AN INTRODUCTION TO THE CONCEPT OF ROVING PUMPS IN THE SYNTHESIS OF MULTIPURPOSE BATCH PLANTS

Shaun Engelbrecht

Master of Science in Engineering by research:
"A dissertation submitted to the Faculty of Engineering and Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.''

Johannesburg, 2018

Contributions

The work contained in this dissertation has been published in the ACS Industrial and Engineering Chemistry Research journal:

Engelbrecht S., Majozi T., 2018. Introducing the Concept of Roving Pumps in the Synthesis of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 57, 2231-2247.

The work was also accepted for inclusion to the PSE 2018 (San Diego, USA) conference proceedings with oral presentation.

Declaration

I declare that this dissertation is my own unaided work. It is being submitted to the Degree of Master of Science in Chemical Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

____

Signature of Candidate

____________ day of _____________ year _________________

To my mom,

Anne-Mariè

who has shown me the value of perseverance and understanding

Acknowledgements

Throughout this endeavor, many people have shared their wisdom, given advice and helped think about the many problems faced. All of their inputs are highly appreciated. Yet it is through the Almighty, Jesus Christ, that all things are possible.

Prof. Thokozani Majozi and my esteemed research colleagues, who always had sound advice and critique to help along the way - you have my thanks. In particular, Joel Croft, a friend, intellectual and open mind, has given up many hours to lend a helping hand and a thinking mind.

Without my family, this work would surely have failed countless times, thank you for your understanding and unrelenting support.

Financially this work would not have been possible without the generous help of the National Research Fund of South Africa. Their inputs help shape the future of research in the country.

Abstract

Multipurpose batch chemical plants are commonly used to produce valuable, high quality products in low volumes. A series of discrete and independently operated tasks allow for flexible production in order to adapt to rapid market fluctuations. Batch plants commonly require pumping equipment in order to transfer liquid material between processing units and storage vessels. However, previous mathematical formulations that address short-term scheduling and synthesis of multipurpose batch plants have largely omitted the importance and impact of the pumping equipment on the batch plant.

Presented in this dissertation is a novel technique to flexibly allocate pumps throughout a batch plant, so called roving pumps. The nature of batch processes allows for the allocation of pumps to different units and tasks depending on pump availability. A Mixed Integer Linear Program mathematical formulation is presented to determine the optimal number and allocation of pumps within the batch plant while simultaneously synthesizing the plant and optimizing the batch production schedule. Also addressed in this dissertation are the material transfer times associated with variable batch sizes during production. By optimally allocating pumps to processing units, this formulation has shown a significant decrease in the number of pumps required within a batch plant, thereby improving the utilization of available equipment. It is shown by way of an example that it is possible to decrease the number of pumps required for operation from eleven to three pumps, while another example identified a decrease in the number of pumps required from seven to two. The formulation was also applied to two case studies, where it was shown that it is possible to reduce the number of pumps required from 17 to 4 and from 19 to 5 for each case, respectively. By using fewer pieces of equipment more frequently, equipment degradation due to nonuse can be reduced and equipment utilization can be maximized. It is, therefore, possible to significantly reduce capital cost investment and maintenance costs by implementing the formulation while still achieving or improving on existing throughputs.

CONTENTS

List of Figures

List of Tables

CHAPTER

1

INTRODUCTION

1.1 Background

Multipurpose batch chemical plants afford great flexibility by producing a number of different products using the same equipment. This flexibility allows for improved equipment utilization while adapting to rapid market fluctuations. In the selection of batch equipment, the processing requirements and production material properties dictate the design. Therefore, most batch chemical plants, such as in the food, pharmaceutical and agrochemical industries, predominantly contain liquids that need to be transferred between processing units using pumps.

Batch processes are commonly used to produce high value products in low volumes (Lee *et al.*, 2015). These products are produced using a series of discrete tasks or steps in order to ensure that product quality and yield is controlled (Pourali *et al.*, 2006). The nature of batch processing affords the ability to be flexible regarding the type and number of products which can be manufactured, as well as the sequencing of tasks. Therefore, optimal scheduling is key for the efficient operation of batch chemical plants. However, historically, the bulk of optimization research has been in continuous processes and not batch processes, largely due to most process industries being designed on a steady state basis. Additionally, the dynamic nature of batch plants makes them challenging to optimize. Nevertheless, significant advances have been made in the area of batch scheduling by use of mathematical modelling in order to facilitate solutions to complex problems (Majozi, 2010).

Batch plants commonly consist of a number of major processing units, such as reactors, heaters and separators, that are ultimately responsible for the transformation of feed materials into high value products. Along with the major equipment, a number of intermediary storage vessels for material are often employed. Connecting the major equipment and storage vessels is a pumping and piping network which is generally responsible for all the liquid transfers within the plant (Loonkar and Robinson, 1970). Combined, the processing units, storage vessels, pumps and piping form the batch plant.

The discrete operation of tasks within batch plants allows for the possibility of reusing processing equipment for various tasks. Similarly, pumping equipment is used intermittently and pumps can, therefore, also be reused for various transfer tasks. Hence, it is possible to reuse pumping equipment more frequently and also reduce the total number of pumps required for material transfer within the plant. Material transfers are generally simplified to fixed time durations when scheduling plant production, regardless of the batch size and costs associated with the transfers (Hegyhati and Friedler, 2011).

The synthesis of multipurpose batch plants requires that the optimal equipment be selected in order to cost effectively produce products. Previous work omitted the pumping equipment from the overall synthesis problem in order to optimally pair material transfer tasks to available pumps.In order to address this gap, a new design method is required that is able to accurately integrate the pumping network with the synthesis and scheduling problems of multipurpose batch plants. In this dissertation, a design method based on a novel mathematical formulation is presented

1.2 Motivation

Conventionally, pumps are fixed between two or more areas in a batch plant as seen in Figure 1-1 below. This results in a potentially large number of pumps needed throughout the plant (Loonkar and Robinson, 1970). These pumps will only be active for limited periods of time when they are required to transfer specific materials between sources and sinks. Consequently, this could result in some pumps being idle for long periods of time and it may, therefore, be possible to repurpose these idle pumps to transfer material for different tasks.

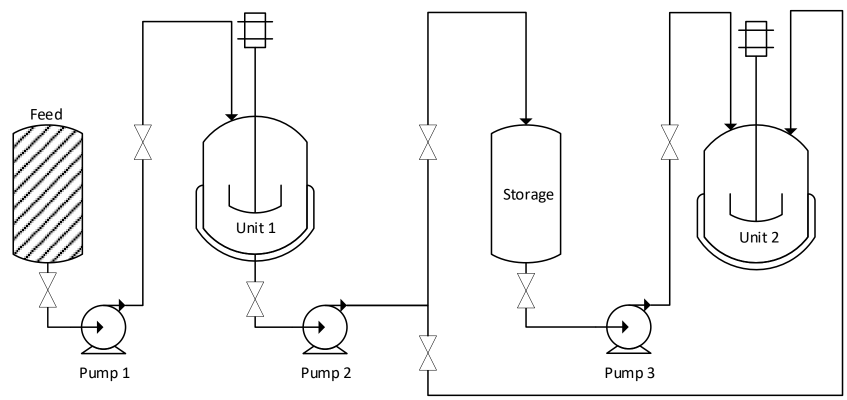


Figure 1-1 Conventional pump allocation within a batch plant

The allocation of pumps is a crucial point to consider, especially when noting that pumps can be expensive components of the plant, both in terms of capital and operational costs, depending on size and design. Additionally, in order to schedule a batch plant as realistically as possible, the inclusion of pumping allocation and the associated material transfer times is paramount. This inclusion is of even greater importance when considering the nature of multipurpose batch plants, wherein material may be transferred between a number of shared processing units. Hence, it is expected that pump allocation and material transfer time be included in the mathematical models that address batch plant scheduling and synthesis.

In this dissertation, the concept of roving pumps is, therefore, introduced. These roving pumps are not necessarily dedicated to particular units or sections in the plant but can, instead, be reassigned to different processing units and sections as required. By using roving pumps, flexibility in the batch plant is increased. Figure 1-2 illustrates how a single pump may be used to perform the various tasks instead of the multiple pumps depicted in Figure 1-1. It is evident from Figure 1-2 that the proposed allocation of pumps would result in increased pumping equipment utilization. This is favorable since prolonged disuse of pumps may increase maintenance costs due to clogging or blockages and structural degradation.

Figure 1-2 (a)–(d) depict that a reduced number of pumps may be used to transfer material between various units when allowing the pumps to be flexibly allocated.

- Figure 1-2 (a) shows that Pump 1 may be used to transfer material from a storage vessel to a unit.
- Figure 1-2 (b) shows that the same pump (Pump 1) may alternatively be used to transfer material from feed stockpiles to units.
- Figure 1-2 (c) shows that the same pump (Pump 1) may also be used to transfer material between different processing units.
- Figure 1-2 (d) shows that the same pump (Pump 1) may be used to transfer material from units to storage vessels as well.

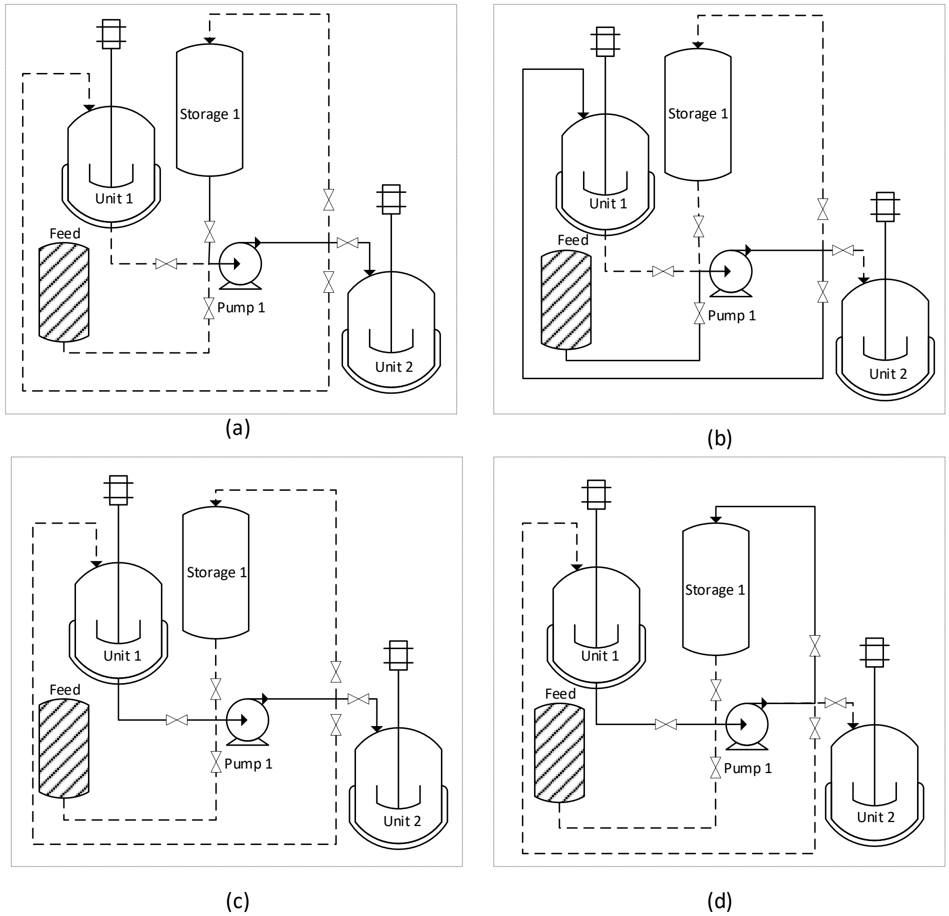


Figure 1-2 Proposed pump allocation

Notably, the flexible allocation of pumps can be achieved while preserving or even improving on the production throughput of cases where flexible pump allocation is not taken into consideration. The flexible allocation of pumps makes it possible to pair the best available pump to a specific material transfer, hence improving on the throughput capabilities of the plant.

1.3 Scope

The focus of this dissertation is, therefore, to present a mathematical model that enables the simultaneous synthesis and scheduling of multipurpose batch plants while incorporating flexible allocation of pumps throughout the plant. Any additional aspects of batch plant scheduling, for example resource minimization or heat integration, are therefore, not part of the scope of the dissertation.

It is expected that the proposed work will aid in the correct scheduling of batch plants by incorporating accurate transfer times between various sources and sinks in a batch chemical facility. Additionally, the synthesis aspect of the work is aimed to optimally select the required equipment and, therefore, decrease the total number of processing equipment, storage vessels, pumps and piping connections in the facility. This is to be achieved while still at least maintaining the original throughput of a conventionally operated facility.

1.4 Objectives

The objectives of the research may be summarized as follows:

i) To develop a deterministic mathematical model that is able to synthesize detailed pumping networks for multipurpose batch plants such that the optimal number of pumps is used.

ii) To develop a mathematical model that accurately determines material transfer times using pumps of different ratings and variable batch production sizes.

iii) To develop a model that accurately schedules pump activity, material transfers and processing for multipurpose batch plants.

iv) To combine the synthesis of the pumping network with that of the major equipment for the overall synthesis of a multipurpose batch plant.

v) To validate the proposed model against literature examples and case studies in order to show the accuracy and benefit of the proposed model.

1.5 Problem Statement

The problem that is addressed in this dissertation can be stated as follows.

Given:

(i) The production recipes of various products,

(ii) The available processing units and storage vessels,

(iii) The available pumps and respective pumping rates, and

(iv) The time horizon of interest.

Determine:

(i) The optimal plant production schedule,

(ii) The optimal number of processing units and piping connections,

(iii) The optimal allocation of pumps to processing units within the time horizon,

(iv) The required material quantity to be processed in each processing unit, and

(v) The time required to transfer the material between areas within the plant.

This should be accomplished in such a way that the performance of the batch processing plant is maximized.

1.6 Structure

This dissertation consists of six chapters, which are outlined as follows.

- Chapter 1 gives the background and purpose of the research presented in the dissertation
- Chapter 2 presents a literature review of optimization theory and the prominent work in the fields of scheduling and synthesis of multipurpose batch plants
- Chapter 3 gives a novel mathematical formulation for the synthesis of multipurpose batch plants and the pumping network
- Chapter 4 presents illustrative examples and the results for the examples when the proposed model was applied
- Chapter 5 entails the limitations and recommendations for the work and possible future developments. Finally, it draws the conclusion for this work.

1.7 References

Hegyhati, M., Friedler, F., 2011. The role of energy consumption in batch process scheduling, in: 21st European Symposium on Computer Aided Process Engineering - ESCAPE 21. pp. 1979–1983.

Lee, J., Seid, E.R., Majozi, T., 2015. Heat integration of intermittently available continuous streams in multipurpose batch plants. Comput. Chem. Eng. 74, 100–114.

Loonkar, Y.R., Robinson, J.D., 1970. Minimization of Capital Investment for Batch Processes Calculation of Optimum Equipment Sizes. Ind. Eng. Chem. Process Des. Dev. 9, 625–629.

Majozi, T., 2010. Batch Chemical Process Integration, Springer.

Pourali, O., Amidpour, M., Rashtchian, D., 2006. Time decomposition in batch process integration. Chem. Eng. Process. Process Intensif. 45, 14–21.

CHAPTER

2

LITERATURE REVIEW

2.1 Introduction

This dissertation focuses on the optimal synthesis of multipurpose batch plants by incorporating flexible pump allocation throughout the plant. Various techniques have previously been employed to optimally synthesize and schedule batch plants. Hence, the pertinent techniques, recipe representations, operational philosophies and time representations that relate to batch processing are discussed. Finally, optimization theory is discussed and a conclusion is drawn to the research gap and the applicability of the proposed model.

Rippin (1983) noted that two types of batch chemical plants exist, that of multiproduct and that of multipurpose. The main distinction between the two types of plants is the manner in which they are operated and the equipment configured.

Multiproduct batch plants

Multiproduct batch plants, also known as flowshop plants, follow a linear sequence of tasks to produce products and are of a simpler configuration compared to multipurpose batch plants. Each of the steps involved has a dedicated unit to perform a particular task. All the raw materials entering the batch plant follow the same sequence of tasks in order to produce the final products. This linear sequencing of tasks in units, as shown in Figure 2-1 (a), allows for relative ease in the scheduling of the batch production. Only a single recipe may be followed per batch, however, different batches need not produce the same product, nor have the same processing durations.

Multipurpose batch plants

Multipurpose batch plants, also known as jobshop plants, do not follow a linear sequence of tasks in units, as shown in Figure 2-1 (b). Various tasks are free to be performed in shared pieces of equipment, thereby greatly increasing the flexibility of the multipurpose batch plant. Multipurpose batch plants are, therefore, more complex than multiproduct batch plants and scheduling becomes distinctly more difficult. Nevertheless, improved profits from multipurpose batch plants are achievable since a larger number of products may be produced using the same equipment.

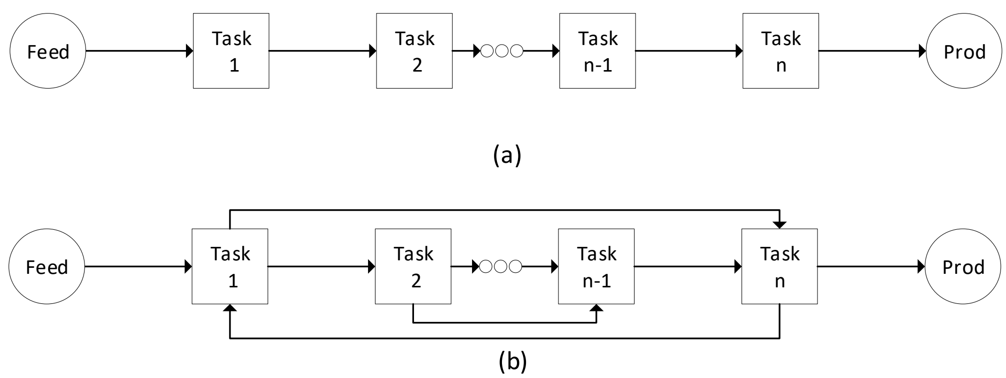


Figure 2-1 (a) Multiproduct versus (b) Multipurpose batch plants

2.2 Recipes

The path followed in order to produce products is known as the recipe. Various methods and techniques of representing the recipe have been discussed in literature (Kondili *et al.* (1993), Pantelides (1994), Ierapetritou and Floudas (1998), Majozi and Zhu (2001)). The manner in which the recipe is represented aides in the understanding of batch plant operation and depicts the manner in which various processing tasks are related. The main recipe representations that have been used extensively include that of the State Task Network, Resource Task Network, State Sequence Network and the State Equipment Network.

State Task Network

Introduced by Kondili *et al.* (1993), the STN recipe representation makes use of two different types of nodes which are linked by arcs. A circle node represents states, or materials, and rectangle nodes represent tasks which need to be performed within the processing units. This node and arc diagram signifies which states are required by various processing tasks. This representation depicts all states that are raw materials, intermediates and final products. Figure 2-2 illustrates the STN representation for a recipe.

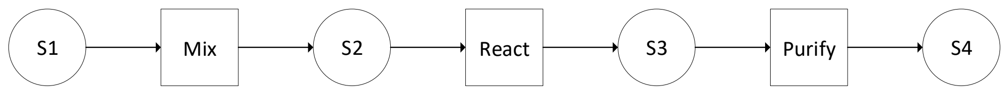


Figure 2-2 State task network

Resource Task Network

Pantelides (1994) based an approach on the STN representation and developed the Resource Task Network (RTN) recipe representation, as shown in Figure 2-3. This representation is similar to that of the STN and allows the distinguishing between states which are raw materials, intermediates and products as well as other resources, such as utility, manpower and storage. As illustrated in Figure 2-3, the equipment that are to be used are specified as part of the recipe. The other distinctions, as mentioned earlier, may be incorporated similarly. Tasks are defined similarly to that of the STN representation and are extended to include aspects such as cleaning, storage and transportation.

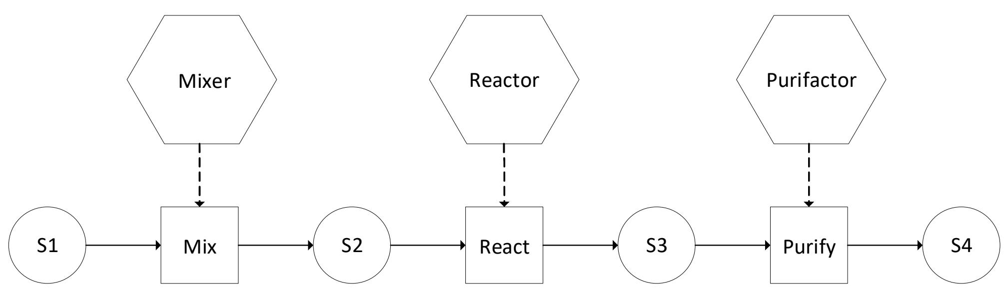


Figure 2-3 Simple RTN representation

Both the STN and RTN representation require a large number of binary variables in order to describe whether a task *i* occurs in a unit *j* at a time point *p*. The resulting number of binary variables are, therefore, $i \times j \times p$. Ierapetritou and Floudas (1998) attempted to decrease the number of binary variables required by decoupling units and tasks by introducing the concept of one-to-one unit and task relationship. They noted that it may be possible to use separate binary variables to represent unit availability and tasks, resulting in $(i + j) \times p$ binary variables.

However, Sundaramoorthy and Karimi (2005) explained that the number of binary variables required will remain the same for the case when tasks and units are decoupled and for the case where a three index variable is employed. This is due to the fact that the method proposed by Ierapetritou and Floudas (1998) requires that a new task to unit relationship be defined when multiple units may perform the same task. Specifically, a new task is required to be defined per processing unit for every shared task. The method of Ierapetritou and Floudas (1998) decreases the number of variables by eliminating the need for variables that pertain to unit activity, yet increases the number of variables again due to the additional tasks that are required to be defined. This method, therefore, does not yield a net decrease in the number of binary variables.

State Sequence Network

The State Sequence Network (SSN) representation introduced by Majozi and Zhu (2001), allowed the introduction of a single-type variable for scheduling. The binary dimension was significantly reduced which facilitated the solution of complex problems (Stamp and Majozi, 2011). Binary variables associated with tasks and units are implicitly incorporated since a change in state implicitly requires a task to occur as shown in Figure 2-4. Furthermore, if more than one state is required for a given processing task, only a single effective state is required to be defined to represent a task. Therefore, the number of binary variables present in the formulation is the number of effective states multiplied by the number of events that take place.

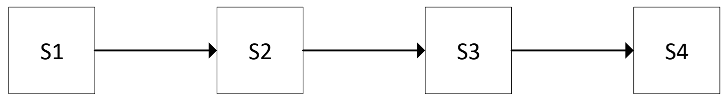


Figure 2-4 State sequence network

State Equipment Network

The State Equipment Network (SEN) (Smith and Pantelides, 1995) represents the batch recipe by making use of a process flowsheet and depicting all states entering the equipment as needed. This method of recipe representation makes it distinctly clear which units are able to accept material and perform certain tasks, as shown in Figure 2-5. Nevertheless, the SEN representation has not received as much attention as other representations and may become cumbersome when large plants are considered.

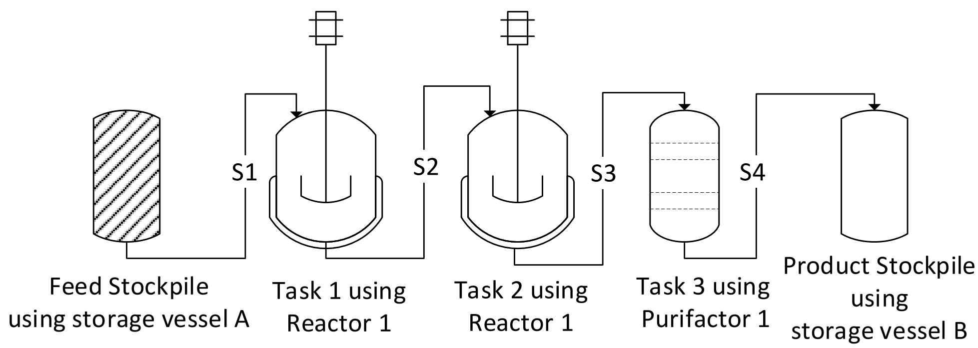


Figure 2-5 Simple SEN recipe representation

2.3 Operational Philosophies

Batch processing units are not continuously active during the time horizon of interest and temporary storage of material is often required to facilitate plant production. The use of storage vessels makes it possible to overcome bottlenecks and buffer the batch production line (Merkert *et al.*, 2015). However, the decisions regarding the use of storage and material transfers are known as the operational philosophies, as detailed below (Pattinson and Majozi, 2010).

The *No Intermediate Storage* (NIS) philosophy states that temporary storage vessels are not available for use before material is sent to the next step in the recipe. This philosophy is usually applied when plant space is limited. Therefore, units retain their material until another unit is free to accept the material. Producing units will only be available for processing once their contents have been removed.

The *Finite Intermediate Storage* (FIS) philosophy is applied when limited storage capacity is available. This philosophy allows produced material to be temporarily stored in storage vessels, yet the storage capacity is not guaranteed. The producing units are, therefore, available to accept fresh material after transferring their contents to storage.

Conversely, the *Unlimited Intermediate Storage* (UIS) philosophy is applied when an unlimited amount of material may be stored. Therefore, storage is always guaranteed and a unit can always distribute its material.

The *Mixed Intermediate Storage* (MIS) philosophy is a hybrid of the aforementioned philosophies. The philosophies are applied as needed to a batch plant depending on the plant requirements and abilities (Wiede and Reklaitis, 1987).

When a limited number of storage vessels are available to serve the entire plant, the *Common Intermediate Storage* (CIS) philosophy may be applied. The philosophy allows intermediate states to share common storage vessels for limited periods of time. However, material contamination is possible if thorough cleaning in place is not available for incompatible material. (Ku and Karimi, 1990; Jung *et al.,* 1996).

The *Process Intermediate Storage* (PIS) philosophy allows processing units to act as temporary storage vessels. Some processing units may be idle for periods of time during batch operation. The idle units may, therefore, be repurposed to temporarily serve as storage vessels for plant material from other units in order to aid with eliminating bottlenecks in the process. This philosophy is useful in cases where plant space or capital is extremely limited (Pattinson and Majozi, 2010).

The susceptibility of material to degrade must in some instances also be accounted for. Commonly, if material is susceptible to degradation, the *Zero-Wait* (ZW) philosophy is implemented. Produced material is immediately transferred from a unit when processing is completed. However, this limits the flexibility of the plant since the accepting vessel is required to be available immediately once the material is produced. Conversely, if the material is very stable, the *Unlimited Wait* (UW) philosophy may be implemented, whereby the material is allowed to remain within the processing units for extended periods of time. If the material degrades after some time, then the *Finite Wait* (FW) philosophy may be implemented so as to allow the materials to remain in the processing units for limited periods of time.

The decision of which operational philosophies to employ, depends on the process nature and availability of equipment. Mathematical formulations are generally adaptable to accommodate various philosophies. However, the choice in operating philosophy not only affects the manner in which storage vessels are utilized, but also how time is interpreted.

2.4 Time Representation

The manner in which the batch plant is scheduled is directly affected by the relevant philosophy, hence, the nature of time and scheduling is crucial to consider. Batch operation is fundamentally different to that of continuous operation as shown in Figure 2-6. Contrary to continuous processes, batch processes are active for short periods of time. Time is, therefore, the critical factor for efficient batch operation. Hence, the manner in which time is represented when scheduling batch plants requires careful consideration.

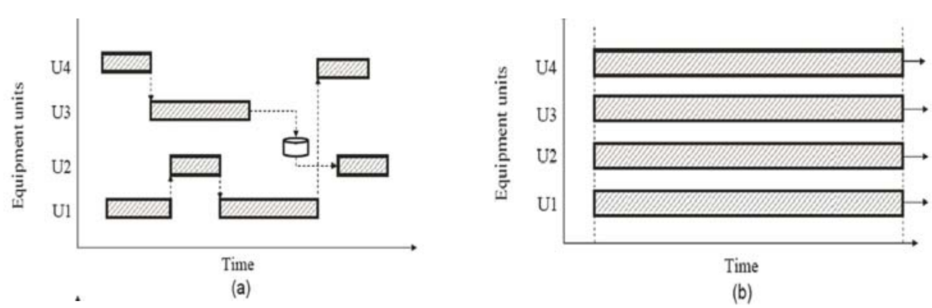


Figure 2-6 (a) Batch mode processing versus (b) Continuous mode processing over a given time horizon (Majozi, 2010)

For batch operation, time can either be represented as evenly discretized or as continuous (uneven discretization). The time over which all processing for a batch cycle must occur, is known as the time horizon of interest.

Discrete-time representation

With discretized time, the time horizon is represented by a series of equal and finite periods which are uniformly spaced. When applying the discrete-time representation, a task may only commence or end at the boundary of a prespecified and fixed period in time (Floudas and Lin, 2004). In batch production, plants rarely operate based on fixed time slots due to varying sizes of batches. Hence, the use of discretized time leads to sub-optimal results.

In mathematical formulations, binary variables are employed to determine whether tasks occur during each time period. The accuracy of discrete time formulations may be increased by increasing the number of intervals within the time horizon. However, by doing so, the binary dimension of the problem becomes explosive. A large number of unnecessary intervals will be present for which binary variables must be assigned.

Continuous-time representation

Zhang and Sargent (1994) and Zhang (1995) proposed the continuous time representation for batch plant scheduling. The continuous-time representation makes uses of time periods of different lengths spread over the duration of the time horizon. Contrary to discretized time representations, the number of time periods and the length of each time period is unknown prior to solving the problem. An iterative procedure is generally required in order to find the optimal number of time periods.

Nevertheless, by using the continuous-time representation, it is possible to eliminate unnecessary binary variables and improve the accuracy of the model to find a truly optimal solution (Floudas and Lin, 2004). The distinction between the time representations is shown in Figure 2-7.

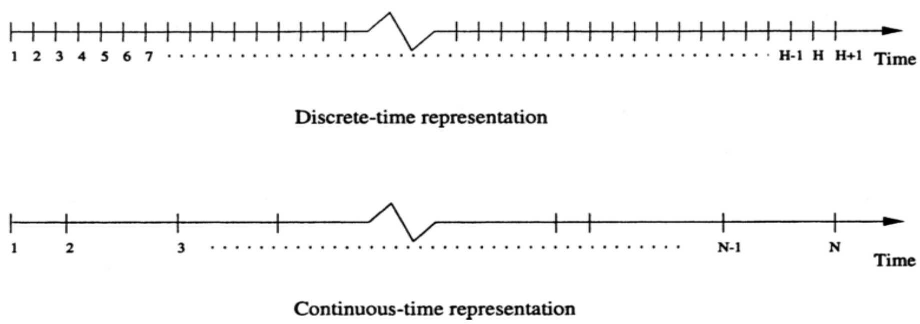


Figure 2-7 Discrete-time and continuous-time representations (Floudas and Lin, 2004)

Additionally, the continuous time models can be classified as slot based, event based or precedence based. Méndez *et al.* (2006), Floudas and Lin (2004) and Shaik *et al.* (2006) gave comprehensive reviews on the use of these models and the manner in which time is treated.

Time slots and time points may be used in continuous time formulations in order to represent specific parts of the time horizon when taking a slot based approach. Pinto and Grossmann (1995) employed varying duration time slots for units to represent the process sequencing. The number of time slots for each processing unit is assumed prior to solving the problem, however, the duration of each time slot is unknown *a priori*. The number of time slots to assign to each unit requires careful consideration.

Time points, as introduced by Schilling and Pantelides (1996), assign a point to a specific time on the time horizon with a time period being the duration between two successive time points. Each time point may represent the beginning or end of a task.

Notably, some models, such as that of Pinto and Grossmann (1994) and Liu and Karimi (2008), allow tasks to start and finish within the same slot, while other models, such as that of Sahinidis and Grossmann (1991), Schilling and Pantelides (1996) as well as Erdirik-Dogan and Grossmann (2008) allow the tasks to continue into the next slots.

Subtly different, event points as used by Ierapetritou and Floudas (1998), assign points on the time horizon that correspond to particular events when applying an event based approach. The duration of unit activity is the time period between two relevant event points. The main distinction between time points and event points, therefore, comes in during the manner in which they are used in the sequence and duration constraints when scheduling batch operations.

Event based models may use a global event time that is common to all the units or an event time that is specific to units. In global event time models, all time periods are shared between the units. Conversely, in unit specific event time models, each unit has a separate event time associated with it, allowing various units to have the same event point at different times. Where the event time is specific to units, less event points are required in total and therefore perform better than models that use an event time that is common to all units (Shaik and Floudas, 2008).

Precedence based approaches require that the relationship between various tasks be exploited so as to schedule unit activity. By employing known sequences within a recipe, it is possible to schedule the tasks in such a manner that they follow the recipe sequence.

By noting the characteristics of discrete-time representations and continuous time representations it is clear that discrete-time representations do not deliver truly optimal solutions. Therefore, the main deficiencies of the discrete-time representations may be summarized as follows.

i) The time horizon is divided into fixed, even intervals to be used as representation for unit activity.
ii) In order to yield an accurate result, a large number of intervals and binary variables are required, which in turn increase the size of the mathematical model.
iii) Variable batch sizes with varying processing times are not accommodated for using discrete intervals.

These shortcomings of discretized time to deliver optimal results is addressed when employing the continuous time representation.

2.5 Scheduling Methods

The order and extent to which batch units are operated impact greatly on the efficiency as well as the profitability of the plant. The aforementioned concepts are related directly to the scheduling of batch operations. Therefore, a number of scheduling techniques have been proposed to determine the optimal sequencing and material allocation within batch plants. The main aim of batch scheduling is to use limited resources in such a manner that some economic objective is met. The objective is usually either that of profit maximization or makespan minimization. When maximizing profit, the time horizon of interest is fixed and the throughput within that time horizon is maximized. Conversely, when aiming to minimize the makespan, the throughput is fixed and the batch production time is minimized. This dissertation focuses on the commonly implemented short-term scheduling of batch plants where it is desired to maximize plant profit over a given time horizon.

The short-term scheduling of batch plants requires efficient rescheduling techniques, since rapid changes within the market, new environmental regulations as well as loss of equipment require that a plant be readily rescheduled to compensate.

Techniques that have commonly been employed to schedule batch plants include deterministic, stochastic, heuristic, metaheuristic as well graphical methods. Deterministic approaches deliver what may be considered to be the best possible, or optimal, solutions to problems with given data under various operating conditions. (Fumero *et al.,* 2015; Corsano and Montagna, 2011).

Sparrow *et al.* (1975) presented an initial formulation to address the general scheduling problem of multiproduct batch plants by considering both heuristic and deterministic branch and bound techniques. The heuristic method delivered a quick, yet sub-optimal solution, whereas the branch and bound technique attempted to find a solution to the resulting Mixed Integer Nonlinear Program (MINLP) problem. Global optimality for the solution could not be guaranteed by virtue of the nonlinear nature of the problem.

As mathematical optimization techniques gained more interest, more models that were applicable to batch scheduling were presented. Notably, a Mixed Integer Linear Program (MILP) formulation for the short term scheduling of multipurpose batch plants was presented by Kondili *et al.* (1993). However, due to the discretized time representation employed and the explicit assignment of states to units, this formulation required a large number of binary variables in order to yield an accurate result. Shah *et al.* (1993) addressed computational issues associated with the model of Kondili *et al.* (1993) and presented a compact formulation with a simpler set of reformulated constraints and solution techniques, yielding a MILP model with a smaller integrality gap. The reformulated constraints allowed for a tightened Linear Program (LP) relaxation problem and the solution techniques employed allowed the reduction of the relaxed LP size, thereby reducing the computational complexity. Nevertheless, the discretized nature of the proposed models by Kondili *et al.* (1993) and Shah *et al.* (1993) inherently led to suboptimal results, hence the continuous time representation became popular in order to yield optimal results.

The aforementioned techniques have been applied to various pertinent scheduling models, each of which exploit particular factors of the batch plant while aiming to continuously increase batch profitability with improved mathematical formulations. Pinto and Grossmann (1995) presented a model that was able to meet intermediate orders during the time horizon while employing cyclic scheduling. The MILP formulation by Schilling and Pantelides (1996) resulted in a large integrality gap and a novel branch and bound algorithm was proposed that decreases the integrality gap by branching off binary and continuous variables. A large integrality gap is undesirable since it increases the computational effort required to solve a give problem. An extended MINLP mathematical formulation based on a continuous time representation was presented by Zhang and Sargent (1996) in order to determine the optimal operating conditions of batch plant such that continuous delivery is met.

Ierapetritou and Floudas (1998) presented a model with continuous time scheduling wherein the number of binary variables associated with event points was decreased by noting additional instances where it is possible to eliminate some binary variables completely. These instances mainly related to the sequencing of tasks over the time horizon and noted the precedence relationship between tasks. However, a large number of continuous variables was required when considering complex problems. Nevertheless, a vastly improved solution time was realized as compared to those of previous models due to the reduced number of binary variables and use of the continuous time representation.

Sundaramoorthy and Karimi (2005) delivered a global event time MILP formulation that was free of Big M constraints. It also required fewer binary variables compared to other formulations that decouple tasks and units in order to improve the formulation (Giannelos and Georgiadis, 2002; Ierapetritou and Floudas, 1998; Maravelias and Grossmann, 2003). Additionally, the formulation allowed tasks to continue over multiple time slots. Susarla *et al.* (2009) extended the work of Sundaramoorthy and Karimi (2005) by using unit specific slots and allowing non-simultaneous transfers to and from units while allowing tasks to span multiple event points. This allowed for improved profitability while decreasing the number of event points required.

Shaik and Floudas (2009) delivered a MILP formulation that is based on a three index binary variable which allowed tasks to continue over multiple event points. The formulation required that an iterative procedure be used to determine both number of event points and the number of event points the tasks span. Vooradi and Shaik (2012) improved on the formulation of Shaik and Floudas (2009) in order to decrease model computational complexity. They incorporated the concept of active task into the formulation, which reduces the number of binary variables required for task monitoring. The active task binary variable monitors over which event points a task is active instead of assigning specific variables to monitor start, processing and end times.

In order to predict the optimal number of required time points, Li and Floudas (2010) presented frameworks that incorporated determining the minimum and maximum number of time points and then applying a branch and bound technique in order to find the optimal number. Seid and Majozi (2012a) also presented a method to determine the optimal number of time points and relied on identifying the critical unit and the number of times the critical unit will be used. Their prediction correlated strongly with the actual number of time points required.

The SSN representation was further employed by Seid and Majozi (2012b) in an unit specific event time formulation which addressed crucial omissions when scheduling batch plants requiring the FIS policy. The FIS policy was violated in models from previous publications evaluated by Seid and Majozi (2012b) when these models exhibited UIS characteristics during the final schedule and were, therefore, found to be suboptimal. The model proposed by Seid and Majozi (2012b) also required fewer event points by basing the continuous representation on unit specific slots instead of process specific slots. This allowed for improved solving performance and longer batch time horizons to be evaluated, due to a decrease in the number of required binary variables.

Recently, Biondi *et al.* (2017) included new aspects such as that of maintenance within the scheduling problem and allowed for effects of plant degradation on the plant schedule. They noted that processing units have a limited lifetime and require intermittent maintenance to be incorporated into the scheduling formulation as a result. However, these inclusions may only be required sparingly and add to the computationally complexity of the problem due to the additional plant considerations required.

Lagzi *et al.* (2017) proposed a MILP formulation that includes multitasking aspects for machines in processing facilities. This inclusion allows for improved throughput and facility profitability. However, batch chemical units perform single tasks at a given time and the formulation is, therefore, limited in its capabilities within these batch plants.

Shaik and Vooradi (2017) presented a unit specific MILP formulation that is capable of scheduling batch plants using fewer event points compared to previous formulations. Notably, the formulation allows material production and consumption to occur during the same unit event and further allows the first unit event to be associated with the first event slot. Previous continuous time formulations required downstream unit events to follow on upstream unit events. However, the proposed formulation is limited to cases without recycle streams. A hybrid approach is proposed to address cases with recycles. Additionally, the formulation only allows material to wait in units during the final event. These limitations exclude a large number of problems to be considered. Nevertheless, the proposed formulation was capable of decreasing the problem size compared to other formulations.

The aforementioned formulations relied on mathematical modelling in order to solve the scheduling problems, however, the S-graph is a graphical method introduced by Sanmartí *et al.* (1998) to solve the scheduling problem for multipurpose plants. The S-graph method is applicable only when employing the NIS or UIS philosophies, since the method schedules units and not storage. Sanmartí *et al.* (1998) aimed to deliver a feasible schedule such that the makespan is minimized. Notably, the S-graph does not require prediction of presupposition of time points, which renders it a truly continuous time technique.

However, the S-graph method assumes that tasks only occur once. Therefore, it is not possible to freely arrange tasks multiple times. Additionally, when multipurpose units are employed, the user is required to pre-specify task-unit assignments. The FIS policy is not supported by the S-graph method and the method, therefore, excludes a number of plant configurations and impacts on profitability. Furthermore, the S-graph technique is not suitable for problems wherein processing has variable duration. This graphical technique is, therefore, limited in its applicability to many scenarios.

Adonyi *et al.* (2003) gave an extensive review of the applicability of the S-graph method to specific cases for use in the scheduling and heat integration within a batch plant. Majozi and Friedler (2006) extended the S-graph method to maximize throughput over a given time horizon. A guided algorithm was presented to find the guaranteed global optimum. The proposed algorithm is applicable only for the NIS policy and when the batch sizes are fixed.

Hegyháti *et al.* (2009) noted that many models overlooked the subtle transfer infeasibility that arose when various operational philosophies were employed. More specifically, it was noted that mathematical models allowed material to be transferred to and from a processing unit at the same time, so called cross-transfer. Hegyháti *et al.* (2009) further demonstrated that the S-graph approach was not susceptible to this infeasibility. However, due to the added dimension of time, graphical techniques are not easily applied to the scheduling of batch plants, especially when various operational philosophies are applied

Both the mathematical and graphical techniques aimed to deliver realistic and improved formulations for the scheduling of batch plants, yet have omitted the impact of available pumping equipment on the schedule. In industrial setups, batch plants are dependent on the rating and availability of pumping equipment. It is therefore crucial to include the chosen pumping equipment as an integral part of the scheduling formulation.

2.6 Synthesis

Batch plant scheduling has been extended and employed rigorously to aid in the synthesis of batch plants (Barbosa-Póvoa, 2007). The optimal synthesis of batch plants entails the selection of the plant configuration from all the available equipment such that the overall plant profit is maximized. Equipment contribute both to capital and maintenance cost, therefore, only profitable equipment should be employed. Furthermore, most batch plants have limited space available, hence, it is ideal to only use the necessary equipment.

The optimum design of batch plants is usually based on a superstructure, which is a superset of feasible combinations for a particular recipe. For a particular processing network, the plant superstructure is all the possible configurations within that network and encompasses all available processing units, storage vessels, piping connections and auxiliary equipment. The optimal combination of these equipment is a subset of the entire superstructure for that network. Hence, the synthesis of batch plants aims to identify this optimal structure within a given superstructure.

Batch plant synthesis may be performed during the grassroot design phase, wherein it is aimed to keep the initial investment and capital costs to a minimum while achieving the best foreseeable profit. Additionally, the synthesis of the batch plant may be performed in retrofit design such that unnecessary equipment can be removed from the plant. By eliminating equipment, savings on maintenance costs is possible. Therefore, the synthesis of batch plants is crucial to plant profitability when considering the costly major equipment such as reactors and separators and the cost contributing auxiliary equipment such as pumping equipment.

Barbosa-Póvoa (2007) gavr a comprehensive review of numerous models that address the design and retrofit of batch plants. In the review, different literature models are tabulated and their characteristics are highlighted. A detailed, chronological order of grassroots design cases is presented for both the basic case design, where only scheduling and sizing is considered, as well as for the extended design of batch plants which includes aspects such as topology and operational characteristics.

2.6.1 Decision Variables

The feature of flexible production that batch plants offer is also what introduces the extreme complexity involved when solving synthesis problems for such plants. As more degrees of flexibility are added to the problem, such as allowing variable pumping transfers or heat integration to be considered along with plant synthesis and scheduling problem, a larger number of variables and possible plant configurations need to be considered. This makes batch plant optimization combinatorically complex (Seid and Majozi, 2015).

When solving batch synthesis and scheduling problems using deterministic models, a number of decisions need to be made. These decisions are incorporated in the models by use of binary variables (Merkert *et al.*, 2015). Hence, most batch synthesis and scheduling formulations have mixed integer constraints. The binary variables are introduced to make allowance for operational and topological choices, whereas continuous variables are used to determine plant properties such as equipment capacities, heat transfers and material usage (Pinto *et al.*, 2003).

With both discrete and continuous time representations, binary variables are required to decide whether a task is active at a particular point in time. Hence, with discretized time, where there is a significant amount of time points present, the number of binary variables is large compared to that of continuous time. Other binary variables may be used to make decisions regarding storage vessel, processing unit, piping or pump existence.

Additionally, the variables enable decisions regarding whether pumping or processing tasks will take place at specific times. Therefore, the synthesis of batch plants may becomes extremely combinatorically complex and binary variables may negatively impact on the solution time of models. Hence only binary variables that are essential should be included in the formulation (Pattinson and Majozi, 2010).

2.6.2 Pertinent Work in Batch Plant Synthesis

Much work has been done on the synthesis of batch plants over the past few decades and the most pertinent are discussed here. Grossmann and Sargent (1979) delivered an initial MINLP model to design sequential multiproduct batch processes.

They further demonstrated that the relaxed subproblem can be solved by treating all variables as continuous. Suhami and Mah (1982) elaborated on the work of Sparrow *et al.* (1975) and Grossmann and Sargent (1979) by presenting a heuristic method to simultaneously schedule and design multipurpose batch plants. A number of configurations were generated, from which the most profitable was chosen as the optimal, however, global optimality could not be guaranteed (Méndez et al., 2006).

Modi and Karimi (1989) presented a MINLP formulation as well as a heuristic procedure for the design of batch processes which included considerations for intermediate storage vessels and semi continuous operations. Again, global optimality could not be guaranteed, however, the procedures were shown to provide close to optimal results. Papageorgaki and Reklaitis (1990), addressed the plant design problem and discussed equipment selection such that a flexible unit-to-task allocation is accounted for. They developed a MINLP set of equations which allowed plant operation to be studied over a number of batch campaigns. The formulation allowed the study of optimal allocation of products to campaigns as well as the optimal duration of each campaign.

Voudouris and Grossmann (1992) reformulated previous MINLP models and replaced continuous equipment sizes with discrete sizes in order to yield a MILP formulation that could be solved with minimal computational effort. The reformulations were included for both multiproduct and multipurpose batch plants.

Shah and Pantelides (1992) noted that the production requirement is generally fixed during the design stage and that a degree of uncertainty may need to be incorporated into the batch design to allow future production requirements. Therefore, they developed a design procedure for multipurpose batch plants which considered variety of operating conditions to meet varying annual production requirements. However, the solutions may not be feasible for large designs or where many different operating conditions are required.

Using an effective continuous time formulation, Lin and Floudas (2001) proposed a formulation for the integrated design, synthesis and scheduling of multipurpose batch plants. They explicitly included sequencing constraints for cases when the same task occurs in the same unit, when different tasks occur in the same unit and when different tasks occur in different units. These interactions are of key importance when dealing with the flexible nature of multipurpose batch facilities. These constraints are captured in various formulation using different forms.

Pinto *et al.* (2008) evaluated a number of models that employed the STN, mSTN and RTN recipe representations so as to determine their comparative effectiveness for the design and synthesis of multipurpose batch plants. They concluded that the mSTN representation performs best. However, the SSN representation, which has since played a significant role in batch plant modelling and optimization, did not form part of their evaluation.

Quaglia *et al.* (2013) presented a stochastic synthesis, design and optimization framework that makes use of Monte Carlo sampling in order to assess systems under uncertainty. The framework guides the user on its use and is capable of being applied to large industrial case studies in order to meet varying production requirements.

The aforementioned formulations largely omitted the necessary piping connection that are required to facilitate material transfer. Barbosa-Póvoa and Macchietto (1994) delivered a detailed design of multipurpose batch plants. They presented a MILP formulation which included plant topology based on a recipe representation adapted from STN, so called maximal-STN (mSTN) in order to depict all possible transfers. The model evaluated the sizing of the units as well as the capital costs and operational costs associated with major equipment and piping.

Barbosa-Póvoa and Macchietto (1994) stated that piping connections must be present between vessels in order to facilitate transfer of material. Therefore, their scheduling formulation included piping availability. Seid and Majozi (2013) presented a MILP mathematical formulation for the scheduling and design of multipurpose batch plants while considering aspects such as piping topology. The model also made provision for the correct scheduling when the FIS policy is observed and included pipe sizing and costing in the overall formulation.

The number of piping connections in a multipurpose batch plant may be large due to the fact that a large number of possibilities exist for material allocation. Therefore, the concept of pipeless batch plants was proposed which allow material to be transferred directly between units by use of transferable vessels and thereby eliminating the need for some pipes and pumps (Niwa, 1993).

Patsiatzis *et al.* (2005) gave a review of the applicability of pipeless batch plants and detailed the various layouts that can be employed in order to achieve a pipeless batch plant. However, pipeless plant layouts require that vessels and processing stations be carefully spaced in both the horizontal and vertical directions. This eliminates the flexibility of adding or removing vessels for different tasks without compromising the production layout. Multipurpose batch plants that rely on a pumping network are, therefore, more flexibly adaptable when it is required to alter the plant layout or product recipes.

Along with the piping, other factors that affect the cost of the overall plant need to be considered. Loonkar and Robinson (1970) noted that the capital cost of a batch plant is also dependent on the rating of pumping equipment, heat exchanger duty and the size of the processing vessels. A general procedure for sizing interdependent equipment was also given.

Additionally, the rating of chosen pumping equipment will affect the transfer time between units. Hegyhati and Friedler (2011) found that material transfers were generally simplified to fixed durations when scheduling plant production, regardless of the batch size and costs associated with the transfers.

Hence, they proposed discrete transfer times and associated energy costs based on the pumping requirement. Lee *et al.* (2015) considered fixed material transfer times in their work in order to allow for heat integration during the transfer. Each transfer time was a fixed parameter which was independent of the amount of material transferred.

Majozi *et al.* (2015) gave a detailed insight into the design and synthesis of batch plants. Various aspects, including that of scheduling, topology consideration, energy minimization as well as waste water generation are included. A number of case studies are given where various models are applicable. However, it is clear that previous formulations did not include pumping equipment and their respective ratings in the overall synthesis problems.

2.7 Mathematical Optimization

The aforementioned scheduling and synthesis formulations mainly employ mathematical optimization techniques. Therefore, the theory behind these techniques is discussed here. Engineering practice and research aim to find the best feasible solution to a given problem. Biegler and Grossmann (2004) and Grossmann and Biegler (2004) gave extensive reviews on the use of mathematical optimization techniques in the fields of chemical and process engineering. They noted that much of the advancement in the field of general optimization theory stem from research in these fields. They further discussed the different optimization methodologies applied to batch production.

Figure 2-8 depicts the classes of mathematical optimization problems encountered in batch processing. Mathematical optimization problems may be discrete, wherein the variables have finite values; continuous, where the variables have values within a range; or a combination of discrete and continuous. However, in order to find a solution to any of the optimization problems, at least one objective function for the problem must be present for evaluation.

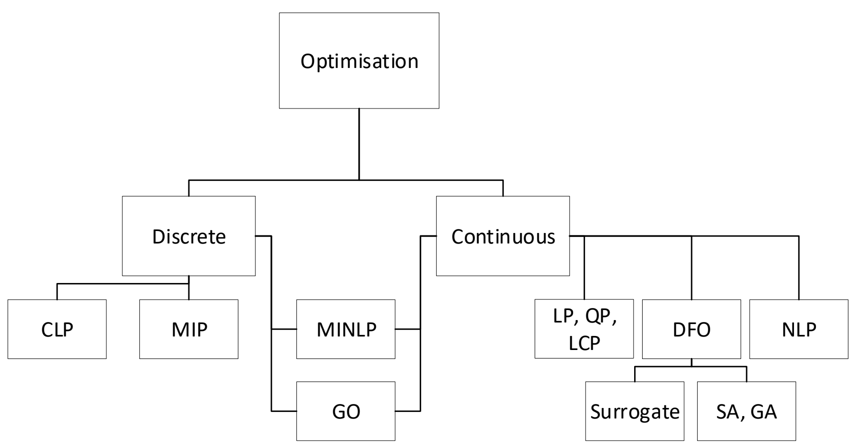


Figure 2-8 Classes of optimization problems (Biegler and Grossmann 2004)

Along with the objective function, a number of constraints are present in most mathematical optimization models. These constraints consist of inequalities and equalities which define the problem to be solved. The problem is generally of the form (Williams, 2013):

Objective:

$$minimize\ g(x, y)$$

Subject to:

$$f(x, y) = 0$$

$$h(x, y) \leq 0$$

$$x \in \mathbb{R}, y \in [0,1]$$

Hence, the variables are evaluated such that the objective $g(x, y)$ is at a minimum and has a feasible solution. The final values of the variables must satisfy all the constraints in order to yield a feasible solution.

Combined, the constraints bound a region of feasible solutions. However, the subset of feasible solutions that finds the best solution to the objective is desired (Edgar *et al.*, 2001).

Integer programming

Problems that contain variables that only take on discrete values are subject to integer programming (IP) optimization and may be defined as follows.

Objective:

$$minimize\ g(x)$$

Subject to:

$$f(x) = 0$$

$$h(x) \leq 0, \qquad x \in Z$$

Other types of problems that may be discretely optimized include that of conjunctive linear programming (CLP) and MILP formulations.

Linear Programming

Similarly, optimization problems that contain variables that only take on continuous or real values, are subject to linear programming and may be defined as follows.

Objective:

$$minimize\ g(y)$$

Subject to:

$$f(y) = 0$$

$$h(y) \leq 0,\ y \in R$$

When the product of two or more continuous variables or expressions involving powers and logarithms are present, a nonlinear programming (NLP) problem is formed.

Mixed Integer Optimization

Since a large number of problems contain both discrete and continuous variables, it is possible to define such optimization problems as follows.

Objective:

$$minimize\ g(x, y)$$

Subject to:

$$f(x, y) = 0$$

$$h(x, y) \leq 0,\ x \in Z, y \in R$$

For cases where the discrete variable is binary, the values of such a variable are limited such that $x \in [0,1]$. In the case where all the constraints are linear, the resulting model is a MILP. However, when nonlinear expressions are present in the formulation, it becomes MINLP. In this dissertation, a MINLP model is mathematically transformed into a MILP model. Therefore, focus is placed on the Branch and Bound Technique for solving this type of problem.

2.7.1 Branch and Bound Technique

A standard method of solving LP problems is by employing the simplex method. Although first introduced in the 1940s, it has since been adapted to become quite advanced and able to solve heavily constrained problems (Grossmann and Biegler, 2004). The simplex algorithm consists of two phases. The first phase finds a basic, yet feasible solution. If a solution does not exist, meaning that the constraints are ineffective at describing the problem, then no solution exists. If a basic solution does exist, it is used as a starting point and used to find a solution such that the objective is minimized. The set of constraints for the problem define a polytope which defines the feasible region. The simplex method used the initial solution and moves along the edges of the defined polytope. This continues until a vertex is reached that is the optimum solution. No final solution can be given should the problem be unbounded (Edgar *et al.*, 2001).

To solve most MILP problems, the simplex LP branch and bound method is generally employed. The technique is based on the approach proposed by Land and Doig (1960) for discrete programming. Tree enumeration is used wherein the integer space is relaxed into LP subproblems and solved at each node in the tree.

For a minimization problem, the lower bound to the solution is found by treating the variables as continuous and solving it as an LP problem and it is, therefore, termed to be relaxed. Various branching rules are then applied if the discrete variables are fractional.

At each branching node, the relaxed LP solution is the lower bound for all descendant nodes. Therefore, whenever a feasible integer solution is present, an upper bound to the problem is found. Branching continues until the lower and upper bounds are within a specified tolerance to each other.

Branching may take place either through a depth-first or a breadth-first approach. The depth-first approach requires branching to continue from the start node through to the end node on a particular branch, before backtracking and continuing with the next branch (Ryoo and Sahinidis, 1996). This approach is illustrated in Figure 2-9.

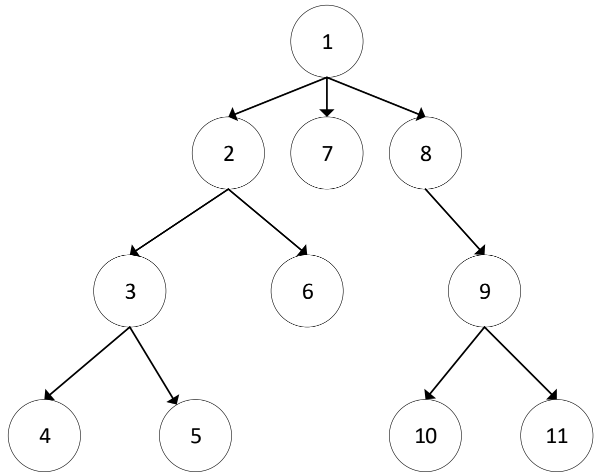


Figure 2-9 Depth-first approach

Conversely, the breadth-first approach requires branching to first follow the branch that has the node with the best solution and then expanding on each of the successor nodes, as illustrated in Figure 2-10. Most branch and bound solution methods make use of breadth first enumeration (Ryoo and Sahinidis, 1996).

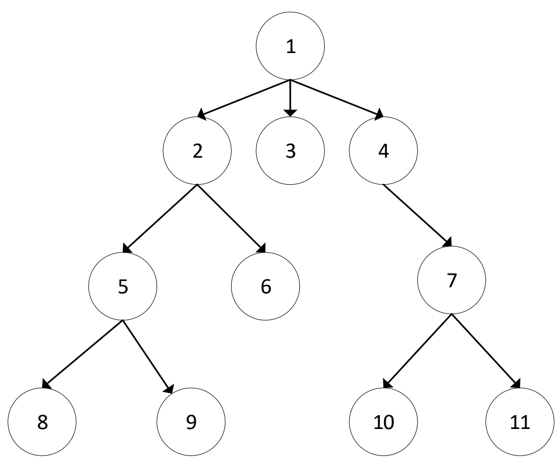


Figure 2-10 Breadth-first approach

The main feature of the branch and bound method is the ability to eliminate inferior or infeasible branches from the tree enumeration procedure, reducing the search space for the global optimum. If at any node the lower bound is no better than the upper bound or if the solution is not feasible, the entire branch is deleted (Ryoo and Sahinidis, 1996). This is illustrated in Figure 2-11.

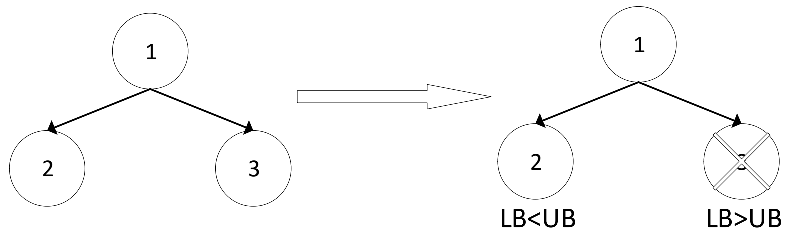


Figure 2-11 Branch elimination in branch and bound

2.7.2 Convexity

The nature of the model determines whether the best solution, i.e. the global optimal, can be found. If models are convex, global optima can be guaranteed, whereas in nonconvex models, the solutions found may only be a number of local optima.

Regions are convex if a continuous line can arbitrarily be drawn between any two points within the feasible solution region, as shown in Figure 2-12 (a). In a nonconvex region, it is not possible to directly join two arbitrary points of the region, as shown in Figure 2-12 (b) (Williams, 2013).

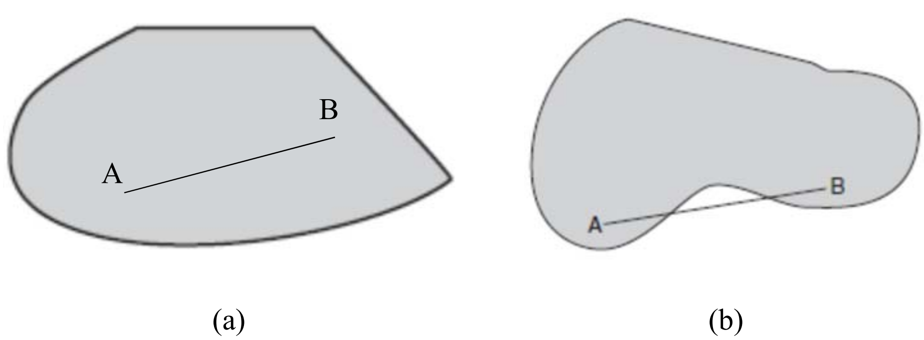
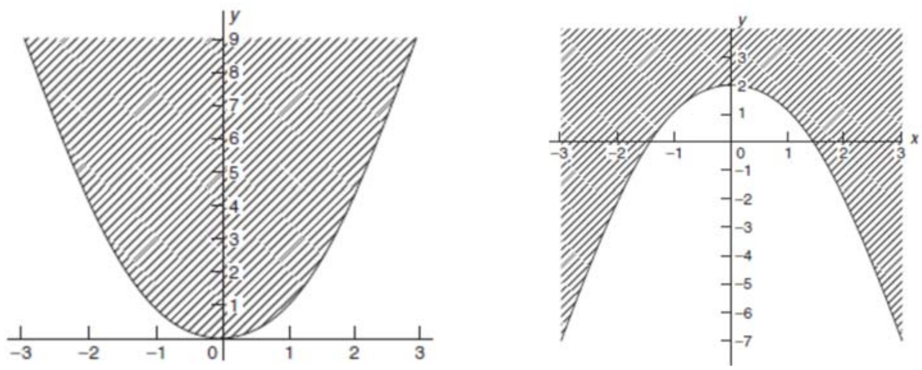


Figure 2-12 (a) A convex region and, (b) a nonconvex region (Williams, 2013)

Similarly, a function is convex if the values that lie above its curve form a convex region, as shown in Figure 2-13 (a). Conversely, a function is nonconvex if all the values that lie above its curve form a nonconvex region, as shown in Figure 2-13 (b).

If the region in Figure 2-13 (a) could be represented by an arbitrary constraint (2.1), then the function may also be classified as strictly convex, wherein the inclusive inequality (≤) is replaced with an exclusive inequality (<). Hence, a strictly convex function has a single optimum solution, and not a set of feasible optimal solutions along the periphery of the curve (Williams, 2013).

$$g(x, y) \leq h \tag{2.1}$$



(a) (b)

Figure 2-13 (a) A function forming a convex region and, (b) a function forming a nonconvex region (Williams, 2013)

Therefore, in order to yield a convex model, all the functions that comprise the model must be convex. The presence of any nonconvex functions within the model will result in the entire model being nonconvex.

In batch scheduling and optimization, the mathematical models may consist of LP, NLP, MILP and MINLP models. NLP and MINLP models are generally nonconvex due to their nonlinear nature, with a limited number of exceptions (Biegler and Grossmann, 2004). A number of convexification methods exist in order to convexify some nonlinear functions in order to facilitate solving and to aid finding a global optimum solution.

These convexification methods may be exact or inexact. When the method is exact, the relevant constraints are transformed into a different form, which when applied, will yield the same result as when no convexification occurred. Inexact methods rely on approximating the original constraints into a form that will yield similar, yet slightly different results. Therefore, it is ideal to only apply exact convexification methods, since there will be no loss in accuracy of the result.

Convexification methods such as that of McCormick (1976) over and under estimators, reformulation-linearization techniques (RLT), piecewise linearization, and Glover (1975) transformations have received interest due to their applicability. The Glover (1975) transformation is the only exact convexification method mentioned, and is directly applicable to the proposed model in this dissertation.

2.7.3 Glover (1975) Transformation

If a nonlinearity is present due to the product of a binary variable and a continuous variable, a Glover (1975) transformation can be applied to yield a linear model without any loss in accuracy. This bilinear term makes the model to be solved a MINLP problem. Therefore, a global optimal solution can be guaranteed. The technique proposed by Glover (1975) stipulates that a given continuous variable w_i is to replace the bilinear term, $x_i d_{ij}$ as defined in constraint (2.2), such that constraints (2.3) and (2.4) hold.

$$\begin{gathered} w_i = x_i d_{ij} \\ \forall x_i \in [0,1], d_{ij} \in R \end{gathered} \tag{2.2}$$

Such that:

$$w_i = 0 \qquad if\ x_i = 0 \tag{2.3}$$

$$w_i = d_{ij} \qquad if\ x_i = 1 \tag{2.4}$$

Constraints (2.3) and (2.4) can be satisfied by employing the transformations shown in constraints (2.5) and (2.6), where it is noted that d_{ij} has upper bound, d^U, and lower bound, d^L.

$$d^l x_i \leq w_i \leq d^U x_i \tag{2.5}$$

$$d_{ij} - d^U(1 - x_i) \leq w_i \leq d_{ij} - d^L(1 - x_i) \tag{2.6}$$

Both constraints (2.5) and (2.6) are linear in terms of their variables. In most instances, the upper and lower bounds for the continuous variable d_{ij} are known. Hence, these constraints combined may be used to replace constraint (2.2). Therefore, it is possible to exactly linearize the bilinear term and convexify a model of the abovementioned nature, at the expense of introducing a new continuous variable and four constraints for each bilinear term. Nevertheless, a global optimal solution can be guaranteed.

2.7.4 Alternative Optimization Techniques

A number of stochastic techniques exist which do not rely solely on the aforementioned mathematical optimization. These techniques are suitable for studies requiring quick solutions without investing much effort and are most effective in problems that are difficult to solve using deterministic mathematical techniques, such as nonconvex MINLP problems. However they do not guarantee global optimality (Biegler and Grossmann, 2004).

Classical direct search methods were the most popular methods for optimization in the field of chemical engineering during the 1960s and 1970s. Methods that are commonly placed within this group are that of the conjugate direction method, simplex and complex methods as well as the adaptive random search method. The best solution from a number of searched output solutions is chosen as the final solution. Modern computer hardware has allowed direct search methods to be reconsidered as a means to find close to optimal solutions to problems. Previously, computer hardware failed when a large number of variables were present, yet the improved computing power has since allowed this problem to be overcome (Biegler and Grossmann, 2004).

Derivative free optimization (DFO) techniques have also attempted to address complex optimization problems by using multiple computer processors. These techniques are not dependent on gradients or stationary points like most mathematical optimization techniques, allowing them to favor globally optimal solutions. However, these techniques require unconstrained problems with simple bounds in order to function. The methods tend to scale very poorly with an increased number of decision, or binary, variables.

Other techniques such as simulated annealing (SA) and genetic algorithms (GA) take analogies from natural phenomena in order to find the best solution to a problem. Simulated annealing draws its analogy from molecular motion during the cooling of metals, whereas genetic algorithms continuously modify generated gene pools of problems in order to improve the population of solutions. Genetic algorithms have received much attention and use in batch plant scheduling (Biegler and Grossmann, 2004).

2.8 Conclusions

From the literature reviewed, it is clear that a number of techniques have been implemented in order to optimally schedule and synthesize batch chemical plants. Many aspects of the flexibility of batch plants have been exploited and use of equipment that may perform a number of different tasks greatly add to the flexible production capabilities within the plant.

Most batch plants rely on a pumping network in order to transfer material between vessels. Nevertheless, previous work has given little consideration to the equipment responsible for these transfers and may, therefore, lead to inaccuracies when attempting to schedule batch plants in practice. The flexible nature of multipurpose batch plants allows varying quantities of material to be processed and transferred between vessels. These transfers may require different amounts of transfer times to complete. Previous work that considered material transfer time, generally included it as a fixed parameter, independent of the amount of material transferred.

Additionally, the discrete operation of tasks within a batch plant, allow the key feature of reducing the number of pumping equipment by reusing pumps throughout the plant. Previous work has largely omitted considerations relating to the selection of appropriate pumping equipment in order to optimally transfer material throughout a batch plant. By considering optimal pump allocation, it is possible to save on capital and maintenance costs as well as reduce the space occupied by plant equipment.

2.9 References

Adonyi, R., Romero, J., Puigjaner, L., Friedler, F., 2003. Incorporating heat integration in batch process scheduling. Appl. Therm. Eng. 23, 1743–1762.

Barbosa-Póvoa, A.P., 2007. A critical review on the design and retrofit of batch plants. Comput. Chem. Eng. 31, 833–855.

Barbosa-Póvoa, A.P., Macchietto, S., 1994. Detailed design of multipurpose batch plants. Comput. Chem. Eng. 18, 1013–1042.

Biegler, L.T., Grossmann, I.E., 2004. Retrospective on optimization. Comput. Chem. Eng. 28, 1169–1192.

Biondi, M., Sand, G., Harjunkoski, I., 2017. Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach. Comput. Chem. Eng. 99, 325–339.

Corsano, G., Montagna, J.M., 2011. Mathematical modeling for simultaneous design of plants and supply chain in the batch process industry. Comput. Chem. Eng. 35, 149–164.

Edgar, T.F., Himmelblau, D.M., Lasdon, L., 2001. Optimization of chemical process, Optimization of chemical processes.

Erdirik-Dogan, M., Grossmann, I.E., 2008. Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers. Ind. Eng. Chem. Res. 47, 1159–1163.

Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Comput. Chem. Eng. 28, 2109–2129.

Fumero, Y., Moreno, M.S., Corsano, G., Montagna, J.M., 2015. A multiproduct batch plant design model incorporating production planning and scheduling decisions under a multiperiod scenario. Appl. Math. Model.

Giannelos, N.F., Georgiadis, M.C., 2002. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. Ind. Eng. Chem. Res. 41, 2178–2184.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Manage. Sci. 22, 455–460.

Grossmann, I.E., Biegler, L.T., 2004. Part II. Future perspective on optimization. Comput. Chem. Eng. 28, 1193–1218.

Grossmann, I.E., Sargent, R.W.H., 1979. Optimum Design of Multipurpose Chemical Plants. AIChE J. 18, 343–348.

Hegyhati, M., Friedler, F., 2011. The role of energy consumption in batch process scheduling, in: 21st European Symposium on Computer Aided Process Engineering - ESCAPE 21. pp. 1979–1983.

Hegyháti, M., Majozi, T., Holczinger, T., Friedler, F., 2009. Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs MILP methods. Chem. Eng. Sci. 64, 605–610.

Ierapetritou, M., Floudas, C., 1998. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. Ind. Eng. Chem. Res. 37, 4341–4359.

Ierapetritou, M.G., Floudas, C.A., 1998. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. Ind. Eng. Chem. Res. 37, 4341–4359.

Jung, J.H., Lee, H.-K., Lee, I.-B., 1996. Completion times algorithm of multi-product batch processes for common intermediate storage policy (CIS) with nonzero transfer and set-up times. Comput. Chem. Eng. 20, 845–852.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—I. MILP formulation. Comput. Chem. Eng. 17, 211–227.

Ku, H.-M., Karimi, I., 1990. Completion time algorithms for serial multiproduct batch processes with shared storage. Comput. Chem. Eng. 14, 49–69.

Lagzi, S., Fukasawa, R., Ricardez-Sandoval, L., 2017. A multitasking continuous time formulation for short-term scheduling of operations in multipurpose plants. Comput. Chem. Eng. 97, 135–146.

Land, A.H., Doig, A.G., 1960. An Automatic Method of Solving Discrete Programming Problems. Econometrica 28, 497–520.

Lee, J., Seid, E.R., Majozi, T., 2015. Heat integration of intermittently available continuous streams in multipurpose batch plants. Comput. Chem. Eng. 74, 100–114.

Li, J., Floudas, C.A., 2010. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. Ind. Eng. Chem. Res. 49, 7446–7469.

Lin, X., Floudas, C., 2001. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. Comput. Chem. Eng. 25, 665–674.

Liu, Y., Karimi, I.A., 2008. Scheduling multistage batch plants with parallel units and no interstage storage. Comput. Chem. Eng. 32, 671–693.

Loonkar, Y.R., Robinson, J.D., 1970. Minimization of Capital Investment for Batch Processes Calculation of Optimum Equipment Sizes. Ind. Eng. Chem. Process Des. Dev. 9, 625–629.

Majozi, T., 2010. Batch Chemical Process Integration, Springer.

Majozi, T., Friedler, F., 2006. Maximization of throughput in a multipurpose batch plant under a fixed time horizon: S-graph approach. Ind. Eng. Chem. Res. 45, 6713–6720.

Majozi, T., Seid, E.R., Lee, J.-Y., 2015. Synthesis, Design, and Resource Optimization in Batch Chemical Plants, First. ed. CRC Press.

Majozi, T., Zhu, X.X., 2001. Novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling. Ind. Eng. Chem. Res. 40, 5935–5948.

Maravelias, C.T., Grossmann, I.E., 2003. New General Continuous-Time State−Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 42, 3056–3074.

McCormick, G.P., 1976. Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Math. Program. 10, 147–175.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913–946.

Merkert, L., Harjunkoski, I., Isaksson, A., Säynevirta, S., Saarela, A., Sand, G., 2015. Scheduling and energy – Industrial challenges and opportunities. Comput. Chem. Eng. 72, 183–198.

Modi, A.K., Karimi, I.A., 1989. Design of multiproduct batch processes with finite intermediate storage. Comput. Chem. Eng. 13, 127–139.

Niwa, T., 1993. Pipeless plants boost batch processing. Chem. Eng. (New York) 100, 102–108.

Pantelides, C., 1994. Unified frameworks for optimal process planning and scheduling, in: Foundations of Computer-Aided Process Operations. Cache Publications, New York.

Papageorgaki, S., Reklaitis, G. V., 1990. Optimal design of multipurpose batch plants. 1. Problem formulation. Ind. Eng. Chem. Res. 29, 2054–2062.

Patsiatzis, D.I., Xu, G., Papageorgiou, L.G., 2005. Layout aspects of pipeless batch plants. Ind. Eng. Chem. Res. 44, 5672–5679.

Pattinson, T., Majozi, T., 2010. Introducing a new operational policy: The PIS operational policy. Comput. Chem. Eng. 34, 59–72.

Pinto, J.M., Grossmann, I.E., 1995. A Continuous Time Mixed Integer Linear Programming Model for Short Term Scheduling of Multistage Batch Plants. Ind. Eng. Chem. Res. 34, 3037–3051.

Pinto, J.M., Grossmann, I.E., 1994. Optimal cyclic scheduling of multistage continuous multiproduct plants. Comput. Chem. Eng. 18, 797–816.

Pinto, T., Barbósa-Póvoa, A.P., Novais, A.Q., 2008. Design of multipurpose batch plants: A comparative analysis between the STN, m-STN, and RTN representations and formulations. Ind. Eng. Chem. Res. 47, 6025–6044.

Pinto, T., Novais, A.Q., Barbosa-Póvoa, A.P., 2003. Optimal Design of Heat-Integrated Multipurpose Batch Facilities with Economic Savings in Utilities: A Mixed Integer Mathematical Formulation. Ann. Oper. Res. 120, 201–230.

Quaglia, A., Sarup, B., Sin, G., Gani, R., 2013. A systematic framework for enterprise-wide optimization: Synthesis and design of processing networks under uncertainty. Comput. Chem. Eng. 59, 47–62.

Rippin, D.W.T., 1983. Simulation of single- and multiproduct batch chemical plants for optimal design and operation. Comput. Chem. Eng. 7, 137–156.

Ryoo, H.S., Sahinidis, N. V, 1996. A branch-and-reduce approach to global optimization. J. Glob. Optim. 8, 107–138.

Sahinidis, N. V., Grossmann, I.E., 1991. MINLP model for cyclic multiproduct scheduling on continuous parallel lines. Comput. Chem. Eng. 15, 85–103.

Sanmartí, E., Friedler, F., Puigjaner, L., 1998. Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation. Comput. Chem. Eng. 22, S847–S850.

Schilling, G., Pantelides, C.C., 1996. A simple continuous-time process scheduling formulation and a novel solution algorithm. Comput. Chem. Eng. 20, S1221–S1226.

Seid, E.R., Majozi, T., 2015. Design and synthesis of batch processing plants: A consideration of utility aspect and using a robust scheduling platform. Comput. Aided Chem. Eng., Computer Aided Chemical Engineering 37, 1169–1174.

Seid, E.R., Majozi, T., 2013. Design and synthesis of multipurpose batch plants using a robust scheduling platform. Ind. Eng. Chem. Res. 52, 16301–16313.

Seid, R., Majozi, T., 2012a. A novel technique for prediction of time points for scheduling of multipurpose batch plants. Chem. Eng. Sci. 68, 54–71.

Seid, R., Majozi, T., 2012b. A robust mathematical formulation for multipurpose batch plants. Chem. Eng. Sci. 68, 36–53.

Shah, N., Pantelides, C.C., 1992. Design of multipurpose batch plants with uncertain production requirements. Ind. Eng. Chem. Res. 31, 1325–1337.

Shah, N., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—II. Computational issues. Comput. Chem. Eng. 17, 229–244.

Shaik, M.A., Floudas, C.A., 2009. Novel unified modeling approach for short-term scheduling. Ind. Eng. Chem. Res. 48.

Shaik, M.A., Floudas, C.A., 2008. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. Comput. Chem. Eng. 32, 260–274.

Shaik, M.A., Janak, S.L., Floudas, C.A., 2006. Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study. Ind. Eng. Chem. Res. 45, 6190–6209.

Shaik, M.A., Vooradi, R., 2017. Short-Term Scheduling of Batch Plants: Reformulation for Handling Material Transfer at the Same Event. Ind. Eng. Chem. Res. 56, 11175–11185.

Smith, E., Pantelides, C., 1995. Design of reaction-separation networks using detailed models. Comput. Chem. Eng. 19, 83–88.

Sparrow, R.E., Forder, G.J., Rippin, D.W.T., 1975. The choice of equipment sizes for multiproduct batch plants. Heuristics vs. branch and bound. Ind. Eng. Chem. Process Des. Devel. 14, 197–203.

Stamp, J., Majozi, T., 2011. Optimum heat storage design for heat integrated multipurpose batch plants. Energy 36, 5119–5131.

Suhami, I., Mah, R., 1982. Optimal design of multipurpose batch plants. Ind. Eng. Chem. Process Des. Devel. 21, 94–100.

Sundaramoorthy, A., Karimi, I.A., 2005. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. Chem. Eng. Sci. 60, 2679–2702.

Susarla, N., Li, J., Karimi, I.A., 2009. A novel approach to scheduling multipurpose batch plants using unit-slots. AIChE J. 56, 1859–1879.

Vooradi, R., Shaik, M.A., 2012. Improved three-index unit-specific event-based model for short-term scheduling of batch plants. Comput. Chem. Eng. 43, 148–172.

Voudouris, V.T., Grossmann, I.E., 1992. Mixed-Integer Linear Programming Reformulations for Batch Process Design with Discrete Equipment Sizes. Ind. Eng. Chem. Res. 31, 1315–1325.

Wiede, W., Reklaitis, G.V., 1987. Determination of completion times for serial multiproduct processes—3. Mixed intermediate storage systems. Comput. Chem. Eng. 11, 357–368.

Williams, H.P., 2013. Model building in mathematical programming, 5th ed. Wiley.

Zhang, X., 1995. Algorithms for optimal scheduling using nonlinear models. University of London.

Zhang, X., Sargent, R.W.H., 1996. The optimal operation of mixed production facilities—a general formulation and some approaches for the solution. Comput. Chem. Eng. 20, 897–904.

Zhang, X., Sargent, R.W.H., 1994. The optimal operation of mixed production facilities - a general formulation and some solution approaches for the solution, in: Proc. Of 5th Intl. Symp. On Process Systems Engineering. Kyongju, Korea, pp. 171–177.

CHAPTER
3
MATHEMATICAL MODEL

3.1 Introduction

This chapter details the development for the mathematical model that can be applied to multipurpose batch plants. A mathematical model that aims to simultaneously schedule the batch plant while synthesizing both the major equipment and the pumping network is presented. The aforementioned concept of roving pumps is applied in the model in order to optimally allocate pumps to perform material transfers. Furthermore, the model determines the required transfer time associated with material transfers, depending on the amount of material transferred and the rating of the pump assigned.

Also addressed are alternative applications of the model. The model may be adapted to schedule conventionally operated batch plants. Additionally, the model is adaptable to optimally synthesize only the pumping network for the batch plant without considering the major equipment. Furthermore, it is possible to adapt the model to various operational philosophies.

3.2 Assumptions

The main assumptions employed in the mathematical formulation are listed below.

- Task duration increases linearly with the amount of material used.
- A higher throughput pump is more expensive.
- All piping connections have the same cost. This assumption may be altered by changing each individual piping connection cost.
- All material transferred is liquid.
- The pumps are all fixed speed pumps with speeds unaffected by liquid properties.
- Cross contamination of material using the same equipment is negligible. Materials are assumed compatible, without the need to clean equipment after tasks.

3.3 Mathematical Formulation

The constraints for the mathematical model that allow for flexible pump allocation, various transfer times, plant synthesis and optimal scheduling are shown below. Constraints (3.1)- (3.15) address allocation of pumping equipment to transfer tasks, whereas constraints (3.16) - (3.41) address the synthesis problem and scheduling of the various processing tasks based on pump availability. The scheduling model is reformulated from the models presented by Lee *et al.* (2015) and Seid and Majozi (2013) to allow for pumping allocation and variable transfer times.

Figure 3-1 illustrates possible material transfers within a given batch plant and depicts which variables are used in this model to represent pertinent material transfers. Notably, each transfer requires an active pump to perform the transfer. Furthermore, the material transfer time is dependent on the pump rating and the quantity of material transferred.

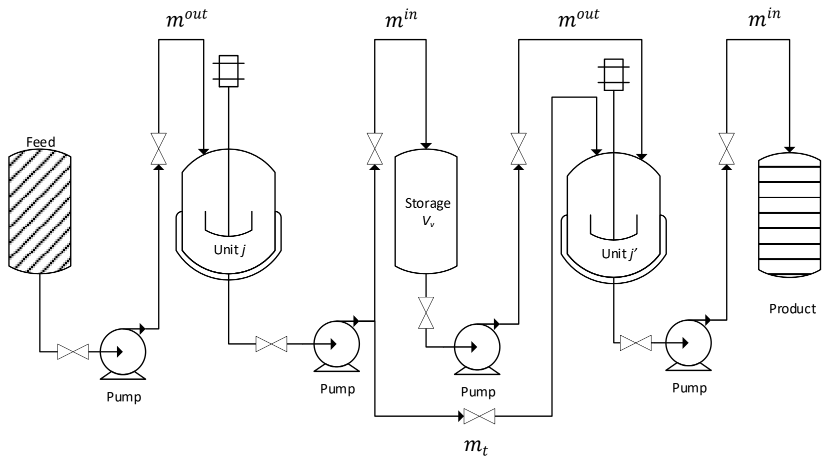


Figure 3-1 Material transfers and pump requirements

3.3.1 Pumping Constraints

Constraints (3.1) and (3.2) are required to determine where the necessary piping connections should be, as shown in Figure 3-2. constraint (3.1) states that pumping into a vessel may occur, provided a piping connection is present between the pump and the destination vessel. However, the presence of a piping connection does not necessarily mean that pumping must occur at a given time point.

$$\mu(v', v, \zeta, p) \leq i(\zeta, v')$$
$$\forall\, p \in P, v, v' \in V, \zeta \in Z \tag{3.1}$$

Similarly, constraint (3.2) states that pumping from a vessel may occur, provided a piping connection is present between the source vessel and the pump. Again, the presence of a piping connection does not necessarily mean that pumping must occur at a given time point.

$$\mu(v', v, \zeta, p) \leq o(v, \zeta)$$
$$\forall\, p \in P, v, v' \in V, \zeta \in Z \tag{3.2}$$

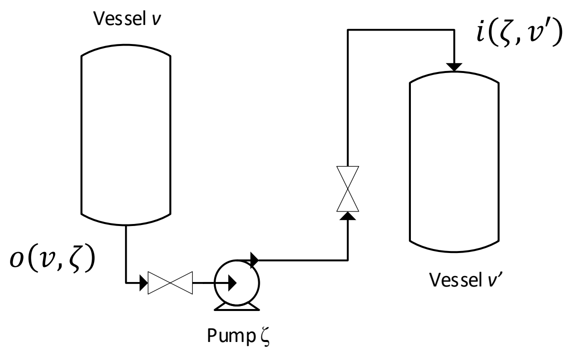


Figure 3-2 Piping requirements

Constraints (3.3) and (3.4) are required to ensure that pumps are not used simultaneously by different units at any given time point. Constraint (3.3) states that if a pump exists and is to be used for pumping from a source to a unit, then that same pump may not be used to pump to any other unit at that time point. However, a processing unit may repeatedly utilize the same pump to receive different states from various sources.

$$\mu(j, v_s, \zeta, p) + \sum_{j'} \mu(j', v_s', \zeta, p) \leq e(\zeta)$$
$$\forall\, p \in P, j, j' \in J, v_s, v_s' \in V_{source}\,, \tag{3.3}$$
$$\zeta \in Z, j \neq j', j' \neq v_s', j \neq v_s$$

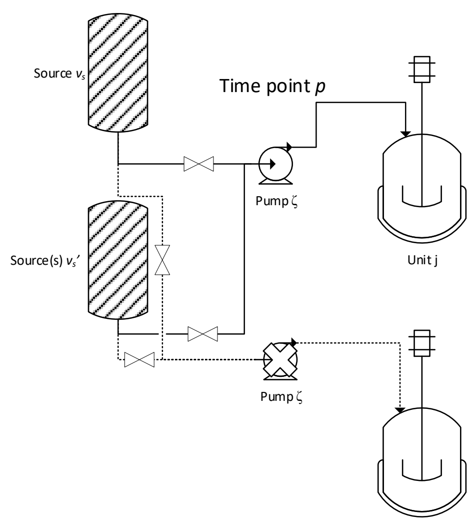


Figure 3-3 Assignment of pumps for transfer into processing units

Similarly, constraint (3.4) states that if a pump exists and is to be used to pump material from a processing unit to another vessel at a time point then that pump may not be used to pump material from other processing units. However, the same processing unit may repeatedly utilize the same pump to transfer different states to various destinations, as shown in Figure 3-4.

$$\mu(v_d, j, \zeta, p) + \sum_{j'} \mu(v_d', j', \zeta, p) \leq e(\zeta)$$
$$\forall\, p \in P, j, j' \in J, v_d, v_d' \in V_{destination}, \quad (3.4)$$
$$\zeta \in \mathrm{Z}, j \neq j', j \neq v_d, j' \neq v_d'$$

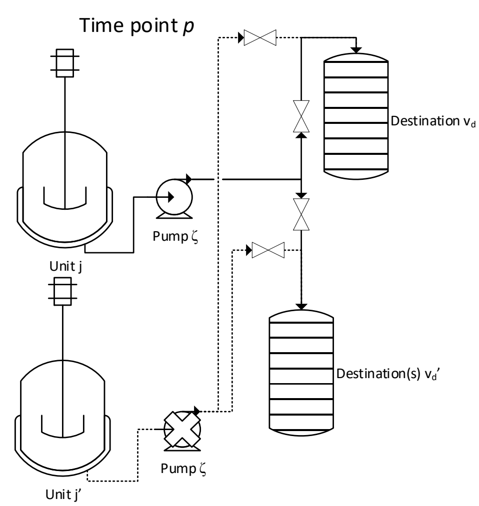


Figure 3-4 Assignment of pumps out of processing units

Constraint (3.5) states that a pump must be active between the storage vessel and the processing unit at a time point for any material to be transferred to a storage vessel from the processing unit.

$$m^{in}(s, j, p) \leq V^U(v_v) \sum_{\zeta} \mu(v_v, j, \zeta, p) \quad (3.5)$$
$$\forall\, s \in S, \zeta \in \mathrm{Z}, v_v \in V_{storage}, p \in P, j \in J$$

Similarly, constraint (3.6) states that a pump must be active in order to transfer product material from a unit to the product stockpile.

$$m^{in}(s,j,p) \leq V^U(j) \sum_{\zeta} \mu\big(v_p, j, \zeta, p\big) \qquad \forall\, s \in S^p, \zeta \in \mathrm{Z}, v_p \in V_{product}, p \in P, j \in J \tag{3.6}$$

Constraint (3.7) states that a pump must be associated with the transfer of material out of a storage vessel to a processing unit.

$$m^{out}(s,j,p) \leq V^U(v_v) \sum_{\zeta} \mu(j, v_v, \zeta, p) \qquad \forall\, s \in S, \zeta \in \mathrm{Z}, v_v \in V_{storage}, p \in P, j \in J \tag{3.7}$$

For the case where material is transferred between different processing units, Constraint (3.8) states that a pump must be associated with the transfer and limits the transfer to the design capacity of the accepting unit.

$$m_t(s,j,j',p) \leq V^U(j) \sum_{\zeta} \mu(j, j', \zeta, p) \qquad \forall\, s \in S, \zeta \in \mathrm{Z}, p \in P, j, j' \in J, j \neq j' \tag{3.8}$$

Constraint (3.9) states that a pump must be associated with the transfer of material out of a feed stockpile and limits the transfer to the design capacity of the accepting unit.

$$m^{out}(s,j,p) \leq V^U(j) \sum_{\zeta} \mu\big(j, v_f, \zeta, p\big) \qquad \forall\, s \in S, \zeta \in \mathrm{Z}, v_f \in V_{feed}, p \in P, j \in J \tag{3.9}$$

Constraint (3.10) assigns the exact transfer time for a state between two different units using a particular pump at a time point. This constraint, which is only active if there is material being pumped between unit j and j', states that the transfer time is related to the rating of the pump that is used and the amount of material transferred. In this manuscript, each pump has a fixed pump rating, which is defined as the transfer time per unit material.

$$\begin{aligned} &R(\zeta)m_t(s,j,j',p) \\ &-M\big(1-\ \mu(j,j',\zeta,p\ )\big) \leq\ t_t(s,j,j',\zeta,p) \leq R(\zeta)m_t(s,j,j',p) \\ &+M\big(1-\ \mu(j,j',\zeta,p\ )\big) \end{aligned} \tag{3.10}$$

$$\forall\, p \in P, j, j' \in J, s \in S, \zeta \in \mathrm{Z}, j \neq j'$$

Similarly, constraint (3.11) states that the transfer time between a processing unit and a storage vessel is dependent on the rating of the active pump and the quantity of material transferred into storage.

$$\begin{aligned} &R(\zeta)m^{in}(s,j,p) \\ &-M\big(1-\mu(v,j,\zeta,p\ )\big) \leq t_t(s,v,j,\zeta,p)\ \leq R(\zeta)m^{in}(s,j,p) \\ &+M\big(1-\mu(v,j,\zeta,p\ )\big) \end{aligned} \tag{3.11}$$

$$\forall\, p \in P, j \in J, v \in V_{storage}\ and\ V_{product}, s \in S, \zeta \in \mathrm{Z}$$

Additionally, constraint (3.12) states that the transfer time between a storage vessel and a processing unit is dependent on the rating of the active pump and the quantity of material transferred out of storage.

$$\begin{aligned} &R(\zeta)m^{out}(s,j,p) \\ &-M\big(1-\mu(j,v,\zeta,p\ )\big) \leq t_t(s,j,v,\zeta,p) \leq\ R(\zeta)m^{out}(s,j,p) \\ &+M\big(1-\mu(j,v,\zeta,p\ )\big) \end{aligned} \tag{3.12}$$

$$\forall\, p \in P, j \in J, v \in V_{storage}\ \&\ V_{feed}, s \in S, \zeta \in \mathrm{Z}$$

In a situation where two consecutive tasks take place in one unit, the transfer time for material already present in the unit is zero, as captured in constraint (3.13).

$$t_t(s, j, j, \zeta, p) = 0$$
$$\forall\, p \in P, j \in J, \zeta \in Z\ s \in S \tag{3.13}$$

Constraint (3.14) allocates time to transfer material to processing tasks. This formulation allows for simultaneous transfers into units from various sources. It may be the case that the transfer times are different for each pump used, thus constraint (3.14) states that transfer duration assigned allows for sufficient time to accommodate all transfers.

$$t_{dur}^{in}(s_{in,j}, p) \geq \sum_{v_S} t_t(s^{sc}, j, v_S, \zeta, p)$$
$$\forall\, p \in P, j \in J, s_{in,j} \in S_{in,j}^{*}, v_s \in V_{source}, s^{sc} \in S, \zeta \in \mathrm{Z} \tag{3.14}$$

Similarly, constraint (3.15) states that the duration assigned for pumping from a processing task must accommodate all the transfer times out of the unit. The constraint is required when transferring material to storage vessels and product stockpiles only, since transfer into other units is captured by constraint (3.14).

$$t_{dur}^{out}(s_{in,j}, p) \geq \sum_{v_v} t_t(s^{sp}, v_v, j, \zeta, p)$$
$$\forall\, p \in P, j \in J, s_{in,j} \in S_{in,j}^{*}, v_v \in V_{storage}, s^{sp} \in S, \zeta \in \mathrm{Z} \tag{3.15}$$

3.3.2 Scheduling

Constraints (3.16) - (3.41) pertain to scheduling and incorporate the pumping constraints. Some of the constraints are not necessarily new, but are included here for completeness purposes. Constraint (3.16) states that only one task may occur within a processing unit at any given time point, provided the processing unit exists.

$$\sum_{s_{in,j}} y(s_{in,j}, p) \leq \mathrm{e}(j)$$
$$\forall\, s_{in,j} \in S_{in,j}^{*}, j \in J, p \in P \tag{3.16}$$

Design capacity constraints

Constraint (3.17) states that the amount of material to be used by a task at any given time point must be within the design capacity of the processing unit.

$$V^L(j)y(s_{in,j},p) \leq m_u(s_{in,j},p) \leq V^U(j)y(s_{in,j},p)$$
$$\forall\, s_{in,j} \in S^*_{in,j}, p \in P, j \in J \tag{3.17}$$

Constraint (3.18) states that the amount of a state already stored in a storage vessel along with the material transferred to that storage vessel at a given time point, must not exceed the maximum design capacity of that storage vessel. Furthermore, it only allows such storage in the case where the storage vessel exists.

$$q(s,p) + \sum_j m^{in}(s,j,p) \leq V^U(v_v)e(v_v)$$
$$\forall\, s \in S, p \in P, j \in J, v_v \in V_{storage} \tag{3.18}$$

Material balance for storage

Constraint (3.19) is the material balance for storage and relates the material entering and leaving storage to that stored during the previous time point. The constraint states that the amount of a state stored at a time point relates to the amount that was previously stored and the material quantities which are leaving and entering the storage vessel.

$$q(s,p) = q(s,p-1) - \sum_{j \in J^{sc}} m^{out}(s,j,p) + \sum_{j \in J^{sp}} m^{in}(s,j,p-1)$$
$$\forall\, s \in S, p \in P, j \in J, p > 1 \tag{3.19}$$

Constraint (3.20) determines the amount of a state stored after it is initially used during the first time point. The constraint states that the material leaving storage during the first time point must be taken from the initial reserve.

$$q(s,p) = Q_0(s) - \sum_{j \in J^{sc}} m^{out}(s,j,p)$$
$$\forall\, s \in S, j \in J, p \in P, p = 1 \tag{3.20}$$

Processing unit material balance

Constraint (3.21) determines the total amount of material consumed by a task in unit *j* at a time point *p*. The constraint states that the state, *s*, to be consumed may either come from a storage vessel or from an upstream unit, j'

$$\rho^{sc}_{s_{in,j}} m_u\left(s_{in,j}, p\right) = m^{out}(s,j,p) + \sum_{j' \in J^{sp}} m_t(s,j,j'\, p)$$
$$\forall\, j, j' \in J, s \in S, p \in P, s_{in,j} \in S^*_{in,j} \tag{3.21}$$

Similarly, constraint (3.22) states that the amount of material produced within a unit is the amount of material that is transferred to storage and to other units.

$$\rho^{sp}_{s_{in,j}} m_u\left(s_{in,j}, p\right) = m^{in}(s,j,p) + \sum_{j \in J^{sc}} m_t(s,j',j,p+1)$$
$$\forall\, j, j' \in J, s \in S, p \in P, s_{in,j} \in S^*_{in,j} \tag{3.22}$$

Constraint (3.23) stipulates that the amount of material that can be produced by a task at any given time point, must be less than the combined available storage capacity and the design capacity of the processing unit. This is required for the case when only a single unit can produce the intermediate state within the batch process at a given time point.

$$\rho^{sp}_{s_{in,j}} m_u\left(s_{in,j}, p-1\right) \leq q(s,p) + V^U(j) l(j,p)$$
$$\forall j \in J, p \in P, s_{in,j} \in S^*_{in,j}, s \in S, p > 1 \tag{3.23}$$

Constraint (3.24) states that the total amount of material that can currently be consumed in a task must be available from storage or from a producing task during the previous time point. This allocation ensures that downstream tasks may receive material from multiple upstream tasks. Constraint (3.24) is nonlinear due to the presence of the bilinear term, however, it can be exactly linearized using the Glover (1975) transformation as discussed in Section 2.7.3. The linearized form of constraint (3.24) may be found in Section 3.3.3.

$$\begin{aligned} \sum_{s_{in,j} \in S_{in,j}^{sc}} \rho_{s_{in,j}}^{sc} m_u(s_{in,j}, p) & \\ \leq q(s, p-1) & \\ + \sum_{s_{in,j} \in S_{in,j}^{sp}} \rho_{s_{in,j}}^{sp} m_u(s_{in,j}, p-1) l(j, p) & \\ \forall j \in J, p \in P, s_{in,j} \in S_{in,j}^{*}, s \in S, p > 1 & \end{aligned} \tag{3.24}$$

Task sequencing

Constraint (3.25) describes the end time for a task and states that the processing duration depends on the batch size.

$$\begin{aligned} t_p(s_{in,j}, p) \geq t_u(s_{in,j}, p) + \tau(s_{in,j}) y(s_{in,j}, p) & \\ + \beta(s_{in,j}) m_u(s_{in,j}, p) & \\ \forall\, p \in P, s_{in,j} \in S_{in,j}^{*} & \end{aligned} \tag{3.25}$$

Constraint (3.26) states that a consuming task may commence in a unit after the required upstream tasks, as well as all the necessary transfers from and to that unit, have been completed.

$$\begin{aligned} t_u(s_{in,j}^{sc}, p) \geq t_p(s_{in,j}^{sp}, p-1) + t_{dur}^{out}(s_{in,j}^{sp}, p-1) + t_{dur}^{in}(s_{in,j}^{sc}, p) & \\ \forall\, p \in P, s_{in,j}^{sc}, s_{in,j}^{sp} \in S_{in,j}^{*} & \end{aligned} \tag{3.26}$$

Similarly, constraint (3.27) states that the same task in a unit may only commence again after processing has concluded and the necessary transfers from the unit have been performed during the previous time point.

$$t_u(s_{in,j}, p) \geq t_p(s_{in,j}, p-1) + t_{dur}^{out}(s_{in,j}, p-1) + t_{dur}^{in}(s_{in,j}, p) \quad \forall\, p \in P, s_{in,j} \in S_{in,j}^* \tag{3.27}$$

Constraint (3.28) pertains to scheduling over multiple time points such that interim material storage is accounted for, as noted by Seid and Majozi (2013). When all produced material is stored during the time point following its production, the constraint states that it can only be used after two time points. This allows sufficient time for the material to fully enter storage and later be removed from storage.

$$\begin{aligned} t_u(s_{in,j}^{sc}, p) \geq t_p(s_{in,j}^{sp}, p-2) + t_{dur}^{in}(s_{in,j}^{sc}, p) \\ - M\left(1 - y(s_{in,j}^{sp}, p-2)\right) \end{aligned} \quad \forall, p \in P, s_{in,j}^{sc}, s_{in,j}^{sp} \in S_{in,j}^*, p > 2 \tag{3.28}$$

Constraint (3.29) s required for the case where the FIS policy is implemented (Seid and Majozi, 2013). Used in conjunction with the above sequencing constraints, constraint (3.29) states that the consuming task begins when all transfers have been completed.

$$\begin{aligned} t_u(s_{in,j}^{sc}, p) \leq t_p(s_{in,j}^{sp}, p-1) + t_{dur}^{in}(s_{in,j}^{sc}, p) \\ + M\left(2 - y(s_{in,j}^{sc}, p) - y(s_{in,j}^{sp}, p-1)\right) \end{aligned} \quad \forall p \in P, s_{in,j}^{sp}, s_{in,j}^{sc} \in S_{in,j}^*, \in S, p > 1 \tag{3.29}$$

If a task is to recommence during a subsequent time point, it may only do so if all the material from the task of the preceding time point has been removed from the unit and fresh material for processing has been transferred to the unit during the current time point. While material is being transferred to the downstream task, material is still present in the source vessel and it is, therefore, not available for use until the transfers have concluded. The downstream task commences when all transfers to it have concluded.

The start of the downstream task must, therefore, precede that of the recommencing upstream task in order to avoid pumping to the upstream task before the source unit has been emptied. This eliminates the possibility of a new task commencing before the unit has finished emptying. Constraint (3.30), therefore, states that a task, $s_{in,j}$, may be initiated again once its dependent downstream tasks, $s'_{in,j'}$, as shown in Figure 3-5, have started and the required material for the current task has been transferred into the processing unit.

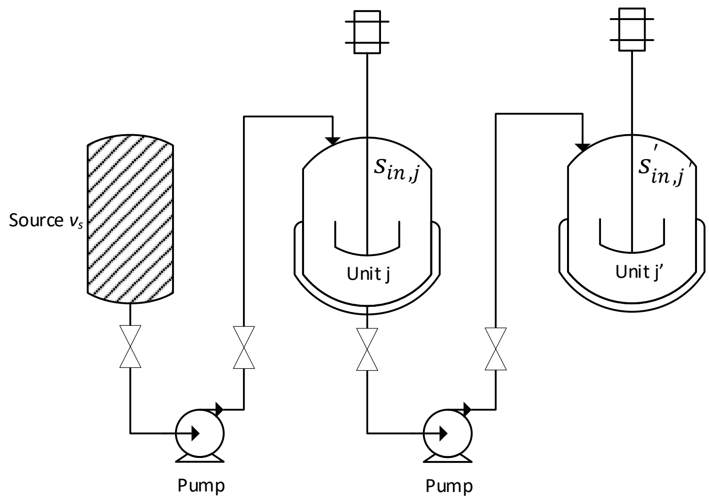


Figure 3-5 Upstream and downstream tasks

$$
\begin{aligned}
t_u(s_{in,j}, p) \geq t_u\left(s'_{in,j'}, p\right) + t^{in}_{dur}(s_{in,j}, p) \\
- M\left(2 - y(s_{in,j}, p) - \mu(j', j, \zeta, p)\right)
\end{aligned}
\qquad (3.30)
$$

$$
\forall p \in P, j, j' \in J, s_{in,j}, s'_{in,j'} \in S^*_{in,j}, \zeta \in Z
$$

Pump Sequencing Constraints

Constraint (3.31) states that the end time for pumping into a unit corresponds to the start time of a processing task, provided that the pump in question is allocated between the relevant vessels and the processing task is active during the time point.

$$
\begin{aligned}
&t_u\left(s_{in,j},p\right)\\
&-M\left(2-\mu(j,v_s,\zeta,p)-y\left(s_{in,j},p\right)\right) \le t_{\zeta p}^{in}(\zeta,j,p) \le t_u\left(s_{in,j},p\right)\\
&+M\left(2-\mu(j,v_s,\zeta,p)-y\left(s_{in,j},p\right)\right)
\end{aligned}
\tag{3.31}
$$

$$\forall\, p \in P, j \in J, v_s \in V_{source}, s_{in,j} \in S_{in,j}^{*}, \zeta \in \mathrm{Z}$$

Constraint (3.32) states that processing for a task in unit *j* may commence after sufficient time for pumping tasks to the task is given.

$$
\begin{aligned}
&t_u\left(s_{in,j},p\right)-t_{dur}^{in}\left(s_{in,j},p\right)\\
&-M\left(2-\mu(j,v_s,\zeta,p)-y\left(s_{in,j},p\right)\right) \le t_{\zeta u}^{in}(\zeta,j,p) \le t_u\left(s_{in,j},p\right)\\
&-t_{dur}^{in}\left(s_{in,j},p\right)+M\left(2-\mu(j,v_s,\zeta,p)-y\left(s_{in,j},p\right)\right)
\end{aligned}
\tag{3.32}
$$

$$\forall\, p \in P, j \in J, v_s \in V_{source}, s_{in,j} \in S_{in,j}^{*}, \zeta \in \mathrm{Z}$$

Constraint (3.33) states that the starting time for pumping out of a unit starts at the end time of a processing task, provided that the pump in question is allocated between the relevant vessels and the producing task is active during the time point.

$$
\begin{aligned}
&t_p\left(s_{in,j},p\right)\\
&-M\left(2-\mu(v_d,j,\zeta,p)-y\left(s_{in,j},p\right)\right) \le t_{\zeta u}^{out}(c,j,p) \ \le t_p\left(s_{in,j},p\right)\\
&+M\left(2-\mu(v_d,j,\zeta,p)-y\left(s_{in,j},p\right)\right)
\end{aligned}
\tag{3.33}
$$

$$\forall\, p \in P, j \in J, v_d \in V_{destination}, s_{in,j} \in S_{in,j}^{\;*}, \zeta \in \mathrm{Z}$$

Constraint (3.34) states that the end time for pumping out of a unit occurs such that sufficient time for pumping is given.

$$t_p(s_{in,j},p) + t_{dur}^{out}(s_{in,j},p)$$
$$-M\left(2-\mu(v_d,j,\zeta,p)-y(s_{in,j},p)\right) \le t_{\zeta u}^{out}(\zeta,j,p) \le t_p(s_{in,j},p)$$
$$+t_{dur}^{out}(s_{in,j},p) + M\left(2-\mu(v_d,j,\zeta,p)-y(s_{in,j},p)\right) \qquad (3.34)$$
$$\forall\, p \in P, j \in J, v_d \in V_{destination}, s_{in,j} \in S_{in,j}^{*}, \zeta \in \mathrm{Z}$$

Constraints (3.35) - (3.39) are required for the correct scheduling of pumping tasks such that there is no overlap using the same pump. Constraint (3.35) states that a particular pump may be used again to pump material into a unit after it has concluded pumping material into a unit during the previous time point.

$$t_{\zeta u}^{in}(\zeta,j,p) \ge t_{\zeta p}^{in}(\zeta,j',p-1)$$
$$\forall\, p \in P, j,j' \in J, \zeta \in \mathrm{Z} \qquad (3.35)$$

Similarly, constraint (3.36) states that a particular pump may be used again to pump material out of a unit after it has concluded pumping material out of any unit during the previous time point.

$$t_{\zeta u}^{out}(\zeta,j,p) \ge t_{\zeta p}^{out}(\zeta,j',p-1)$$
$$\forall\, p \in P, j,j' \in J, \zeta \in \mathrm{Z} \qquad (3.36)$$

Additionally, constraint (3.37) stipulates that a pump may be used to pump material into a unit after it has concluded pumping material out of any unit during the previous time point.

$$t_{\zeta u}^{in}(\zeta,j,p) \ge t_{\zeta p}^{out}(\zeta,j',p-1)$$
$$\forall\, p \in P, j,j' \in J, \zeta \in \mathrm{Z} \qquad (3.37)$$

Constraint (3.38) states that pumping into a particular unit may commence only after all pumping out of the same unit has concluded during the previous time point.

$$t_{\zeta u}^{in}(\zeta, j, p) \geq t_{\zeta p}^{out}(\zeta', j, p-1) \qquad \forall\, p \in P, j \in J, \zeta, \zeta' \in Z \tag{3.38}$$

Constraint (3.39) states that pumping from a vessel into another unit may only occur after the vessel has concluded receiving its material from all pumping tasks during the previous time point.

$$\begin{aligned} t_{\zeta u}^{in}(\zeta, j, p) \geq\ & t_{\zeta p}^{out}(\zeta', j', p-1) \\ & - M\big(2 - \mu(v_d, j', \zeta', p-1) - \mu(j, v_s, \zeta, p)\big) \end{aligned} \tag{3.39}$$

$$\forall\, p \in P, j, j' \in J, \zeta, \zeta' \in Z, v_s \in V_{source}, v_d \in V_{destination}, v_s = v_d$$

Constraint (3.40) states that all pumping and, therefore, processing must conclude within the time horizon of interest.

$$t_{\zeta p}^{out}(\zeta, j, p) \leq H \qquad \forall\, p \in P, j \in J, \zeta \in Z \tag{3.40}$$

Objective function

The objective function is the maximization of annualized profit available from the batch plant. In this work it is assumed that 7200 working hours are available per annum. Profit is described as the difference between the revenue from the products and the cost of all feed material, equipment and piping connections.

The total cost of each pump is a combination of a base cost and a cost associated with the rating of the pump. Higher rated pumps are taken to be more expensive than lower rated pumps. The combined costs of each pump is deducted from the total annualized profit, provided the pump is chosen to be present.

The cost of the major processing vessels is based on a fixed cost and is deducted from the total annualized profit, provided the processing unit is chosen to be present. Similarly, a fixed cost is associated with each of the storage vessels which is deducted from the profit when chosen to be part of the resulting system.

The costs of the piping connections are represented by connection costs into vessels from the pumps and out of vessels into pumps, respectively. This is deducted from the profit when the relevant connection is chosen to be present. It is possible to individually include each piping cost for the relevant piping connection based on piping design data obtained elsewhere. This individual piping cost is beyond the scope of the dissertation and a fixed cost is assumed in the applications.

$$\begin{aligned} &maximize\ Profit = \\ &= \left(\sum_{p} \sum_{s_{in,j} \in S^p} \rho^{sp}_{s_{in,j}} m_u\left(s_{in,j}, p\right) C\left(S^p\right) \right. \\ &\left. - \sum_{p} \sum_{s_{in,j} \in S^f} \rho^{sc}_{s_{in,j}} m_u\left(s_{in,j}, p\right) C\left(S^f\right) \right) H_{yr}/H \\ &- \sum_{\zeta} (C(\zeta) + C_{PR}/R(\zeta)) e(\zeta) - \sum_{j} \left(e(j) C(j)\right) \\ &- \sum_{v_v} \left(e(v_v) C(v_v)\right) - \sum_{v} \sum_{\zeta} \left(i(\zeta, V) C_{\zeta V} + o(V, \zeta) C_{V\zeta}\right) \end{aligned} \tag{3.41}$$

$$\forall\ p \in P, j \in J, v_v \in V_{storage}, v \in V, s_{in,j} \in S^*_{in,j}, S^p, S^f \in S, \zeta \in Z$$

3.3.3 Linearization

Constraint (3.24) is nonlinear due to the presence of the bilinear term. However, using a linearization technique developed by Glover (1975) as discussed in Section 2.7.3, the nonlinear constraint may be exactly linearized, allowing a MILP formulation that can be solved to global optimality. Constraints (3.42)-(3.44) form the linearization of constraint (3.24).

$$\sum_{s_{in,j} \in S_{in,j}^{sc}} \rho_{s_{in_j}}^{sc} m_u\left(s_{in_j}, p\right) \leq q(s, p-1) + \sum_{s_{in,j} \in S_{in,j}^{sp}} \rho_{s_{in_j}}^{sp} \Gamma\left(s_{in,j}^{sp}, p\right) \tag{3.42}$$

$$\forall j \in J, p \in P, s_{in,j} \in S_{in,j}^{*}, s \in S, p > 1$$

$$m_u\left(s_{in,j}^{sp}, p-1\right) - V^U(j)\left(1 - l(j,p)\right) \leq \Gamma\left(s_{in,j}^{sp}, p\right) \leq m_u\left(s_{in,j}^{sp}, p-1\right) - V^{L\prime}(j)\left(1 - l(j,p)\right) \tag{3.43}$$

$$\forall j \in J, p \in P, s_{in,j}^{sp} \in S_{in,j}^{*}, p > 1$$

$$l(\mathrm{j}, \mathrm{p}) V^{L\prime}(j) \leq \Gamma\left(s_{in,j}^{sp}, p\right) \leq l(j,p) V^U(j) \tag{3.44}$$

$$\forall j \in J, p \in P, s_{in,j}^{sp} \in S_{in,j}^{*}, p > 1$$

Notably, in most applications of this transform, $V^{L\prime}(j)$ will be zero, since it is allowable to transfer any amount of material from the producing task to the consuming task, as long as it does not exceed the upper capacity bound of the accepting unit. Hence, if $V^{L\prime}(j)$ is nonzero, it signifies the minimum amount of material that must be removed from a producing unit for use in a consuming task.

3.4 Scenarios

The proposed mathematical formulation in Section 3.3 is applicable to cases where it is desired to simultaneously schedule and synthesize the entire batch plant. The formulation includes aspects on the major equipment, storage vessels, piping connections and pumping equipment. However, the formulation may be adapted to correctly schedule existing batch plants. It can also be applied to only synthesize the pumping network, leaving the major equipment and storage vessels unaffected.

3.4.1 Plant Scheduling

The proposed formulation may be adapted for use when pumps are conventionally placed as shown in Figure 1-1. The formulation allows transfer times for pumping tasks to be accurately determined where either fixed or variable batch sizes are processed. Previous published literature either omitted the transfer times entirely, or used a fixed parameter to represent the transfers. These cases lead to inaccurate results when applying the scheduling models in practice.

The accurate scheduling of a batch plant can be achieved by adapting the proposed formulation by using the following constraints. Constraints (3.3) and (3.4) are to be replaced by constraints (3.45) and (3.46). Since all pumps are present when only scheduling the plant, the binary variable associated with the existence of pumping equipment takes the value of one.

$$\mu(j, v_s, \zeta, p) + \sum_{j'} \mu(j', v_s', \zeta, p) \leq 1$$

$$\forall\, p \in P, j, j' \in J, v_s, v_s' \in V_{source}\,, \qquad (3.45)$$

$$\zeta \in \mathrm{Z}, j \neq j', j' \neq v_s', j \neq v_s$$

$$\mu(v_d, j, \zeta, p) + \sum_{j'} \mu(v_d', j', \zeta, p) \leq 1$$

$$\forall\, p \in P, j, j' \in J, v_d, v_d' \in V_{destination}, \qquad (3.46)$$

$$\zeta \in \mathrm{Z}, j \neq j', j \neq v_d, j' \neq v_d{}'$$

Constraint (3.5) is adapted to produce constraint (3.47), since a fixed pump is present to transfer the material from a unit to a storage vessel. Here it is required to specify which pump will be responsible for all material transfer between the destination vessel and the source vessel.

$$m^{in}(s,j,p) \leq V^U(v_v)\mu(v_v,j,\zeta,p)$$
$$\forall\ s \in S, \zeta \in \mathrm{Z}, v_v \in V_{storage}, p\ \in P, j \in J \tag{3.47}$$

Similarly, constraints (3.6) - (3.9) are replaced by constraints (3.48) - (3.51).

$$m^{in}(s,j,p) \leq V^U(j)\mu\big(v_p,j,\zeta,p\big)$$
$$\forall\ s \in S^p, \zeta \in \mathrm{Z}, v_p \in V_{product}, p\ \in P, j \in J \tag{3.48}$$

$$m^{out}(s,j,p) \leq V^U(v_v)\mu(j,v_v,\zeta,p)$$
$$\forall\ s \in S, \zeta \in \mathrm{Z}, v_v \in V_{storage}, p\ \in P, j \in J \tag{3.49}$$

$$m_t(s,j,j',p) \leq V^U(j)\mu(j,j',\zeta,p)$$
$$\forall\ s \in S, \zeta \in \mathrm{Z}, p\ \in P, j, j' \in J, j \neq j' \tag{3.50}$$

$$m^{out}(s,j,p) \leq V^U(j)\mu\big(j,v_f,\zeta,p\big)$$
$$\forall\ s \in S, \zeta \in \mathrm{Z}, v_f \in V_{feed}, p\ \in P, j \in J \tag{3.51}$$

Constraint (3.10) may be replaced by constraint (3.52), since a fixed pump will perform all transfers between multiple points. It is required to specify which pump will be employed to perform the transfer.

$$R(\zeta)m_t(s,j,j',p) = t_t(s,j,j',\zeta,p)$$
$$\forall\, p \in P, j,j' \in J, s \in S, \zeta \in \mathrm{Z}, j \neq j' \tag{3.52}$$

Similarly, constraints (3.11) and (3.12) may be replaced by constraints (3.53) and (3.54).

$$R(\zeta)m^{in}(s,j,p) = t_t(s,v,j,\zeta,p)$$
$$\forall\, p \in P, j \in J, v \in V_{storage}\ and\ V_{product}, s \in S, \zeta \in \mathrm{Z} \tag{3.53}$$

$$R(\zeta)m^{out}(s,j,p) = t_t(s,j,v,\zeta,p)$$
$$\forall\, p \in P, j \in J, v \in V_{storage}\ and\ V_{feed}, s \in S, \zeta \in \mathrm{Z} \tag{3.54}$$

Constraint (3.16) is replaced by constraint (3.55), since all processing units in the original layout are present.

$$\sum_{s_{in,j}} y(s_{in,j},p) \leq 1$$
$$\forall\, s_{in,j} \in S^*_{in,j}, p \in P \tag{3.55}$$

Similarly, due to the fact that all storage vessels are present, constraint (3.18) is replaced by constraint (3.56).

$$q(s,p) + \sum_j m^{in}(s,j,p) \leq V^U(v_v)$$
$$\forall\, s \in S, p \in P, j \in J, v_v \in V_{storage} \tag{3.56}$$

Lastly, the objective function for the model is to be replaced by constraint (3.57) which maximizes the profit from the plant and excludes all capital investment from the original formulation.

$$maximize\ Profit = \sum_{p} \sum_{s_{in,j} \in S^p} \rho^{sp}_{s_{in,j}} m_u(s_{in,j}, p) C(S^p) - \sum_{p} \sum_{s_{in,j} \in S^f} \rho^{sc}_{s_{in,j}} m_u(s_{in,j}, p) C(S^f) \tag{3.57}$$

$$\forall\ p \in P, s_{in,j} \in S^*_{in,j}, S^p, S^f \in S, \zeta \in Z$$

3.4.2 Pumping Network Synthesis

The proposed model may also be adapted for cases where it is desired to only synthesize the pumping network. A case where a limited number of pumps is available to serve the entire batch plant can thus be accurately scheduled such that the plant is operated optimally with the available equipment.

For this case, the major equipment does not form part of the synthesis problem, hence, constraint (3.16) is replaced by constraint (3.58), which states that tasks may be performed in relevant units.

$$\sum_{s_{in,j}} y(s_{in,j}, p) \leq 1 \tag{3.58}$$

$$\forall\ s_{in,j} \in S^*_{in,j}, p \in P$$

Similarly, since all storage vessels are present, constraint (3.18) is replaced by constraint (3.59).

$$q(s,p) + \sum_{j} m^{in}(s,j,p) \leq V^U(v_v) \tag{3.59}$$

$$\forall\ s \in S, p\ \in P, j \in J, v_v \in V_{storage}$$

Finally, the objective function for this case is to maximize the annualized profit by including the cost of pumping and piping equipment, as shown by constraint (3.60).

$$
\begin{aligned}
&maximize\ Profit \\
&\quad = \left(\sum_{p} \sum_{s_{in,j} \in S^p} \rho^{sp}_{s_{in,j}} m_u\left(s_{in,j}, p\right) C(S^p) \right. \\
&\quad \left. - \sum_{p} \sum_{s_{in,j} \in S^f} \rho^{sc}_{s_{in,j}} m_u\left(s_{in,j}, p\right) C(S^f) \right) H_{yr}/H \\
&\quad - \sum_{\zeta} \left(C(\zeta) + C_{PR}/R(\zeta)\right) e(\zeta) \\
&\quad - \sum_{v} \sum_{\zeta} \left(i(\zeta, V) C_{\zeta V} + o(V, \zeta) C_{V\zeta}\right) \\
&\forall\ p \in P, v \in V, s_{in,j} \in S^*_{in,j}, S^p, S^f \in S, \zeta \in Z
\end{aligned}
\qquad (3.60)
$$

3.4.3 *Different Operational Philosophies*

Since a number of operational philosophies exist, as discussed in Section 2.3, it is expected that the proposed model will be adapted to different cases. The proposed model is based on the FIS policy since it is the most common and practical. Further discussed below is the applicability of the model to other common operational philosophies.

For the NIS operational philosophy, no intermediate storage vessels are available for use, meaning that material must be transferred directly between processing units. Hence, the existence variables of all the storage vessels will be zero and no cost can be associated with such units.

Hence in order to allow for the NIS policy, Equation (3.61) is required. By stating that no storage vessels exist, no material can be stored nor be transferred to or from storage.

$$e(v_v) = 0 \qquad \forall\, v_v \in V_{storage} \tag{3.61}$$

In cases where an unlimited amount of storage capacity is available for use, the UIS philosophy is applied. In order to employ the philosophy in the model, the upper bound for the capacity of storage vessels must be infinite or a sufficiently large number, as shown by Equation (3.62)

$$V^U(v_v) = \infty \qquad \forall\, v_v \in V_{storage} \tag{3.62}$$

When a material must be transferred directly from a processing unit to another vessel after production, the ZW policy is used. Hence, the material is not allowed to remain in the producing vessel after production. Therefore, constraint (3.25) needs to be replaced by constraint (3.63) that states that a task ends immediately after processing concludes.

$$\begin{aligned} t_p(s_{in,j}, p) = t_u(s_{in,j}, p) + \tau(s_{in,j}) y(s_{in,j}, p) \\ + \beta(s_{in,j}) m_u(s_{in,j}, p) \end{aligned} \qquad \forall\, p \in P, s_{in,j} \in S_{in,j}^{*} \tag{3.63}$$

If it is desired to have various operational philosophies present during production, the necessary constraints need to be added or adapted as outlined above.

3.5 Nomenclature

Sets

J	$\{j \mid j$ is a processing unit/piece of equipment$\}$
J^{sc}	$\{j^{sc} \mid j^{sc}$ is a consuming unit$\}$
J^{sp}	$\{j^{sp} \mid j^{sp}$ is a producing unit$\}$
P	$\{p \mid p$ is a time point$\}$
S	$\{s \mid s$ is any state/material$\}$
S^{sc}	$\{s \mid s$ is any state/material that is consumed$\}$
S^{sp}	$\{s \mid s$ is any state/material that is produced$\}$
$S_{in,j}$	$\{s_{in,j} \mid s_{in,j}$ is a task in a processing unit$\}$
$S^{*}_{in,j}$	$\{s^{*}_{in,j} \mid s^{*}_{in,j}$ is an effective task in a processing unit$\}$
$S^{sc}_{in,j}$	$\{s^{sc}_{in,j} \mid s^{sc}_{in,j}$ is a task which consumes state s$\}$
$S^{sp}_{in,j}$	$\{s^{sp}_{in,j} \mid s^{sp}_{in,j}$ is a task which produces state s$\}$
V	$\{v \mid v$ is any vessel and includes processing units (j), storage vessels (v_v), source vessels (v_s), destination vessels (v_d) as well as material feeds (v_f) and product stockpiles $(v_p)\}$
Z	$\{\zeta \mid \zeta$ is a pump$\}$

Binary Variables

$y(s_{in,j}, p)$ $\begin{cases} = 1 \text{ if task } s_{in,j} \text{ occurs during time point } p \\ = 0 \text{ otherwise} \end{cases}$

$l(j, p)$ $\begin{cases} = 1 \text{ if product from unit } j \text{ will be used in the next time point} \\ = 0 \text{ otherwise} \end{cases}$

$\mu(v, v', \zeta, p)$ $\begin{cases} = 1 \text{ if material is transferred from vessel } v' \text{to } v \text{ using pump } \zeta \text{ at time point } p \\ = 0 \text{ otherwise} \end{cases}$

$e(j)$ $\begin{cases} = 1 \text{ if unit } j\ exists \\ = 0 \text{ otherwise} \end{cases}$

$e(v_v)$ $\begin{cases} = 1 \text{ if unit } v\ exists \\ = 0 \text{ otherwise} \end{cases}$

$e(\zeta)$ $\begin{cases} = 1 \text{ if pump } \zeta\ exists \\ = 0 \text{ otherwise} \end{cases}$

$o(v, \zeta)$ $\begin{cases} = 1 \text{ if piping connection from unit } v \text{ to pump } \zeta \text{ exists} \\ = 0 \text{ otherwise} \end{cases}$

$i(\zeta, v)$ $\begin{cases} = 1 \text{ if piping connection from pump } \zeta \text{ to unit } v \text{ exists} \\ = 0 \text{ otherwise} \end{cases}$

Continuous Variables

$m_u\left(s_{in_j}, p\right)$	Amount of state *s* used during task s_{in_j} at time point *p*
$q(s,p)$	Amount of state *s* stored at time point *p*
$m^{in}(s,j,p)$	Amount of state *s* transferred from unit *j* to storage
$m^{out}(s,j,p)$	Amount of state *s* transferred to unit *j* from storage
$m_t(s,j,j'\,p)$	Amount of state *s* transferred to unit *j* from unit *j'*
$t_p\left(s_{in,j}, p\right)$	Time a task ends during time point *p*
$t_u\left(s_{in,j}, p\right)$	Time a task starts during time point *p*
$t_t(s, v_d, v_s, \zeta, p)$	Transfer time for state *s* from source to destination using pump ζ
$\Gamma\left(s_{in,j}, p\right)$	Glover transform variable
$t_{\zeta p}^{in}(\zeta, j, p)$	Time that pumping into unit *j* must end using pump ζ
$t_{\zeta u}^{in}(\zeta, j, p)$	Time that pumping into unit *j* may start using pump ζ
$t_{\zeta u}^{out}(\zeta, j, p)$	Time pumping from a unit *j* may start using pump ζ
$t_{\zeta p}^{out}(\zeta, j, p)$	Time pumping from a unit *j* must end using pump ζ
$t_{dur}^{in}\left(s_{in,j}, p\right)$	Duration to pump into a task
$t_{dur}^{out}\left(s_{in,j}, p\right)$	Duration to pump out of a task

Parameters

$V^L(v)$	Lower design capacity bound for vessel *v*
$V^U(v)$	Upper design capacity for vessel *v*
$Q_0(s)$	Initial reserve of stored state *s*
$\rho^{sp}_{s_{in_j}}$	Fraction of material produced from total used material
$\rho^{sc}_{s_{in_j}}$	Fraction of material consumed from total used material
$\tau\left(s_{in,j}\right)$	Fixed processing time for task $s_{in,j}$
$\beta\left(s_{in,j}\right)$	Batch size dependent variable processing time of task $s_{in,j}$
H	Time horizon of interest
H_{yr}	Annualized working hours. (*7200 hours*)
M	Big M constraint, any sufficiently large number (*100 000*)
$R(\zeta)$	Pumping rate of pump ζ measured in hours per unit amount material used
$C(s^p)$	Price of product states
$C(s^f)$	Price of feed states
$C(\zeta)$	Base cost of pump existence
C_{PR}	Base cost of pump rating
$C_{\zeta V}$	Cost for piping connection from pump ζ to vessel *v*
$C_{V\zeta}$	Cost for piping connection from vessel *v* to pump ζ
$C(v)$	Cost for vessel *v*

3.6 References.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Manage. Sci. 22, 455–460.

Lee, J., Seid, E.R., Majozi, T., 2015. Heat integration of intermittently available continuous streams in multipurpose batch plants. Comput. Chem. Eng. 74, 100–114.

Seid, E.R., Majozi, T., 2013. Design and synthesis of multipurpose batch plants using a robust scheduling platform. Ind. Eng. Chem. Res. 52, 16301–16313.

CHAPTER

4

APPLICATIONS

4.1 Introduction

In order to assess the proposed MILP model, benchmark literature examples and case studies were evaluated. All results were obtained using a computer with 16 GB of RAM, 2.59 GHz, Intel I7 Processor with Windows 10 operating system and the GAMS 24.8.3 software suite. The CPLEX solver in the GAMS suite was utilized to solve the problems.

The main aim of applying the proposed model to the various problems is to verify the model applicability and use. Furthermore, it is desired to evaluate how the proposed mode of operation will differ to that of conventional operation. Due to the novel nature of incorporating pump allocation and pump rating, it is not possible to directly compare the proposed formulation result values with previous literature.

4.2 First Illustrative Example

The first example is adapted from the scenario given by Lin and Floudas (2001) which involves a multipurpose batch plant capable of performing 4 distinct tasks. The recipe representation for the scenario is given in Figure 4-1. The production recipe also allows a product state, S5, to be consumed to produce another product, S6. Three major processing units are available to perform the tasks along with a storage vessel for state 4 (S4) to be chosen from for synthesis.

Data pertaining to the equipment may be found in Table 4-1 and economic data in Table 4-2, as adapted from Lin and Floudas (2001). The data pertaining to pumping rates for the various available pumps are given in Table 4-3.

The illustrative pumping rates were calculated such that no pump will be active for more than five minutes when the pumps are placed conventionally with the plant operating at full capacity. A time horizon of 12 hours was considered while employing the unlimited wait philosophy for operation. The processing duration is independent of batch size.

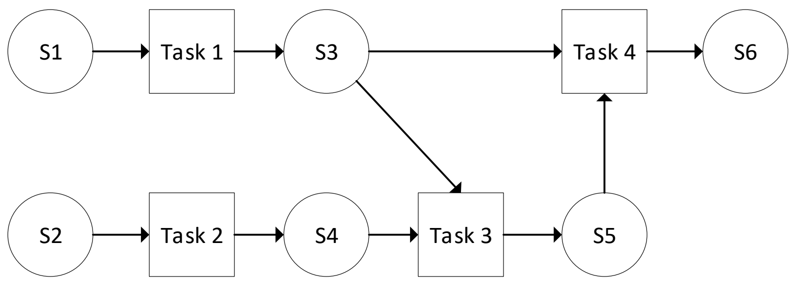


Figure 4-1 STN representation for the first illustrative example

Table 4-1 Equipment data for the first illustrative example

Unit/ Storage Vessel	Capacity	Suitability (Task/Storage)	Task Duration (Hours)	Cost (Cost Units)
Unit J1	50-150 mu	T1 T2	2 2	100
Unit J2	50-150 mu	T1 T2	2 2	150
Unit J3	50-200 mu	T3 T4	4 2	120
Vessel 4	100 mu	S4	-	15

Table 4-2 Economic data for the first illustrative example

Material/Equipment	**Price (Cost units)**
S1, S2	0.1 / material unit (mu)
S5 (Product 1)	0.4 /mu
S6 (Product 2)	0.6 /mu
Pump base cost	10
Pump rating cost	0.001
Piping cost per connection	5

Table 4-3 Pump ratings for the first illustrative example

Pump	**Pump Rating (seconds / material unit)**
A1	2.00
A2	2.00
A3	1.50
B1	2.00
B2	2.00
B3	1.50
B4	1.50

Figure 4-2 shows the typical flowsheet for the first illustrative example and depicts the pumps which would conventionally be used to transfer material between different areas within the plant. In order to simplify the flowsheet, it is deconstructed such that the pumps required to pump material out of processing units and the pumps required to pump material into processing units are separated, as shown in Figure 4-3 and Figure 4-4, respectively. These deconstructed views are useful for larger plants, where the piping connections are difficult to identify and follow. A total of seven pumps are required to pump material throughout the plant when the plant is conventionally operated.

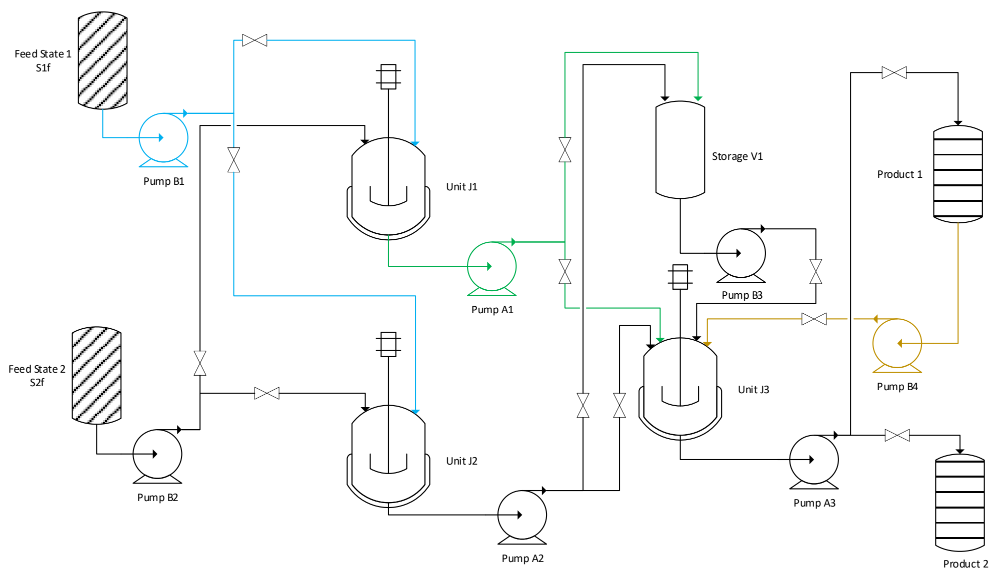


Figure 4-2 First illustrative example: conventional pump allocation and plant superstructure

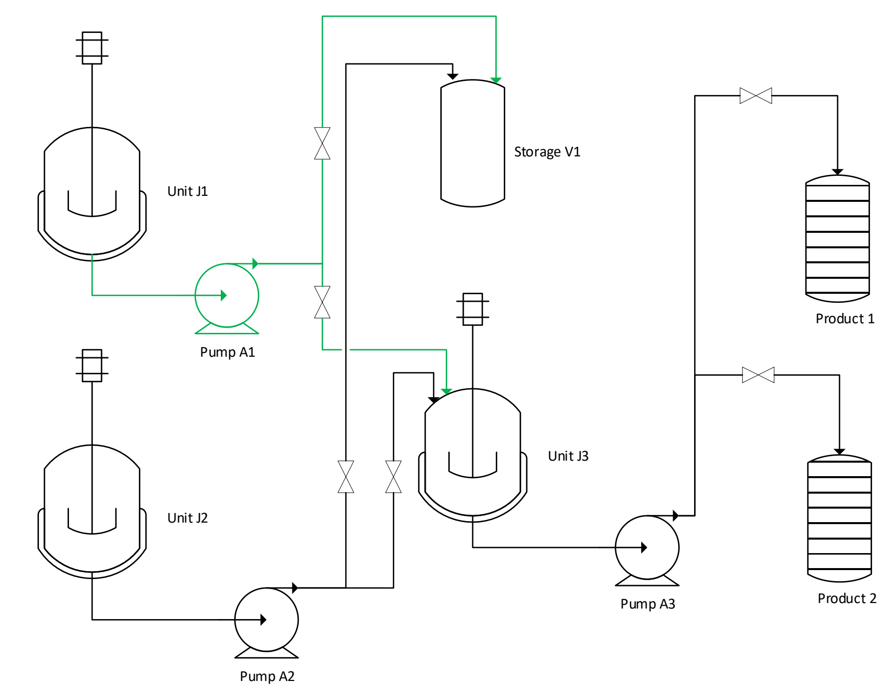


Figure 4-3 First illustrative example: conventional pump allocation for pumping from units

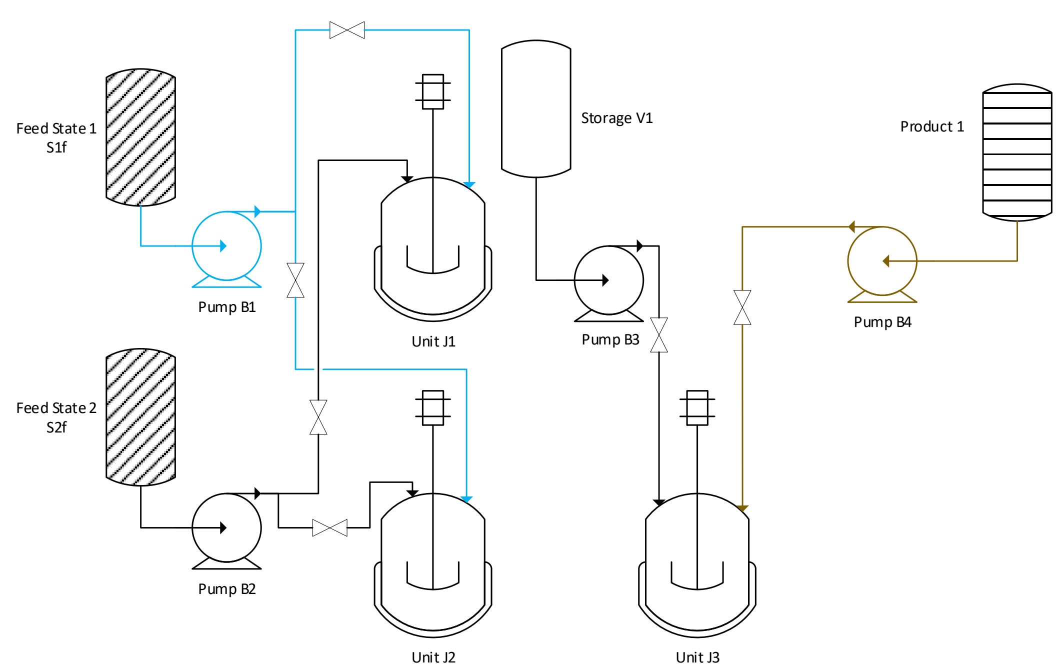


Figure 4-4 First illustrative example: conventional pump allocation for pumping to units

In solving the example, constraints (3.3) and (3.4) are to be applied. These constraints are key for the assignment of pumps for transfer tasks. For the first illustrative example, constraint (3.3) is employed as follows, when the destination unit is taken as unit J1 from source vessel J2, using a pump A1 at a given time point, *p*. This explicit statement dictates that no other pumping transfer may occur elsewhere in the plant that is not feeding unit J1, without compromising possible feeds to other units.

$$\mu(J1, J2, A1, p) + \mu(J2, J1, A1, p) + \mu(J3, J1, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J3, J2, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, J3, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, V1, A1, p) + \mu(J3, V1, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, S1f, A1, p) + \mu(J3, S1f, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, S2f, A1, p) + \mu(J3, S2f, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J3, Prod\ 1, A1, p) \leq e(A1)$$

Constraint (3.3) is similarly employed to cater for any other destination unit *j*, from source vessel v_s, using pump ζ, at time point *p*. Similarly, constraint (3.4) is employed as follows to transfer material to destination storage vessel V1, from source unit J1, using pump A1 at time point *p*.

$$\mu(V1, J1, A1, p) + \mu(Prod\ 1, J2, A1, p) + \mu(Prod\ 1, J3, A1, p) \leq e(A1)$$
$$\mu(V1, J1, A1, p) + \mu(Prod\ 2, J2, A1, p) + \mu(Prod\ 2, J3, A1, p) \leq e(A1)$$
$$\mu(V1, J1, A1, p) + \mu(V1, J2, A1, p) + \mu(V1, J3, A1, p) \leq e(A1)$$

Constraint (3.4) is similarly employed to cater for any other destination storage vessel v_d, from source unit *j*, using pump ζ, at time point *p*.

4.2.1 *Results for First Illustrative Example*

The MILP problem was solved using the CPLEX solver Figure 4-5 shows the resulting optimal plant pumping structure needed to cater for the example. Deconstructing the optimal pumping structure in a similar fashion as with the flowsheet, yields the simplified view for the pumping inlet connections and the pumping outlet connections, as shown in Figure 4-6 and Figure 4-7 respectively. By flexibly allocating pumps, the plant required two pumps to transfer material throughout the batch plant whereas seven pumps were required when flexible pump allocation was not considered.

The problem was solved yielding the model results as shown in Table 4-4, along with the results of Lin and Floudas (2001).

Table 4-4 First illustrative example: results summary

	This work	Lin and Floudas (2001)
Objective value (cost units)	1.27×10^6	195.6
CPU solution time (seconds)	1.63	0.4
Discrete Variables	4 246	59
Continuous Variables	5 315	175
Time points used	4	4
Notes	-	Not annualized cost objective, excludes feed costs, pump cost, piping connections, includes minimum throughput requirements, varying design costs.

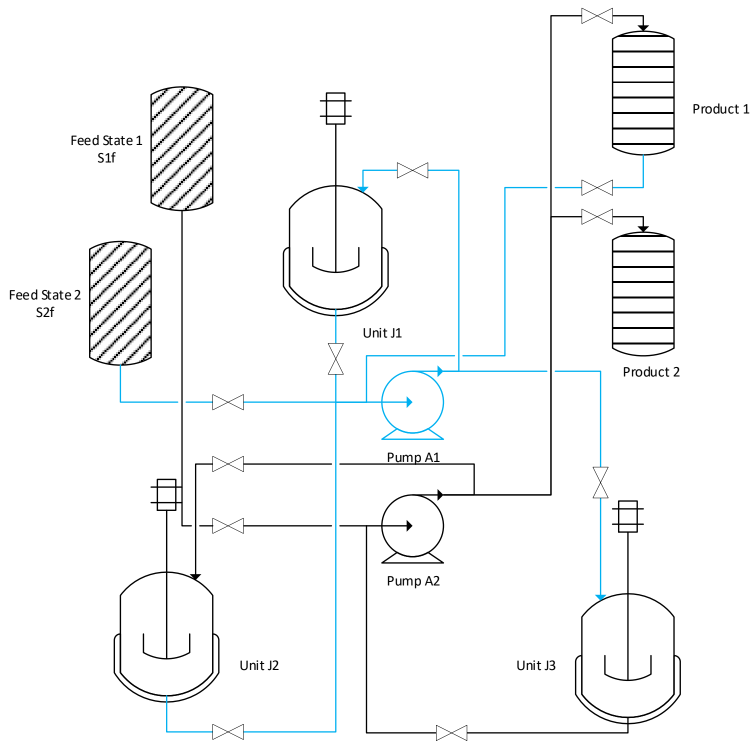


Figure 4-5 First illustrative example: optimal roving pump allocation and connections

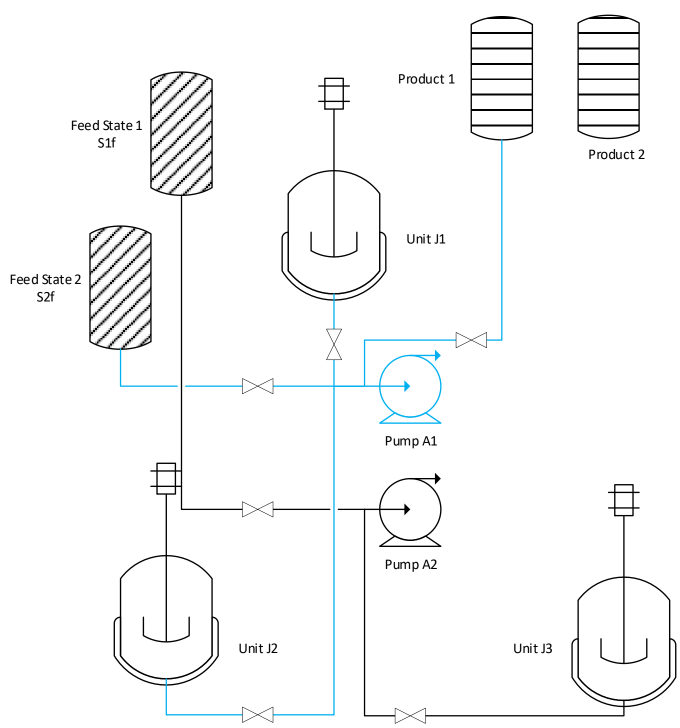


Figure 4-6 First illustrative example: optimal pump allocation (pump inlet)

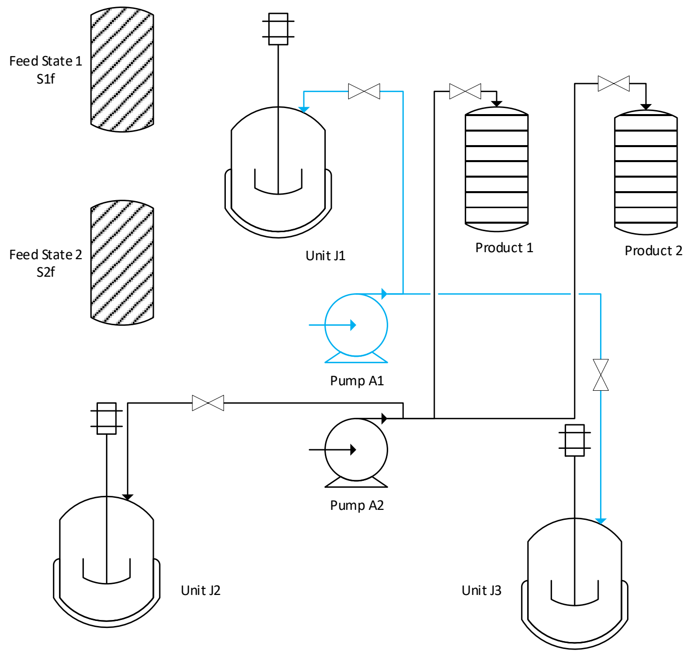


Figure 4-7 First illustrative example: optimal pump allocation (pump outlet)

Figure 4-8 shows the optimal pumping schedule required for the first illustrative example when pumps are flexibly assigned. As seen on the schedule, a number of pumping tasks performed by a same pump are highlighted. These correspond to the processing tasks that require multiple pumping tasks. The order in which material is to be pumped is of no consequence, since all raw feeds need to enter or leave a unit before operation may commence. Therefore, the pumping order is left to the user. The optimal production schedule using roving pumps may be found in Figure 4-9.

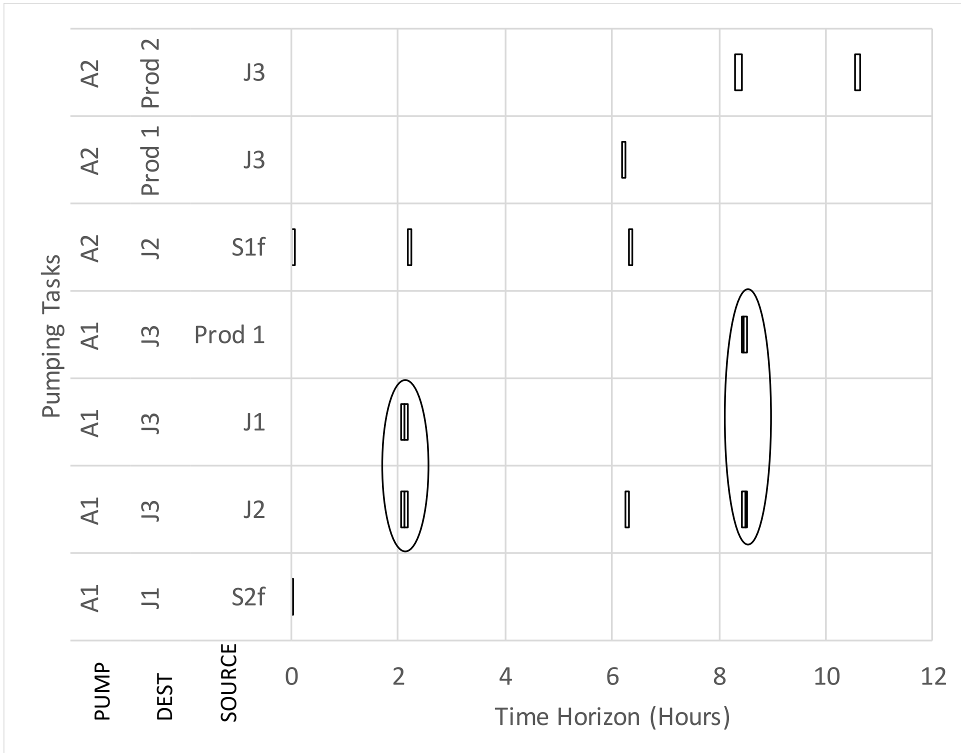


Figure 4-8 Optimal pumping schedule obtained for the first illustrative example using roving pumps

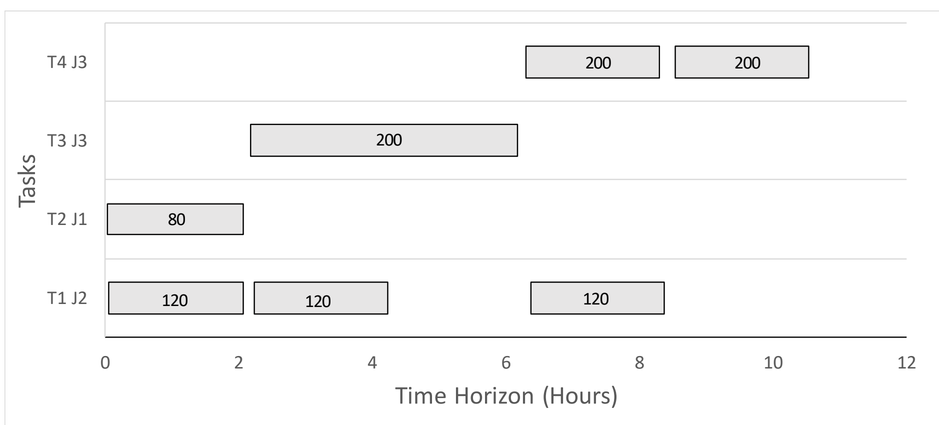


Figure 4-9 Optimal production schedule obtained for the first illustrative example using roving pumps

Figure 4-10 is the optimal production schedule for when the plant is not optimally synthesized and translates to an annualized profit of 126 665 cost units. The optimal production schedules for both the case when roving pumps are allocated and for the case when the plant is operated in a conventional manner, are very similar. Both cases yielded the same throughput within the time horizon of interest. Therefore, by allocating pumps flexibly throughout the plant it is possible to achieve at least the same throughput as when the plant is operated conventionally.

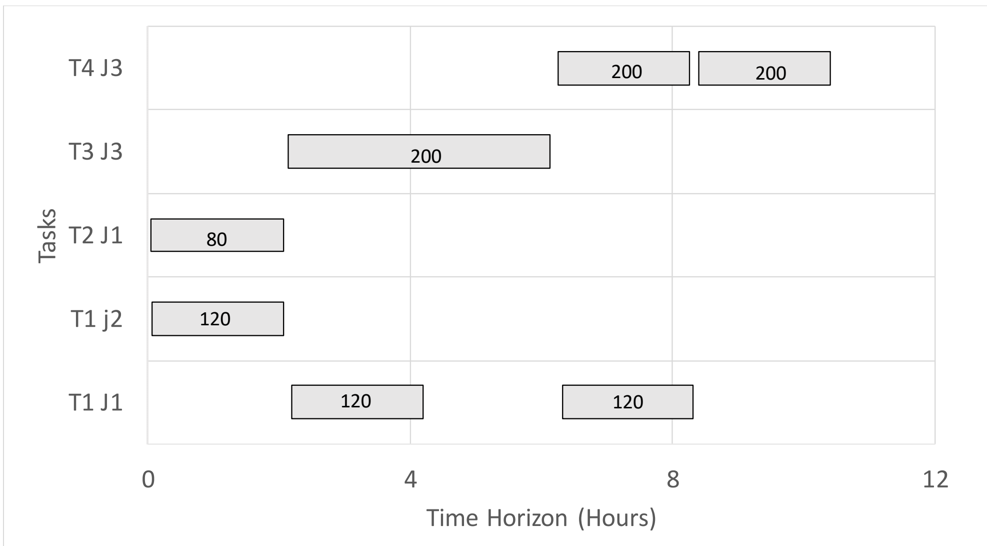


Figure 4-10 Optimal production schedule obtained for the first illustrative example using conventional operation

The optimal production schedule as presented by Lin and Floudas (2001) may be found in Figure 4-11. Noticeably, it clearly illustrates that transfer time is not incorporated in the model since processing tasks may commence immediately.

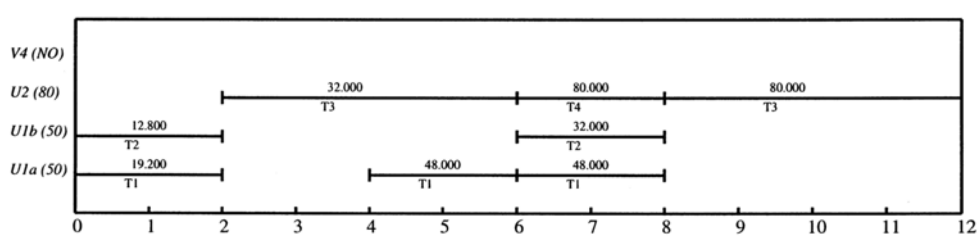


Figure 4-11 Optimal schedule for the first illustrative example as presented by Lin and Floudas (2001)

In order to assess the scalability of the proposed formulation, the first illustrative example was subjected to varying conditions. Both the number of available pumps and the number of time points were varied in order to evaluate the solution time required. A standard pumping rate of 6 seconds per unit material was allocated for each pump. Table 4-5 shows the achievable objective value and the required solution time when a varying number of pumps are used. Four time points were used for each entry. To evaluate the effect that a varying number of time points has on the required solution time, three standard pumps were used with an increasing number of time points. The scalability results for a varying number of time points may be seen in Table 4-6.

Table 4-5. Effect of available pumps on the scalability of the first illustrative example

Available number of pumps	**Objective value (c.u)**	**CPU solution time (s)**
1	35 709	0.6
3	126 753	0.9
5	126 753	1.5
7	126 753	2.8
9	126 753	3.6
11	126 753	5.1

Table 4-6. Effect of number of time points on the scalability of the first illustrative example

Number of time points	**Objective value (c.u)**	**CPU solution time (s)**
1	0	0.3
2	35 564	0.6
3	81 159	0.7
4	126 753	0.9

Both an increase in the number of available pumps and the number of time points increases the required solution time. However, as seen in Table 4-5, the number of pumps significantly impacts on the required solution time. This is mainly due to the increased number of binary variables in the formulation that makes the problem combinatorically complex.

4.3 Second Illustrative Example

Originally presented by Kondili *et al.* (1993), multipurpose batch plant is presented that involves a heating unit, two reactors which can perform three different reactions each, followed by a separation unit. The STN representation of the recipe for the given example may be found in Figure 4-12. Data pertaining to the equipment may be found in Table 4-7, and data pertaining to the economic aspects may be found in Table 4-8 as adapted from Lin and Floudas (2001). Pumping data for the eleven pumps present may be found in Table 4-9. A time horizon of 12 hours was considered while allowing unlimited wait in the processing units. The batch duration is material amount dependent.

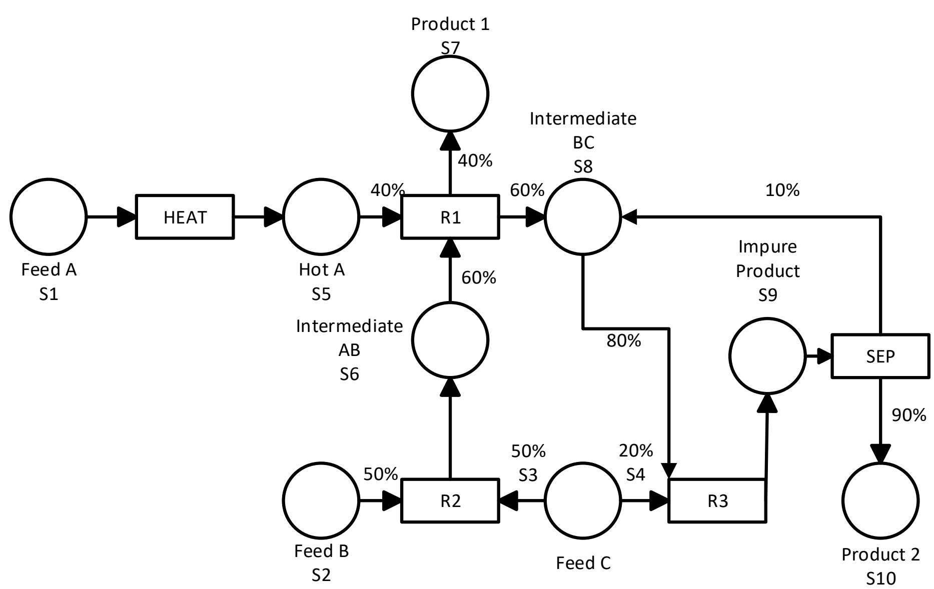


Figure 4-12 STN representation for the second illustrative example (Kondili *et al.*, 1993)

Table 4-7 Equipment data for the second illustrative example

Unit / Storage	Capacity (mu)	Suitability (Task/Storage)	Task Duration (Hours)	Cost (Cost Units)
Unit J1	20-50	HEAT	1 + 0.0067mu	100
Unit J2	50-70	R1 R2 R3	2 + 0.0267mu 2 + 0.0267mu 1 + 0.0133mu	150
Unit J3	70	R1 R2 R3	2 + 0.0167mu 2 + 0.0167mu 1 + 0.0083mu	120
Unit J4	50-80	SEP	2 + 0.0033mu	150
Vessel 1	30	S5	-	30
Vessel 2	60	S6	-	15
Vessel 3	70	S8	-	10
Vessel 4	100	S9	-	20

Table 4-8 Economic data for the second illustrative example

Material/Equipment	Price (Cost units) (Thousands)
S1	0.001 /mu
S2	0.002 /mu
S3	0.0015 /mu
S4	0.0015 /mu
S7 (Product 1)	0.02 /mu
S10 (Product 2)	0.03 /mu
Pump base cost	10
Pump rating cost	0.001
Piping cost per connection	5

Table 4-9 Pump ratings for the second illustrative example

Pump	Pump Rating (seconds / unit material)
A1	6.00
A2	4.29
A3	4.29
A4	3.75
B1	6.00
B2	4.29
B3	4.29
B4	3.75
B5	4.29
B6	4.29
B7	3.75

Figure 4-13 and Figure 4-14 are deconstructed views of the superstructure and pumping connections for the second illustrative example when the pumps are placed according to convention. Due to the large number of piping connections, it is difficult to identify the structure clearly, hence the deconstructed views aid with plant layout representation. A total of eleven pumps are required to service all areas of the plant.

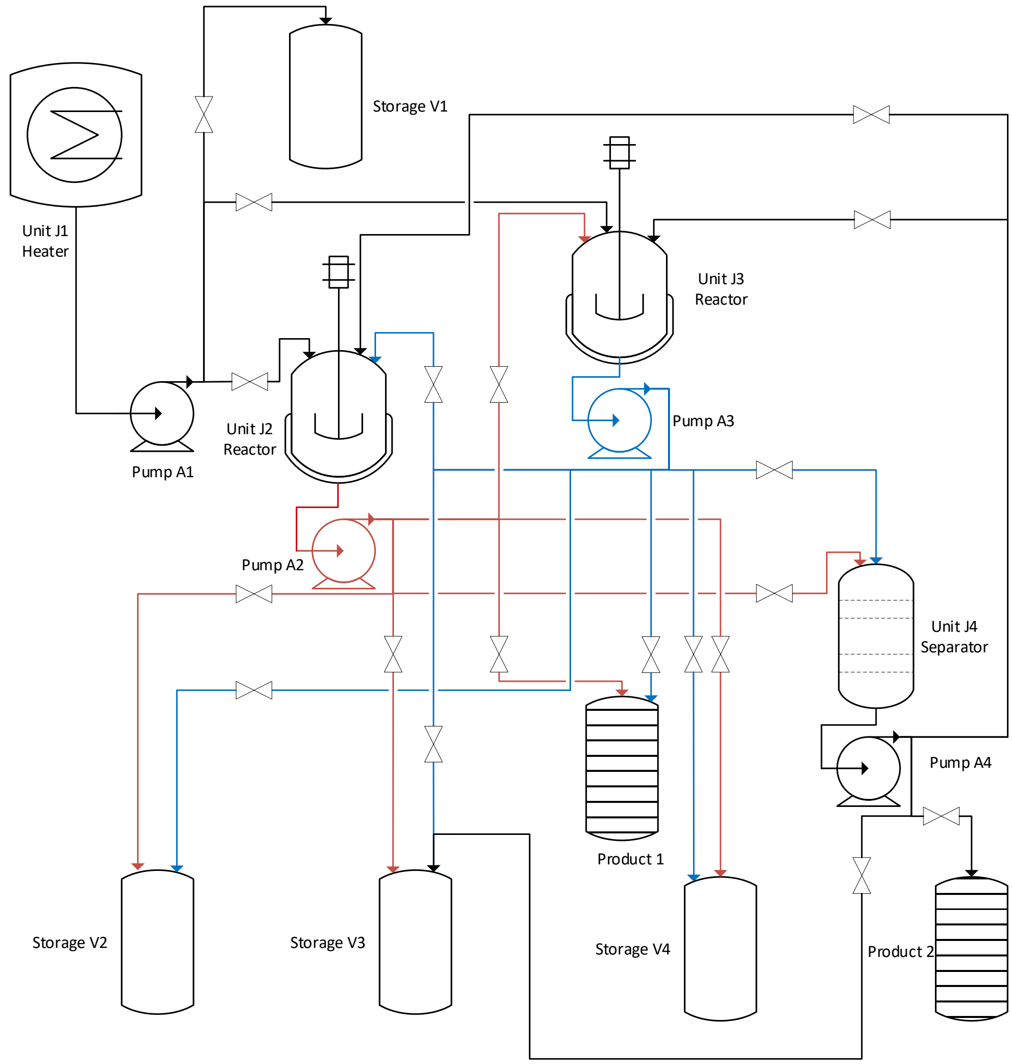


Figure 4-13 Second illustrative example: conventional pump allocation for pumping from units

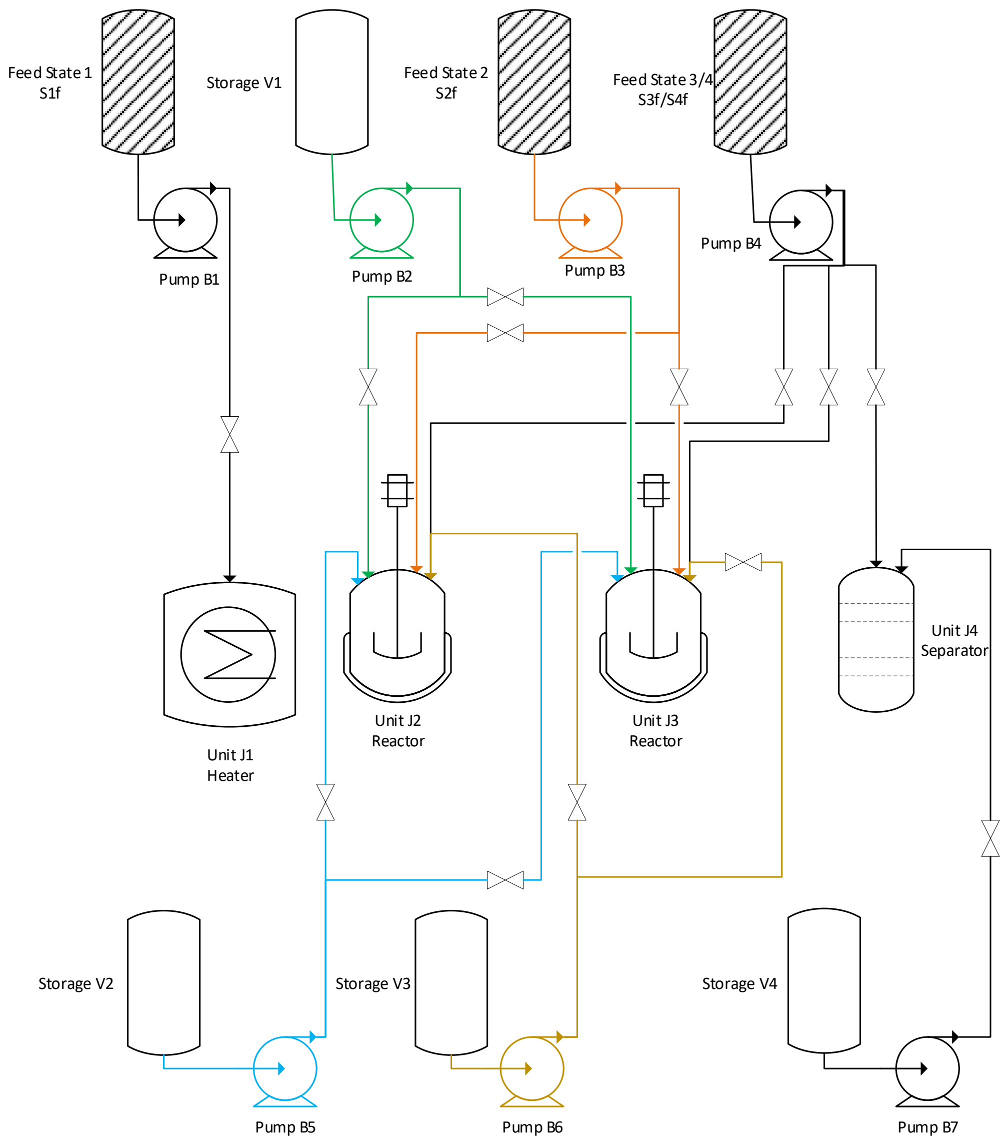


Figure 4-14 Second illustrative example: conventional pump allocation for pumping to units

As shown in the first illustrative example, constraints (3.3) and (3.4) are key for pump allocation. For the second illustrative example, constraint (3.3) is employed as follows, when the destination unit is J1 from source vessel J2, using a pump A1 at a given time point, p.

$$\mu(J1, J2, A1, p) + \mu(J2, J1, A1, p) + \mu(J3, J1, A1, p) + \mu(J4, J1, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J3, J2, A1, p) + \mu(J4, J2, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, J3, A1, p) + \mu(J4, J3, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, J4, A1, p) + \mu(J3, J4, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, V1, A1, p) + \mu(J3, V1, A1, p) + \mu(J4, V1, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, V2, A1, p) + \mu(J3, V2, A1, p) + \mu(J4, V2, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, V3, A1, p) + \mu(J3, V3, A1, p) + \mu(J4, V3, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, V4, A1, p) + \mu(J3, V4, A1, p) + \mu(J4, V4, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, S2f, A1, p) + \mu(J3, S2f, A1, p) \leq e(A1)$$
$$\mu(J1, J2, A1, p) + \mu(J2, S3f, A1, p) + \mu(J3, S3f, A1, p) \leq e(A1)$$

Constraint (3.3) is similarly employed to cater for any other destination unit j, from source vessel v_s, using pump ζ, at time point p. constraint (3.4) is employed as follows to transfer material to destination storage vessel V2, from source unit J2, using pump A1 at time point p.

$$\mu(V2, J2, A1, p) + \mu(Prod\ 1, J3, A1, p) \leq e(A1)$$
$$\mu(V2, J2, A1, p) + \mu(Prod\ 2, J4, A1, p) \leq e(A1)$$
$$\mu(V2, J1, A1, p) + \mu(V1, J1, A1, p) \leq e(A1)$$
$$\mu(V2, J2, A1, p) + \mu(V2, J3, A1, p) \leq e(A1)$$
$$\mu(V2, J2, A1, p) + \mu(V3, J3, A1, p) + \mu(V3, J4, A1, p) \leq e(A1)$$
$$\mu(V2, J2, A1, p) + \mu(V4, J3, A1, p) \leq e(A1)$$

Constraint (3.4) is similarly employed to cater for any other destination storage vessel v_d, from source unit j, using pump, ζ, at time point p.

4.3.1 Results for Second Illustrative Example

The MILP problem was solved using the CPLEX solver. Figure 4-15 and Figure 4-16 show the deconstructed view of the resultant optimal allocation of pumps to the batch plant. A total of three pumps are required, which is significantly fewer than the eleven pumps required when the plant is operated in a conventional manner.

The model results are summarized in Table 4-10. The results for the case study as reported by Lin and Floudas (2001) are presented for comparison.

Table 4-10 Second illustrative example: results summary

	This work	Lin and Floudas (2001)
Objective value (cost units)	1.83×10^6	573
CPU solution time (seconds)	22.9	22.49
Discrete Variables	8 995	128
Continuous Variables	11 793	341
Time points used	4	6
Notes		Not annualized cost objective, excludes feed costs, pump cost, piping connections, includes minimum throughput requirements, varying design costs.

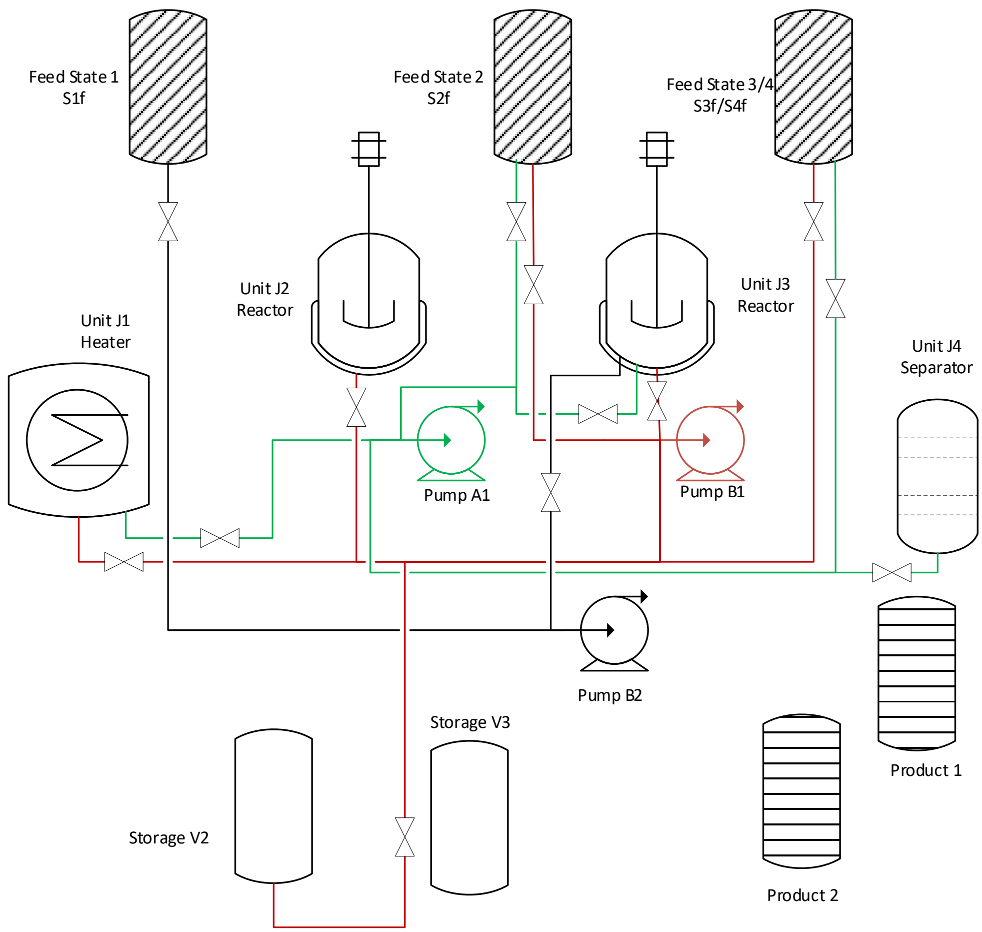


Figure 4-15 Second illustrative example: optimal pump inlet allocation

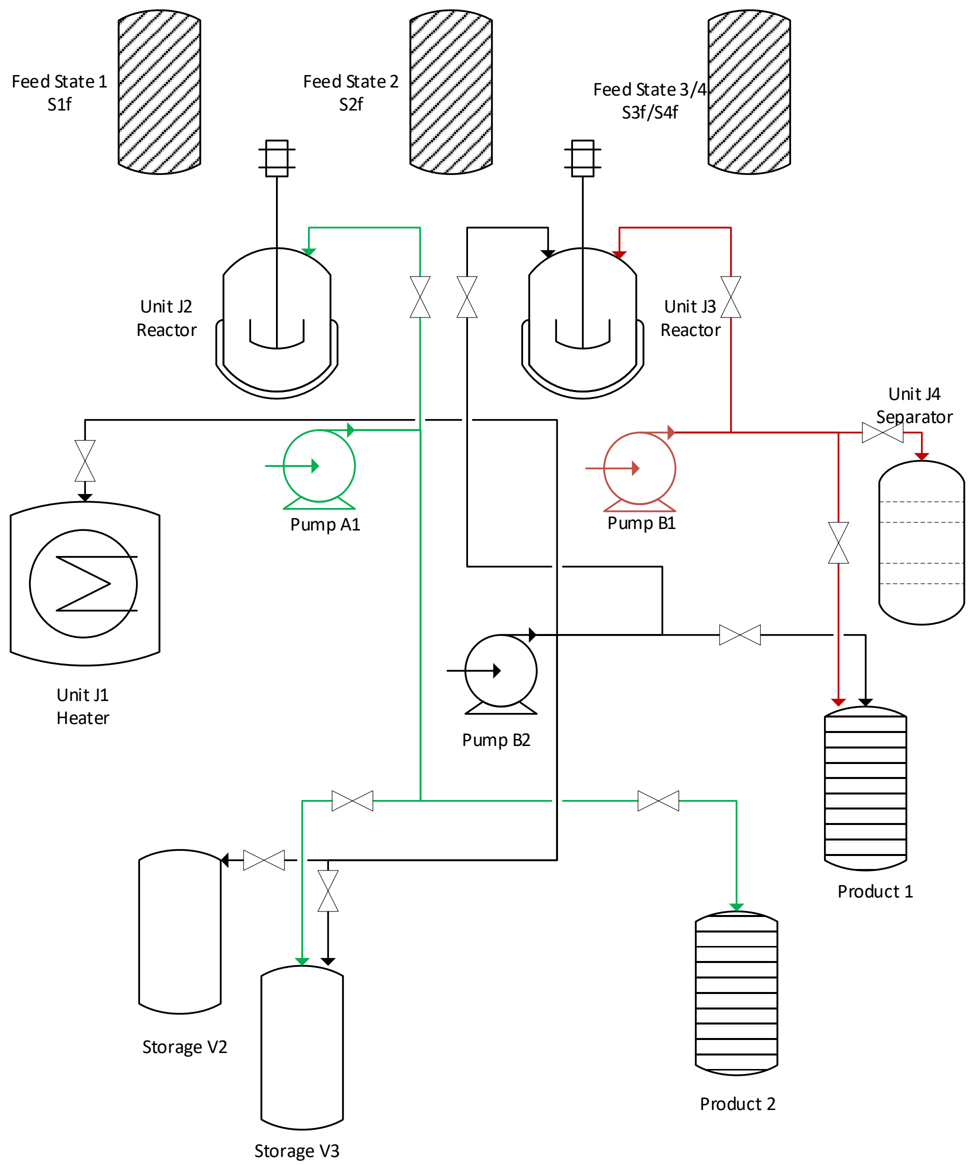


Figure 4-16 Second illustrative example: optimal pump outlet allocation

The pumping schedule for the second illustrative example may be found in Figure 4-17. Similarly to the pumping schedule obtained for the first illustrative example, it leaves the order of material transfer using the same pump to the discretion of the user. The resultant production schedule may be found in Figure 4-18.

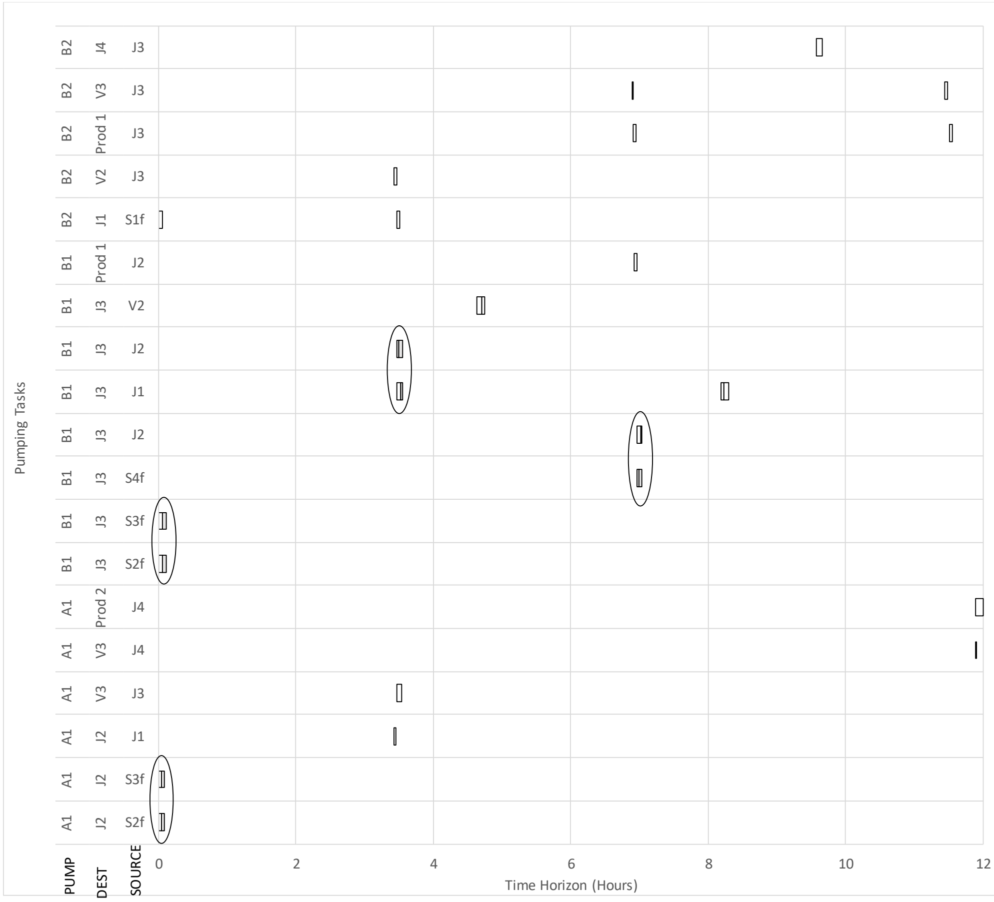


Figure 4-17 Optimal pumping schedule obtained for the second illustrative example using roving pumps

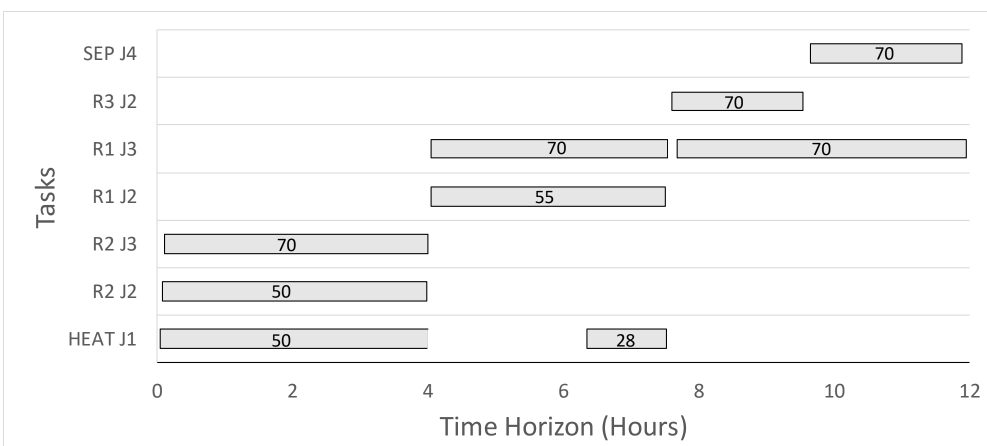


Figure 4-18 Optimal production schedule obtained for the second illustrative example using roving pumps

When the plant is operated conventionally under the same conditions, an optimal objective value of 1.3 x 10^6 cost units is obtained. The corresponding optimal production schedule for when the batch plant is conventionally operated may be found in Figure 4-19. For this case it is clear that by incorporating roving pumps, it is possible to increase the batch plant throughput significantly.

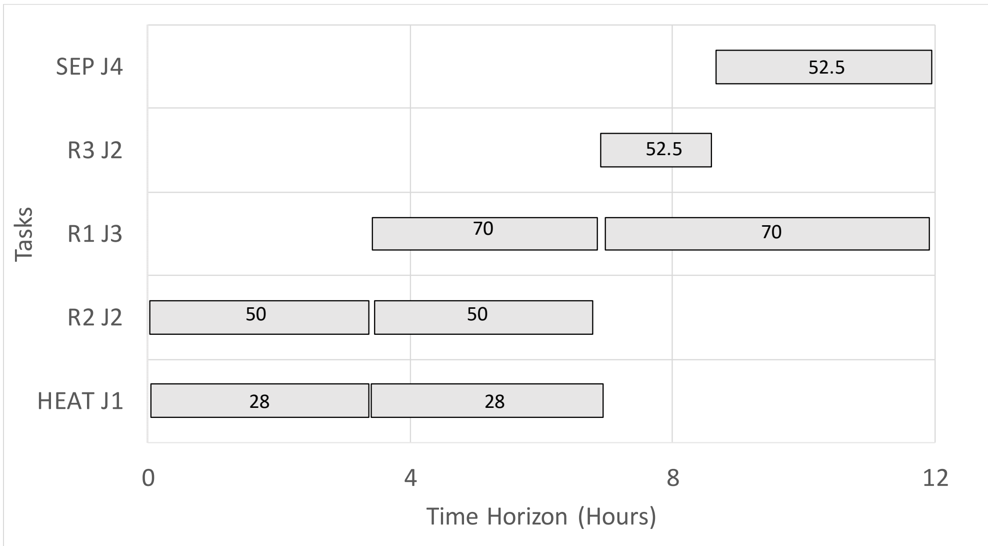


Figure 4-19 Optimal production schedule obtained for the second illustrative example using conventional operation

The optimal production schedule for the second illustrative example as given by Lin and Floudas (2001) is shown in Figure 4-20. It again illustrates that transfer time and pumping allocation was not part of the original formulation. Furthermore, the production schedule of Lin and Floudas (2001) shows many tasks of smaller quantity compared to the schedule of this work, which shows fewer tasks of greater quantity.

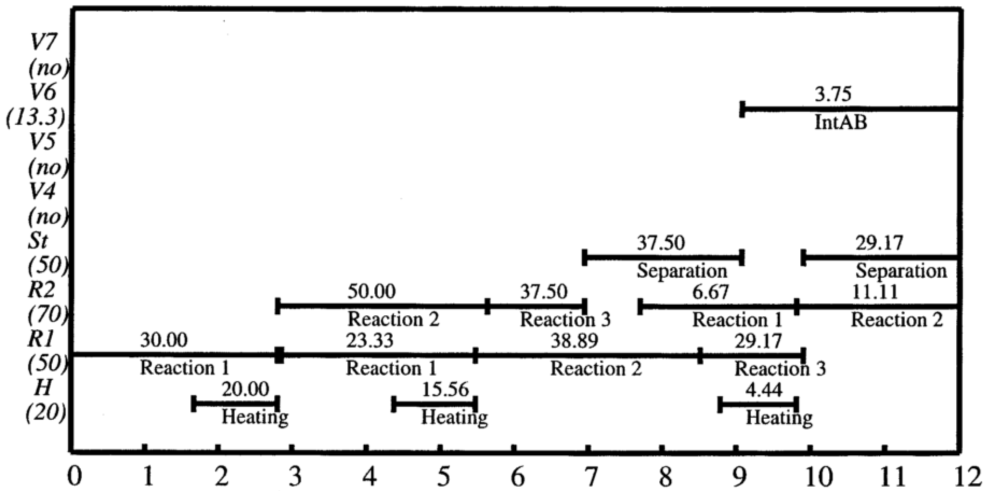


Figure 4-20 Optimal schedule for the second illustrative example as presented by Lin and Floudas (2001)

Similar to the first illustrative example, the scalability of the proposed formulation on the second illustrative example was evaluated. The effect of the number of available pumps on the second illustrative example may be seen in Table 4-11, while the effect of a varying number of time points may be seen in Table 4-12.

Table 4-11. Effect of available pumps on the scalability of the second illustrative example

Available number of pumps	**Objective value (c.u)**	**CPU solution time (s)**
1	0.4×10^6	0.6
3	1.8×10^6	3.1
5	1.8×10^6	7.7
7	1.8×10^6	16.2
9	1.8×10^6	18.7
11	1.8×10^6	31.6

Table 4-12. Effect of number of time points on the scalability of the second illustrative example

Number of time points	Objective value (c.u)	CPU solution time (s)
1	0	0.3
2	0.2×10^6	0.8
3	0.7×10^6	2.1
4	1.8×10^6	3.1

The second illustrative example is by its nature more complex than the first illustrative example. Therefore, an increase in the number of pumps and the number of time points have a greater impact on the required solution time for the second illustrative example. Nevertheless, while more significant in the second illustrative example, the required solution times are not excessive.

Therefore, in general, more complex multipurpose batch plants with a large number of processing equipment, storage vessels, pumps and piping connections will require longer solution times. There may, however, be exceptions to this. Ultimately, the solution time is dependent on the structure of the particular case.

4.4 Case Study I

The formulation was also applied to an industrial case study given by Majozi (2010). In the case study, it is desired to produce a monosodium-based herbicide. Three different reactions take place to produce the desired intermediates. Settlers and evaporators then assist to purify the final product. The original flowsheet provides two reactors to manufacture an arsenate salt solution from States 1 and 2 (S1 & S2). The produced arsenate salt is then transferred to another set of reactors. These two reactors are each able to conduct two different reactions. The first of which is to transform the arsenate salt into a disodium salt and the second is to transform the disodium salt into a monosodium salt using States 4 and 6 (S4 & S6), respectively. The monosodium salt liquor is then sent to three settlers, where the solid byproduct is discarded as waste. Only the liquid streams appear on the flowsheet as usable states. The solution is sent downstream to the two evaporators where excess liquid is evaporated.

The final solution from the evaporators is the product (S10). Four storage vessels are originally available to be used, the first is for the arsenate salt solution, the second for the disodium salt solution, the third for monosodium salt solution and the final storage vessel may be used for the decanted monosodium salt solution. The recipe representation may be seen in Figure 4-21 and the original deconstructed views of the original flowsheet in Figure 4-22 and Figure 4-23.

Data pertaining to the equipment used in the case study may be found in Table 4-13. Relevant economic data may be found in Table 4-14. In this case study all of the pumps are identical and are capable of transferring material at a rate of 12 metric tons/hour. A time horizon of 24 hours is to be considered, allowing for unlimited wait in the processing units. Processing time is batch size dependent.

The results from the previous examples showed that a very limited number of pumps are required, therefore, it may be sufficient to introduce a limited number of these pumps to the formulation in order to facilitate solution times. Eight of the original seventeen pumps were introduced into the model to be potentially used. However, if the solution requires all eight pumps to be used, then it is advisable to reintroduce all seventeen identical pumps into the problem in order to determine the optimal number of pumps. Additionally, if all the pumps were not identical, it would be advisable to add all seventeen distinct pumps into the formulation.

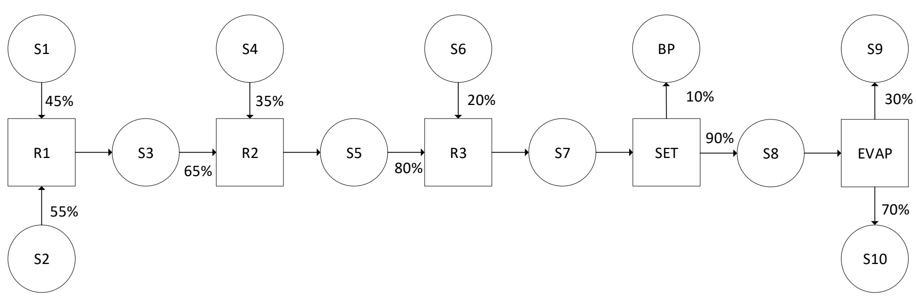


Figure 4-21 STN representation for Case Study I

Table 4-13 Equipment data for Case Study I

Unit/ Storage Vessel	**Capacity (ton)**	**Suitability (Task/Storage)**	**Task Duration (Hours)**	**Cost (Cost Units)**
Unit J1	2-10	R1	2 + 0.1mu	10 000
Unit J2	2-10	R1	2 +0.1 mu	10 000
Unit J3	2-10	R2 R3	3 + 0.1mu 1 + 0.05mu	12 000
Unit J4	2-10	R2 R3	3 + 0.1mu 1+0.05mu	12 000
Unit J5	2-10	SET	1 + 0.11mu	20 000
Unit J6	2-10	SET	1 + 0.11mu	20 000
Unit J7	2-10	SET	1 + 0.11mu	20 000
Unit J8	2-10	EVAP	3 + 0.1mu	30 000
Unit J9	2-10	EVAP	3 + 0.1mu	30 000
Vessel 1	100	S3	-	3 000
Vessel 2	100	S5	-	1 500
Vessel 3	100	S7	-	1 000
Vessel 4	100	S8	-	2 000

Table 4-14 Economic data for Case Study I

Material/Equipment	**Price (Cost units)**
S1	200 / ton
S2	100 / ton
S4	50 / ton
S6	60 / ton
S10 (product)	700 / ton
Pump base cost	1 000
Pump rating cost	10
Piping cost per connection	500

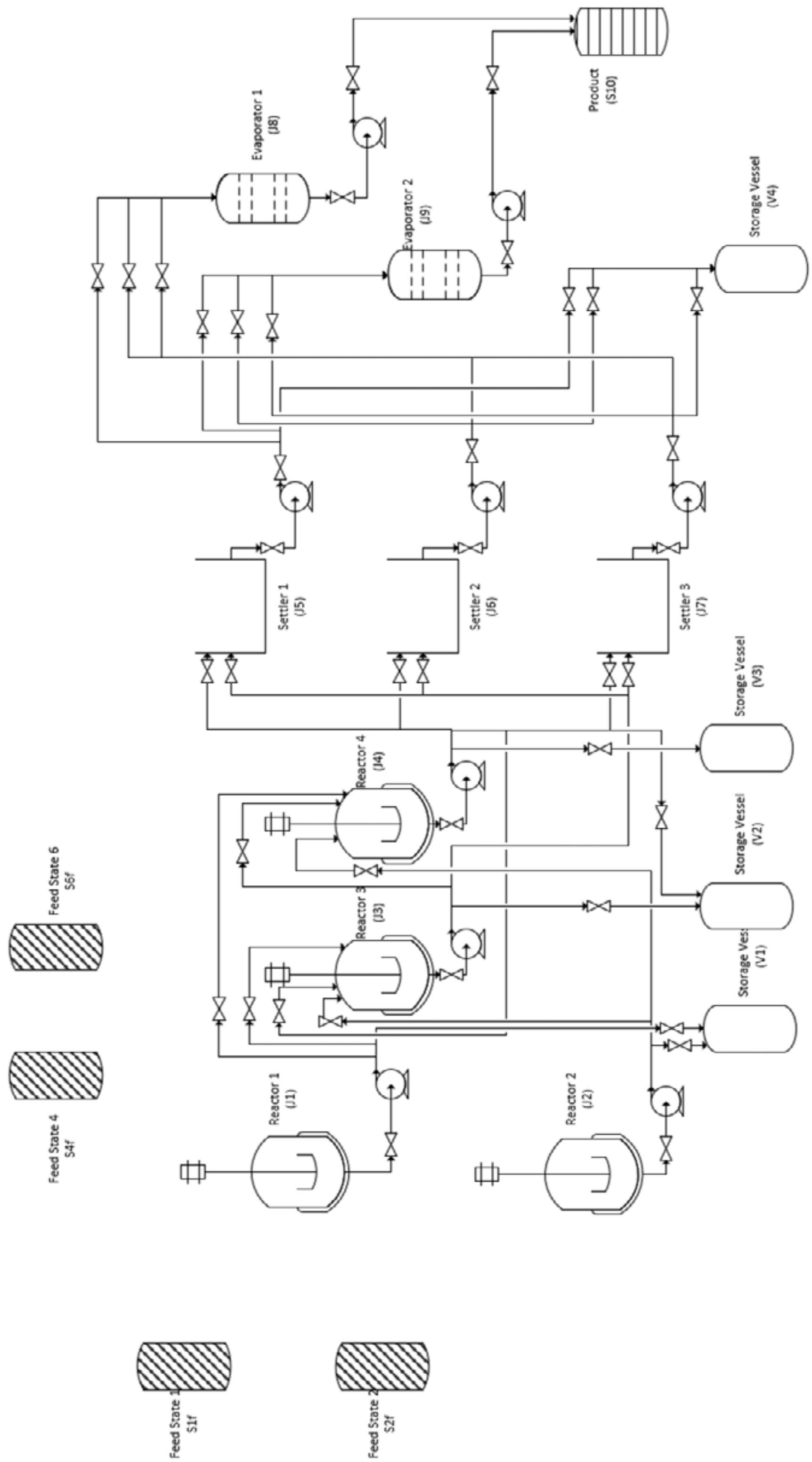


Figure 4-22 Case Study I: conventional pump allocation for pumping from units

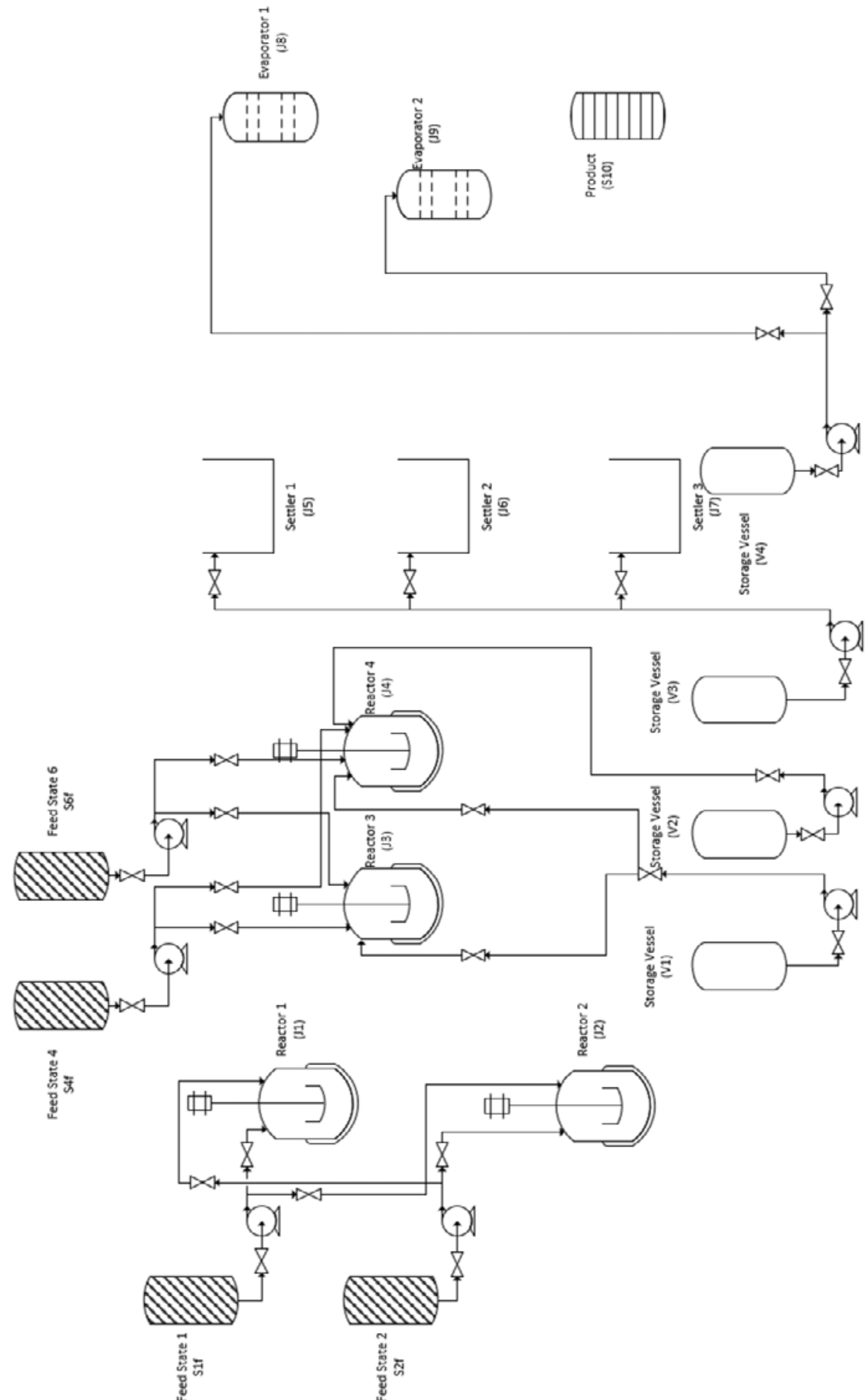


Figure 4-23 Case Study I: conventional pump allocation for pumping to units

4.4.1 Results for Case Study I

The MILP problem was solved using the CPLEX solver. Deconstructing the optimal pumping structure in a similar fashion as with the flowsheet, yields the simplified view for the pumping inlet connections and the pumping outlet connections, as shown in Figure 4-24 and Figure 4-25 respectively. By flexibly allocating pumps, the plant required four pumps to transfer material throughout the batch plant whereas 17 pumps were required when flexible pump allocation was not considered.

The results for the case study are outlined in Table 4-15. Majozi (2010) did not report results for the given case study for comparison.

Table 4-15 Case Study I: results summary

	This work	Majozi (2010)
Objective value (cost units)	2.90×10^6	-
CPU solution time (seconds)	221	-
Discrete Variables	15 972	-
Continuous Variables	20 840	-
Time points used	6	
Notes		Results not provided

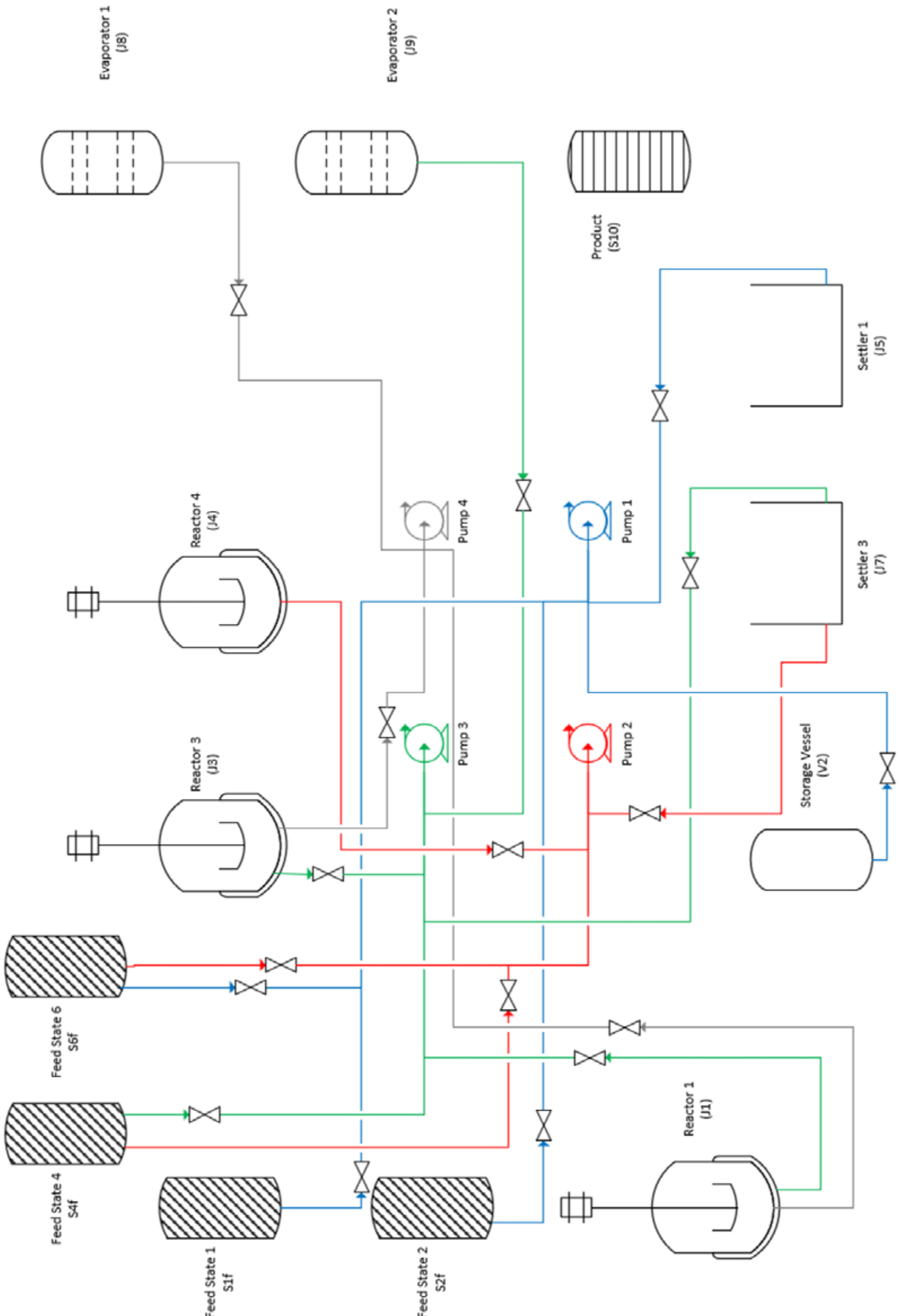


Figure 4-24 Case Study I: optimal pump allocation (pump inlet)

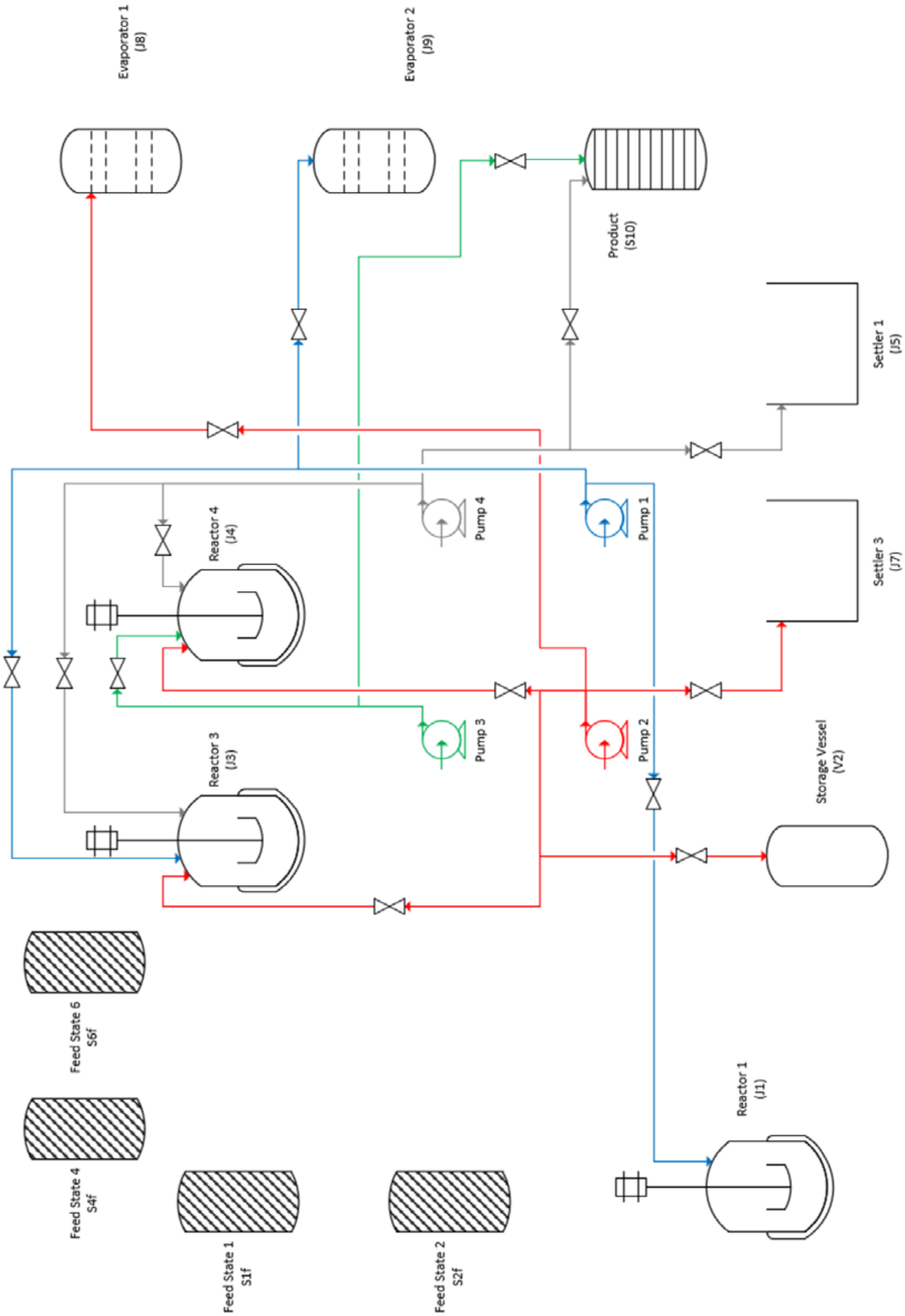


Figure 4-25 Case Study I: optimal pump allocation (pump outlet)

Notably, the formulation eliminated the need for 1 reactor (J2), one Settler (J6), three storage vessels (V1, V3 and V4) as well as 13 pumps. Figure 4-26 shows the optimal pumping schedule required for Case Study I when pumps are flexibly assigned. As with the literature examples, the pumping tasks that are at the discretion of the user, are highlighted. These correspond to the processing tasks that require multiple pumping tasks. The optimal production schedule using roving pumps may be found in Figure 4-27.

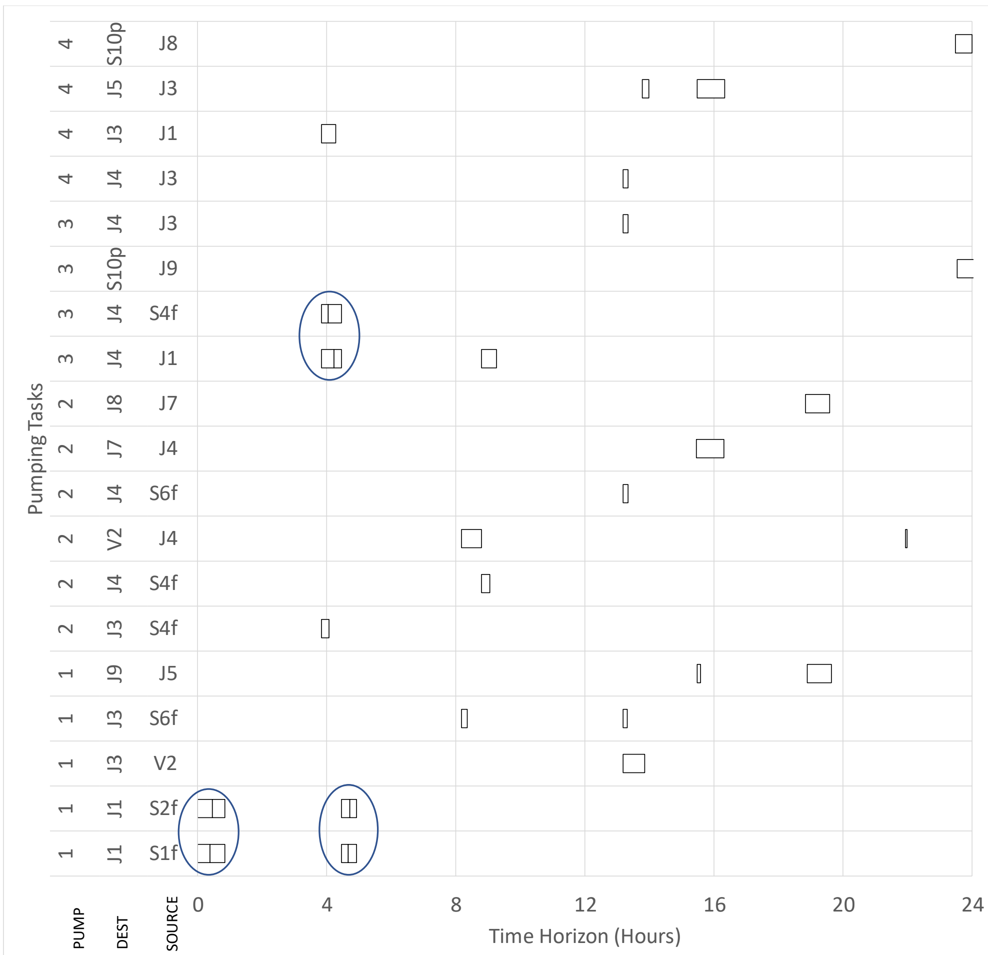


Figure 4-26 Optimal pumping schedule obtained for Case Study I using roving pumps

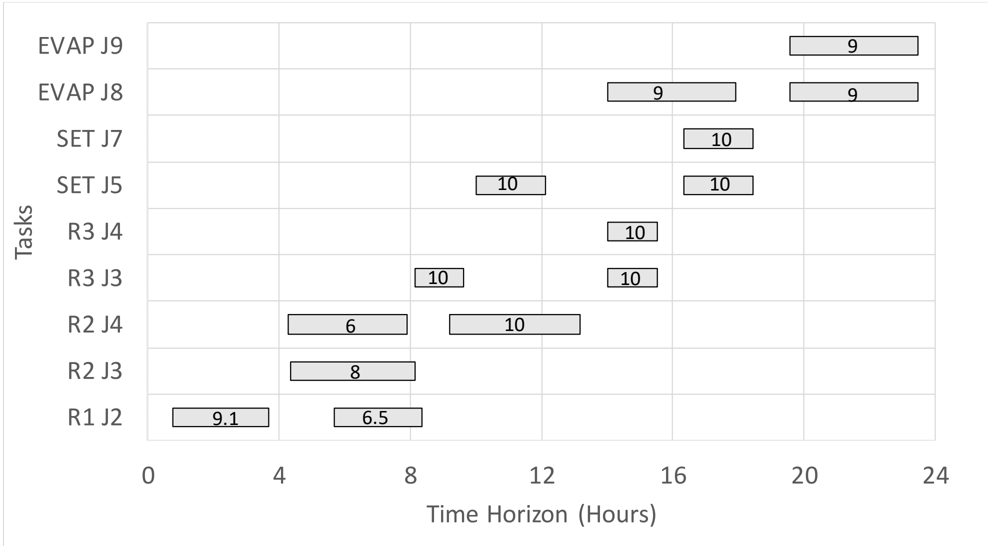


Figure 4-27 Optimal production schedule obtained for Case Study I using roving pumps

In this case, when the conventional operation is employed, an optimal objective value of 1.82 x 10^6 is obtained along with the optimal production schedule shown in Figure 4-28. Employing roving pumps again allows for improved throughput compared to conventional operation.

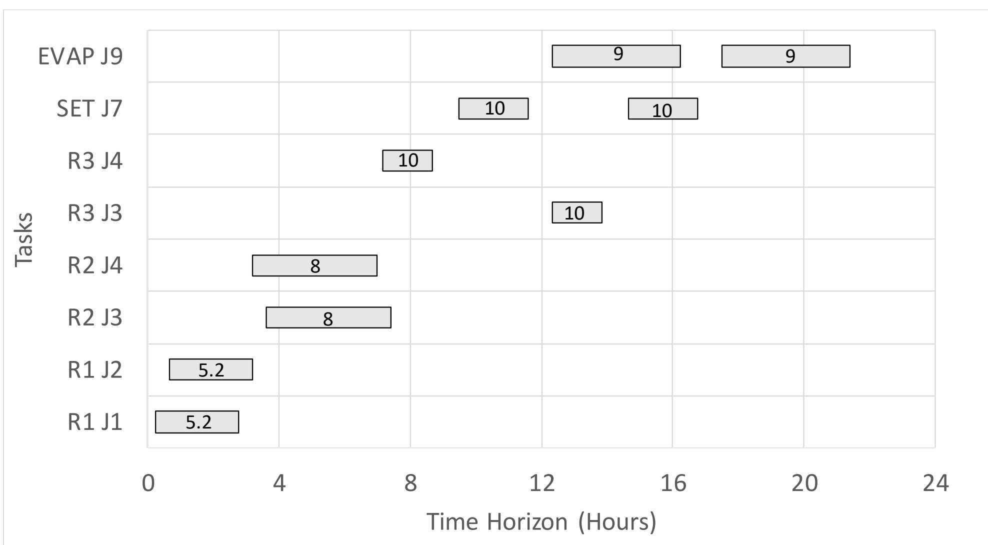


Figure 4-28 Optimal production schedule obtained for Case Study I using conventional operation

4.5 Case Study II

The formulation was also applied to a benchmark case study presented by Seid and Majozi (2014). The case study pertained to a petrochemical plant and was adapted to include flexible pumping allocation. The recipe representation may be seen in Figure 4-29 and the original deconstructed views of the original flowsheet in Figure 4-30 and Figure 4-31. The case study is considered to be very complex due to its non-serial nature. The original flowsheet contains 10 reactors, 11 storage vessels, 19 states and 20 pumps.

Relevant economic data may be found in Table 4-16. Data pertaining to the equipment used in the case study may be found in Table 4-17. In this case study all of the pumps are identical and are capable of transferring material at a rate of 60 metric tons/hour. Eight of the original 20 pumps were introduced into the model to be potentially used. A time horizon of 60 hours is to be considered. The case study allows for the Mixed Intermediate Storage (MIS) policy to be applied with finite storage, unlimited storage and no intermediate storage. The zero-wait philosophy is to be applied to operations. Processing units require that their full capacity be utilized for each batch.

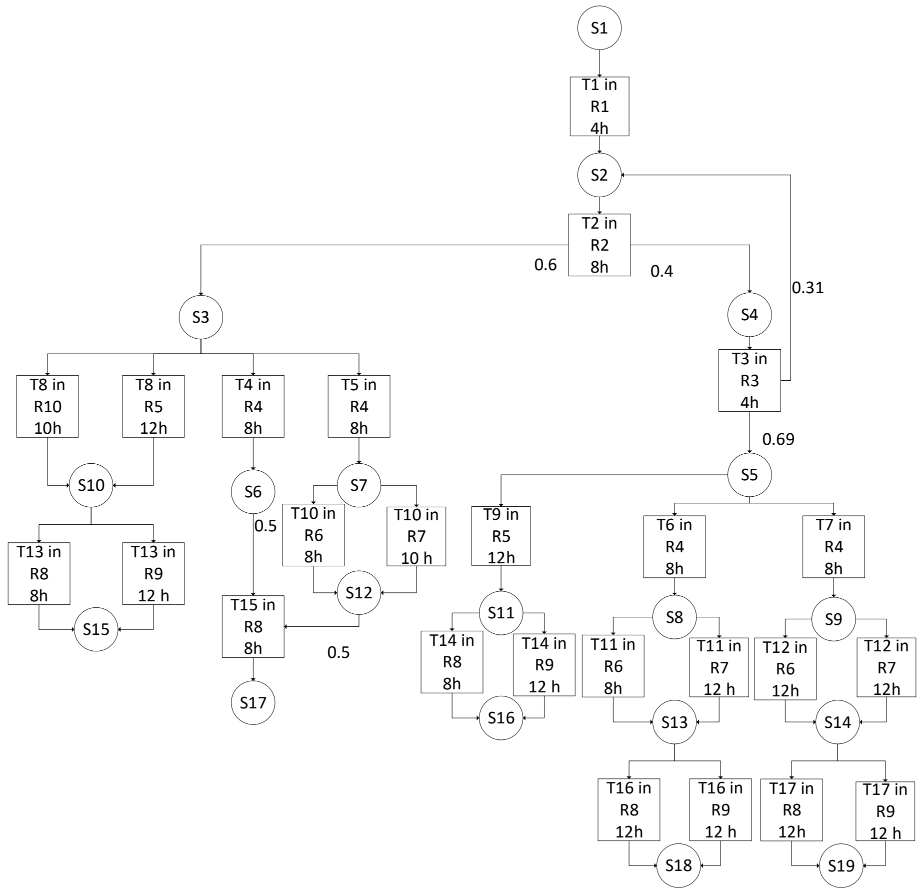


Figure 4-29 STN representation for Case Study II

Table 4-16 Economic data for Case Study II

Material/Equipment	Price (Cost units) [thousands]
S1	4 / ton
S15	5 / ton
S16	5 / ton
S17	10 / ton
S18	7.5 / ton
S19	7.5 / ton
Pump base cost	8
Pump rating cost	0.05
Piping cost per connection	1

Table 4-17 Equipment data for Case Study II

Unit/ Storage Vessel	**Capacity (ton)**	**Suitability (Task/Storage)**	**Task Duration (Hours)**	**Cost (Cost Units) [thousands]**
Unit R1	100	T1	4	100
Unit R2	100	T2	8	100
Unit R3	100	T3	4	12
Unit R4	100	T4	8	12
		T5	8	
		T6	8	
		T7	8	
Unit R5	150	T8	12	20
		T9	12	
Unit R6	50	T10	8	20
		T11	8	
		T12	12	
Unit R7	50	T10	10	20
		T11	12	
		T12	12	
Unit R8	100	T13	8	30
		T14	8	
		T15	8	
		T16	12	
		T17	12	
Unit R9	100	T13	8	30
		T14	12	
		T16	12	
		T17	12	
Unit R10	100	T8	10	20
Vessel 2	100	S2	-	2
Vessel 3	100	S3	-	1
Vessel 4	100	S4	-	5
Vessel 5	100	S5		6
Vessel 7	100	S7	-	2
Vessel 8	100	S8	-	1
Vessel 9	100	S9	-	4
Vessel 12	100	S12	-	3
Vessel 13	100	S13	-	2

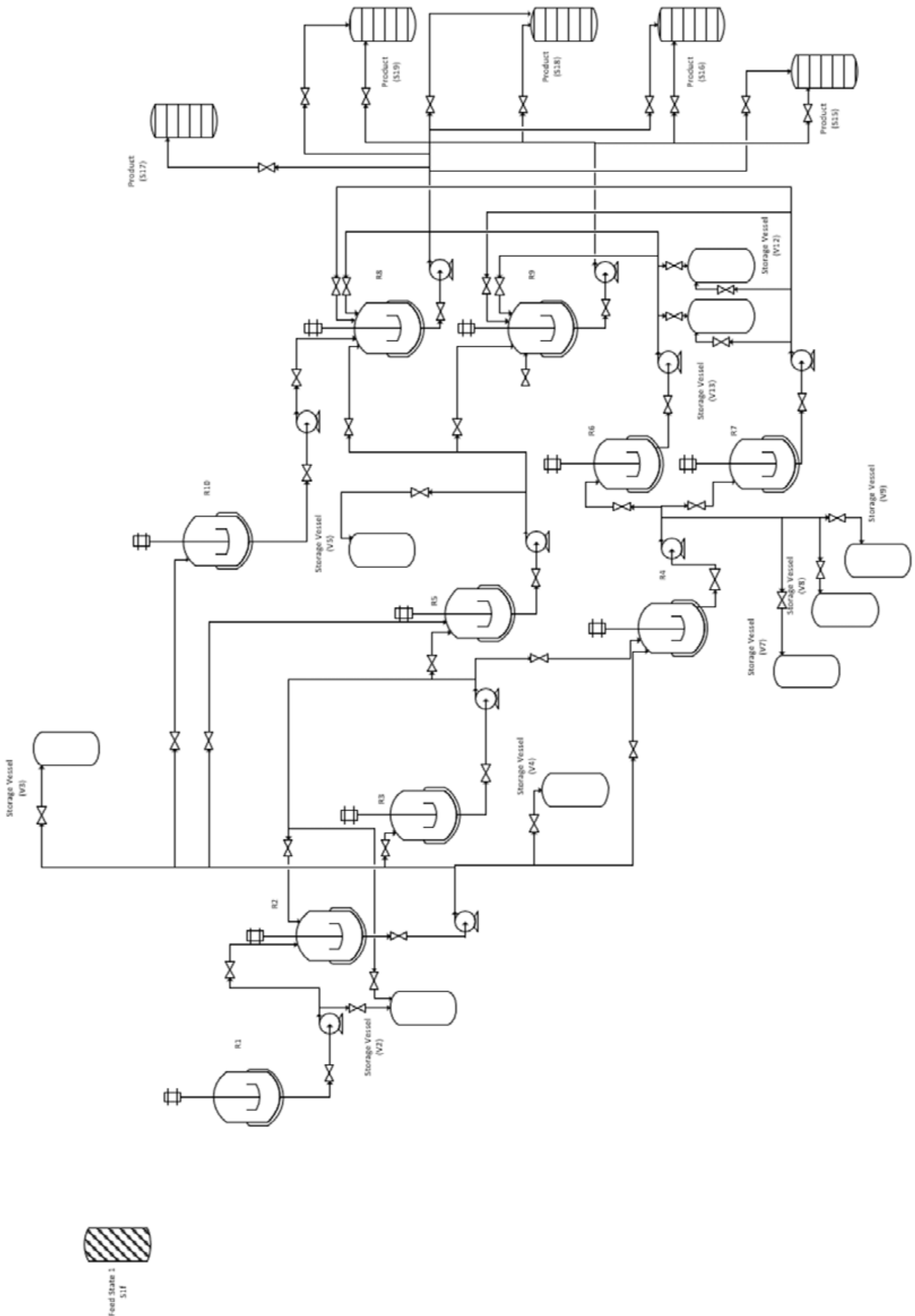


Figure 4-30 Case Study II: conventional pump allocation for pumping from units

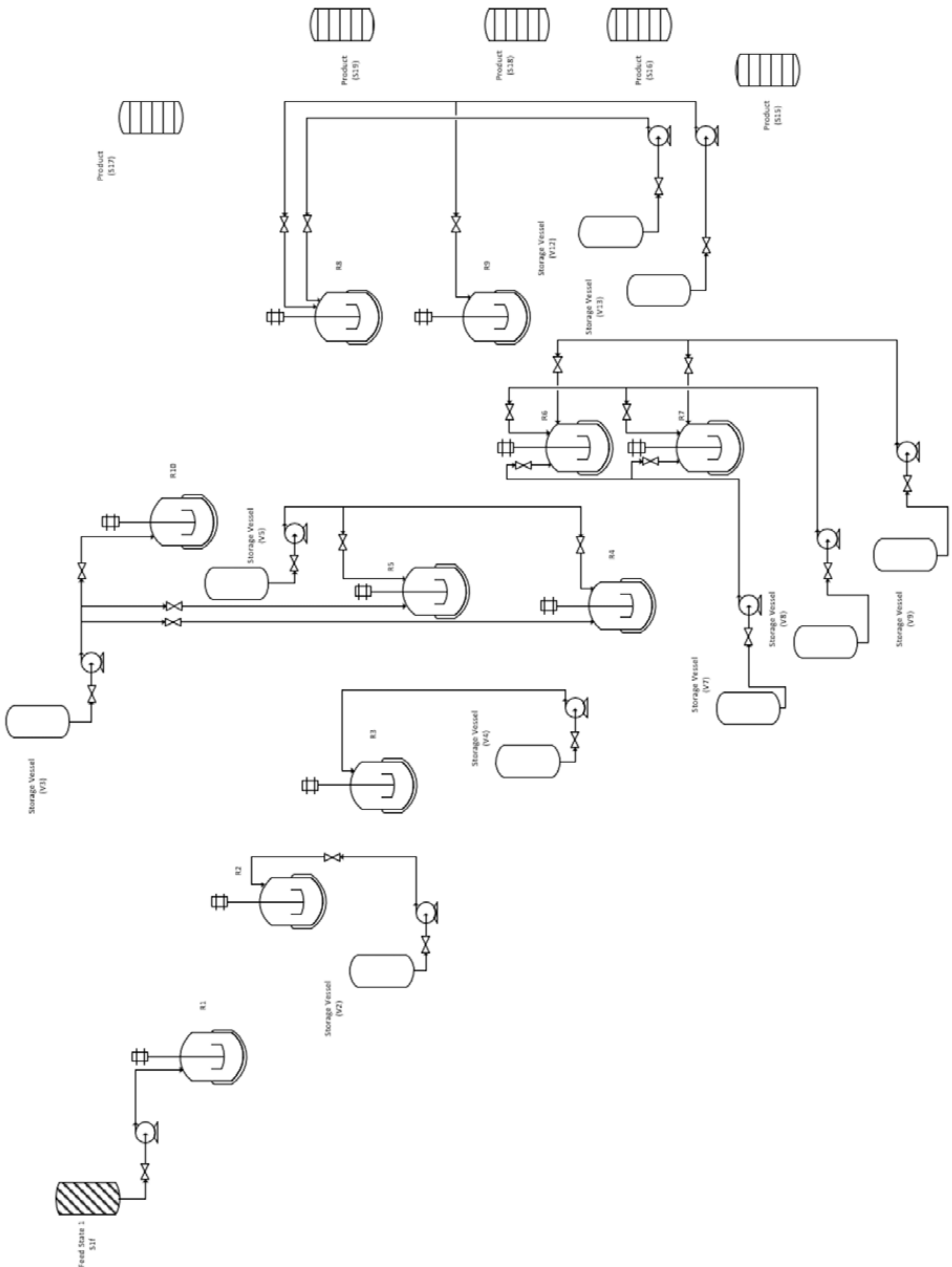


Figure 4-31 Case Study II: conventional pump allocation for pumping to units

4.5.1 *Results for Case Study II*

The MILP problem was solved using the CPLEX solver. As before, deconstructing the optimal pumping structure yields the simplified view for the pumping inlet connections and the pumping outlet connections, as shown in Figure 4-32 and Figure 4-33, respectively. By flexibly allocating pumps, the plant required three pumps to transfer material throughout the batch plant whereas 20 pumps were required when flexible pump allocation was not considered.

The case study results are summarized in Table 4-18, along with the results reported by Seid and Majozi (2014) for their case with the highest objective value.

Fundamentally the approach to the problems are different. Where this work aims only for scheduling, the work of Seid and Majozi (2014) aimed to address heat integration problems.

Table 4-18 Case Study II: results summary

	This Work	Seid and Majozi (2014) (Standalone case with direct heat integration)
Objective value (cost units)	23.7 x 10^6	7.91 x 10^6
CPU solution time (seconds)	15.2	5000 (specified)
Discrete Variables	30 617	1315
Continuous Variables	37 191	-
Time points used	6	17
Notes	60 hour time horizon	120 hour time horizon, included utility aspects and heat integration, does not include synthesis. Nonlinear.

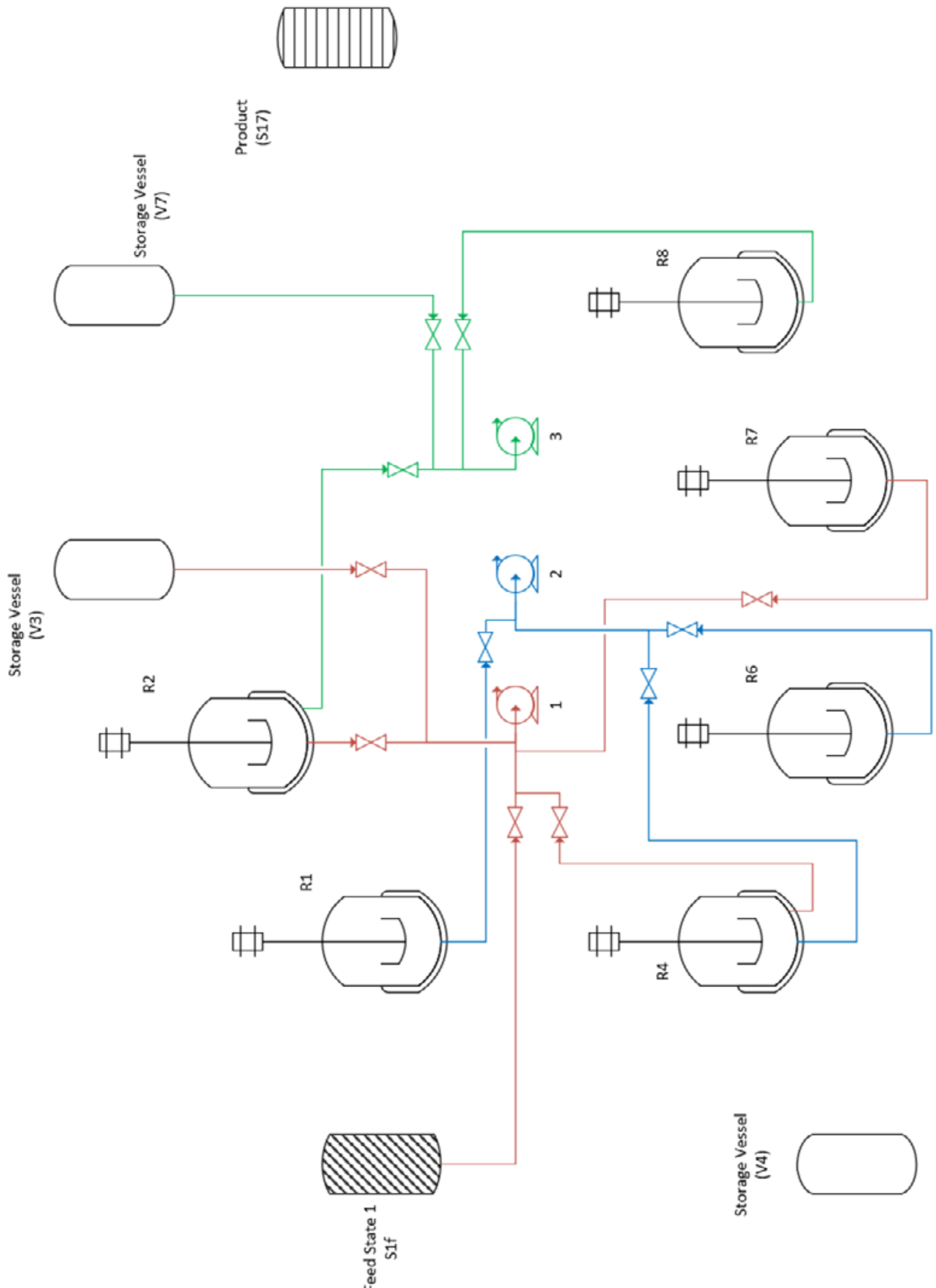


Figure 4-32 Case Study II: optimal pump allocation (pump inlet)

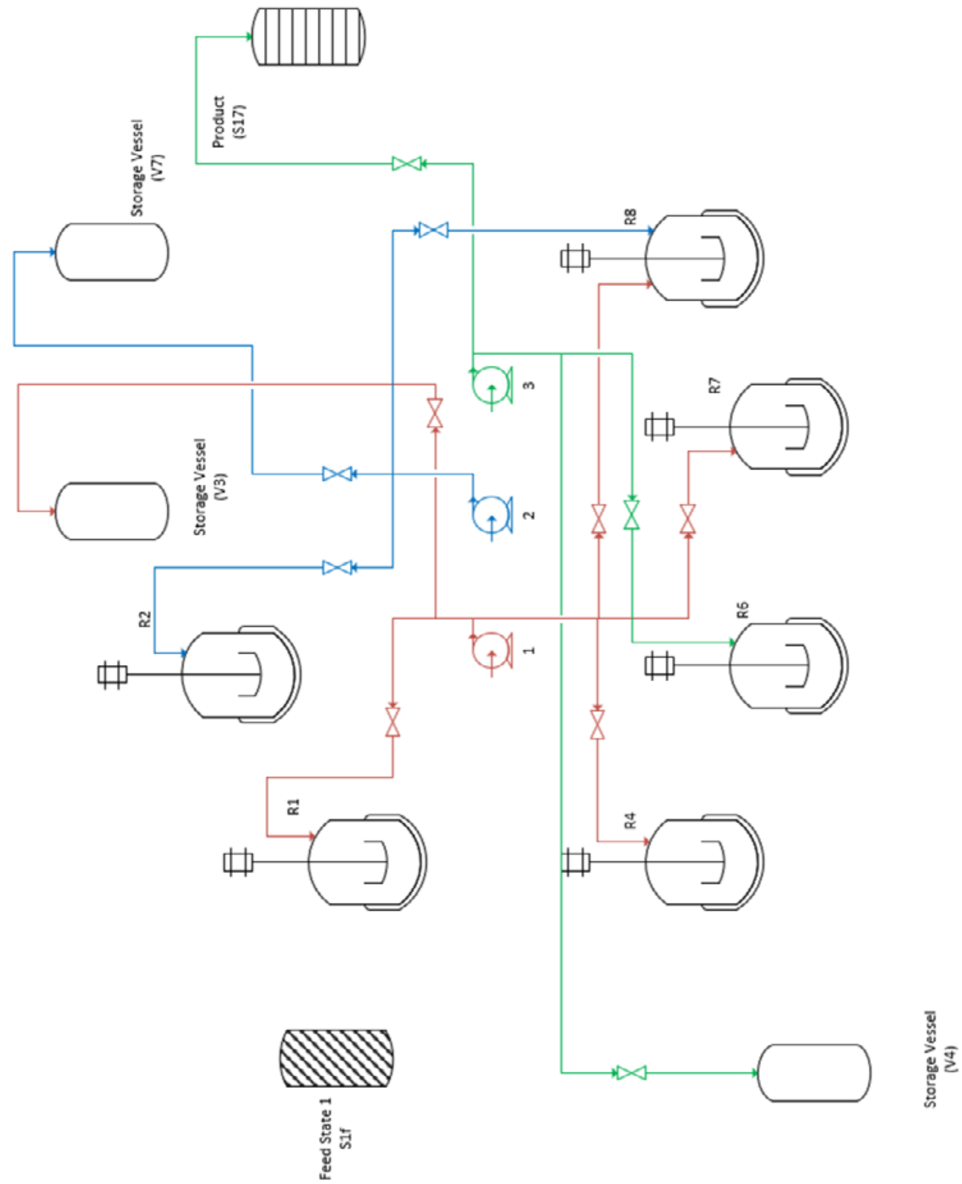


Figure 4-33 Case Study II: optimal pump allocation (pump outlet)

Notably, the formulation eliminated the need for 4 processing units (R3, R5, R9 and R10), 6 storage vessels (V2, V5, V8, V9, V12 and V13) as well as 17 pumps. Figure 4-34 shows the optimal pumping schedule required for the case study when pumps are flexibly assigned. As with the previous cases, the pumping tasks that are at the discretion of the user, are highlighted, since these correspond to the processing tasks that require multiple pumping tasks. The optimal production schedule using roving pumps may be found in Figure 4-35.

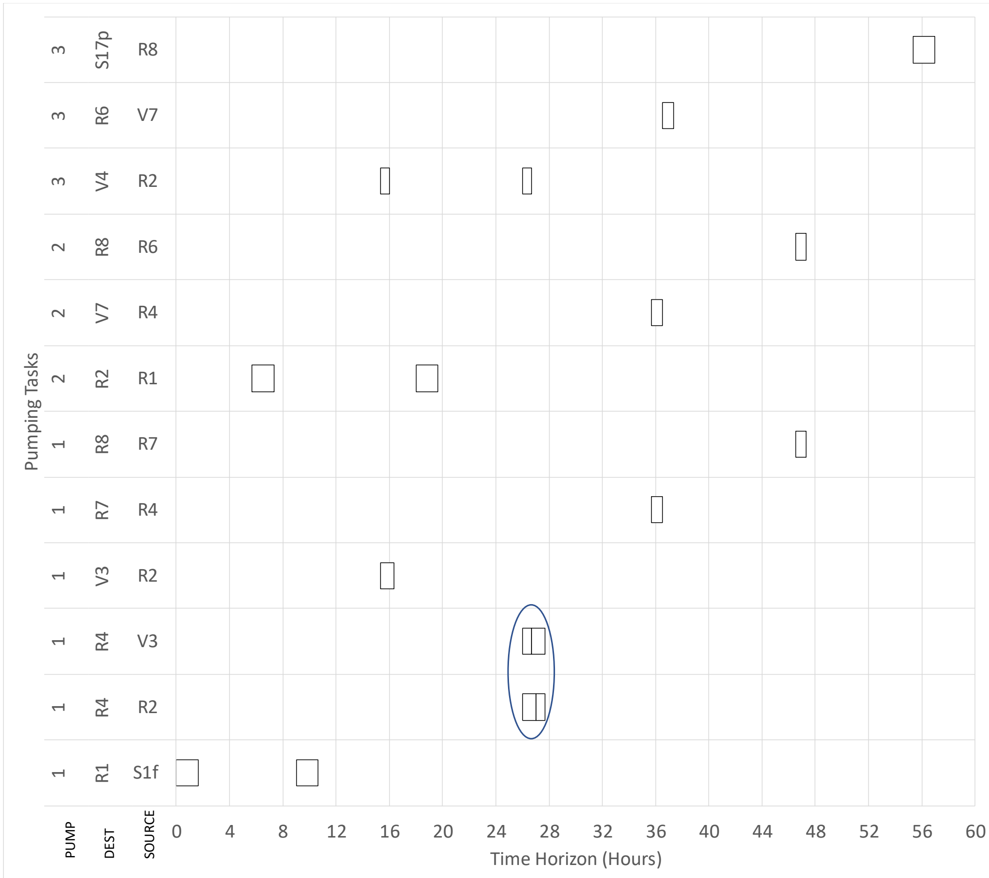


Figure 4-34 Optimal pumping schedule obtained for Case Study II using roving pumps

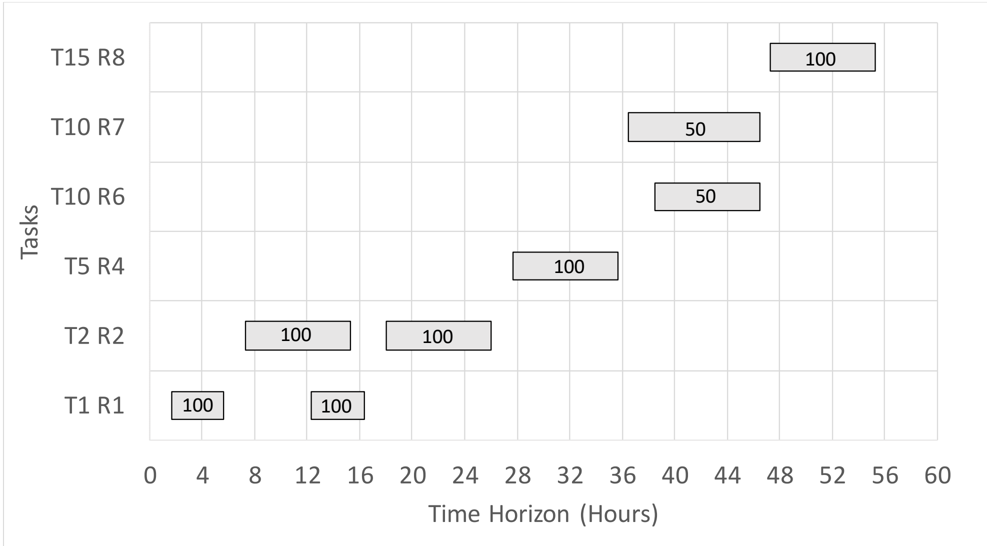


Figure 4-35 Optimal production schedule obtained for Case Study II using roving pumps

The optimal production schedule reported by Seid and Majozi (2014). Synthesis was not part of the scope of the model, with the main focus being identifying heat integration opportunities using their formulation. Their production schedule, as with the other schedules shown of literature work, illustrates again that transfer time is not incorporated between tasks.

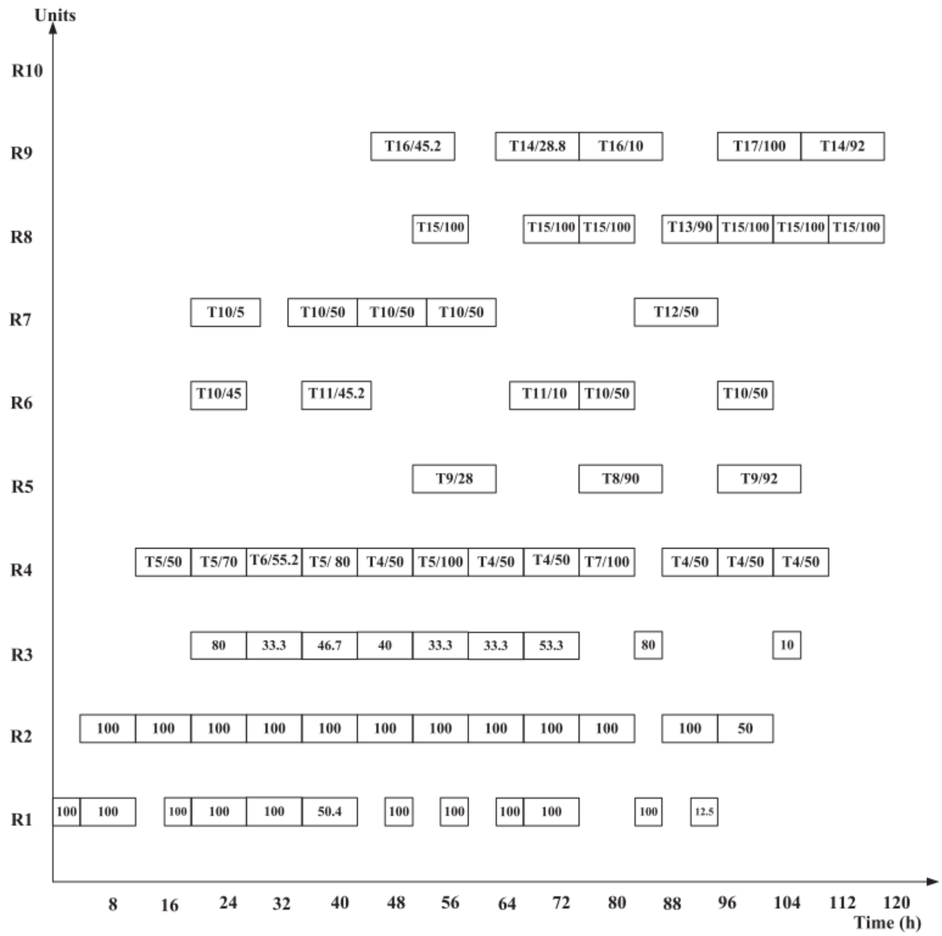


Figure 4-36 Optimal schedule for the case study as presented by
Seid and Majozi (2014)

The optimal production schedule for conventional operation of the plant under similar conditions may be found in Figure 4-37. An objective value of 23.3 x10^6 cost units is achievable with conventional operation. In this case the throughput when employing roving pumps is the same as the throughput under conventional operation. However, when employing roving pumps, fewer pumps are required.

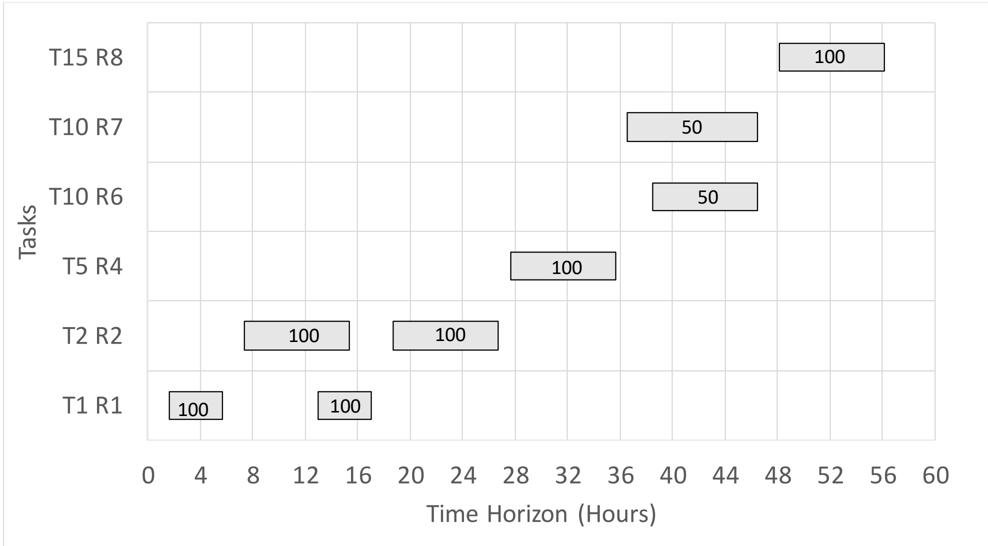


Figure 4-37 Optimal production schedule obtained for Case Study II using conventional operation.

All the cases considered were able to achieve the throughput of the conventional operated plants or improve on them significantly. This was achievable despite requiring fewer pumps than conventional plants.

Notably, in comparison to the other models found in literature, this work requires significantly more binary and continuous variables in order to yield solutions.

The proposed formulation is able to accurately schedule pumping transfers. Previous literature work did not include transfer times between sources and sinks in the batch plant when variable amounts of material is processed. This makes the proposed formulation more reliable for batch scheduling. Additionally, the proposed formulation is capable of eliminating unnecessary equipment, which can save on capital expenses during the design phase. Furthermore, the increased utilization of available equipment allows for decreased maintenance costs of redundant equipment.

4.6 References

Kondili E, Pantelides CC, Sargent RWH (1993) A general algorithm for short-term scheduling of batch operations—I. MILP formulation. Comput Chem Eng 17:211–227

Lin X, Floudas C (2001) Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. Comput Chem Eng 25:665–674

Majozi T (2010) Batch Chemical Process Integration. Springer

Seid ER, Majozi T (2014) Heat integration in multipurpose batch plants using a robust scheduling framework. Energy 71:302–320

CHAPTER 5 RECOMMENDATIONS AND CONCLUSION

5.1 Introduction

In this chapter, possible improvements to the mathematical model and solution procedure are discussed. Further addressed are the shortcomings and implications of the model. Additionally, possible future work is proposed in order to further the use of the proposed concept of roving pumps. Finally, conclusion of the dissertation is drawn.

5.2 Implications and Shortcomings

The concept of roving pumps introduces a novel method for the operation of batch chemical plants. By using and reusing a smaller number of pumps, the operation of the batch plant becomes more complex than conventionally operated plants. When transitioning from conventional operation, where pumps are used to transfer material between fixed points, to a pumping network wherein pumps are flexibly allocated to pumping tasks, it may be difficult to control operation.

Additionally, by reusing the same pumping and piping equipment, there exists the possibility of cross-contamination of material from different tasks. This may lead to quality implications if the materials are not compatible. In this work it was assumed that the materials are compatible and that cleaning is not required after tasks. However, should material contamination be a concern, cleaning in places processes are in place even for conventionally operated batch plants.

Furthermore, it may appear counterintuitive that a reduced number of pumps can accomplish the same, if not improve, the throughput of a batch plant. However, throughput can be optimized due to the optimal allocation of pumps to the relevant material transfer task.

The proposed mathematical formulation in Chapter 3 requires a large number of decisions to be made. These decisions are in the form of binary variables which make the problem complex to solve. Prominent in the proposed formulation is the binary variable that is associated with the decision on whether a particular pump is to perform a material transfer at a given time point.

Hence, an increased number of available pumps has a significant impact on the combinatorial complexity of the problem. Therefore, the formulation is best suited to small and medium batch plants with a moderate number of available pumps to choose from. Further research into mathematical optimization may be required in order to decrease the number of these binary variables to aid solution times.

Additionally, different liquids have varying viscosities and densities that affect the rate at which they can be transported using pumps. This is especially true when employing centrifugal pumps. In this formulation, the pump rating was assumed to be fixed, independent of the liquid properties. Therefore, the transfer times may be slightly affected if these considerations are included.

Modern solvers make use of improved and powerful computing hardware, while employing ever-improving techniques in order to most efficiently find the optimal solution. However, many solvers and their techniques are proprietary and expensive to obtain a license for. It is therefore recommended that research be done to find improved solving algorithms which are openly available.

5.3 Future Work

The proposed formulation addresses the base scheduling and synthesis problem and does not include aspects on utility and resource minimization. Therefore, it is recommended that the proposed formulation be incorporated with other mathematical formulations which address these aspects.

Previously, when it was aimed to exploit heat integration opportunities within the batch plant through intermittently available streams, the material transfer times were a fixed parameter, independent of the amount of material transferred (Lee *et al.*, 2015). It is therefore suggested that these formulations be applied with the variable transfer times proposed in this work in order to improve the accuracy of results.

Resource minimization generally aims to reduce the amount of water and energy used within a batch plant so as to negate the impact on the environment and reduce operational costs (Fernández *et al.*, 2012). Multipurpose batch plants generally require cleaning after batches in order to prevent product contamination. Pumping equipment and piping are not exempt from this. Therefore, it is recommended to include considerations for cleaning in place for pumping equipment and apply resource minimization techniques within the proposed formulation.

Additionally, the implications of running such a tightly operated batch plant on the ability of the plant to adapt to market fluctuations should be investigated. It may be the case that regular pump additions or removals occur in order to maintain market viability.

5.4 Conclusion

This work addresses the optimal simultaneous scheduling and synthesis of multipurpose batch chemical plants by introducing the concept of roving pumps. The proposed formulation allows the flexible relocation of limited pumping equipment to units in order to transfer material throughout the batch plant. The formulation is capable of determining the transfer times associated with variable batch sizes and incorporates the transfer times in the overall scheduling problem.

The proposed model was applied to two literature examples and two case studies. The CPLEX solver in the GAMS software suite was used in order to find optimal solutions to the problems posed in the literature examples. The results from the first example studied showed that it is possible to maintain the product throughput using a reduced number of pumps. Only two pumps were required when flexibly allocating pumping equipment compared to seven that are required when the plant is conventionally operated. The other example showed that it is possible to reduce the number of pumps from eleven to three pumps. When applied to the first case study, only 4 of the original 17 pumps were required, while in the second case study only 3 of the original 20 pumps were required.

Along with significant reduction in number of required pumps, the formulation also assists in eliminating unnecessary pieces of equipment as seen in the output from the literature examples and the case studies. This contributes to the reduced capital and maintenance costs of batch facilities.

Hence, the formulation allows significant savings on capital investment and maintenance costs, while maintaining or improving on existing product throughputs. However, the proposed method of operation is more complex than conventional operation.

It was noted that the formulation is combinatorically complex when a very large number of pumps is available to be chosen from or if the problem is significantly complex. Hence, the CPU solution time is negatively impacted when very large problems are considered. Nevertheless, the introduced concept is expected to be applied when future work on the synthesis of batch plants is studied.

5.5 References

Fernández, I., Renedo, C.J., Pérez, S.F., Ortiz, A., Mañana, M., 2012. A review: Energy recovery in batch processes. Renew. Sustain. Energy Rev. 16, 2260–2277.

Lee, J., Seid, E.R., Majozi, T., 2015. Heat integration of intermittently available continuous streams in multipurpose batch plants. Comput. Chem. Eng. 74, 100–114