exceptionally low associated truck availability. Alternatives to the equal treatment of these rates by the cost penalty fitness function, as implemented in this research, may be preferred in such cases. Analysts may also wish to assign a weighting to rate card records according to the expected number of truckloads to discourage rate hedging on routes with high anticipated total costs when network simplification can be achieved by combining lower-volume lanes elsewhere in the network. Furthermore, a shipper may be interested in minimizing the disparity of the cost impact between lanes. The evolutionary approach presented in this paper can be easily tested using these or other functions.

The cost penalty function, used as one of only two fitness functions, had a significant bearing on the behavior of the algorithms and results. This function assumed that, when compelled to submit a rate for a lane with a large origin and/or destination, a carrier would be risk averse and quote the highest applicable rate. However, the function could be adapted to reflect a more realistic value. For example, a weighted equation between the highest and lowest rates could be used, with weighting factors based on the analyst's assessment and likelihood from past experiences.

The evolutionary approach used two distance categories, namely local and long-distance, demarcated by a single distance bracket to allow different definitions for lanes of varying lengths. Further research could investigate whether this is, indeed, the correct approach, especially when applied to different regions or modes of transport. The relationship between the number of distance brackets, the complexity of the evolutionary approach, the computational cost, and the performance of MOEAs would be interesting to establish.

This research integrated the single evolutionary approach with five of the six MOEAs mentioned by Coello Coello *et al*. (2007). The sixth solution approach, Micro-GA, was excluded due to the number of unique parameters that would have required tuning. However, this MOEA has potential to outperform the other approaches. The use of Micro-GA to solve the LECP is therefore a topic for further investigation.

The exploratory component of the empirical testing of the five MOEAs involved testing each solution approach with only one of the three available shipper rates cards. Further testing to include all problem instances in the exploration of the behaviour of these algorithms may be conducted to increase confidence in the findings and possibly detect additional opportunities for adapting the procedures to better fit the problem domain.

Additional MOEA enhancement strategies that lend themselves to the LECP could also be implemented and tested. These include coevolution techniques, such as predator-prey, parasitism and symbiotic relationships. The distinct cluster sets of an LECP solution, each representing a fixed-length building block of the solution's corresponding chromosomal representation, make cooperative coevolution an especially promising strategy. This technique could be implemented such that each cluster set evolves independently and simultaneously on dedicated processors. Cluster sets would then be sampled randomly from the subpopulations and combined to form full chromosomes for fitness evaluation (Coello Coello *et al*., 2007). Cooperative coevolution therefore represents another variation of parallelization and has potential to outperform the island pMOEA model tested for this thesis.

The experimental procedure used in the three case studies focused on optimizing the results from each shipper in isolation. However, as they operate in the same region using equivalent equipment types for transportation, it would be possible investigate solving a combined case study where all shipper rate card data are considered. The results of such an analysis could provide a more general view of suggested lane definitions for a southern African shipper and further collaboration at a regional level.

The LECP was formulated under the assumption that only flat transport rates would be used as the cost input. Flat rates, however, are one of many pricing mechanisms used for defining transport prices. A

minor adaptation to the LECP's cost penalty objective function could be implemented to test whether lane elimination is more effective when using a distance-based rate basis. This would quantify the mitigating effect of applying a single cost per kilometre or mile for transport to or from clusters and thereby demonstrate that much larger clusters and significantly fewer lanes could be implemented for a given degree of cost impact. The simplification gains achieved by transitioning to distance-based costing could therefore offset or possibly exceed the additional complexity of defining and operationalizing precise distance definitions for each node-to-node route of the shipper's network for planning and financial processes.

Although the effectiveness of the evolutionary approach to solving the LECP was demonstrated by the research, the benefit of clustering solutions generated by the MOEAs for subsequent transport optimization activities can be tested. For example, the use of nondominated LECP solutions could be used to define the lanes for a number of winner determination problem (WDP) variations and problem instances. As the concept of a lane already exists in most WDP formulations, this would involve only a pre-processing step of clustering nodes into the LECP solution structures and aggregating forecasted volumes and rates accordingly prior to optimization. These new lane definitions should allow the optimization method to reduce cost further while satisfying constraints such as the minimum number of carriers to allocate to a lane and the minimum loads to award to an active carrier on a lane.

Nondominated LECP solutions should also be tested as features for machine learning (ML) approaches to transport cost analyses. For example, the local and long-distance origin and destination clusters generated by one of the tested MOEAs may be favoured by ML algorithms that attempt to identify transport cost levers. This could greatly enhance the efficacy of ML applications to transport costing and budgeting.

References

1. Acocella, A., and Caplice, C. (2021). Research on truckload transportation procurement: A review, framework, and future research agenda. *Journal of Business Logistics*, 44:228-256. doi: 10.1111/jbl.12333.
2. Acocella, A., and Caplice, C. (2022). The hidden costs of not-so-friendly ghost lanes. *MIT Working Paper*, No. 2022-mitscale-ctl-03, Cambridge, MA.
3. Acocella, A., Caplice, C. and Sheffi, Y. (2020). Elephants or Goldish?: An Empirical Analysis of Carrier Reciprocity in Dynamic Freight Markets. *MIT Working Paper*, No. 2020-mitscale=ctl-01, Cambridge, MA. doi: 10.1016/j.tre.2020.102073.
4. Adjavor, H., Al-Hihi, S, and Wang, Y. (2020). A Classification of Winners and Losers of Shippers' Weekly Transportation Capacity Procurement. *IISE Annual Conference. Proceedings; Norcross*, 2020:346-351.
5. Adra, S.F., Griffin, I., and Fleming, P.J. (2005). Hybrid Multi-objective Genetic Algorithm with a New Adaptive Local Search Process. *GECCO'05*, Washington DC, USA, 25-29 June 2005. doi: 10.1145/1068009.1068180.
6. Aemireddy, N.R., and Yuan, X. (2019). *Root cause analysis of unplanned procurement on truckload transportation costs*. M.S. thesis, Massachusetts Institute of Technology, Massachusetts, U.S.A.
7. Affenzeller, M., Wagner, S., and Winkler, S. (2008). *Evolutionary Systems Identification: New Algorithmic Concepts and Applications*. In *Advances in Evolutionary Algorithms*, 1st edition. Vienna: I-Tech Education and Publishing. ISBN: 978-953-7619-11-4.
8. Aguiree, H.E., and Tanaka, K. (2009). Adaptive ε-Ranking on many-objective problems. *Evolutionary Intelligence*, 2:183-206. doi: 10.1007/s12065-009-0031-2.
9. Albadr, M.A., Tiun, S., Ayob, M., and AL-Dhief, F. (2020). Genetic Algorithm Based on Natural Selection Theory for Optimisation Problems. *Symmetry*, 12(11):1-31. doi: 10.3390/sym12111758.
10. Aldstadt, J. (2010). *Spatial clustering*. In *Handbook of Applied Spatial Analysis*, 1st edition. Berlin: Springer. ISBN: 978-3-642-03647-7.
11. Almannaa, M. H., Elhenawy, M., and Rakha, H. A. (2018). A Novel Supervised Clustering Algorithm for Transportation System Applications. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):222-232. doi: 10.1109/TITS.2018.2890588.
12. Almufti, M., Ahmad Shaban, A., Ismael Ali, R. and Dela Fuente, J. (2023). Overview of Metaheuristic Algorithms. *Polaris Global Journal of Scholarly Research and Trends*, 2(2):10-32. doi: 10.58429/pgjsrt.v2n2a144.
13. Andersson, J., and Krus, P. (2004). Multi-objective Optimisation of Mixed Variable Design Problems, in Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., and Corne, D. (eds) *First*

International Conference on Evolutionary Multi-Criterion Optimisation. Springer, Berlin, Heidelberg, pp.624-638.

14. Ankerst, M., Breunig, M.M., Kreigel, H-P., and Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. *Computers & Operations Research*, 34:1551-1560. doi: 10.1145/304182.304187.
15. Antonio, L.M., and Coello Coello, C.A. (2018). Coevolutionary Multi-objective Evolutionary Algorithms: Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation,* 22(6):851-865. doi: 10.1109/TEVC.2017.2767023
16. Antonio, L.M., and Coello Coello, C.A. (2018). Indicator-based cooperative coevolution for multi-objective optimisation. *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canda, 24-29 July 2016.
17. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243-256. doi: 10.1016/j.patcog.2012.07.021.
18. Arunachalam, D., Kumar, N., and Kawalek, J. P. (2017). Understanding big data analytics capabilities in supply chain management: Unravelling the issues, challenges and implications for practice. *ACM SIGMOD Record*, 28(2):49-60. doi: 10.1016/j.tre.2017.04.001.
19. Arunapuram, S., Mathur, M., and Solow, D. (2003). Vehicle Routing and Scheduling with Full Truckloads. *Transportation Science INFORMS*, 37(2):170-182. doi: 10.1287/trsc.37.2.170.15248.
20. Assunção, R. M., Neves, M. C., Cămara, G., and da Costa Freitas, C. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797-811. doi: 10.1080/13658810600665111.
21. Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., and Salomon, L. (2020). Performance Indicators in Multi-objective Optimisation. *European Journal of Operational Research*, 292:397-422. doi: 10.1016/j.ejor.2020.11.016.
22. Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. *Foundations of Genetic Algorithms (FOGA 2009)*, Orlando, Florida, USA, 9-11 January 2009.
23. Balachandran, P., Xue, D., Theiler, J., Hogden, J., and Lookman, T. (2016). Adaptive Strategies for Materials Design using Uncertainties. *Scientific Reports*, 6(1):1-9. doi: 10.1038/srep19660.
24. Bakhsh, A., (2013). *A Posteriori And Interactive Approaches For Decision-making A Posteriori And Interactive Approaches For Decision-making With Multiple Stochastic Objectives With Multiple Stochastic Objectives*. Phd thesis, University of the Central Florida, Orlando.
25. Bandaru, V. R., and Dolci, E. (2020). *Going awry, understanding transportation budget failures*. M.S. thesis, Massachusetts Institute of Technology, Massachusetts, U.S.A.

26. Basu, R.J., Subramanian, N., and Jie, F. (2010). Full Truck Load (TL) Transportation Service Procurement – A Literature Review Approach. *ORSI 2010*, Madurai, India, 15-17 December 2010.
27. Bandyopadhyay, S., Maulik, U., and Mukhopadhyay, A. (2007). Multi-objective Genetic Clustering for Pixel Classification in Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.,* 45(5):1506–1511. doi: 10.1109/TGRS.2007.892604.
28. Bandyopadhyay, S., and Pal, S.K. (1998). Incorporating Chromosome Differentiation in Genetic Algorithms. *INFORMATION SCIENCES*, 104:293-319. doi: 10.1016/S0167-8655(97)00005-6.
29. Basu, R.J., Subramanian, N., Gunasekaran, A., and Palaniappan, P.L.K. (2016). Influences of non-price and environmental sustainability factors on truckload procurement processes. *Annals of Operations Research*, 250(2):363-388. doi: 10.1007/s10479-016-2170-z.
30. Bérubé, J-F., Gendreau, and M., Potvin, J-Y. (2007). An Exact ε-constraint Method for Bi-objective Combinatorial Optimisation Problems – Application to the Traveling Salesman Problem with Profits. *European Journal of Operational Research,* 194(1):39-50. doi: 10.1016/j.ejor.2007.12.014.
31. Bilmes, J., Vahdat, A., Hsu, W., and Im. E.-J. (19997). *Empirical observations of probabilistic heuristics for the clustering problem*. Technical Report TR-97-018, International Computer Science Institute, University of California, Berkeley, CA.
32. Bishop, C.M. (1994). Neural networks and their applications. *Review of Scientific Instruments,* 65:1803-1832. doi: 10.1063/1.1144830.
33. Borndörfer, R., Grötschel, M., and Löbel, A. (1998). Optimisation of Transportation systems. *ACTA Forum Engelburg*, 98(09).
34. Bosman, P.A.N., and de Jong, E.D. (2005). Exploiting gradient information in numerical multi--objective evolutionary optimisation. *Genetic and Evolutionary Computation Conference, GECCO 2005*, Washington DC, USA, 25-29 June 2005.
35. Bourke, P. (1997). Determining if a point lies on the interior of a polygon. *Data structures and algorithms.* Available at: https://www.eecs.umich.edu/courses/eecs380/HANDOUTS/PROJ2/InsidePoly.html. Accessed 23 December 2022. Last updated 2001.
36. Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004). Finding Knees in Multi-objective Optimisation, in Yao, x., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., and Schwefe, H-P. (eds) *Parallel Problem Solving from Nature - PPSN VIII.* Springer Berlin, Heidelberg, pp.722-731. ISBN: 978-3-540-23092-2.
37. Branke, J., Schmeck, H., Deb K., and Reddy. M. (2005). Parallelizing multi-objective evolutionary algorithms: cone separation. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, Portland, OR, USA, 19-23 June 2004.

38. Braun, M.A., (2018). *Scalarized Preferences in Multi-objective Optimisation*. Phd thesis, Karlsruhe Institute of Technology, Karlsruhe.
39. Buczkowska, S., Coulombel, N., and de Lapparent, M. (2017). Euclidean versus network distance in business location: A probabilistic mixture of hurdle-Poisson models. *Annals of Regional Science*, 2016. doi: 10.13140/RG.2.1.4875.6241.
40. Bührmann, J.H., (2016). *The Effects of Clustering on the Medium and Large-Scale Capacitated Location-Routing Problem*. Phd thesis, University of the Witwatersrand, Johannesburg.
41. C.H. Robinson (2021) Post-18 education and funding review: Interim conclusion (CP. 358). Available at: https://www.gov.uk/government/publications/post-18-education-and-funding-review-interim-conclusion (Accessed: 22 January 2021).
42. Caponio, A., and Neri, F. (2009). Integrating Cross-Dominance Adaptation in Multi-Objective Memetic Algorithms., in Goh, CK., Ong, YS., and Tan, K.C. (eds) *Multi-Objective Memetic Algorithms. Studies in Computational Intelligence*, vol 171. Springer, Berlin, Heidelberg, pp. 325–351.
43. Caplice, C., (2007). Electronic Markets for Truckload Transportation. *Production and Operations Management, [e-journal]* 16(4):423–436. doi: 10.1111/j.1937-5956.2007.tb00270.x.
44. Caplice, C. (2021). Reducing uncertainty in freight transportation procurement. *Journal of Supply Chain Management, Logistics and Procurement*, 4(1):137-155.
45. Caplice, C., and Sheffi, Y. (2003). Optimisation-based Procurement for Transportation Services. *Journal of Business Logistics*, 24(2):109-128. doi: 10.1002/j.2158-1592.2003.tb00048.x.
46. Carmia, M., and Dell'Olmo, P. (2008). Increasing Capacity, Service Level and Safety with Optimisation Algorithms. *Multi-objective Management in Freight Logistics*, ed. 1. London: Springer. ISBN: 978-1-84800-381-1.
47. Carr, J. (2014). *An Introduction to Genetic Algorithms. Whitman University*, Available at: www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf. (Accessed: 29 May 2023)
48. Catini, R., Karamshuk, D., Penner, O., and Riccaboni, M. (2015). Identifying geographic clusters: A network analytic approach. *Research Policy,* 44(9):1749-1762. doi: 10.1016/j.respol.2015.01.011.
49. Chand, S., and Wagner, M. (2015). Evolutionary Many-Objective Optimisation: A Quick-Start Guide. *Surveys in Operations Research and Management Science*, 20(2):35-42. doi: 10.1016/j.sorms.2015.08.001.
50. Chen, J., and Liu, Y. (2022). Neural Optimisation Machine: A Neural Network Approach for Optimisation. https://arxiv.org/abs/2208.03897. Cited 30 May 2023. Last updated 8 August 2022.

51. Chen, E., and Wang, F. (2005). Dynamic clustering using multi-objective evolutionary algorithm. *Proceedings of the 2005 international conference on Computational Intelligence and Security - Volume Part I (CIS'05)*, Xi'an, China, 15-19 December 2005.
52. Cheng, R., Yaochu, J., Olhofer, M., and Sendhoff, B. (2017). Test Problems for Large-Scale Multi-objective and Many-Objective Optimisation. *IEEE Trans. Cybern*, 47(12):4108-4121. doi: 10.1109/TCYB.2016.2600577.
53. Cheong, S-H., and Si, Y-W. (2020). CWBound: boundary node detection algorithm for complex non-convex mobile ad hoc networks. *The Journal of Supercomputing*, 74:5558-5577. doi: 10.1007/s11227-018-2494-3.
54. Chircop, K., and Zammit-Mangion, D. (2013). On Epsilon-Constraint Based Methods for the Generation of Pareto Frontiers. *Journal of Mechanics Engineering and Automation*, 3:279-289.
55. Coello Coello, C.A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering,* 191(11-12): 1245-1287. doi: 10.1016/S0045-7825(01)00323-1.
56. Coello Coello, C.A., González Brambila, S., Figueroa Gamboa, J., Castillo Tapia, M.G., and Hernández Gómez, R.H. (2020). Evolutionary multi-objective optimisation: open research areas and some challenges lying ahead. *Complex and Intelligent Systems,* 6:221-236. doi: 10.1007/s40747-019-0113-4.
57. Coello Coello, C.A., Lamont, G., and Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edition. New York: Springer. ISBN 978-0-387-33254-3.
58. Coello Coello, C.A., and Lamont, G.B. (2004). An introduction to multi-objective evolutionary algorithms and their applications, in Coello Coello, C.A., and Lamont, G.B. (eds) *Applications of Multi-Objective Evolutionary Algorithms.* World Scientific Publishing Company, pp.1-28. ISBN: 978-9812561060.
59. Coello Coello, C.A., and Pulido, G.T. (2001). A micro-genetic algorithm for multi-objective optimisation. *Evolutionary Multi-Criterion Optimisation*, Zurich, Switzerland, March 7-9 2001.
60. Coello Coello, C.A., and Pulido, G.T. (2001). Multi-objective Optimisation using a Micro-Genetic Algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, San Francisco, California, 7-11 July 2001.
61. Coello Coello, C.A., and Pulido, G.T. (2003). The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multi-objective Optimisation. *Evolutionary Multi-Criterion Optimisation. Second International Conference, EMO 2003*, Faro, Portugal, 8-11 April 2003.
62. Coello Coello, C.A., and Pulido, G.T. (2005). Multi-objective structural optimisation using a microgenetic algorithm. *Structural and Multidisciplinary Optimisation,* 30(5):388-403. doi: 10.1007/s00158-005-0527-z.

63. Coello Coello, C.A., and Reyes Sierra, M. (2003). A coevolutionary multi-objective evolutionary algorithm. *Evolutionary Computation, 2003. CEC '03*, Canberra, Australia, 8-12 December 2003.
64. Cohen, J. (1979). Non-Deterministic Algorithms. *Computing Surveys*, 11(2):79-94.
65. Cohon, J.L., and Mark, D.H. (1975). A review and evaluation of multi-objective programming techniques. *Water Resources Research,* 11(2):208-220.
66. Cole, R.M., (1998). *Clustering with genetic algorithms*. M.S. thesis, University of Western Australia, Perth, W.A.
67. Collins, J.M., and Quinlan, R.R. (2010). *The Impact of Bidding Aggregation Levels on Truckload Rates*. Master of Engineering Logistics thesis, Massachusetts Institute of Technology, Massachusetts.
68. Corne, D., Knowles, J.D., and Oates, M.J. (2000). The Pareto Envelop-based Selection Algorithm for Multi-Objective Optimisation. *Lecture Notes in Computer Science,* 1917:839-848. doi: 10.1007/3-540-45356-3_82.
69. Coyle, J.J., Bardi, E.J., and Langley, C.J. (2003). *Management of Business Logistics: A Supply Chain Perspective*, 7th edition. Mason, OH: South-Western College Publishing. ISBN 978-0324007510.
70. Cuevas, E., Echavarría, A., and Ramírez-Ortegón, M.A. (2014). An optimisation algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied Intelligence*, 40(2):256-272. doi: 10.1007/s10489-013-0458-0.
71. Cui, Y., Geng, Z., Zhu, Q., and Han, Y. (2017). Multi-objective optimisation methods and application in energy saving. *Energy,* 125:681-704. doi: 10.1016/j.energy.2017.02.174.
72. Curtin, K.M. (2005). 'Operations Research', in Kempf-Leonard, K. (ed.) *Encyclopedia of Social Measurement.* Elsevier, pp. 925-931.
73. Custódio, A.L., Aguilar Madeira, J.F., and Emmerich, M. (2012). Recent Developments in Derivative-Free Multi-objective Optimisation. *Computational Technology Reviews*, 5(1):1-31. doi: 10.4203/ctr.5.1.
74. Cvetkovic, D., and Parmee, I.C. (2002). "Preferences and their application in evolutionary multi-objective optimisation. *IEEE Transactions on Evolutionary Computation,* 6(1):42-57. doi: 10.1016/bs.adcom.2015.03.001.
75. Daglia, Z.C., and Mirchev, M. (2020). Chapter 15 - When Evolutionary Computing Meets Astro- and Geoinformatics, in Škoda, P, and Adam, F. (eds) *Knowledge Discovery in Big Data from Astronomy and Earth Observation.* Elsevier, pp.283-306. ISBN: 9780128191545.
76. Das, I., and Dennis, J. (1997). A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimisation Problems. *Structural Optimisation*, 14:63-69. doi: 10.1007/BF01197559.

77. De Buck, V., López, C.A.M., Nimmegeers, P., Hashem, I., Van Impe, J. (2019). 'Multi-objective optimisation of chemical processes via improved genetic algorithms: A novel trade-off and termination criterion', in Kiss, A.A., Zondervan, E., Lakerveld, R., and Özkan, L. (eds) *Computer Aided Chemical Engineering.* Elsevier, pp.613-618. ISBN: 9780128186343.
78. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering,* 186(2):311-338. doi: 10.1016/S0045-7825(99)00389-8.
79. Deb, K. (2002). A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation,* 6(2):182-197. doi: 10.1109/4235.996017.
80. Deb, K. (2003). 'Multi-objective evolutionary algorithms: Introducing bias among Pareto-optimal solutions', in Ghosh, A., and Tsutsui, S. (eds) *Advances in evolutionary computing: theory and applications.* Berlin, Heidelberg: Spring Berlin Heidelberg, pp. 263-292.
81. Deb, K. (2011). 'Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction', in Wang, L., Ng, A., Deb, K. (eds) *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing.* London: Springer, ISBN: 978-0-85729-617-7.
82. Deb, K., Chaudhuri, S., and Miettinen, K. (2006). Towards estimating nadir objective vector using evolutionary approaches. *GECCO 2006 - Genetic and Evolutionary Computation Conference*, Seattle, Washington, USA, 8-12 July 2006.
83. Deb, K., and Gupta, H. (2006). Introducing robustness in multi-objective optimisation. *Evolutionary Computation,* 14(4):463-494. doi: 10.1162/evco.2006.14.4.463.
84. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., and Chaudhuri, S. (2006). Reference Point Based Multi-Objective Optimisation Using Evolutionary Algorithms. *International Journal of Computational Intelligence Research*, 2(3):273-286.
85. Della Cioppa, A., and Marcelli, A., and Napoli, P. (2011). Speciation in evolutionary algorithms: Adaptive Species Discovery. *GECCO 2011 - Genetic and Evolutionary Computation Conference*, Dublin, Ireland, 12-16 July 2011.
86. Dellbrügge, M., Brilka, T., Kreuz, F. and Clausen, U. (2022). Auction Design in Strategic Freight Procurement. *Proceedings of the Hamburg International Conference of Logistics*, Hamburg, Germany, 20-23 September 2022.
87. Demir, G.N., Uyar, A.S., and Gündüz-Őğüdücü, S. (2010). Multi-objective evolutionary clustering of Web user sessions: a case study in Web page recommendation. *Soft Computing*, 14:579-597. doi: 10.1007/s00500-009-0428-y.
88. Dimária Silva, E.M., Mario Fernandes, D.C., and Thomaz, J.C. (2014). Logistics Service Providers In Brazil: a study of geographic clustering from the perspective of logistics chain integration. *JOSCM : Journal of Operations and Supply Chain Management*, 7(1):47-67. doi: 10.12660/joscmv7n1p47-67.

89. Dorronsoro, B., Danoy, G., Nebro, A.J., and Bouvry, P. (2013). Achieving super-linear performance in parallel multi-objective evolutionary algorithms by means of cooperative coevolution. *Computers & Operations Research*, 40(6):1552-1563. doi: 10.1016/j.cor.2011.11.014.
90. D'Souza, R.G.L., Sekaran, K.C., and Kandasamy, A. (2010). Improved NSGA-II Based on a Novel Ranking Scheme. *Journal of Computing*, 2(1):91-95. doi: 10.48550/arXiv.1002.4005.
91. Du Perron, S.F. (2008). *Applying a Virtual Organisation on IKEA´s Transport Organisation*. Phd thesis, Delft University of Technology, Delft.
92. Durillo, J.J., Alba, E., Zhang, Y., and Harman, M. (2011). A study of the bi-objective next release problem. *Empirical Software Engineering,* 16(1):29-60. doi: 10.1007/s10664-010-9147-3.
93. Duque, J. C., Church, R. L., and Middleton, R. S. (2011). The p-regions problem. *Geographical Analysis*, 43(1):104-126. doi: 10.1111/j.1538-4632.2010.00810.x.
94. Dutta, D., Sil, J., and Dutta, P. (2019). Automatic clustering by multi-objective genetic algorithm with numeric and categorical features. *Expert Systems with Applications*, 137:357-379. doi: 10.1016/j.eswa.2019.06.056.
95. Eshelman, L.J. (1997). 'Genetic Algorithms", in De Jong, K., Fogel, L., and Schwefel, H-P. (ed.) *The Handbook of Evolutionary Computation.* Taylor & Francis, pp. B1.2:1-B1.2:11. ISBN: 0750308958.
96. Emmerich, M.T.M., and Deutz, A.H. (2018). A tutorial on multi-objective optimisation: fundamentals and evolutionary methods. *Natural Computing*, 17:585-609. 10.1007/s11047-018-9685-y.
97. Ergun, O., Kuyzu, G., and Savelsbergh, M. (2007). *Shipper collaboration*. *Computers & Operations Research*, 34:1551-1560. doi: 10.1016/j.cor.2005.07.026.
98. Erickson, M., Mayer, A., and Horn, J. (2001). 'The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems', in Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., and Corne, D. (eds). *Evolutionary Multi-Criterion Optimisation. EMO 2001. Lecture Notes in Computer Science*, vol 1993. Springer, Berlin, Heidelberg. ISBN: 978-3-540-41745-3.
99. Ester, M., Kiregel, H-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portal, Oregon, USA, 2-4 August 1996.
100. Everitt, B., Landau, S. L., Leese, M., and Stahl, D. (2011). *Cluster analysis*, 5th edition, West Sussex: John Wiley & Sons, Ltd. ISBN: 978-0-470-74991-3.

101. Faceli, K., de Carvalho, A.C.P.L.F., and de Souto, M.C.P. (2007). Multi-objective clustering ensemble. *International Journal of Hybrid Intelligent Systems*, 4(3):145-156. doi: 10.1109/HIS.2006.264934.

102. Falcón-Cardona, J.G., Hernández Gómez, R., Coello Coello, C.A., and Castillo Tapia, M.G. (2021). Parallel Multi-Objective Evolutionary Algorithms: A Comprehensive Survey. *Swarm and Evolutionary Computation*, 67. doi: 10.1016/j.swevo.2021.100960.

103. Falkenauer, E. (1998). *Genetic Algorithms and Grouping Problems*, 1st edition. Wiley. ISBN: 978-0471971504.

104. Fang, M., Tang, L., Kan, Z. Yang, X., Pei, T., Li, Q., and Li, C. (2021). An adaptive Origin-Destination flows cluster-detecting method to identify urban mobility trends. https://arxiv.org/abs/2106.05436. Cited 3 June 2022. Last updated 10 June 2021.

105. Fazackerley, S., Paeth, A., and Lawrence, R. (2009). Cluster head selection using RF signal strength. *Canadian Conference on Electrical and Computer Engineering*, St. John's, Newfoundland, Canada, 3-6 May 2009.

106. Filho, J.L.R., Treveleaven, C., and Alippi, C. (1994). Genetic algorithm programming environments. *IEEE Comput*, 27:28-43. doi: 10.1109/2.294850.

107. Fleming, P., Purshouse, R.C., and Lygoe, R.J. (2002). May-Objective Optimisation: An Engineering Design Perspective. *Evolutionary Multi-Criterion Optimisation: Third International Conference*, Guanjuato, Mexico, 9-11 March 2005.

108. Fonseca, C., and Fleming, P. (1993). Genetic Algorithms for Multi-objective Optimisation: Formulation Discussion and Generalization. *The Fifth International Conference on Genetic Algorithms*, San Francisco, California, USA, 1 June 1993.

109. Fonseca, C., and Fleming, P. (1995). An Overview of Evolutionary Algorithms in Multi-objective Optimisation. *Evolutionary Computation*, 3(1): 1-16. doi: 10.1162/evco.1995.3.1.1

110. Garza-Fabre, M., Toscano-Pulido, G., Coello Coello, C.A., and Rodriguez-Tello, E. (2011). Effective ranking+ speciation= many-objective optimisation. *2011 IEEE Congress of Evolutionary Computation (CEC).* Carlton, New Orleans, 5-8 June 2011.

111. Gero, J. and Kazakov, V.A. (1995). Evolving building blocks for genetic algorithms using genetic engineering. *Journal of Clinical Neuroscience - J CLIN NEUROSCI*, 34(10). doi: 10.1109/ICEC.1995.489170.

112. Ghoreishi, S., Clausen, A., and Joergensen, B. (2017). Termination Criteria in Evolutionary Algorithms: A Survey. *9th International Joint Conference on Computational Intelligence*, Funchal, Madeira, Portugal, 1-3 November 2017.

113. Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning*, 1st edition. Boston: Addison-Wesley Professional. ISBN 978-0201157673.

114. Gong, Y.J., Chen, W.N., Zhan, Z.H., Zhang, J., Li, Y., Zhang, Q., and Li, J.J. (2015). Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing,* 34:286-300. doi: 10.1016/j.asoc.2015.04.061.

115. Gonzalez-Feliu, J., Palacios-Argüello, L., and Suarez-Nũnez, C. (2020). Links between Freight Trip Generation Rates, Accessibility and Socio-Demographic Variables in Urban Zones. *Archives of Transport*, 53(1):7-20. doi: 10.5604/01.3001.0014.1738.

116. Greco, S., Matarazzo, B., and Slowiński, R. (2010). Interactive Evolutionary Multi-objective Optimisation using Dominance-based Rough Set Approach. *WCCI 2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, 18-23 July 2010.

117. Griffis, S.E., and Goldsby, T.J. (2007). Transportation management systems: an exploration of progress and future prospects. *Journal of Transportation Management*, 18(1):14.

118. Groenendijk, R. (2016). *Love me tender? Successful transport tender management*, Available at: https:// www.aeb.com/blog/2016/03/04/successful-transport-tender-management/index.php. Accessed 3 November 2017. Last updated 2017.

119. Guha, S., Rastogi, R., and Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. *ACM SIGMOD Record*, 27(2):73-84. doi: 10.1145/276304.276312.

120. Gunantara, N. (2018). A review of multi-objective optimisation: Methods and its applications. *Cogent Engineering*, 5:1-16. doi: 10.1080/23311916.2018.1502242.

121. Gurney, K. (1997). *An introduction to neural networks*, 1st edition, London: UCL Press. ISBN: 1-85728-503-4.

122. Grazia Speranza, M. (2018). Trends in transportation and logistics. *European Journal of Operational Research*, 264:830-836. doi: 10.1016/j.ejor.2016.08.032.

123. Haimes, Y.Y, and Hall, W.A. (1974). Multi-objectives in water resources systems analysis: The surrogate trade-off method. *Water Resources Research*, 10(4):615-624. doi: 10.1029/WR010I004P00615.

124. Hakimi-Asiabar, M., Ghodsypour, S.H., and Kerachian, R. (2009). Multi-objective genetic local search algorithm using Kohonen's neural map. *Computers & Industrial Engineering*, 56:1566-1576. doi: 10.1016/j.cie.2008.10.010.

125. Handl, J., and Knowles, J.D. (2004). Evolutionary Multi-objective Clustering. *Proc. 8th Int. Conf. Parallel Problem Solving in Nature*, Birmingham, UK, 18-22 September 2004.

126. Handl, J., and Knowles, J.D. (2006). Multi-Objective Clustering and Cluster Validation. *Multi-Objective Machine Learning. Studies in Computational Intelligence*, vol 16. Heidelberg: Springer. ISBN: 978-3-540-30676-4.

127. Hassanat, A.B., Prasath, V.B.S., Abbadi, M.A., Abu-Qdari, S.A., and Faris, H. (2018). An Improved Genetic Algorithm with a New Initialization Mechanism Based on Regression Techniques. *Information*, 9(7):167. doi: 10.3390/info9070167.

128. He, B., Zhang, Y., Chen, Y., and Gu, Z. (2018). A Simple Line Clustering Method for Spatial Analysis with Origin-Destination Data and Its Application to Bike-Sharing Movement Data. *International Journal of Geo-Information*, 7(6). doi: 10.3390/ijgi7060203.

129. Hedvall, K., Dubois, A., and Lind, F. (2017). Variety in freight transport service procurement approaches. *Transportation Research Procedia*. 25:806-823. doi: 10.1016/j.trpro.2017.05.459.

130. Hernández Aguirre, A., Botello Rionda, S., Lizárraga Lizárraga, G., and Coello Coello, C.A. (2003). 'S-PAES: A Constraint-Handling Technique Based on Multi-objective Optimisation Concepts', in Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., and Deb, K. (eds). *Evolutionary Multi-Criterion Optimisation. EMO 2003. Lecture Notes in Computer Science*, vol 2632. Springer, Berlin, Heidelberg. ISBN: 978-3-540-01869-8.

131. Hindsberger, M., and Vidal, R. (2000). Tabu search-a guided tour. *Control and Cybernetics*, 29(3):631-651.

132. Hinneburg, A., and Keim, D.A. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *Knowledge Discovery and Data Mining*, 98:58-65.

133. Holland, J. (2000). Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation*, 8(4):373-391. doi: 10.1162/106365600568220.

134. Holmes, T.J., and Lee, S. (2010). Cities as Six-By-Six-Mile Squares: Zipf's Law?. *Agglomeration Economics*, 105-131.

135. Holter, A.R., Grant, D.B., Ritchie, J., and Shaw, N. (2016). A framework for purchasing transport services in small and medium size enterprizes. *International Journal of Physical Distribution & Logistics Management*, 38(1):21-38. doi: 10.1108/09600030810857193.

136. Hoos, H.H., and Stützle, T. (2013). Evaluating Las Vegas Algorithms - Pitfalls and Remedies. https://arxiv.org/abs/1301.7383. Cited 30 May 2023. Last updated 30 January 2013.

137. Hopfield, J.J., and Tank, D.W. (1985). "Neural" computation of decisions in optimisation problems. *Bioloical Cybernetics*, 52:141-152. doi: 10.1007/0-306-48056-5_15.

138. Horn, J. (1997). 'Multicriterion Decision making', in Bäck, T., Fogel, D., and Michalewicz, Z. (eds) *Handbook of Evolutionary Computation.* IOP Publishing Ltd. and Oxford University Press, pp. F1.9:1-F1.9:15.

139. Horn, J., Nafpliotis, N., and Goldberg, D.J. (1994). *A Niched Pareto Genetic Algorithm for Multi-objective Optimisation*. The First IEEE Conference on Evolutionary Computation, Orlanda, Florida, USA, 27-29 June 1994.

140. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A.. and Ponce Leon F. de Carvalho, A.C. (2009). A Survey of Evolutionary Algorithms for Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2): 133-155. doi: 10.1109/TSMCC.2008.2007252.

141. Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., and Prasad, S. (2015). Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems,* 54:240-254. doi: 10.1016/j.compenvurbsys.2015.09.001.

142. Hu, X., Huang, Z., and Wang, Z. (2003). Hybridization of the multi-objective evolutionary algorithms and the gradient based algorithms. *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, Canberra, ACT, Australia, 8-12 December 2003.

143. Huang, W., Zhang, Y., and Li, L. (2019). Survey on multi-objective evolutionary algorithms. In *Journal of physics: Conference series* (Vol. 1288, No. 1, p. 012057). IOP Publishing.

144. Hussain, A., and Kim, H-M. (2020). Goal-Programming-Based Multi-Objective Optimisation in Off-Grid Microgrids. *Sustainability,* 12(19):1-18. doi: 10.3390/su12198119.

145. Ignizio, J.P. (1981). The determination of a subset of efficient solutions via goal programming. *Computers and Operations Research,* 3:9-16.

146. Ingber, L. (1993). Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11):29-57.

147. Iorio, A.W., and Li, X. (2004). 'A Cooperative Coevolutionary Multi-objective Algorithm Using Non-dominated Sorting', in Deb, K. (eds) *enetic and Evolutionary Computation – GECCO 2004. GECCO 2004. Lecture Notes in Computer Science*, vol 3102 Springer, Berlin, Heidelberg, 978-3-540-22344-3.

148. Ishibuchi, H., Yoshida, T., and Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multi-objective permutation flowshop scheduling. *IEEE Trans. Evol. Comput.*, 7:204-223. doi: 10.1109/TEVC.2003.810752.

149. Izzo, D., Ruciński, and Biscani, F. (2012). The Generalized Island Model. *Studies in Computational Intelligence*, 415:151-169. doi: 10.1007/978-3-642-28789-3_7.

150. Jain, A.K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters,* 31(8):651-666. doi: 10.1016/j.patrec.2009.09.011.

151. Jain, A.K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*, 1st edition. New Jersey: Prentice Hall. ISBN: 0-13-022278-X.

152. Jain, A.K., Murty, M.N., and Flynn, P.J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264-323. doi: 10.1145/331499.331504.

153. Jain, B. J., Pohlheim, H., and Wegener, J. (2001). *On Termination Criteria of Evolutionary Algorithms*. GECCO 2001 Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, California, USA, 7-11 July 2001.

154. Jain, S., and Parashar, V. (2022). Comparison of priori and posteriori approach of multi-objective optimisation for WEDM on Ti6Al4V alloy. *Material Research Express,* 9(7):1-19. doi: 10.1088/2053-1591/ac7f83.

155. Jaskiewicz, A. (2001). Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem. *IEEE Transactions on Evolutionary Computation*, 7(2):133-143. doi: 10.1109/TEVC.2003.810759.

156. Jaskiewicz, A. (2003). Comparison of Local Search-Based Metaheuristics on the Multiple Objective Knapsack Problem. *Foundations of Computing and Decision Sciences*, 26(1):99-120.

157. Jiamthapthaksin, R., Eick, C.F., and Vilalta, R. (2009). A Framework for Multi-Objective Clustering and Its Application to Co-Location Mining, in uang, R., Yang, Q., Pei, J., Gama, J., Meng, X., and Li, X. (eds) *Advanced Data Mining and Applications. ADMA 2009. Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg, 978-3-642-03347-6.

158. Jiang, S., Yang, S., and Li, M. (2016). *On the use of hypervolume for diversity measurement of Pareto front approximations*. 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6-9 December 2016.

159. Jiang, S., Zhou, J., Yang, S., and Yao, X. (2022). Evolutionary Dynamic Multi-objective Optimisation: A Survey. *ACM Computing Surveys,* 55(4): 1-47. doi: 10.1145/3524495.

160. Jiao, L.C., Wang, H., Shang, R.H., and Liu, F. (2013). A co-evolutionary multi-objective optimisation algorithm based on direction vectors. *Information Sciences*, 228:90-112. doi: 10.1016/j.ins.2012.12.013.

161. Jing, R., Wang, M., Zhang, Z., Liu, J., Liang, H., Meng, C., Shah, N., Li, N., and Zhao, Y. (2019). Comparative study of posteriori decision-making methods when designing building integrated energy systems with multi-objectives. *Energy and Buildings,* 194: 123-139. doi: 10.1016/j.enbuild.2019.04.023.

162. José-García, A., and Gómez-Flores, W. (2021). *A survey of cluster validity indices for automatic data clustering using differential evolution*. GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference, Lille, France, 10-14 July 2021.

163. Julter, H. (1967). Liniejnaja modiel z nieskolkimi celevymi funkcjami (Linear Model with Several Objective Functions). *Ekonomika i matematiceckije Metody,* 11(2):208-220.

164. Karamooz, M.R., Forouzan, M.R., and Moosavi, H. (2011). Flow irregularity and wear optimisation in epitrochoidal gerotor pumps. *Meccanica*, 3:397–406. doi: 10.1007/s11012-011-9473-6.

165. Kang, S., and Kumar, R. (2008). Magellan: A Search and Machine Learning-based Framework for Fast Multi-core Design Space Exploration and Optimisation. *2008 Design, Automation and Test in Europe*, Munich, Germany, 10-14 March 2008.

166. Kang, Y., Wub, K., Gaoa, S., Ngc, I., Raoa, J., Yed, S., Zhange, F., and Feib, T. (2022). STICC: A multivariate spatial clustering method for repeated geographic pattern discovery with

consideration of spatial contiguity. https://arxiv.org/abs/1912.00643. Cited 16 January 2022. Last updated 31 March 2022.

167. Kazimipour, B., Li, X., and Qin, K. (2009). A Review of Population Initialization Techniques for Evolutionary Algorithms. *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 6-11 July 2014.

168. Keerativuttitumrong, N., Chaiyaratana, N., and Varavithya, V. (2002). 'Multi-objective Co-operative Co-evolutionary Genetic Algorithm', Guervós, J.J.M., Adamidis, P., Beyer, HG., Schwefel, HP., and Fernández-Villacañas, JL. (eds). *Parallel Problem Solving from Nature*, vol 2439. Springer, Berlin, Heidelberg. ISBN: 978-3-540-44139-7.

169. Khare, M., Patnaik, T., and Khare, A. (2009). A Comparison of Multi-objective Evolutionary Algorithms. *National Conference on Emerging Trends in Software & Networking Technologies*, Noida, India, 17-18 April 2009.

170. Khazaee, A., and Naimi, H.M. (2011). Two Multi-Objective Genetic Algorithms for Finding Optimum Design of an I-Beam. *Engineering*, 3(10):1054-1061. doi: 10.4236/eng.2011.310131.

171. Kirkland, O., Rayward-Smith, V., and de la Iglesia, B. (2011). A Novel Multi-Objective Genetic Algorithm for Clustering., in Yin, H., Wang, W., and Rayward-Smith, V. (eds) *Intelligent Data Engineering and Automated Learning - IDEAL 2011*, vol 6936.. Springer, Berlin, Heidelberg, pp. 317–326.

172. Kim, J., and Yoo, S. (2018). Software review: DEAP (Distributed Evolutionary Algorithm in Python) library. *Genetic Programming and Evolvable Machines*, 33:1455-1465. doi: 10.1007/s10710-018-9341-4.

173. Kim, K., Kim, H., and Denis, J.D. (2015). Spatial Optimisation for Regionalization Problems with Spatial Interaction: A Heuristic Approach. *International Journal of Geographical Information Science*, 30:1-23. doi: 10.1080/13658816.2015.1031671.

174. Kiser, J., (2018). Developing Optimisation Techniques for Logistical Tendering Using Reverse Combinatorial Auctions. MSc thesis, East Tennessee State University, USA.

175. Knarr, M.R., Golts, M., Lamont, G.B., and Huang, J. (2004). In situ bioremediation of perchlorate-contaminated groundwater using a multi-objective parallel evolutionary algorithm. *The 2003 Congress on Evolutionary*, Canberra, ACT, Australia, 8-12 December 2003.

176. Knowles, J., and Corne, D. (2000). M-PAES: A memetic algorithm for multi-objective optimisation. *Proc. Congr. Evol. Comput. 2000*, La Jolla, CA, USA, 16-19 July 2000.

177. Konak, A., Coit, D.W., and Smith, A.E. (2006). Multi-objective optimisation using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 91(9):992-1007. doi: 10.1016/j.ress.2005.11.018.

178. Kora, P., and Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. *International Journal of Computer Applications*, 162(10):34-36. doi: 10.5120/ijca2017913370.

179. Koza, J.R. (1995). Survey of genetic algorithms and genetic programming. *WESCON'95*, San Francisco, California, USA, 7-9 November 1995.

180. Kreige, N., Mutzel, P., and Schäfer, T. (2014). Practical SAHN Clustering for Very Large Data Sets and Expensive Distance Metrics. *Journal of Graph Algorithms and Applications*, 18(4):577-602. doi: 10.7155/jgaa.00338.

181. Kul'ka, J., Mantič, M., and Onofrejova, D. (2022). Case study of production optimisation using the enumerative method. *MM Science Journal*, 2022(2): 5656-5661. doi: 10.17973/MMSJ.2022_06_2022057.

182. Kumar, A. (2013). Encoding schemes in genetic algorithm. *International Journal of Advanced Research in IT and Engineering*, 2(3):1-7.

183. Kumar, M., Husian, M., Upreti, N., and Gupta, D. (2010). Genetic Algorithm: Review and Application. *International Journal of Information Technology and Knowledge Management*, 20:139-142.

184. Kumar, R., and Rockett, P. (2002). Improved sampling of the pareto-front in multi-objective genetic optimisations by steady-state evolution: a Pareto converging genetic algorithm. *Evolutionary Computation*, 10(3):283-314. doi: 10.1162/106365602760234117.

185. Kumar S G, V., and Panneerselvam, R. (2017). A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator. *International Journal of Computational Intelligence Research*, 13(7):1717-1733.

186. Kuyzu, G. (2007). Procurement in Truckload Transportation. Phd thesis, Georgia Institute of Technology, Georgia.

187. Laili, Y., Zhang, L., and Li, Y. (2019). Parallel transfer evolution algorithm. *Applied Soft Computing*, 75:686-701. doi: 10.1016/j.asoc.2018.11.044.

188. Laszczyk, M., and Myszkowski, P.B. (2019). Survey of quality measures for multi-objective optimisation: Construction of complementary set of multi-objective quality measures. *Swarm and Evolutionary Computation*, 48:109-133. doi: 10.1016/j.swevo.2019.04.001.

189. Lara, A., Sanchez, G., Coello Coello, C.A., and Schütze, O. (2010). HCS: A New Local Search Strategy for Memetic Multi-objective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112-132. doi: 10.1109/TEVC.2009.2024143.

190. Lee, C-G., Kwon, R.H., and Ma, Z. (2007). A carrier's optimal bid generation problem in combinatorial auctions for transportation procurement. *Transportation Research Part E: Logistics and Transportation Review*, 43(2):173-191. doi: 10.1016/j.tre.2005.01.004.

191. Leiva, H.A., Esquivel, S.C., and Gallard, R.H. (2000). Multiplicity and Local Search in Evolutionary Algorithms to Build the Pareto Front. *Proceedings of the XX International Conference of the Chilean Computer Science Society*, Santiago, Shile, 16-18 November 2000.

192. Li, G., Wang, G-G., and Wang, S. (2021). Two-Population Coevolutionary Algorithm with Dynamic Learning Strategy for Many-Objective Optimisation. *Mathematics,* 9(4):420. doi: 10.3390/math9040420.

193. Li, H., Ye, X., Imakura, A., and Sakurai, T. (2020). Ensemble Learning for Spectral Clustering. *2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, 17-20 November 2020.

194. Li, L., Li, M., and Lin, D. (2007). A Novel Epsilon-Dominance Multi-objective Evolutionary Algorithms for Solving DRS Multi-objective Optimisation. *Third International Conference on Natural Computation*, Haikou, China, 24-27 August 2007.

195. Li, J., and Rhinehart, R.R. (1998). Heuristic random optimisation. *Computers and chemical engineering*, 22(3):427-444.

196. Li, X., and Wong, K.-C. (2019). Evolutionary Multi-objective Clustering and Its Applications to Patient Stratification. *IEEE Transactions on Cybernetics*, 49(5):1680-1693. doi: 10.1109/TCYB.2018.2817480.

197. Li, X., Claramunt, C., and Xiao, N. (2011). Initialization strategies to enhancing the performance of genetic algorithms for the p-median problem. *Computers and Industrial Engineering*, 61:1024-1034. doi: 10.1016/j.cie.2011.06.015.

198. Li, Y., Zhou, A., and Zhang, G. (2014). An MOEA/D with multiple differential evolution mutation operators. *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 6-11 July 2014.

199. Lim, A., Xu, Z., and Rodrigues, B. (2008). Transportation Procurement with Seasonally Varying Shipper Demand and Volume Guarantees. *Operations Research*, 56:758-771. doi: 10.1287/opre.1070.0481.

200. Lindsey, K.A., Erera, A.L., and Savelsbergh, M.W.P. (2013). A pickup and delivery problem using crossdocks and truckload lane rates. *EURO Journal on Transportation and Logistics*, 2(1-2):5-27. doi: 10.1007/s13676-012-0013-x.

201. Liu, Y.X., and Miller, A.C. (2021). Should shippers be afraid of ghost freight? An empirical analysis of a customer portfolio from tmc, a div. Of CH Robinson. MSc thesis, Massachusetts Institute of Technology, Massachusetts.

202. Liu, K., Qiu, P., Gao, S., Lu, F., Jiang, J., and Yin, L. (2020). Investigating urban metro stations as cognitive places in cities using points of interest. *Cities*, 2020. doi: 10.1016/j.cities.2019.102561.

203. Liu,Y., Őzyer, T., Alhajj, R., and Barker, K. (2005). Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering. *Informatica*, 29:33-40.

204. Lohn, J.D., Kraus, W.F., and Haith, G.L. (2001). Comparing a Coevolutionary Genetic Algorithm for Multi-objective Optimisation. *Congress on Evolutionary Computation (CEC'2002)*, Honolulu, HI, USA, 12-17 May 2002.

205. Lopes, RJFSB. (2011). Location-routing problems of semi-obnoxious facilities. Phd thesis, University of Aveiro, Portugal.

206. Luo, Z-Q., and Y, W. (2006). An Introduction to Convex Optimisation for Communications and Signal Processing. *IEEE Journal on Selected Areas in Communications*, 24(8):1426-1438. doi: 10.1109/JSAC.2006.879347.

207. Maaranen, H., Miettinen, K., and Penttinen, A. (2007). On initial populations of a genetic algorithm for continuous optimisation problems. *Journal of Global Optimisation*, 37:405-436. doi: 10.1007/s10898-006-9056-6.

208. Magee, L. (2011). "2 - Frameworks for knowledge representation", in Cope, B., Kalantzis, M,. and Magee, L. (ed.). *Towards a Semantic Web.* Cambridge:Chandos Publishing, pp.15-34.

209. Mai, G., Janowicz, K., Gao, S., and Hu, Y. (2017). ADCN: An Anisotropic Density-Based Clustering Algorithm for Discovering Spatial Point Patterns with Noise. *Transactions in GIS*, 22(1). doi: 10.1111/tgis.12313.

210. Maier, H. R., Razavi, S., Kapelan, Z., and Matott, L. S. (2019). Introductory overview: Optimisation using evolutionary algorithms and other metaheuristics. *Environmental Modelling and Software*, 114. doi: 10.1016/j.envsoft.2018.11.018.

211. Maneeratana, K., Boonlong, K., and Chaiyaratana, N. (2004). Multi-objective optimisation by co-operative co-evolution. *Parallel Problem Solving from Nature-PPSN VIII: 8th International Conference*, Birmingham, UK, 18-22 September 2004.

212. Marler, R.T., and Arora, J.S. (2010). The weighted sum method for multi-objective optimisation: New insights. *Structural and Multidisciplinary Optimisation*, 41(6):853-862. doi: 10.1007/s00158-009-0460-7.

213. Martí, L., Garcia, J., Berlanga, A., and Molina, J. (2009). An Approach to Stopping Criteria for Multi-objective Optimisation Evolutionary Algorithms: The MGBM Criterion. *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, Trondheim, Norway, 18-21 May 2009.

214. Martinez, L.M., Viegas, J.M., and Silva, E.A. (2009). A traffic analysis zone definition: a new methodology and algorithm. *Transportation*, 36(6):581-599.

215. Mashwani, W.K., Salhi, A., Jan, M.A., Khanum, R., and Sulaiman, M. (2015). Impact Analysis of Crossovers in Multi-objective Evolutionary Algorithm. *Science International (Lahore)*, 27(6):4943-4956.

216. Mavortas, G. (2009). Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 213(2):455-465. doi: 10.1016/j.amc.2009.03.037.

217. Mao, J., Hirasawa, K., Hu, J., and Murata, J. (2000). Genetic Symbiosis Algorithm for Multi-objective Optimisation Problems. T*ransactions of the Society of Instrument and Control Engineers*, 37(9):893-901. doi 10.1109/ROMAN.2000.892484.

218. Maoh, H., and Kanaroglou, P. (2007). Geographic clustering of firms and urban form: a multivariate analysis. *Journal of Geographical Systems*, 9:29-52. doi: DOI:10.1007/s10109-006-0029-6.

219. Matake, N., Hiroyasu, T., Miki, M,. and Senda, T. (2007). Multi-objective clustering with automatic k-determination for large-scale data. *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO'07)*, London, UK, 7-11 July 2007.

220. Merkle, L.D., and Lamont, G.B. (2014). A Random Function Based Framework for Evolutionary Algorithms. In Bäck, W. (ed.) *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Mateo, California: Morgan Kaufmann Publishers, pp. 105-112.

221. Maulik, U., and Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, 33:1455-1465. doi: 10.1016/S0031-3203(99)00137-5.

222. Maulik, U., Bandyopadhyay, S., and Mukhopadhyay, A. (2001). *Multi-objective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*, 1st edition. Springer. ISBN: 978-3-642-16614-3.

223. Maulik, U., Bandyopadhyay, S., and Mukhopadhyay, A. (2009). Combining Pareto-Optimal Clusters using Supervised Learning for Identifying Co-expressed Genes. *BMC Bioinformatics*, 10(27):1-16. doi: 10.1186/1471-2105-10-27.

224. Mesa-Arango, R., and Ukkusuri, S.V. (2015). Demand clustering in freight logistic networks. *Transportation Research Part E*, 81:36-51. doi: 10.1016/j.tre.2015.06.002.

225. Michalewicz, M. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edition. Verlag: Springer. ISBN: 3-540-60676-9.

226. Miettinen, K. (1998). *Nonlinear multi-objective optimisation*, 1st edition. New York: Springer. ISBN: 978-1-4613-7544-9.

227. Mishra, B.S.P., Dehuri, S., Mall, R., and Ghosh, A. (2011). Parallel Single and Multiple Objectives Genetic Algorithms: A Survey. *International Journal of Applied Evolutionary Computation*, 2(2):21-57. doi: 10.4018/jaec.2011040102.

228. Mishra, S.K., Panda, G., Meher, S., Majhi, R., and Singh, M. (2011). Portfolio management assessment by four multi-objective optimisation algorithm. 2*011 IEEE Recent Advances in Intelligent Computational Systems*, Kerala, India, 22-24 September 2011.

229. Mitchell, M. (1999). *An Introduction to Genetic Algorithms*, 1st edition. Cambridge: The MIT Press. ISBN: 0−262−13316−4.

230. Mosayebi, M*.,* and Sodhi, M. (2020). Tuning genetic algorithm parameters using design of experiments. *GECCO '20: Genetic and Evolutionary Computation Conference*, Cancún, Mexico, 8-12 July 2020.

231. Mostaghim, S., and Teich, J.R. (2004). The Role of ε-Dominance in Multi Objective Particle Swarm Optimisation Methods. *Proc of IEEE Swarm Intelligence Symposium*, Indianapolis, USA, Indianapolis, USA. 26 April 2003.

232. Mukhopadhyay, A., and Maulik, U. (2011). A multi-objective approach to MR brain image segmentation. *Applied Soft Computing*, 11:872-880.

233. Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2015). A Survey of Multiobjective Evolutionary Clustering. *ACM Computing Surveys*, 47(4):1-46. doi: 10.1145/2742642.

234. Murata, T., Kaige, S., and Ishibuchi, H. (2003). Generalization of Dominance Relation-Based Replacement Rules for Memetic EMO Algorithms. *Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference*, Chicago, IL, USA, 12-16 July 2003.

235. Nagahara, M. (2010). 'Algorithms for Convex Optimisation', in Soldatos, J. (ed.) *Security Risk Management for the Internet of Things: Technologies and Techniques for IoT Security, Privacy and Data Protection.* Now Publishers, pp. 54-85.

236. Najundan, S., Sankaran, S., Arjun, C.R., and Paavai Anand, G. (2019). Identifying the number of clusters for K-Means: A hypersphere density based approach. https://arxiv.org/abs/1912.00643. Cited 12 December 2022. Last updated 4 December 2019.

237. Neef, M., Thierens, D., and Arciszewski, H. (2007). A case study of a multi-objective recombinative genetic algorithm with coevolutionary sharing. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Washington DC, USA, 6-9 July 1999.

238. Nievergelt, J. (2000). Exhaustive Search, Combinatorial Optimisation and Enumeration: Exploring the Potential of Raw Computing Power. *SOFSEM 2000: Theory and Practice of Informatics*, Milovoy, Czech Republic, 25 November – 2 December 2000.

239. Őzyer, T., Liu, Y., Alhajj, R., and Barker, K. (2004). Multi-objective genetic algorithm based clustering approach and its application to gene expression data. *Proceedings of the Third international conference on Advances in Information Systems (ADVIS'04)*, Izmir, Turkey, 20-22 October 2004.

240. Openshaw, S., Alvanides, S., and Whalley, S. (1998). *Some Further Experiments with Designing Output Areas for The 2001 UK Census. Report (*Leeds, UK: University of Leeds).

241. Openshaw, S. (1977) A geographical solution to scale and aggregation problems in region building, partitioning and spatial modelling. *Transactions of the Institute of British Geographers (New Series)*, 2:459–472. doi: 10.2307/622300.

242. Otman, A., Abouchabaka, J., and Tajani, C. (2012). Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem. *International Journal of Emerging Sciences*, 2(1):61-77.

243. Papadimitriou, C.H. and Steiglitz, K. (1998). *Combinatorial Optimisation: Algorithms and Complexity*, 2nd edition. New York: Dover Publications. ISBN 0-486-40258-4.

244. Paredis, J. (1995). Coevolutionary computation. *Artificial Life*, 2(4):355-375. doi: 10.1162/artl.1995.2.355.

245. Parmee, I.C., and Watson, A.H. (1999). 'Preliminary airframe design using co-evolutionary multi-objective genetic algorithms', in Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., and Smith, R.E. (eds) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, vol 2. Morgan Kaufmann, San Francisco, California, pp. 1657–1665.

246. Parsa, D. (2019). *Should-cost analysis as an alternative to open book accounting*. M.S. thesis, Tampere University, Tampere, Finland. doi: 10.13140/RG.2.2.28935.34723.

247. Pirim, H., Bayraktar, E., and Eksioglu, B. (2008). *Tabu Search: A Comparative Study*. in Jaziri, W. (ed.) *Local Search Techniques: Focus on Tabu Search*, 1st edition. Croatia: In-Teh. ISBN: 978-3-902613-34-7.

248. Pencheva, T., Atanassov, K., and Shannon, A. (2013). Modelling of a Roulette Wheel Selection Operator in Genetic Algorithms Using Generalized Nets: *Bio Automation*, 13(4):257-264

249. Ponsich, A., Pibouleau, L., Azzaro-Pantel, C., and Domenech, S. (2008). Constraint handling strategies in Genetic Algorithms application to optimal batch plant design. *Chemical Engineering and Processing: Process Intensification*, 47(3):420-434. doi: 10.1016/j.cep.2007.01.020.

250. Pogančić, M.V., Paulus, A., Musil, V., Matrius, G., and Rolinek, M. (2020). Differentiation of Blackbox Combinatorial Solvers. *ICLR Conference 2020 - Genetic and Evolutionary Computation Conference*, Online, 26 April – 1 May June 2020.

251. Potter, M.A. and Jong, K.A.D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1-29. doi: 10.1162/106365600568086.

252. Punia, P., and Kaur, M. (2013). Various Genetic Approaches for Solving Single and

Multi-Objective Optimisation Problems: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7):1014-1020.

253. Purshouse, R.C., and Fleming, P.J. (2002). Why use elitism and sharing in a multi-objective genetic algorithm. *the 4th Annual Conference on Genetic and Evolutionary Computation*, New York, USA, 9-13 July 2002.

254. Rahnamayan, S., Mahdavi, S., Deb, K., and Bidgoli, A. A. (2020). Ranking Multi-Metric Scientific Achievements Using a Concept of Pareto Optimality, *Mathematics*, 8(956). doi: 10.3390/math8060956.

255. Rana, S., and Caplice, C. (2020). Forecasting Long Haul Truckload Spot Market Rates. *MIT Global Supply Chain and Logistics Excellence Network working Paper Series*, 2020-mitscale-ctl-02.

256. Rani, D., Jain, S.K., Srivastava, D.K., and Perumal, M. (2013). "3 - Genetic Algorithms and Their Applications to Water Resources Systems", in Yang, X-S., Gandomi, A.H., Talatahari, S., and Alavi, A.H. (ed.) *Metaheuristics in Water, Geotechnical and Transport Engineering.* Elsevier, pp. 43-78. ISBN 9780123982964.

257. Raquel, C. and Naval, P. (2005). An effective use of crowding distance in multi-objective particle swarm optimisation. *GECCO 2005 - Genetic and Evolutionary Computation Conference*, Washington, USA, 25-29 June 2005.

258. Ravindran, A., Ragsdell, K.M., and Reklaitis, G.V. (2006). *Engineering Optimisation: Methods and Applications*, 2nd edition. New Jersey: John Wiley & Sons. ISBN: 978-0-471-55814-9.

259. Reyes Fernández de Bulnes, D., Bolufé-Röhler, A., and Tamayo-Vera, D. (2019). Multi-objective optimisation approach based on Minimum Population Search algorithm. *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología*, 7(2):1-19.

260. Ripon, K. S. N., and Siddique, M. N. H. (2009). Evolutionary multi-objective clustering for overlapping clusters detection. *Proceedings of the Eleventh conference on Congress on Evolutionary Computation (Proc. IEEE CEC'09)*, Trondheim, Norway, 18-21 May 2009.

261. Rogers, A., and Prugel-Bennett, A. (1999). Genetic Drift in Genetic Algorithm Selection Schemes. *IEEE Transactions on Evolutionary Computation,* 3:298-303. doi: 10.1109/4235.797972.

262. Rozenfeld, H.D., Rybski, D., Gabaix, X., and Makse, H.A. (2009). The Area and Population of Cities: New Insights from a Different Perspective on Cities. *NBER Working Paper* Series, 15409. Available at: https://www.nber.org/papers/w15409/. Cited 3 June 2022. Last updated October 2009.

263. Saha, S., and Bandyopadhyay, S. (2012). A generalized automatic clustering algorithm in a multi-objective framework. *Applied Soft Computing*, 13(2013):89-108. doi: 10.1016/j.asoc.2012.08.005.

264. Sahoo, D., Pati, J.K., Tripathy, A.K., and Panda, P.K. (2022). Effect of Pareto Optimality in Multi Objective Optimisation Under Fuzzy Environment Using Rank and Diversity. *SRN*. doi: 10.2139/ssrn.4142152.

265. Saini, N. and Saha, S. (2021). Multi-objective optimisation techniques: a survey of the state-of-the-art and applications. *The European Physical Journal Special Topics*, 230:2319-2335. doi: 10.1140/epjs/s11734-021-00206-w.

266. Santana-Quintero, L.V., Montaño, A.A., and Coello Coello, C.A. (2010). A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-objective Optimisation. *Computational Intelligence in Expensive Optimisation Problems*, 2:29-59. doi: 10.1007/978-3-642-10701-6_2.

267. Sarma, G.V., Sellami, L., and Houam, K.D. (1993). Application of Lexicographic Goal Programming in Production Planning—Two case studies. *Opsearch,* 30(2):141-162.

268. Sasket, S. and Pandya, S. C. (2016). An Overview of Partitioning Algorithms in Clustering Techniques. *International Journal of Advanced Research in Computer Engineering & Technology*, 5(6):1943-1946.

269. Schaffer, J.D. (1984). *Multi-objective Objective Optimisation with Vector Evaluated Genetic Algorithms*. Phd thesis, Vanderbilt University, Nashville, Tennessee.

270. Scheepers, C. and Engelbrecht, A.P. (2016). Misleading Pareto Optimal Front Diversity Metrics: Spacing and Distribution. *2016 IEEEE Symposium Series on Computational Intelligence*, Athens, Greece, 6-9 December 2016.

271. Seiler, T. (2012). *Operative Transportation Planning: Solutions in Consumer Goods Supply Chains*, 1st edition. Heidelberg: Springer. ISBN: 978-3-7908-2791-0.

272. Sheffi, Y. (2004). Combinatorial Auctions in Procurement of Transportation services. *International Journal of Geographical Information Science*, 35:689-716. doi: 10.1287/inte.1040.0075.

273. Shirakawa, S. and Nagao, T. (2009). Evolutionary image segmentation based on multi-objective clustering. *Proceedings of the Eleventh conference on Congress on Evolutionary Computation (Proc. IEEE CEC'09)*, Trondheim, Norway, 18-21 May 2009.

274. Shu, H., Pei, T., Song, C., Chen, X., Guo, S., Liu, Y., Chen, J., Wang, X., and Zhou, C. (2020). L-function of geographical flows. *INFORMS*, 34(4):245-252. doi: 10.1080/13658816.2020.1749277.

275. Shukla, A., Pandey, H.M., and Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, Greater Noida, India, 25-27 February 2015.

276. Simchi-Levi, D., Chen, X., and Bramel, J. (2013). *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*, 3rd edition. Heidelberg: Springer. ISBN: 978-1-4614-9148-4.

277. Solich, R. (1969). Zadanie programowania liniowego z wieloma funkcjami celu (Linear Programming Problem with Several Objective Functions). *Przeglad Statystyczny*, 16:24–30.

278. Song, C., Pei, T., and Shu, H. (2019). Identifying flow clusters based on density domain decomposition. *IEEEE Access*, 8:5236-5243. doi: 10.1109/ACCESS.2019.2963107.

279. Suominen, M. (2008). *Innovative approaches to transportation service procurement*. Bachelor's thesis, Tampere University of Applied Sciences, Tampere.

280. Srinivas, N., and Deb, K. (1994). Multi-objective optimisation using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221-248. doi: 10.1162/evco.1994.2.3.221.

281. Statistics South Africa. *South African Census Community Profiles 2011*. Pretoria: Statistics SA [producer], 2014.Cape Town: DataFirst [distributor], 2015. doi: 10.25828/6n0m-7m52, cited 13 January 2025. Last updated 29 March 2020.

282. Stork, J., Eiben, A.E., and Bartz-Beielstein, T. (2020). A new taxonomy of global optimisation algorithms. *Natural Computing*, 21:219-242. doi: 10.1007/s11047-020-09820-4.

283. Suman, B. (2004). Study of simulated annealing based algorithms for multi-objective optimisation of a constrained problem. *Computers & Chemical Engineering*, 28(9):1849-1871. doi: 10.1016/j.compchemeng.2004.02.037.

284. Taha, H.A. (2017). *Operations Research: An Introduction*, 10th edition. New York: Pearson. ISBN: 78-1-292-16554-7.

285. Talbi, E-G. (2019). A unified view of parallel multi-objective evolutionary algorithms *Journal of Parallel and Distributed Computing*, 113: 349-358. doi: 10.1016/j.jpdc.2018.04.012.

286. Talbi, E-G., Nebro, A-J., Basseur, M., and Alba, E. (2012). Multi-objective optimisation using metaheuristics: Non-standard algorithms. *International Transactions in Operational Research*, 19:283-305. doi: 10.1111/j.1475-3995.2011.00808.x.

287. Tao, R. and Thrill, J-C.F. (2016). A Density-Based Spatial Flow Cluster Detection Method. *International Conference on GIScience Short Paper Proceedings*, Montreal, Canada, 27-30 September 2016.

288. Tao, R. and Thill, J-C.F. (2016). Spatial Cluster Detection in Spatial Flow Data. *Geographic Analysis*, 40:1-18.

289. Tan, K.C., Yang, Y.J., and Goh, C-K. (2006). A distributed Cooperative coevolutionary algorithm for multi-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 10(5):527-549. doi: 10.1111/gean.12100.

290. Tan, T.G., Lau, H.K., and Teo, J. (2007). 'Cooperative Versus Competitive Coevolution for Pareto Multi-objective Optimisation, in Li, K., Fei, M., Irwin, G.W., and Ma, S. (eds). *Bio-*

Inspired Computational Intelligence and Applications. LSMS 2007. Lecture Notes in Computer Science, vol 4688. Springer, Berlin, Heidelberg. ISBN: 978-3-540-74768-0.

291. Tian, Y., Zhang, T., Xiao, J., Zhang, X. and Jin, Y. (2020). A coevolutionary framework for constrained multi-objective optimisation problems. *IEEE Transactions on Evolutionary Computation*, 25(1):102-116. doi: 10.1109/TEVC.2020.3004012.

292. Triki C., Oprea S., Beraldi P., and Crainic T.G. (2014). The stochastic bid generation problem in combinatorial transportation auctions. *European Journal of Operational Research*, 236(3):991–999. doi: 10.1016/j.ejor.2013.06.013

293. Truchet, C., Philippe, C. and Richoux, F. (2013). Prediction of Parallel Speed-ups for Las Vegas Algorithm. https://arxiv.org/abs/1212.4287. Cited 30 May 2023. Last updated 18 December 2012.

294. Umbarkar, A.J., and Sheth, P.D. (2015). Crossover operators in genetic algorithms: a review. *CTACT Journal on Soft Computing*, 6(1):1083-1092. doi: 10.21917/ijsc.2015.0150.

295. Umarusman, N. (2019). Using Global Criterion Method to Define Priorities in Lexicographic Goal Programming and an Application for Optimal System Design. *Manas Sosyal Araştırmalar Dergisi*, 8(1):326-341. doi: 10.33206/mjss.519112.

296. Van Veldhuizen, D.A. (1999). *Multi-objective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Phd thesis, Air Force Institute of Technology, Ohio.

297. Van Veldhuizen, D.A., and Lamont, G.B. (1998). *Multi-objective evolutionary algorithm research: A history and analysis*. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio.

298. Van Veldhuizen, D.A., and Lamont, G.B. (2000). Multi-objective Optimisation with Messy Genetic Algorithms. *2000 ACM Symposium on Applied Computing*, Como, Italy, 19-21 March 2000.

299. Vas, F., Lavinas, Y., Aranha, C., and Ladeira, M. (2021). Exploring Constraint Handling Techniques in Real-World Problems on MOEA/D with Limited Budget of Evaluations. *Evolutionary Multi-Criterion Optimisation: 11th International Conference*, Shenzen, China, 28-31 March 2021.

300. Verma, S., Pant, M., and Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimisation Problems. *IEEE Access*, 9:57757-57791. doi: 10.1109/ACCESS.2021.3070634.

301. Wagner, T., Martí, L., and Trautmann, H. (2011). A taxonomy of online stopping criteria for multi-objective evolutionary algorithms. *Evolutionary Multi-Criterion Optimisation: 6th International Conference, EMO 2011*, Ouro Preto, Brazil, 5-8 April 2011.

302. Wallin, D., Ryan, C., and Azad, R. (2005). Symbiogenetic coevolution. *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2-5 September 2005.

303. Wang, H., He, S., and Yao, X. (2017). Nadir point estimation for many-objective optimisation problems based on emphasized critical regions. *Soft Computing*, 21:2283-2295. doi: 10.1007/s00500-015-1940-x.

304. Wiem Mkaouer, M., and Kessentini, M. (2014). 'Chapter Four - Model Transformation Using Multi-objective Optimisation', in Hurson, A. (eds). *Advances in Computers*, vol 92. Elsevier. ISBN: 9780124202320.

305. Wierzchoń, S., and Kłopotek, M. (2017). *Modern Algorithms of Cluster Analysis*, 1st edition. Springer Cham. ISBN 978-3-319-69307-1.

306. Wolpert, D. H. and Macready, W. G. (1997). No Free Lunch Theorems for Optimisation. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. doi: 10.1109/4235.585893.

307. Wu, J. (2012). Advances in K-means Clustering: A Data Mining Thinking. Phd thesis, Beihang University, China.

308. Xiang, Q. and Wu, Q. (2019). Tree-Based and Optimum Cut-Based Origin-Destination Flow Clustering. *International Journal of Geo-Information*, 8(11). doi: 10.3390/ijgi8110477.

309. Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., and Liu, B. (2018). Interactive Multi-objective Optimisation: A Review of the State-of-the-Art. *IEEE Access*, 6:41256-41279. doi: 10.1109/ACCESS.2018.2856832.

310. Xiujuan, L. and Zhongke, S. (2004). Overview of multi-objective optimisation methods. *Journal of Systems Engineering and Electronics*, 15(2):142-146.

311. Yadati, C., Oliveira, C.A., and Pardalos, P.M. (2007). An Approximate Winner Determination Algorithm for Hybrid Procurement Mechanisms Logistics, in Kontoghiorghes, E.J., and Gatu, C. (eds) *Optimisation, Econometric and Financial Analysis. Advances in Computational Management Science, vol 9*. Springer, Berlin, Heidelberg. doi: 10.1007/3-540-36626-1_3.

312. Yang, F., Huang, Y-H., and Li, J. (2019). Alternative solution algorithm for winner determination problem with quantity discount of transportation service procurement. *European Journal of Operational Research*, 112(2):432:459. doi: 10.1016/j.physa.2019.122286.

313. Yang, J-B. (1999). Gradient projection and local region search for multi-objective optimisation. *European Journal of Operational Research*, 112(2):432-459.

314. Yang, X-S. (2014). *Chapter 14 - Multi-Objective Optimisation*, 1st edition. Elsevier. ISBN: 9780124167438.

315. Yang, X-S. (2014). *Chapter 4 - Simulated Annealing* in Yang, X-S. (ed.) *Nature-Inspired Optimisation Algorithms*, 1st edition. Elsevier. ISBN: 9780124167438.

316. Yao, X.J., Shao, Y.C., Chen, H., Wang, D.Z., and L.W. Tian. (2015). Predator-prey coevolution applied in multi-objective optimisation for micro-grid energy management. *015 International Conference on Testing and Measurement Techniques (TMTA 2015)*, Phuket Island, Thailand, 716-17 January 2015.

317. Ye, F., Neumann, F., de Nobel, J., Neumann, A., and Bäck, T. (2023). Towards Self-adaptive Mutation in Evolutionary Multi-Objective Algorithms. *Foundations of Genetic Algorithms (FOGA '23)*, Potsdam, Germany, 30 August-1 September 2023.

318. Zavoianu, C., Lughofer, E., Koppelstaetter, W.,Weidenholzer, G., Amrhein, W., and Klement, E.P. (2013). 'On the Performance of Master-Slave Parallelization Methods for Multi-Objective Evolutionary Algorithms', in Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds) *Artificial Intelligence and Soft Computing. ICAISC 2013.* vol 7895. Heidelberg: Springer. ISBN: 978-3-642-38609-1.

319. Zhang, W., and Fujimura, S. (2010). Improved Vector Evaluated Genetic Algorithm with Archive for Solving Multi-objective PPS Problem. *2010 International Conference on E-Product E-Service and E-Entertainment*, Henan, China, 7-9 November 2010.

320. Zhang, L., Hou, S-S., Hu, J-J., Xie, T. and Mei, H. (2010). Is operator-based mutant selection superior to random mutant selection? *ICSE '10: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, 1:435-444. doi: 10.1145/1806799.1806863.

321. Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25(2):103-114. doi: 10.1145/233269.233324.

322. Zheng, A., and Oliver, J. (2023). *Development of Market-Based Routing Guide Strategy*. M.S. thesis, Massachusetts Institute of Technology, Massachusetts, U.S.A.

323. Zhou, A., Qu, B-Y., Li, H., Zhao, S-Z., Suganthan, P.N., and Zhang, Q. (2011). Multi-objective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32-49. doi: 10.1016/j.swevo.2011.03.001.

324. Zhu, X., and Guo, D. (2014). Mapping Large Spatial Flow Data with Hierarchical Clustering. *Transactions In GIS*, 18(3):1-14. doi: 10.1111/tgis.12100.

325. Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multi-objective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173-195. doi: 10.1162/106365600568202.

326. Zitzler, E., Laumanns, M., and Thiele, L. (2001). 'Improving the Strength Pareto Evolutionary Algorithm', in Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., and Fogarty, T. (eds) *EUROGEN 2001. Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems.* Athens, Greece. pp.95-100.

327. Zitzler, E., and Thiele, L. (1999). Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257-271. doi: 10.1109/4235.797969.

328. Zydallis, J.B., Van Veldhuizen, D.A., and Lamont, G.B. (2001). 'A Statistical Comparison of Multi-objective Evolutionary Algorithms Including the MOMGA-II', in Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., and Corne, D. (eds) *Evolutionary Multi-Criterion Optimisation. EMO 2001. Lecture Notes in Computer Science.* vol 1993. Heidelberg: Springer. ISBN: 978-3-540-41745-3.