

https://dspace.org/

Advances in genetic algorithm optimization of traffic signals

Kesur, Khewal Bhupendra

2008-05-29T10:13:54Z
http://hdl.handle.net/10539/4900

Downloaded from a DSpace repository. https://dspace.org/

ADVANCES IN GENETIC ALGORITHM OPTIMIZATION OF TRAFFIC SIGNALS

Khewal Bhupendra Kesur

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, in fulfilment of the requirements for the degree of Master of Science.

Johannesburg, 2007

Declaration

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

_____

(Signature of candidate)

_____ day of _____ 20 _____

Abstract

Recent advances in the optimization of fixed time traffic signals have demonstrated a move towards the use of genetic algorithm optimization with traffic network performance evaluated via stochastic microscopic simulation models. This dissertation examines methods for improved optimization. Several modified versions of the genetic algorithm and alternative genetic operators were evaluated on test networks. A traffic simulation model was developed for assessment purposes. Application of the CHC search algorithm with real crossover and mutation operators were found to offer improved optimization efficiency over the standard genetic algorithm with binary genetic operators. Computing resources are best utilized by using a single replication of the traffic simulation model with common random numbers for fitness evaluations. Combining the improvements, delay reductions between 13%-32% were obtained over the standard approaches. A coding scheme allowing for complete optimization of signal phasing is proposed and a statistical model for comparing genetic algorithm optimization efficiency on stochastic functions is also introduced. Alternative delay measurements, amendments to genetic operators and modifications to the CHC algorithm are also suggested.

Dedication

This dissertation is dedicated to my family

Acknowledgements

I would like to thank my supervisor, Professor L.P. Fatti for his patience and useful comments over the long period that this research was conducted. I would like to thank my former employer Quindiem Consulting for granting me a 3 month leave of absence for completion of the proposal phase in 2005. I would also like to thank Professor J. Galpin, for granting me ample research time in 2006. Most of all, I am thankful to my parents who support me in all my endeavours.

Note on referencing system

This dissertation makes use of the numerical referencing system. Numeric references are given in superscript enclosed in brackets. We make use of footnotes which are also referenced in superscript, but without brackets.

Table of Contents

Contents Page

List of Figures ..... **xiv**

List of Tables ..... **xvi**

List of Symbols ..... **xviii**

1 Chapter One – Introduction **1**

1.1 General Introduction 1

1.2 Background 1

1.2.1 Traffic signal operation [2, 3] 1

1.2.2 Optimization criteria 1

1.3 Study limitations 2

1.4 Objectives 2

1.5 Organization of material 3

2 Chapter Two – Undersaturated Conditions, Single Isolated Intersection **4**

2.1 Introduction 4

2.2 Terminology 4

2.2.1 Vehicular terminology [2, 6] 4

2.2.2 Traffic signal terminology [5, 6] 4

2.3 The Traffic Model 5

2.3.1 Arrival process 5

2.3.2 Departure process [5, 9] 5

2.3.3 Opposed-turning movements [2, 10, 11] 7

2.3.4 Queue evolution 8

2.4 Webster's formula 9

2.4.1 Additional terminology [2] 9

2.4.2 Computing steady-state delay 10

2.5 Deriving the Optimal Timing Plan 11

2.5.1 Two-phase intersection 11

2.5.2 More complicated junction layouts, multiphase sequencing and vehicle mix 12

2.5.3 Optimization of stage composition and sequence 13
2.5.4 Optimization criteria besides delay 14

3 Chapter Three – Undersaturated Conditions, Signalized Networks 15

3.1 Introduction 15
3.2 Network Terminology [9] 15
3.3 Bandwidth methods 16
3.3.1 Introduction [1, 2, 26] 16
3.3.2 Bandwidth maximization algorithms for arterial networks 19
3.3.3 Extensions to grid networks 20
3.4 Delay-based methods 21
3.4.1 Modelling methodology [40, 41] 21
3.4.2 Local optimization methods 24
3.4.3 Global optimization models 25
3.5 Hybrid methods 27
3.5.1 Contrast between bandwidth and delay-based approaches 27
3.5.2 Modifying bandwidth methods to consider delay 28
3.5.3 Modifying delay based methods to consider bandwidth 29

4 Chapter Four – Oversaturated Conditions 30

4.1 Introduction 30
4.2 Single Intersection 31
4.2.1 Method of Gazis and Potts 31
4.2.2 Extensions to methodology of Gazis and Potts 32
4.2.3 Other models 32
4.3 Signalized networks 33
4.3.1 Extensions to the model of Gazis and Potts 33
4.3.2 Other analytical models 33
4.3.3 Modification to the bandwidth approach 33
4.3.4 Modifications to TRANSYT-7F for oversaturated conditions 34

5 Chapter Five – Genetic Algorithms 35

5.1 Introduction 35
5.2 What are genetic algorithms and how do they work? [78, 79] 35

5.3 Simple Genetic Algorithm [78] 37
5.3.1 Introduction 37
5.3.2 Problem encoding 37
5.3.3 Initialization 38
5.3.4 Selection 38
5.3.5 Crossover 38
5.3.6 Mutation 39
5.3.7 Replacement 39
5.4 Why do genetic algorithms work? [78, 79] 39
5.4.1 Introduction 39
5.4.2 Qualitative justifications of optimization steps 39
5.4.3 Genetic algorithm theory 40
5.5 Improvements to the simple genetic algorithm 42
5.5.1 Introduction 42
5.5.2 The elitist method 43
5.5.3 Uniform crossover [86, 87] 43
5.5.4 Ranking and tournament selection 43
5.5.5 Alternative algorithms 45
5.5.6 Alternative problem encodings 48
5.5.7 Optimization in noisy environments 52
5.6 Genetic algorithms in traffic signal optimization 53
6 Chapter Six – Research Questions 55
6.1 Introduction 55
6.2 Motivation 55
6.2.1 Why do we need a better genetic algorithm? 55
6.2.2 Why use a stochastic traffic simulation model? 56
6.3 Research Questions 56
6.3.1 Introduction 56
6.3.2 Re-evaluation of fitness in GGA's 56
6.3.3 Common Random Numbers 57
6.3.4 Number of replications 58
6.3.5 Alternative problem encodings 59

6.3.6 Algorithm type 59
6.3.7 Optimal parameter tunings 60

7 Chapter Seven – Methodology 61

7.1 Introduction 61
7.2 MSTRANS 61
7.3 Test networks 62
7.4 Simulation run length 63
7.5 Optimization objective 64
7.6 Genetic algorithm problem encoding 65
7.6.1 Introduction 65
7.6.2 Phasing 65
7.6.3 Genetic encoding 69
7.7 Modifications to real genetic operators 72
7.7.1 Introduction 72
7.7.2 Blend crossover 73
7.7.3 Real mutation 73
7.8 Genetic algorithms and parameter values 74
7.9 Comparisons of genetic algorithms 75
7.9.1 Introduction 75
7.9.2 Multiple runs 76
7.9.3 Number of function evaluations 76
7.9.4 Obtaining an unbiased measure of solution quality 77
7.9.5 Construction of sample statistics 77
7.9.6 Statistical tests 78
7.10 Experimental design 80

8 Chapter Eight – Results 81

8.1 Introduction 81
8.2 Re-evaluation of fitness in GGA's 81
8.3 Common Random Numbers 83
8.4 Number of replications 85
8.5 Alternative problem encodings 87

8.5.1 Gray coding 87
8.5.2 Real Crossover 88
8.5.3 Real mutation 89
8.5.4 Real CHC 91
8.6 Algorithm Type 92
8.7 Optimal parameter tunings 93
8.7.1 Blend crossover parameter 94
8.7.2 Population size 95
8.8 Overall improvement 96
9 Chapter Nine – Conclusions and Recommendations 99
A1 Appendix One – Microscopic Stochastic Traffic Network Simulator 105
A1.1 Introduction 105
A1.2 Algorithmic structure 105
A1.2.1 Initialization 105
A1.2.2 Simulation 106
A1.2.3 Finalization 106
A1.3 MSTRANS components 106
A1.3.1 Simulation time step 107
A1.3.2 Free flow acceleration 107
A1.3.3 Stopping 111
A1.3.4 Car-following 112
A1.3.5 Lane Changing 118
A1.3.6 Routing 120
A1.3.7 Vehicle generation 120
A1.3.8 Turning movements 121
A1.3.9 Start-up delay 124
A1.3.10 Response to amber 125
A1.3.11 Behaviour in oversaturated conditions 126
A1.3.12 Computation of measures of effectiveness 127
A1.3.13 Random number generation 128
A1.4 Validation of MSTRANS 128
A1.4.1 Animation 128

A1.4.2 Examination of queue discharge 129
A1.4.3 Comparisons with CORSIM 131

A2 Appendix Two – Common Random Numbers [136] 137

A2.1 Introduction 137
A2.2 Variability in stochastic experiment output 137
A2.3 CRN's for comparing alternatives 137
A2.4 Variance reduction 137

A3 Appendix Three – Test Networks 140

A3.1 Introduction 140
A3.2 Arterial Network 140
A3.2.1 Network diagram 141
A3.2.2 Flow rates 142
A3.2.3 Turning proportions 143
A3.3 Grid Network 144
A3.3.1 Network diagram 144
A3.3.2 Flow rates 145
A3.3.3 Turning proportions 145

A4 Appendix Four – Formulae for evaluating magnitude of optimization problem ... 147

A4.1 Introduction 147
A4.2 Number of decision variables 147
A4.3 Bit string length 147

A5 Appendix Five – Real CHC 149

A5.1 Introduction 149
A5.1.1 Distance calculation 149
A5.1.2 Initial value of mating threshold 150
A5.1.3 Decrements to mating threshold 151
A5.1.4 Value of mating threshold after cataclysmic mutation 152

A6 Appendix Six – Statistical model for comparing genetic algorithm performance .. 154

A6.1 Introduction 154
A6.2 Setting assumptions of statistical model 154

A6.2.1 Variability in search trajectories 154
A6.2.2 Variability in the objective function 158
A6.2.3 Normally distributed objective function 159
A6.3 Statistical model 160
A6.3.1 Model describing distribution of observed performance measures 160
A6.3.2 Estimation of model parameters 161
A6.3.3 Testing for significant differences in performance 163
A6.4 Estimation of *a* and *b* 163
A7 Appendix Seven – Experimental Output 168
A7.1 Introduction 168
A7.2 Re-evaluation of fitness in GGA's 168
A7.3 Common Random Numbers 169
A7.3.1 GGA 170
A7.3.2 SSGA 171
A7.3.3 CHC 172
A7.4 Number of replications 172
A7.5 Alternative problem encodings 173
A7.5.1 Gray coding 173
A7.5.2 Real Crossover 174
A7.5.3 Real mutation 175
A7.5.4 Real CHC 176
A7.6 Algorithm Type 176
A7.7 Optimal parameter tunings 177
A7.7.1 Blend crossover parameter 177
A7.7.2 Population size 178
References 179

List of Figures

Figure Page

Figure 1: Departure Process 7

Figure 2: Simplified Departure Process 7

Figure 3: Opposed-turning movement 8

Figure 4: Isolated intersection with no turning movements 11

Figure 5: Arterial network 17

Figure 6: Space-time diagram for main arterial 17

Figure 7: Bandwidth in both directions 18

Figure 8: Flow at entry of an internal link 22

Figure 9: Flow at a point further downstream 23

Figure 10: Graphical illustration of blend crossover 51

Figure 11: Traffic movements at a single intersection 65

Figure 12: Staging of green for traffic movements on approach W. 66

Figure 13: Staging of green sequences for traffic movements on approach E. 67

Figure 14: Nine phasing possibilities during E/W green phase. 69

Figure 15: Comparison of algorithm performance – Arterial Undersaturated scenario 97

Figure 16: Comparison of algorithm performance – Grid Undersaturated scenario 97

Figure 17: Comparison of algorithm performance – Arterial Oversaturated scenario 98

Figure 18: Comparison of algorithm performance – Grid Oversaturated scenario 98

Figure 19: Vehicle kinematics according to the linear acceleration model 109

Figure 20: Polynomial acceleration model 110

Figure 21: Gipps model free flow acceleration curve 117

Figure 22: Turning arc's for right hand rule of the road 122

Figure 23: Comparison of CORSIM and MSTRANS – sample mean of link delay 133

Figure 24: Comparison of CORSIM and MSTRANS – sample standard deviation of link delay 134

Figure 25: Histogram of θ_i 158

Figure 26: Scatter plot of sample mean and sample variance values in Table 23 159

Figure 27: Histogram of extended network delay 160

Figure 28: Statistical model describing distribution of observed performance measures 161

Figure 29: Fit of linear variance model – Arterial Undersaturated scenario 166

Figure 30: Fit of linear variance model – Arterial Oversaturated scenario 167

Figure 31: Fit of linear variance model – Grid Undersaturated scenario 167

Figure 32: Fit of linear variance model – Grid Oversaturated scenario 167

List of Tables

Table Page

Table 1: Binary coding and Gray coding 49

Table 2: Assessment of re-evaluation mechanism in GGA 82

Table 3: Effectiveness of CRN's in GGA 83

Table 4: Effectiveness of CRN's in SGGA 84

Table 5: Effectiveness of CRN's in CHC 84

Table 6: Impact of additional replications on GGA (Using CRN's) 86

Table 7: Impact of additional replications on SSGA 86

Table 8: Impact of additional replications on CHC 87

Table 9: Impact of Gray code on genetic algorithms 88

Table 10: Impact of real crossover on genetic algorithm performance 89

Table 11: Evaluation of real mutation on performance of GGA and SSGA 90

Table 12: Evaluation of Real CHC 92

Table 13: Comparison of GGA and SSGA with Real CHC 93

Table 14: Impact of blend crossover parameter on performance of Real CHC (relative to $\alpha = 0.5$) 94

Table 15: Impact of population size of performance of Real CHC 95

Table 16: Relative improvement of enhanced genetic algorithm over the standard genetic algorithm 96

Table 17: Constant deceleration rates 112

Table 18: Amber stopping probabilities 126

Table 19: Discharge headways for a standing queue 130

Table 20: MSTRANS and CORSIM network delay in undersaturated conditions. 132

Table 21: MSTRANS and CORSIM extended network delay in oversaturated conditions. 135

Table 22: Specifications of genetic algorithm for evaluating variability in search trajectories . 154

Table 23: Sample mean and variance of extended network delay over 100 independent replications of genetic algorithm 157

Table 24: Estimated values of *a* and *b* for each test network 166

List of Symbols

Separate lists are provided for each chapter and appendix.

Chapter Two – Undersaturated Conditions, Single Isolated Intersection

q = Arrival rate

C = Cycle time

s = Saturation flow rate

g = Effective green time

x = Degree of saturation

d = Steady-state vehicle delay

N, E, S and W = Various traffic movements

q_i = Arrival rate for movement i (vehicles/hour)

$g_{N/S}$ = Effective green time for N and S movements

$g_{E/W}$ = Effective green time for E and W movements

Chapter Five – Genetic Algorithms

x = Real or integer valued decision variable

a = Lower limit of search domain

b = Upper limit of search domain

l = Length of binary string

k = Index of particular bit in binary string

y_k = Value of k'th bit in binary string

y = Decision variable in binary code

j = Index of decision variable

x_j = Value of j'th decision variable

a_j = Lower limit of search domain of j'th decision variable

b_j = Upper limit of search domain of j'th decision variable

z_j = Value of j'th decision variable in binary code

l_j = Length of binary string representing the j'th decision variable

$\underline{x}$ = Vector of decision variables

L = Length of concatenated binary string

N_{Pop} = Population size

i = Index of individual in the population

f_i = Objective function value for the i'th individual

c_k = Value of k'th bit in first parent

d_k = Value of k'th bit in second parent

c = Binary string of first parent

d = Binary string of second parent

u = Discrete uniform random variable

c^* = Binary string of first child

d^* = Binary string of second child

p_c = Probability of crossover

p_m = Probability of mutation

* = "Do not care" symbol in schema definition

t = Generation number

$f_1(x)$ and $f_2(x)$ = Example objective functions

$f(x)$ = Positive objective function to be minimized

C = Constant to convert minimization problem into a maximization problem

β = Bias factor for linear rank selection

m = Tournament size in tournament selection

r = CHC divergence rate

y_k^b = Value of k'th bit in standard binary coding of decision variable

y_k^g = Value of k'th bit in Gray coded representation of decision variable

α = Blend crossover parameter

x_i = Value of i'th decision variable in first parent

y_i = Value of i'th decision variable in second parent

z_i = Value of i'th decision variable in offspring

Chapter Seven – Methodology

N, E, S and W = Approaches at an intersection

N/S and E/W = Alternating green phases at an intersection

ρ = Duration of E/W green phase (s)

g_W^1 = Duration of stage with green indications for all traffic movements on approach W (s)

g_W^2 = Duration of stage allowing for protected left turn movements for left turning vehicles on approach E (s)

δ_W = Sequence of green stages for traffic movements on approach W

g_E^1 = Duration of stage with green indications for all traffic movements on approach E (s)

g_E^2 = Duration of stage allowing for protected left turn movements for left turning vehicles on approach W (s)

δ_E = Sequence of green stages for traffic movements on approach E

N_s = Number of traffic signals in the network under study

i = Index of traffic signal

j = Index of green phase

k = Index of approach at signal

$G(k)$ = Green phase corresponding to a particular approach

A = Duration of amber interval (s)

R = Duration of all-red interval (s)

$C_{\min}$ = Minimum cycle length to search (s)

$C_{\max}$ = Maximum cycle length to search (s)

$g_{\min,stage}$ = Minimum duration of any green stage (s)

$g_{\min,phase}$ = Minimum duration of green phase including signal change period (s)

C = Cycle length (s)

ϕ_i = Offset of traffic signal i with respect to beginning of the N/S green phase relative to the start of the simulation (s)

$\rho_{i,j}$ = Duration of green phase j at traffic signal i including amber and all-red transition period (s)

$\delta_{i,k}$ = Sequence of stages for approach k at traffic signal i

$g_{i,k}^{1}$ = Duration of stage with green indications for all traffic movements on approach k at traffic signal i (s)

$g_{i,k}^{2}$ = Duration of stage allowing for protected left turn movements on approach opposing approach k at traffic signal i (s)

x = Value of decision variable in the problem space

l = Number of bits in the binary encoding for representing x and

y_k = Value of k'th bit in binary string representing x

y = Binary value mapping to x

$D(x)$ = Normalized value of decision variable x

$y_{i,k,1}, y_{i,k,2}$ = Bits encoding $\delta_{i,k}$

p_m = Probability of mutation

N_{Pop} = Population size

p_c = Probability of crossover

β = Bias factor for linear rank selection

r = CHC divergence rate

F = Number of objective function evaluations

X, Y = Particular genetic algorithms

$\mu_{X,F}$ = Population mean of extended network delay of best individual produced by genetic algorithm X on average after F function evaluations

N = Number of independent runs of genetic algorithm

n = Number of independent replications of the best individual

$X_{i,F,j}$ = Extended network delay on the *j*'th independent replication of the best individual after F function evaluations on the *i*'th independent run of genetic algorithm *X*

$X_{i,F}$ = Sample mean of extended network of the best individual after *F* function evaluations on the *i*'th independent run of genetic algorithm *X* over the *n* independent replications of the best individual.

X_F = Mean of the sample means of extended network of the best individual after *F* function evaluations over the *N* independent runs of genetic algorithm *X*

Chapter Eight – Results

F = Number of objective function evaluations

N_{Pop} = Population size

α = Blend crossover parameter

Chapter Nine – Conclusions and Recommendations

α = Blend crossover parameter

N_{Pop} = Population size

r = CHC divergence rate

Appendix One – Microscopic Stochastic Traffic Network Simulator

A = Acceleration (ft/s^2)

V = Speed (ft/s)

t = Time (s)

α , β = Parameters of the linear acceleration model

X = Distance (ft)

f = Desired free flow speed (ft/s).

d = Constant deceleration rate

$X_L(t)$ = Position of lead vehicle at time t in terms of distance from the upstream node (ft)

$X_F(t)$ = Position of following vehicle at time t in terms of distance from the upstream node (ft)

$\Delta X(t)$ = Distance between leader and follower at time t (ft)

$V_L(t)$ = Speed of leader at time t (ft/s)

$V_F(t)$ = Speed of follower at time t (ft/s)

$\Delta V(t)$ = Relative speed difference between leader and follower at time t (ft/s)

$a_F(t)$ = Acceleration of follower at time t (ft/s^2)

τ = Driver reaction time

$\alpha^+, l^+, m^+, \alpha^-, l^-$ and m^- = Parameters of the stimulus response car-following model

$X_L^S(t)$ = Stopped position of the leader should the leader undergo emergency braking at time *t* (ft)

$X_F^S(t)$ = Stopped position of the follower allowing for a reaction time before braking in response to the leader's deceleration (ft)

dl = Emergency deceleration rate of the leader (ft/s^2)

df = Emergency deceleration rate of the follower (ft/s^2)

L = Effective length of leading vehicle (ft)

b, *c* = Computed quantities for the Gipps car-following model

l = Lane width

m = Number of lanes.

V_T = Maximum turning speed (ft/s)

F = Coefficient of friction

r = Turning radius (ft)

g = Acceleration of gravity (ft/s^2)

π = Probability of accepting a gap

x = Size of the gap (ft)

β_0, β_1 = Parameters of gap acceptance model

T_U = Uninterrupted travel time (s)

D = Distance traversed on path through the network (ft)

Appendix Two – Common Random Numbers [136]

n = Number of replications

i = Index of replication

X_i = Performance measure for the first signal timing policy on the *i*'th replication

Y_i = Performance measure for the second signal timing policy on the *i*'th replication

μ_X = Population mean of X_i

μ_Y = Population mean of Y_i

$\overline{X}$ = Sample mean of X_i

$\overline{Y}$ = Sample mean of Y_i

μ_Z = Mean difference between the timing plans

$Z_i, \overline{Z}$ = Quantities for measuring mean difference

Appendix Four – Formulae for evaluating magnitude of optimization problem

C = Cycle length (s)

N_s = Number of traffic signals in the network under study

i = Index of traffic signal

ϕ_i = Offset of traffic signal *i* with respect to beginning of the N/S green phase relative to the start of the simulation (s)

N/S and E/W = Alternating green phases at an intersection

$\rho_{i,N/S}$ = Duration of N/S green phase at traffic signal *i* including amber and all-red transition period (s)

k = Index of approach at signal

$\delta_{i,k}$ = Sequence of stages for approach *k* at traffic signal *i*

$g^1_{i,k}$ = Duration of stage with green indications for all traffic movements on approach *k* at traffic signal *i* (s)

N_d = Number of decision variables

l_C = Number of bits for encoding cycle time

l_ϕ = Number of bits for encoding an offset

l_ρ = Number of bits for encoding length of N/S green phase

l_δ = Number of bits for encoding a stage sequence

l_g = Number of bits for encoding duration of stage with green indications for all traffic movements

N_b = Total length of binary string in the binary encoding

Appendix Five – Real CHC

x = Value of particular decision variable in the problem space in first individual

y = Value of particular decision variable in the problem space in second individual

$D(x)$ = Normalized value of decision variable x

l = Number of bits in the binary encoding of decision variable

x_i = Value of i'th bit encoding binary decision variable in first individual

y_i = Value of i'th bit encoding binary decision variable in second individual

N_b = Total length of binary string in the binary encoding

N_{Real} = Number of real decision variables

N_{Binary} = Number of decision variables that utilize a binary encoding.

r = CHC divergence rate

κ = Constant in the range $[0,1]$

κ_i = Normalized value of the *i*'th real decision variable in the best individual

Appendix Six – Statistical model for comparing genetic algorithm performance

$X_{i,j}$ = Extended network delay on *j*'th independent replication of the best individual after 2000 function evaluations on the *i*'th replication of the genetic algorithm in Table 22 (seconds/vehicle)

θ_i = Population mean of extended network delay for the best individual after 2000 function evaluations on the *i*'th replication of the genetic algorithm in Table 22 (seconds/vehicle)

n_i^* = Estimate of the number of replications to perform to ensure that the sample mean of extended network delay is within 2 seconds of θ_i with 95% confidence

$t_{n-1,1-\alpha} = 100(1-\alpha)$ Percentile of the t-distribution with $n-1$ degrees of freedom

$\overline{X_i}$ = Sample mean of $\{X_{i,j}\}_{j=1}^{20}$

S_i^2 = Sample variance of $\{X_{i,j}\}_{j=1}^{20}$

$\pi(\theta)$ = Probability density function of Translated Gamma distribution

α, μ, τ = Parameters of the Translated Gamma distribution

a, b = Parameters of the linear variance model

X, Y = Particular genetic algorithm algorithms

N = Number of independent runs of genetic algorithm

n = Number of independent replications of the best individual

i = Index of independent run of genetic algorithm

j = Index of independent replication of best individual

F = Number of objective function evaluations

$X_{i,F,j}$ = Extended network delay on the j'th independent replication of the best individual after F function evaluations on the i'th independent run of genetic algorithm X

θ_i = Population mean of extended network delay of the best individual after F function evaluations on the i'th independent run of genetic algorithm X

$\alpha_{X,F}, \mu_{X,F}, \tau_{X,F}$ = Parameters of the Translated Gamma distribution governing θ_i

$\Gamma(\alpha, \mu, \tau)$ = Translated Gamma distribution

$x_{i,j}$ = Realized or observed values of the $X_{i,j}$

$\underline{x_i}$ = Vector formed by $\{x_{i,j}\}_{j=1}^{n}$

X = Matrix formed by the vectors $\underline{x_i}$

$L_i(\underline{x_i}, \alpha, \mu, \tau)$ = Likelihood of observing the sample $\underline{x_i}$

$l_i(\underline{x_i}, \alpha, \mu, \tau)$ = Log-likelihood of sample $\underline{x_i}$

$l(X, \alpha, \mu, \tau)$ = Total log-likelihood of the sample X

$f(x_{i,j} \mid \theta_i)$ = Probabilitity density function of $x_{i,j}$ given θ_i

z = Dummy variable in integration

$I(\alpha, \mu, \tau, c, d)$ = Integral requiring numerical evaluation

χ^2 = Test statistic

χ_1^2 = Chi-squared distribution with one degree of freedom

m = Number of paired values of sample mean and sample variance of extended network delay, from all genetic algorithms and all output intervals

x_i = Sample mean of extended network delay of the *i*'th sample based on the *n* independent replications

y_i = Sample variance of extended network delay of the *i*'th sample based on the *n* independent replications

θ_i = Population mean of extended network delay of the *i*'th sample

$\underline{y}$ = Vector formed by $\{y_i\}$

$L(\underline{y}, a, b)$ = Likelihood of $\underline{y}$

$l(\underline{y}, a, b)$ = Log-likelihood of $\underline{y}$

K = Constant independent of a and b

Appendix Seven – Experimental Output

X = Particular genetic algorithm

F = Number of objective function evaluations

X_F = Mean of the sample means of extended network of the best individual after F function evaluations over the independent runs of genetic algorithm X

α = Blend crossover parameter

N_{Pop} = Population size

1 Chapter One – Introduction

1.1 General Introduction

Vehicular traffic passing through an intersection must to be controlled in order to overcome the conflicts arising between different directional flows of traffic. The installation and operation of traffic lights provide an effective method of controlling traffic when vehicle flows are relatively high (1). A problem the traffic engineer then faces is to determine the timing of these traffic lights so as to optimize the flow of traffic. The latest research into traffic signal optimization demonstrates a move towards the use of stochastic simulation models for the evaluation of alternative signal timing policies and genetic algorithm optimization procedures[1].

Stochastic traffic simulation models require a high level of detail to realistically predict the impact of a particular signal timing plan. Vehicles are modelled individually and their status and behaviour are updated periodically. Consequentially, these models are computationally expensive, placing a limit on the number of signal timing policies that can be examined in the genetic algorithm search. In this dissertation we examine several untested modifications to the genetic algorithm optimization procedure in the search for more effective optimization strategies. The amendments tested include alternative search algorithms, genetic operators, problem encodings and tunings of the parameters governing the search process.

In this chapter, we first give some background on the different methods of signal operation and the commonly used optimization criteria. The limitations to the scope of the study are discussed next. We then outline the objectives of the study and finally, we discuss the structure of this dissertation.

[1] The literature will be discussed in detail in later chapters.

1.2 Background

1.2.1 Traffic signal operation [2, 3]

A breakdown of the different methods of operating traffic lights is given below:

Off-line	On-Line
Fixed Time	Traffic Actuated
Multi-Dial	Centralized Computer Controlled

Off-line operation

In off-line operation, traffic signals utilize pre-computed timing plans. Fixed time systems run continuously with a singe pre-specified timing plan. An improvement over this approach can be obtained by using multi-dial systems which allow for separate timing schemes to be employed at different times of the day. A common approach with multi-dial systems is to have three separate timing plans: one for the morning rush period, one for afternoon rush period and one for average conditions. The transition from one dial to another is carried out at fixed times during the day.

On-line operation

On-line systems can adjust timing plans based on actual traffic flows. Real-time vehicle flow measurements are obtained from vehicle detectors. With traffic actuated systems, each intersection[2] is controlled in isolation based on the detector information. Better coordination can be obtained by controlling the entire network simultaneously and this is accomplished with the centralized computer controlled approach.

1.2.2 Optimization criteria

The following measures are typically used as indicators of performance when evaluating signal timing schemes [4]:

- Vehicular delay
- Number of vehicle stops

[2] In this dissertation, it is assumed that all intersections are controlled by traffic signals.

- System throughput
- Capacity
- Queue lengths
- Fuel consumption
- Vehicle Emissions

Optimal signal settings are obtained by minimizing or maximizing one, or a combination of the above measures. The most commonly used criterion is delay. Typically, two different types of delay are considered [5]:

- Stopped delay: This is the total time spent by a vehicle in a stationary queue.
- Control delay: This is the difference between the actual travel time and the uninterrupted travel time. It includes the delay from deceleration, stopped delay, queue move-up time and acceleration delay.

1.3 Study limitations

This dissertation will be limited to the consideration of off-line traffic signal optimization, in terms of both literature review and empirical work. The findings may still be applicable to on-line optimization models which utilize genetic algorithms. We will not consider models that cater for alterations in demand or changes to routing patterns induced by different signal timing policies.

1.4 Objectives

Our goals in this study are:

- Provide an up-to-date literature review on optimization methodologies for fixed time traffic signals.
- Review the literature on genetic algorithms and identify modifications that offer potential for improving optimization of traffic signals.
- Develop a microscopic traffic simulation model for evaluating the quality of individual timing plans.
- Develop a statistical model for the comparing the mean search efficiency of alternative optimization policies.

- Test the modifications empirically on a test-bed of signal networks, identifying the amendments that contribute to improved search efficiency.

1.5 Organization of material

Chapters 2, 3 and 4 provide a summary of the analytical and computerized models that have been developed to solve the traffic signal timing problem. While not claiming to exhaustively cover all the literature, this summary aims at covering the major contributions with an emphasis on methods that have been used in practice to obtain signal settings. Chapter 2 details the contributions for a single intersection. The more general problem for a network of intersections is then discussed in chapter 3. The methods discussed in these two chapters apply to undersaturated conditions where[3]:

- Traffic flow levels are "low"
- Mean traffic flow levels are constant over time

The case of heavy traffic flows or oversaturated conditions is discussed in chapter 4.

In chapter 5 we give an overview of the genetic algorithm search mechanism. We introduce the alternative algorithms, search operators and problem encodings that we will be testing together with the unanswered questions surrounding the setting of certain search parameters. We also review the ongoing applications of genetic algorithms in traffic signal optimization. In chapter 6 we discuss the research questions. In chapter 7, we elaborate on the research methodology. The results of the empirical work are given in chapter 8. A summary of the findings and ideas for further research are given in chapter 9.

This dissertation also contains several appendices. We make reference to these appendices in the main chapters. The dissertation also includes a compact disc with the code listings for the computer programs that were written for performing the experiments.

[3] A formal definition is given in Chapter 2.

2 Chapter Two – Undersaturated Conditions, Single Isolated Intersection

2.1 Introduction

This chapter covers research on methods that produce optimal signal timings for an isolated intersection in undersaturated conditions. The intersection is "isolated" in the sense that there are no other junctions in close vicinity. We first introduce some terminology and then discuss the basic traffic model used by the majority of researchers. Contributions to the derivation of an optimal signal plan for a single intersection problem will then be discussed. Additional terminology will be introduced as required.

2.2 Terminology

2.2.1 Vehicular terminology [2, 6]

Headway: Time interval between the fronts (or rears) of two successive vehicles measured at a fixed point on a road.

Traffic Movement: As a vehicle approaches a signal controlled junction, it selects a lane according to the manoeuvre to be performed. If the same manoeuvre can be performed from more than one lane, those vehicles will in effect form a single queue. The segment of traffic using a set of lanes shared in this way is called a stream or movement.

Arrival Rate or Flow (q): This is the average rate at which vehicles arrive on a particular stream.

2.2.2 Traffic signal terminology [5, 6]

Signal Group: Set of traffic streams controlled as a single unit by the controller.

Stage or Phase: Period during which signal indications remain constant for all groups[4].

[4] Some authors differentiate between stage and phase. For all sections of this dissertation excluding 7.6, we use the terms interchangeably.

Each stage is characterized by the following two parameters:

Green Time: The length of time the movements are given right of way is called the green time or the display green time

Inter-Green Time: Time from the end of green of a particular phase to the beginning of green on the next phase. The inter-green time is made up of the amber and all-red periods.

Cycle Time (C): Sum of all phase green and inter-green times.

Green Split: The set of green time allocations for all signal groups.

Phase Sequence: Ordering of stages in a cycle, starting from some arbitrary stage.

2.3 The Traffic Model

The model describing vehicle behaviour at a signal is captured by the assumptions of the arrival and departure process. The assumptions underlying these processes determine queue evolution for each traffic movement.

2.3.1 Arrival process

Adams [7] was the first to demonstrate that the number of vehicles in light traffic passing a point in equal intervals of time follows a Poisson distribution. This corresponds to vehicle headways that are exponentially distributed. This result has been confirmed in various other studies such as (8). The analytical tractability of this model has led to wide adoption of the hypothesis that vehicle arrivals on a particular approach at an isolated intersection occur according to a Poisson Process with intensity equal to the arrival rate q.

2.3.2 Departure process [5, 9]

Studies have demonstrated that after a signal has turned green, allowing a queue of vehicles to start from rest, the flow across the stop-line increases rapidly to a maximum rate. The departure

rate remains fairly constant at this maximum rate until either the queue has been entirely discharged or the green period ends. This maximum departure rate is called the saturation flow rate (*s*). Vehicles arriving at a green signal when no queue is present are not delayed. Figure 1 demonstrates this process graphically for a queue that cannot be fully serviced in a single cycle. The departure rate is lower during the first few seconds as a result of the reaction time of drivers and the time needed for vehicles to accelerate to normal running speeds. When a queue is still present at the end of the green period, the departure rate does not immediately drop to zero as vehicles may still cross the intersection during the amber and all-red periods.

A simplified approximation can be obtained by replacing the departure flow curve in Figure 1 by a rectangle with height *s* and the same area[5] as illustrated in Figure 2. The width of the rectangle of the rectangle is called the effective green time (*g*). With the simplified departure process, it is assumed that vehicles may only depart during effective green at the saturation flow rate. The time difference between the start of display green and effective green is called the starting loss and allows for the lower initial departure rate at the onset of display green. The time interval between the end of display green and effective green is called the end gain and allows for the additional departures during the amber and all red periods. Figure 1 and Figure 2 illustrate the relationship between the various quantities graphically.

[5] The area under each curve represents the number of vehicle departures.

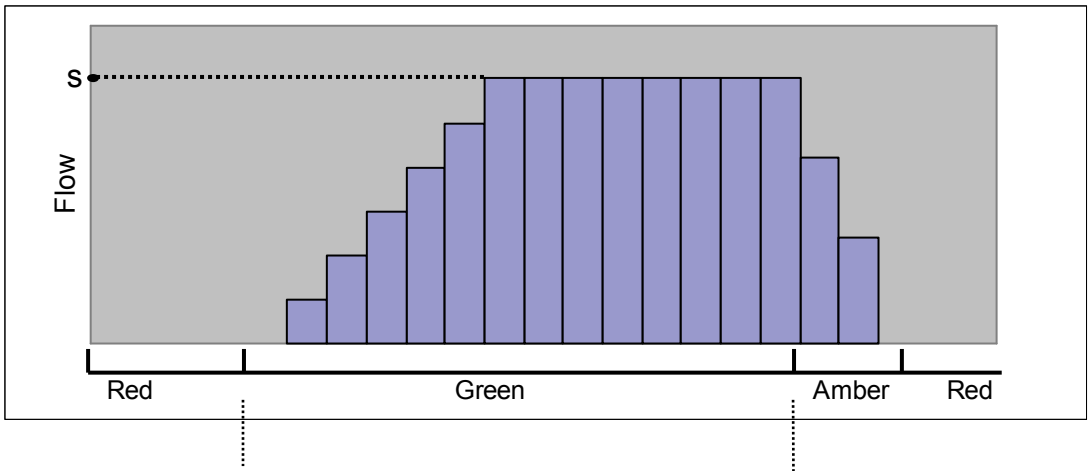


Figure 1: Departure Process

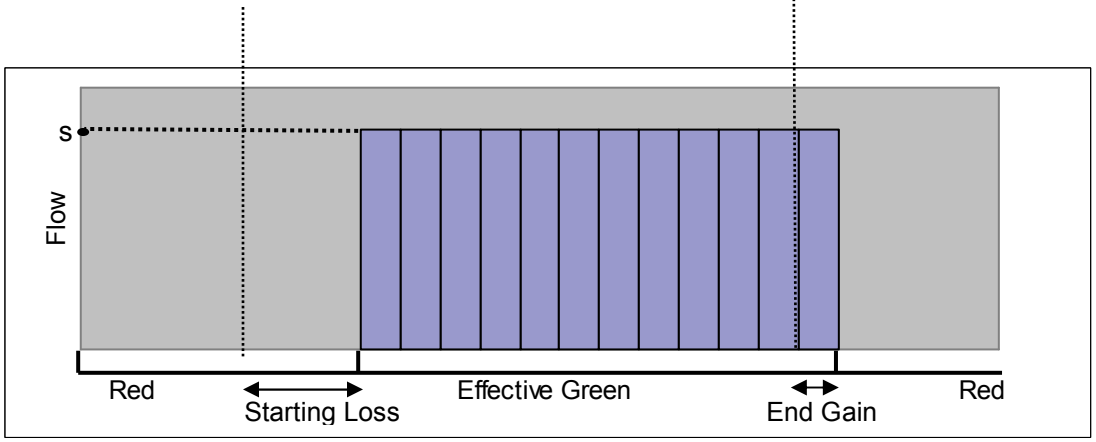


Figure 2: Simplified Departure Process

2.3.3 Opposed-turning movements [2, 10, 11]

Suppose the traffic streams illustrated in Figure 3 receive green simultaneously.

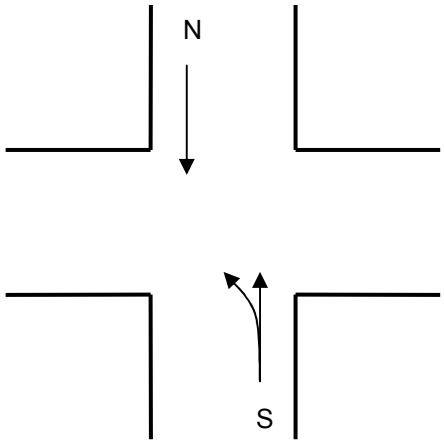


Figure 3: Opposed-turning movement[6]

The vehicles arriving on approach *S* and turning left must yield to through traffic arriving from approach *N*. The turning vehicles may only proceed when there is a suitable "gap" in the oncoming traffic stream. Vehicles that have crossed the stop line may turn during the amber and all-red periods. These traffic streams are known as opposed turning movements. The behaviour of vehicles in these streams can be described using gap-acceptance logic (see section A1.3.8). As an approximation, these movements can be modelled by applying the same assumptions for the departure process discussed in 2.3.2 with a reduced value for the saturation flow and an extension of the effective green time[7].

2.3.4 Queue evolution

The process of queue formation and dissipation for each stream can be analyzed using queuing theory [(2)]. Queuing theory describes the behaviour of queues of customers as they arrive according to some specified statistical arrival distribution at a server (or group of servers) [(12, 10)].

For the signal optimization problem, the vehicles arriving at the intersection are the customers and the traffic signal is the server. The inter-arrival times of the customers (vehicles) correspond to the vehicle headways which are assumed to have an exponential distribution. Vehicles are

[6] The figure is drawn for the right-hand rule of the road. All illustrations and terminology introduced throughout this dissertation apply to traffic operating under the right-hand rule of the road.
[7] Although such an approximation may not be sufficiently realistic, a discussion of the shortcomings in the methodology of analytical models for undersaturated isolated intersections is not the intended purpose of this study.

serviced during the effective green period. The service times for queued vehicles are assumed to be constant, equal to s^{-1}, the inverse of the saturation flow rate. Vehicles arriving when no queue is present are served instantaneously. There is no service during the red period.

2.4 Webster's formula

Now that the commonly used traffic model has been introduced, we can discuss Webster's average delay formula [13]. Some additional terminology is first required.

2.4.1 Additional terminology [2]

Degree of saturation (x): The degree of saturation for a particular movement is defined as:

$$x = \frac{qC}{gs} \tag{2-1}$$

The numerator is the mean number of vehicles arriving per cycle. The denominator is the maximum number of vehicles that may pass through the intersection per cycle.

Capacity: The arrival rate that produces $x = 1$ i.e. $q = \frac{gs}{C}$.

Undersaturated conditions: Corresponds to traffic conditions where the arrival rate for all movements is below capacity.

Oversaturated conditions: If the arrival rate is above capacity for any of the streams, then the intersection is said to be oversaturated.

Critical movement: During a particular phase, more than one movement may receive right of way. The movement with the largest degree of saturation is called the critical movement for that phase.

Critical degree of saturation: The degree of saturation for the critical movement.

2.4.2 Computing steady-state delay

As already mentioned, queuing theory can be used to analyze the system and compute performance measures such as delay. Results are generally most easily obtained for steady-state quantities [12]. The steady-state vehicle delay (*d*) is the stopped delay a vehicle would experience on average, assuming that the system has been operating for an infinite amount of time. Steady-state delay may only be considered as a performance measure in undersaturated conditions as queues of infinite length would develop for volumes above capacity. When referring to steady-state delay in undersaturated conditions, we will use the term interchangeably with average delay.

Derivation of an exact equation for the average delay is complex. This is due to the non-homogenous nature of the departure process[8]. An exact expression can be obtained in terms of the complex roots of certain transcendental equations [2].

The difficulty in obtaining a simple, easily computable expression for the average delay prompted researchers to look for approximations and bounds [2]. Webster [13] used simulation to calibrate an approximate formula for computing average delay at a movement level which is given below[9]:

$$d = \frac{C\left(1-\frac{g}{C}\right)^2}{2\left(1-\frac{gx}{C}\right)} + \frac{x^2}{2q(1-x)} - 0.65\left(\frac{C}{q^2}\right)^{\frac{1}{3}} x^{2+5\frac{g}{c}}, \quad x<1 \qquad (2\text{-}2)$$

The expression was not derived entirely empirically. The first term of equation (2-2) is the expression for average delay assuming deterministic arrivals and a continuous approximation to the arrival and departure process [15]. The second term is the steady state delay for a queuing system with random arrivals and departures at constant intervals throughout the cycle [12]. The third term is an empirical correction term. A simplified version of the formula can be obtained by

[8] The departure rate is a function of the traffic signal indication which changes over time.
[9] Other approximate expressions have been obtained. Numerical differences between the various formulae have been found to be small [14].

ignoring the third term and multiplying the result by 0.9 [13]. This simplified form is often referred to as Webster's two-term delay formula.

The total intersection delay is obtained by multiplying the average delay for each movement by the corresponding arrival rate and summing over all movements.

2.5 Deriving the Optimal Timing Plan

2.5.1 Two-phase intersection

Webster [13] considered a single isolated intersection with no turning movements as illustrated in Figure 4.

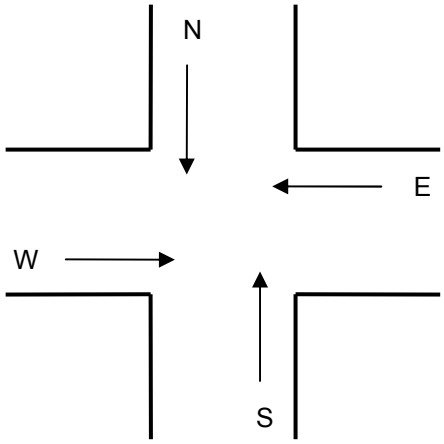


Figure 4: Isolated intersection with no turning movements

Assuming two phase control with two groups[10], he applied his two-term delay formula and demonstrated that the optimal green split (in terms of minimizing intersection delay) for any given cycle length can be approximated by allocating green time so as to equalize critical degrees of saturation. We demonstrate how this result can be used to compute the optimal green split via an example:

[10] *N* and *S* movements in one group, *E* and *W* movements in the other.

Let q_i = arrival rate (vehicles/hour) for movement *i*,

$g_{N/S}$ = effective green time for *N* and *S* movements and

$g_{E/W}$ = effective green time for *E* and *W* movements.

Suppose that $q_N = 600, q_S = 400, q_E = 150, q_W = 300$ and that the saturation flow rate for all movements is identical and equal to *s*. For these inputs, the *N* movement is the critical movement during the N/S green phase and the *W* movement is the critical movement during the E/W green phase. Equate critical degrees of saturation we get:

$$\frac{q_N C}{g_{N/S} s} = \frac{q_W C}{g_{E/W} s}. \qquad (2\text{-}3)$$

Manipulating this equation we obtain $\frac{g_{N/S}}{g_{E/W}} = \frac{q_N}{q_W} = 2$. Thus the optimal green split is obtained by allocating exactly twice as much effective green time to the N/S green phase than the E/W green phase.

We see that Webster's method [13] provides a simple procedure for computing the optimal green split. He also derived an approximate formula for the optimal cycle time.

2.5.2 More complicated junction layouts, multiphase sequencing and vehicle mix

Allsop [10] extended Webster's work to provide optimal signal settings for any number of approaches and stages. Webster's two-term delay formula is used to compute the average delay for each movement in a mathematical programming formulation of the problem. The mathematical programming approach accommodates constraints on the cycle length and effective green times. The minimum delay cycle length and green splits are obtained using an iterative method similar to the method of feasible directions.

For movements that receive green in more than one stage, Yagar [11] extended Allsop's formulation to allow for differing saturation flows during each stage[11]. He suggested an alternative solution method via convex programming.

Traffic composition is a mix of passenger cars and heavier vehicles such as buses and trucks. The saturation flow for heavy vehicles is usually lower than that of passenger cars. The different types of vehicles can be accounted for by using a weighted average of the corresponding saturation flows, where the weightings applied are the observed proportions of each vehicle type [5, 9].

2.5.3 Optimization of stage composition and sequence

The methods of Webster [13], Allsop [10] and Yagar [11] are called stage based approaches in the sense that they require the stage structure and sequence of the controller to be pre-specified [6]. For junctions with many approaches and traffic streams, the composition and order of the signal stages can significantly impact the best achievable performance. In such cases, optimization of these control parameters together with the cycle length and green splits can be beneficial. This is called the group-based approach to signal optimization. The optimization problem can be treated in two steps [16]:

- First, an enumeration of all possible "valid" control schemes is obtained[12].
- Optimal cycle length and green splits are computed for each of the possible control sequences using one of the stage based methods. The optimal sequence is identified from the enumeration.

Gallivan and Heydecker [17] noted several problems with this sequential approach and devised a modified optimization method which could be used to evaluate all possible control sequences as well as reduce the total number of control sequences to be considered [6, 17, 18]. Improta and

[11] This is needed for opposed-turning movements which receive both a protected phase as well as a phase where they have to yield to opposing traffic. The saturation flow rate will be higher in the protected phase.

[12] A control sequence is "valid" if [16]:

- Each signal group receives a green signal indication at least once during a cycle
- As many groups as possible should receive green simultaneous, provided there is no conflict
- Each group should have a single continuous period of green within a single cycle (i.e. stages which contain a common signal group should appear consecutively in the cycle)

Reference (16) provides an overview of literature on methods that generate such sequences.

Cantarella [16] devised an alternative approach which simultaneously optimizes the stage composition, stage sequence and green timings. Binary integer variables are used to represent the order in which groups have right of way. Branch and bound techniques are applied to generate the optimal solution.

2.5.4 Optimization criteria besides delay

While delay has generally been the most commonly used optimization objective, other measures have also been considered, either directly or via constraints. Allsop [19] and Yagar [20] have devised linear programming formulations to maximize reserve capacity. The reserve capacity is the largest factor by which arrival rate for all movements can be increased while still maintaining operation below capacity. The methods discussed in 2.5.3 can be applied to either delay minimization or reserve capacity maximization. For the case of delay minimization, these methods allow for constraints on the maximum allowable degree of saturation for each stream. Ohno and Mine [21, 22] considered minimization of the intersection degree of saturation which is the maximum degree of saturation over all approaches.

In addition, there are several computer packages available that consider both average delay and average number vehicle stops[13]. Signal settings that minimize delay and stops in combination with other criteria listed in section 1.2.2 are obtained using numerical methods [23, 24, 25].

[13] The steady-state number of vehicle stops are computed using approximate formulae such as those proposed by Webster [13] or Akcelik [5]

3 Chapter Three – Undersaturated Conditions, Signalized Networks

3.1 Introduction

Considering the difficulty in obtaining an exact expression for the steady-state delay for a single intersection, it is not surprising that a tractable analytical model for a network of intersections with stochastic inputs has not yet been developed. In this chapter, we discuss the methods that have been used in practice to derive timings schemes for networks of traffic signals. The approaches can be broken down into three different classes:

- Bandwidth methods
- Delay-based methods
- Hybrid methods

Before we discuss the various approaches, some additional terminology is required to supplement that introduced in chapter 2.

3.2 Network Terminology [9]

Relative Offset: Together with the decision variables, cycle time, green split and stage structure and sequence, a relative offset must be specified for each traffic signal in the network. The relative offset determines the starting time for a particular stage of operation for a particular traffic signal with reference to some specified point in time in the cycle. Alternatively, the reference time point for a particular signal can be the starting time of a particular stage at an adjacent signal. The relative offsets are sometimes defined with respect to the mid-point of stages rather than the beginning.

It is important that the relative offsets for all signals in the network remain fixed from cycle to cycle in order to maintain a fixed synchronization. This can be achieved by assigning a common

cycle length to all signals in the network[14]. All network optimization models discussed in this dissertation make use of this assumption[15].

Links and Nodes: Traffic networks are usually represented diagrammatically using links and nodes. The links represent streets and the nodes are the intersections. Two-way streets are represented by two links, one for each direction of travel.

Upstream and Downstream: For a particular traffic movement, the direction from which the traffic originates is called the upstream direction. The direction of travel of the stream is called the downstream direction.

Arterial Network: An arterial network is a traffic network consisting of a main street that intersects several side streets. Generally, the majority of traffic flows along the main street.

Grid Network: A grid network consists of a group of parallel streets which are orthogonal to and intersect another set of parallel streets. The traffic network structure in a city centre is an example of a grid network.

3.3 Bandwidth methods

3.3.1 Introduction [1, 2, 26]

This is the oldest network synchronization scheme and is still widely used. The method is generally applied to arterial networks such as the one illustrated in Figure 5.

[14] Synchronization can also be achieved by the selection of cycle lengths which have a common multiple.
[15] This requirement is generally always applied in off-line signal control. With responsive traffic control (see section 1.2.1), synchronization need not remain fixed and a common cycle time is not always imposed.

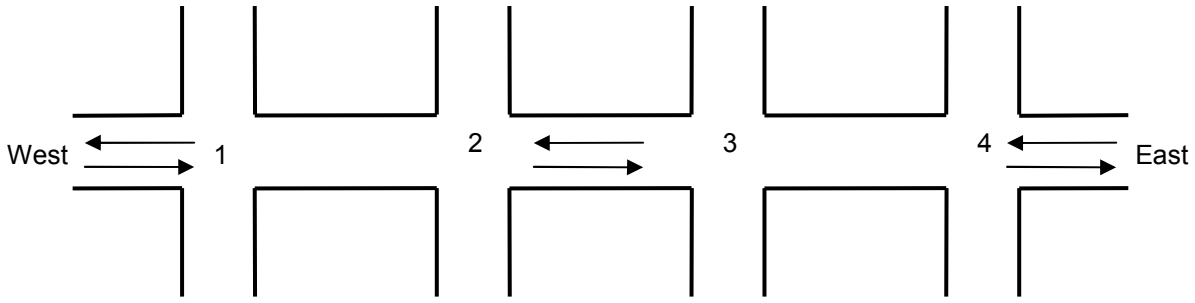


Figure 5: Arterial network

The bandwidth approach aims to take advantage of the following traffic phenomena:

- The on/off nature of traffic signals tends to create bunches or platoons of vehicles.
- The platooning effect is accentuated on arterials with predominantly through traffic and signalized intersections at frequent intervals.

In such cases, it seems reasonable to set the offsets of the signals in such a way so as to allow continuous progression of platoons through successive signals along the arterial without stopping. Figure 6 illustrates the problem for the arterial network in Figure 5 via a space-time diagram.

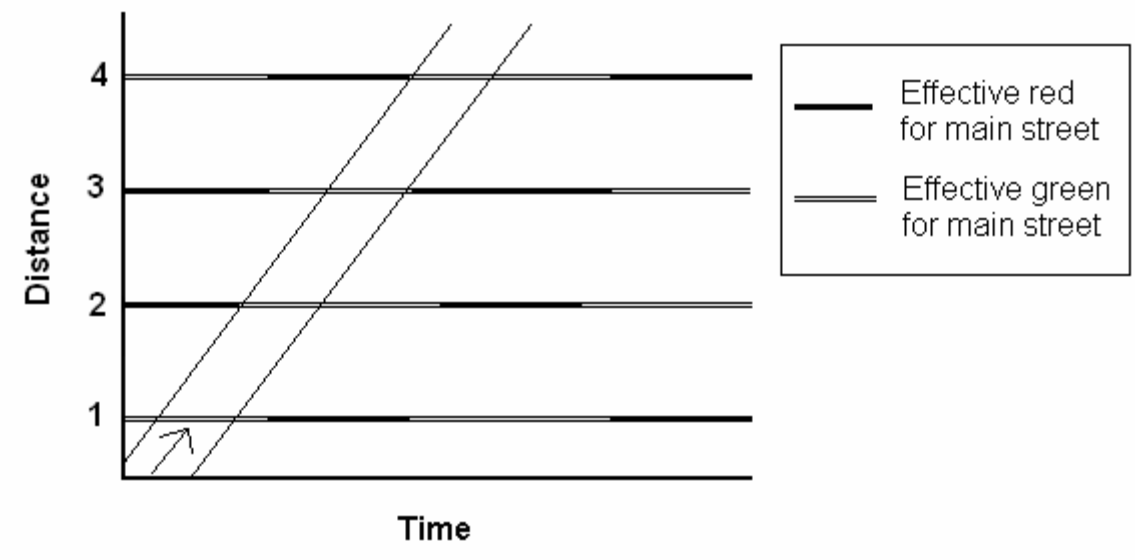


Figure 6: Space-time diagram for main arterial

The numbering on the y-axis corresponds to the numbered intersections in Figure 5. The diagonal arrow represents a vehicle heading East on the arterial, travelling at a constant design speed. This vehicle will not have to stop at any of the intersections. This is true for any vehicle passing between the two diagonal lines and travelling at the design speed. The horizontal distance between the two diagonal lines is called the bandwidth. The bandwidth or progression method seeks to maximize the opportunity for uninterrupted travel along the arterial by maximizing the bandwidth in both directions as demonstrated in Figure 7.

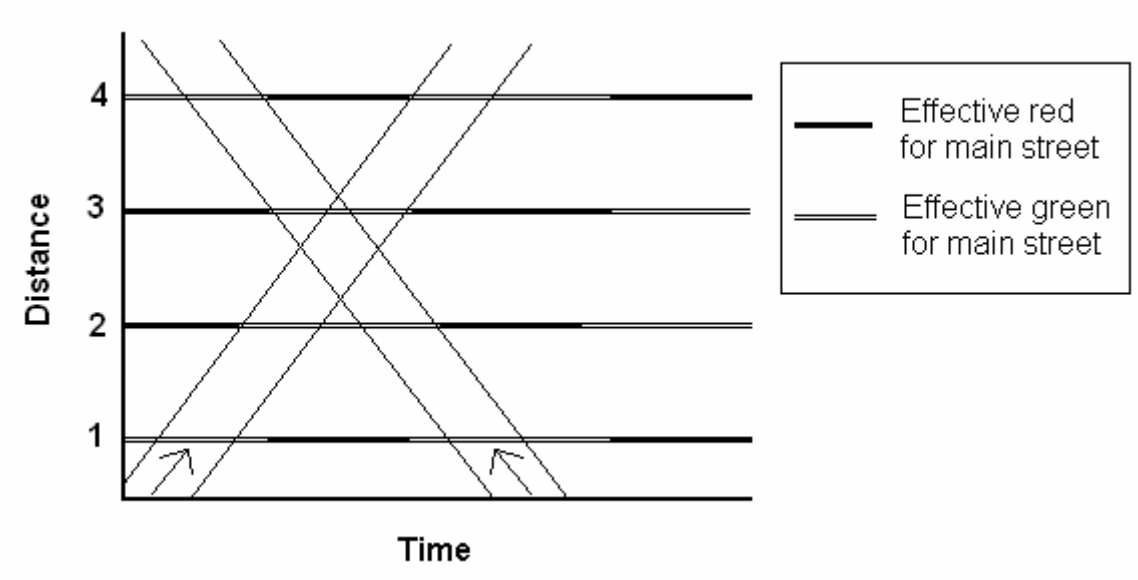


Figure 7: Bandwidth in both directions

Bandwidth based methods cannot optimize green time allocations[16]. An optimal cycle length can be obtained by running the bandwidth maximization algorithm for several cycle lengths and choosing the one which maximizes the bandwidth efficiency which is the ratio of the bandwidth to the cycle length[17].

[16] This is because the maximum possible bandwidth can be obtained by allocating the maximum allowable green time to the arterial movements, disadvantaging cross-street movements. When utilizing the bandwidth method, green splits are obtained by applying Webster's method (13) independently at each signal or some other heuristic method.

[17] Alternatively Webster and Cobbe (27) suggest considering the most heavily loaded intersection as the basis for computing the common optimal cycle length for the entire network.

3.3.2 Bandwidth maximization algorithms for arterial networks

Determining the relative offsets that maximize bandwidth is a trivial problem for one-way streets[18]. For two-way streets, traffic engineer's initially used graphical trial-and-error procedures [9].

Morgan and Little [28] were the first to develop a computational procedure for generating maximal equal bandwidths. Their algorithm allows for different design speeds along each link of the arterial. For the case of equal design speeds in both directions, they demonstrated that the maximal bandwidth solution is achieved in a solution where the relative offsets of adjacent signals is either zero or half the cycle time (where the relative offset is measured from the centre of red for a particular signal to the closest centre of red of an adjacent signal). This situation is illustrated in Figure 7. For the case of arterials with unbalanced directional flows, they provide an algorithm to adjust the solution to provide wider bandwidth for the direction with larger flow.

Little [29] cast the progression design problem in the framework of mixed integer linear programming. His formulation was used as the basis for the MAXBAND [30] computer model which provides the following extensions:

- Design speed on each link can be included as decision variables in the optimization process.
- For multiphase signals, the optimal ordering of stages for through arterial movements can be determined[19].
- A time advance in the through band can be introduced to permit queues of stationary vehicles to clear the intersection before the arrival of a platoon from upstream.

Relative offsets that maximize bandwidth are obtained using a branch and bound solution method.

[18] This can be demonstrated by considering Figure 6. The largest that bandwidth can be made is the shortest main street green which occurs at intersection 3. We can draw the diagonal lines representing this maximal bandwidth through the green interval for intersection 3 (These lines have slope equal to the deisgn speed). We then set the offsets of the other signals in such a way that the main street green interval for each signal begins where the left diagonal of the maximal through band starts [28].

[19] This is incorporated into the formulation via the use of binary integer variables to represent alternate phase sequencing.

Another widely used bandwidth model is the PASSER-II model developed by Messer et al [31]. PASSER-II uses an efficient heuristic optimization technique based on the concept of minimizing interference to the progression bands and has capabilities similar to MAXBAND.

As already mentioned, directional bandwidths can be adjusted to account for differences in directional volumes. A common practice is to allocate the total bandwidth in proportion to the directional volumes, although more detailed weighting schemes do exist [32]. However, with turn-in and turn-out traffic at each intersection, the volumes along each direction of an arterial are not constant. Thus a single directional volume figure may not be appropriate. This problem was noted by Gartner et al [33] who developed the MULTIBAND model, an extension to MAXBAND that computes bandwidth separately for each directional link. The total bandwidth can be weighted by the flow on each link and maximized.

3.3.3 Extensions to grid networks

Chang et al [34] extended the MAXBAND model for application to a network of arterials[20]. The objective function is a weighted average of all the one-way bandwidths. A similar extension to the MULTIBAND model has also been developed [35].

Bandwidth models define relative offsets of signals with respect to adjacent signals. In the case of grid networks, this requires the additional constraint that the sum of relative offsets for each loop in the network is an integer multiple of the cycle length [36]. This constraint is needed to ensure that the synchronization stays fixed. These are called the network closure constraints [36]. The large number of additional integer constraints in the mixed integer linear program increase the computation time of the branch and bound algorithm to the point where the models can no longer be practically applied [37]. Several heuristic algorithms have been proposed to improve computational efficiency [37, 38, 39].

[20] As in the case of single arterials, the green allocations for each signal must be specified.

3.4 Delay-based methods

Unlike bandwidth methods which rely on the assumption that an increase in bandwidth corresponds to lower delays and stops, delay-based methods aim to achieve these objectives directly[21]. These methods can be classified into one of two types [(9)]:

- Local optimization methods
- Global optimization methods

Before we proceed with a discussion of these two methods, we first discuss the general modelling methodology.

3.4.1 Modelling methodology [(40, 41)]

Macroscopic Models

The majority of delay-based models apply a macroscopic approximation to total vehicle flow. The flow at any point on a link is represented by a single continuous variable. Vehicles enter the network continuously from the boundaries. The entry flow is deterministic and occurs at a uniform rate equal to the arrival rate. Figure 8 illustrates flow at the entry point of an internal link.

[21] Delay-based methods usually incorporate delay in combination with one or more of the other measures listed in 1.2.2.

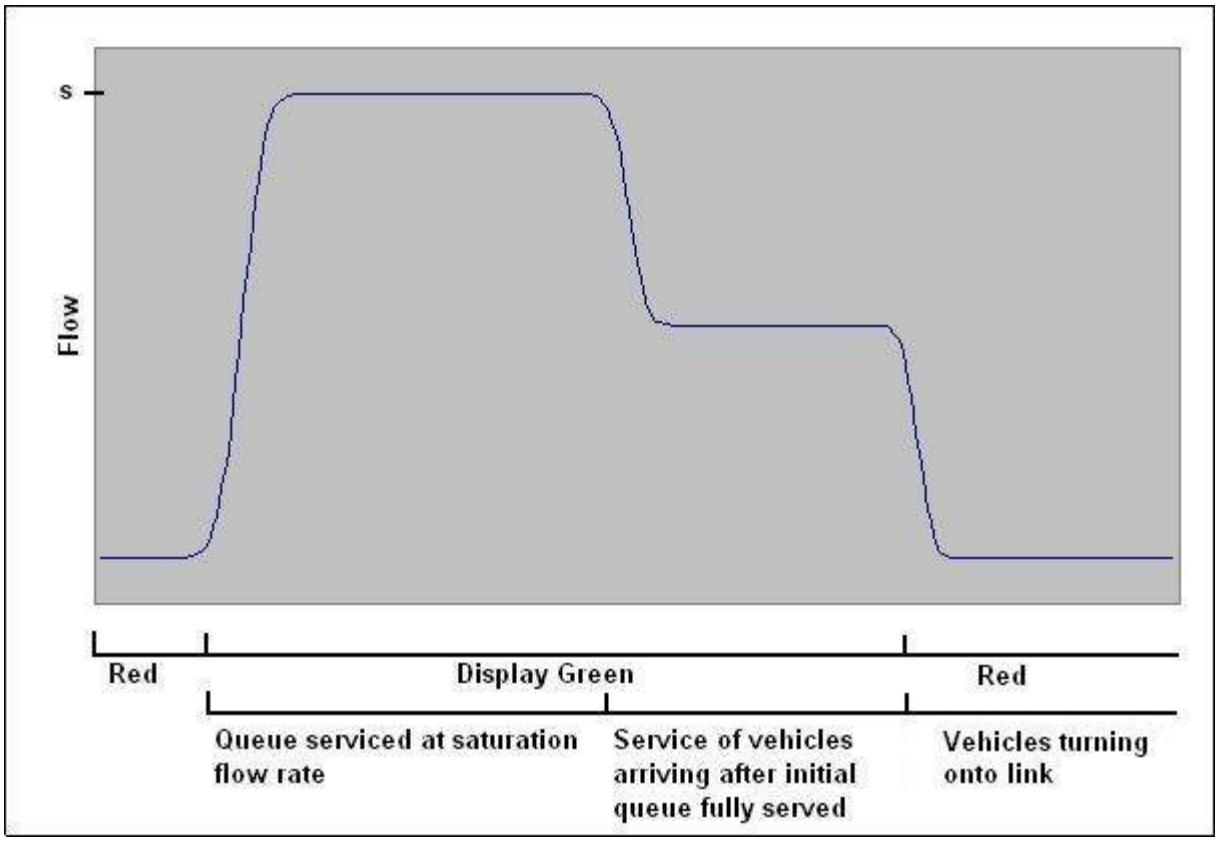


Figure 8: Flow at entry of an internal link

Traffic entering the link constitutes the vehicles that have departed from upstream links. The signal indications given in the figure are for through traffic entering the link[22]. The large concentration of queued vehicles that depart together is called a platoon. Due to the different speeds of individual vehicles, the platoon disperses as it moves further downstream. Figure 9 illustrates the effect this may have on the platoon pattern.

[22] The amber and all-red periods have not been labelled for the sake of making the illustration more legible.

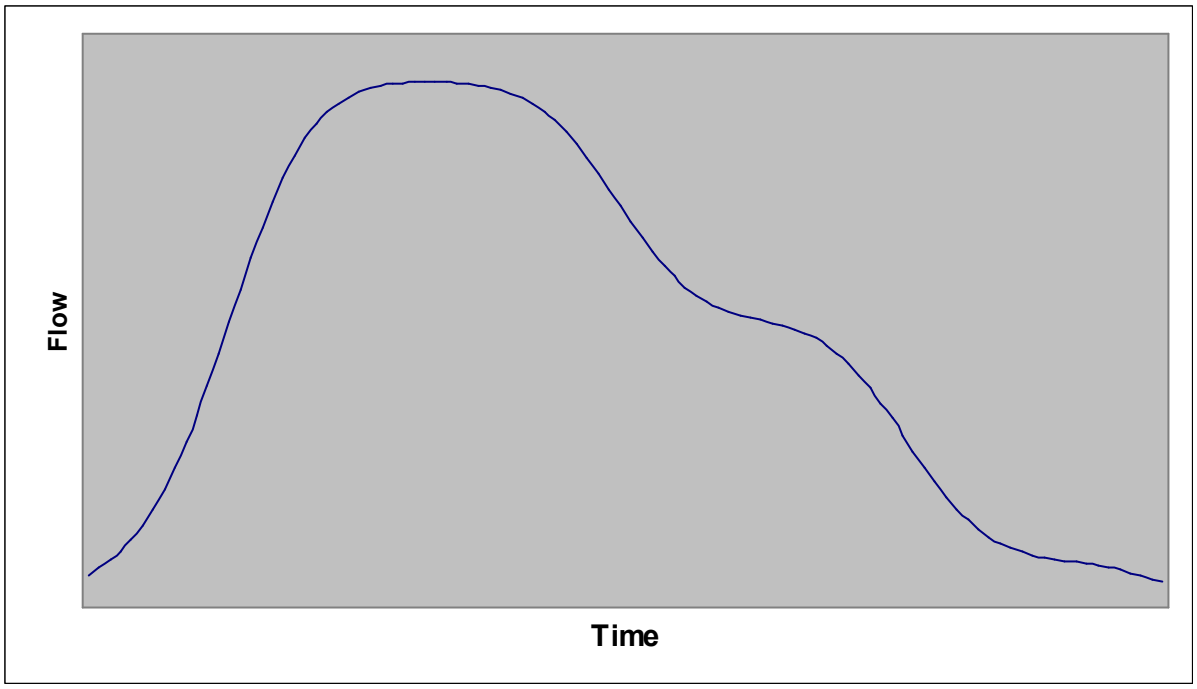


Figure 9: Flow at a point further downstream

This phenomenon is known as platoon dispersion and is essential to the realistic modelling of traffic behaviour along links in macroscopic models.

Microscopic models

Microscopic models model traffic behaviour in larger detail. Vehicles are modelled individually and their behaviour is determined by car-following and gap-acceptance logic (Appendix A1 provides a discussion of these topics). Vehicle flows into the network from the boundaries are stochastic and the behaviour of individual vehicles includes stochastic components.

Analytical formulae

Another alternative is to base delay computations on analytical formulae such as Webster's delay formula [13]. These delay formulae are usually applicable to isolated intersections. However, the assumption of random arrivals is not applicable on internal links of the network where platooning effects create a more regular arrival pattern. Adjustments to account for the quality of

progression of delay are made by revising delay estimates downward in the case of favourable progression and increasing the delay assessment when progression is poor.

3.4.2 Local optimization methods

Introduction

With local optimization, an attempt is made to simplify the network optimization problem into a series of single intersection optimization problems. In order for this reduction to occur, the following assumptions are required:

- The volume of traffic flowing along any link is constant (i.e. independent of signal settings)
- The delay on any link is a function of the settings of the signals at the two ends of the link.

Two noteworthy methods are the Combination method developed by Hillier [42] and the MITROP model developed by Gartner, Little and Gabbay [43]. Both methods are limited to the optimization of networks with two-phase signal operation.

Combination method [42, 44, 45, 46]

The combination method is limited to the optimization of relative offsets. A common cycle time and green splits at each intersection must be specified. The objective function to be minimized is a linear combination of average delay and stops. For each link, the delay and number of stops for a single representative cycle are computed either by analytical formulae or via a crude macroscopic simulation[23]. The relative offsets that minimize the objective function can be obtained by dynamic programming or the branch and bound method.

MITROP model [43]

MITROP simultaneously optimizes cycle time, green splits and relative offsets using delay minimization as the objective. The model uses a macroscopic approximation to vehicle flow with

[23] The analytical formula assumes constant travel time between nodes. The simulation uses platoon dispersion and is more realistic.

a rectangular platoon dispersion function[24]. The computed delay for the generated deterministic flow is adjusted to allow for the fact that arrivals are in fact stochastic. The adjustment made is similar to the adjustment to the deterministic delay term in Webster's formula (13) (see 2.4.2). The model is formulated as a mixed integer convex program. The optimal values of the decision variables are obtained by the method of branch and bound.

Shortcomings of the local optimization approach

The most serious limitation of local optimization models is the assumption that the performance measure on any link is a function of the settings of the signals at the two ends of the link. This assumption is known to be poor, particularly in conditions of low traffic flow[25].

3.4.3 Global optimization models

Introduction

Global techniques attempt to simultaneously optimize settings at all traffic signals. The impact of changes to signal settings are assessed by their impact on the network as a whole by considering overall network wide statistics[26]. We discuss SYNCHRO and TRANSYT-7F, two global optimization models in widespread use.

SYNCHRO (48)

Average delay for a specified network signal timing policy in SYNCHRO is obtained via an analytical formula. The delay formula is applied separately to each traffic movement on each link and assumes deterministic arrivals. An adjustment to account for the quality of progression is made by multiplying delay figures by an internally computed progression factors. Variability in vehicle arrival patterns is accounted through the use of percentile delay. The traffic volumes

[24] A rectangular platoon dispersion function assumes the platoon to be of uniform density. The length of the platoon increases as a function of the distance traveled according to a dispersion equation.

[25] By ignoring other signals, a traffic flow pattern at the link entry point must be assumed. Under conditions of high traffic, we can be fairly sure that a queue of vehicles will be discharged from the upstream link at the onset of the green interval of the upstream controller. In less saturated traffic conditions, the discharge pattern at the upstream signal is highly dependent on the relationship of that controller with signals further upstream (44). Local optimization methods have been found to produce unreasonable timing schemes in less saturated conditions (47).

[26] Unlike the network models discussed so far, global optimization models can define relative offsets of signals with reference to a specified point in time in the cycle. Thus the network closure constraints (see 3.3.3) are not required, making the optimization procedure a lot more straight forward.

entered by the user are taken as the mean of Poisson random variables of the observed traffic volumes. Approximating the Poisson distribution with the Normal distribution, 5 flow levels based on the 10th, 30th, 50th, 70th and 90th percentiles are obtained. The percentile delay value for a particular movement is taken as the average of these 5 scenarios.

SYNCHRO determines optimal signal timings by minimizing percentile delay using a quasi-exhaustive search. Various cycle lengths, offsets and phase sequences are examined in the search process. Green split allocations are determined via Webster's method.

TRANSYT / TRANSYT-7F

The TRANSYT model (Traffic Network Study Tool) was originally developed in the late 1960's by Robertson [40] at the U.K. Road Research Laboratories. The model has subsequently been subject to continuous development and improvement. We consider the U.S. version TRANSYT-7F [49], which is considerably more advanced than its U.K counterpart.

TRANSYT-7F employs a more realistic modelling of network traffic flow than other models. A simulation model is employed to track flow patterns from external links all the way through the network. Through and turning proportions are specified at the end of each link.

The model seeks to minimize an objective called the performance index. The performance index used in earlier versions was a linear combination of delay and stops. Newer versions can incorporate any of the criteria listed in section 1.2.2. The performance index for a specified network timing plan is computed by a macroscopic simulation model with an allowance for platoon dispersion via an exponential smoothing function. The simulation process begins with an empty system (i.e. no vehicles or queues present). The deterministic flows are introduced on the external links and the simulation is run for several cycles until a periodic pattern of queue growth and decay is obtained. Delay and stops are computed for this average flow representation and adjusted to account for stochastic effects.

Earlier versions required the cycle time and phase sequence to be specified. Optimal green splits and relative offsets were obtained using the following hill-climbing procedure:

- First, a set of initial timings must be given.
- The signals in the network are then considered one at a time.
- The green splits and offsets of a particular signal are adjusted while keeping the settings on all other signals fixed. The green splits and offsets are adjusted independently in a sequence of large and small steps. Adjustments that improve the performance index are retained.
- Each signal is adjusted in turn using the same procedure until all signals have been considered.
- The entire process is repeated a number of times.

The hill climbing procedure has several drawbacks:

- The final solution is sensitive to the quality of the initial solution.
- A globally optimum solution is not guaranteed.
- Phase sequencing cannot be optimized for multiphase signals.

These deficiencies were resolved by release 10 of TRANSYT-7F, introduced in 2004, which features optimization by genetic algorithms, allowing for global optimization and consideration for phase sequence optimization[27]. In addition, release 10.2, allows the performance index to be computed by CORSIM (50), a detailed stand-alone microscopic simulation model.

3.5 Hybrid methods

3.5.1 Contrast between bandwidth and delay-based approaches

Bandwidth methods rely on the assumption that control strategies with larger bandwidth produce improved performance in terms of delay and stops. However, one cannot conclude that a maximal bandwidth strategy corresponds to one with minimal journey time (9). In addition, bandwidth methods require the green splits to be specified. Delay-based methods directly optimize the pertinent performance measures. Furthermore, delay-based methods employ traffic models which explicitly model vehicle interactions, driver behaviour and queue effects. Bandwidth methods ignore these traffic phenomena by assuming constant link travel times.

[27] Genetic algorithms and their applications in traffic signal optimization are discussed in detail in Chapter 5.

Thus one would expect the delay-based approach to be the one preferred by traffic engineers. However, in practice, consideration is still given to bandwidth as a measure of system performance for the following reasons [30]:

- Bandwidth models rely on relatively little input and the progressive timings that they produce are thought to be operationally robust.
- Space-time diagrams allow the traffic engineer to visualize the quality of results.
- Drivers expect signal progression and use it as a measure of the quality of signal timing.

This has led to the development of hybrid approaches that incorporate both delay and bandwidth. The development of hybrid models has followed two distinct paths:

- Modifying bandwidth methods to consider delay.
- Modifying delay-based methods to consider bandwidth.

3.5.2 Modifying bandwidth methods to consider delay

Bandwidth maximizing solutions are not unique [51]. This can be observed in Figure 7 where it is possible to modify the relative offset of the signal at intersection 1 in a small range without reducing bandwidth in either direction. The limits of these ranges are called slack times [51]. Bandwidth programs generally produce relative offsets so that the green bands are centred, with an equal amount of slack time on both sides of the two-way bands. The existence of these slack times prompted Chang et al [51] to extend PASSER II so that relative offsets are "fine-tuned" within the slack times to minimize delay without reducing bandwidth. Delay for each link is computed by an analytical formula, assuming fixed link volumes.

Lan et al [4] developed the COMBAND model, an extension of the MULTIBAND model which incorporates a delay equation based on constant link volumes and deterministic flow. The objective function, a combination of bandwidth and delay, is optimized by mixed integer convex programming.

The approach that has been found to be more successful is the one discussed next, which is to modify a delay based model (namely TRANSYT or TRANSYT-7F) to cater for bandwidth.

3.5.3 Modifying delay based methods to consider bandwidth

Initial efforts involved concurrent use of TRANSYT / TRANSYT-7F with a bandwidth model [52, 53, 54, 55]. A timing scheme with good progression is first obtained from a bandwidth program such as PASSER II or MAXBAND. This scheme is then used as the initial timing plan for the TRANSYT / TRANSYT-7F hill-climbing procedure. The motivation for this approach is that the final solution generated by the delay-based model will have small delay while still maintaining the good progression inherent in the initial solution.

Cohen and Liu [56] and Liu [57] later modified the TRANSYT hill-climbing procedure to preserve the bandwidth of the initial solution. This was accomplished by modifying the rectangular optimization procedure applied to each signal to only accept those changes that improve the performance index without degrading the through bands.

Release 7 of TRANSYT-7F resolved the problem of having to use the model in conjunction with a bandwidth program by introducing explicit modelling of progression within the model via the Progression Opportunities (PROS) measure [58]. PROS is a generalized bandwidth measure that makes allowance for short term progression opportunities along an arterial [59]. PROS can be included in the performance index calculation in combination with the usual measures such as delay and stops.

4 Chapter Four – Oversaturated Conditions

4.1 Introduction

The models discussed in the previous two chapters are applicable in undersaturated conditions. These are conditions where flow rates are below capacity level (see 2.4.1). During "rush periods", flow rates may exceed capacity [2], a situation known as oversaturation. Oversaturated conditions are typified by the following traffic phenomena [60, 61]:

- Time dependent flow and dynamic control

  Flow conditions are no longer stationary. Because more traffic is demanding service than can be accommodated, conditions at any point in time are a function of past flow patterns and signal control decisions. This consideration dictates time dependent signal control.

- Reduced saturation flow

  With flow rates exceeding capacity, links may experience excessive queue build-up. This causes motorists entering such links from upstream to hesitate, resulting in increased departure headways and consequentially a reduction in the saturation flow rate. This reduces capacity and may propagate the congestion to upstream links.

- Spillback

  In extreme cases, queue build-up can lead to complete blockage of links, where no traffic can discharge from upstream on a green signal. This condition is known as spillback or de facto red.

- Intersection Blockage

  Even when links are completely blocked, vehicles may still discharge from the upstream signal and queue in the intersection area. If the queue does not move before the end of the green period, these vehicles will block cross-street movements. This phenomenon is known as intersection blockage or grid-lock.

During oversaturated conditions, delay-based methods such as Webster's method [13] and TRANSYT [40] are no longer applicable as the idea of a steady state representative cycle, which can be optimized and then applied for the entire study period, is no longer applicable. Bandwidth methods are also invalid as the queue effects which they ignore are further accentuated in heavy traffic [62].

In this chapter, we review the analytical methods that have been developed to obtain signal timings in oversaturated conditions. Optimal control schemes are obtained using techniques from Control Theory [63]. Using terminology from control theory, the variables determining signal settings, which are to be optimized, are known as control variables. We first discuss the case of a single intersection and then the network signal timing problem. All analytical models discussed assume a continuous approximation to vehicle flow and a deterministic arrival process. We also discuss the modifications to the bandwidth method and the enhancements that have been made to the TRANSYT-7F [49] simulation model to extend their applicability to oversaturated conditions.

4.2 Single Intersection

4.2.1 Method of Gazis and Potts

Gazis and Potts [64] considered the signal timing problem for an intersection of two one-way streets under two-phase control. In addition to the assumptions mentioned in section 4.1, their formulation makes the following additional assumptions and simplifications:

- The arrival rates at the start of the control period are large enough to exceed capacity and eventually reduce to levels below capacity, signalling the end of the "rush period".
- The cumulative departure curve for each approach can be approximated by a smooth curve[28].
- The optimal control policy must simultaneously dissolve queues on both approaches[29]. The point of elimination of queues designates the end of the control period.

[28] The succession of red and green signal indications results in a cumulative departure curve with a "saw-tooth" shape. This is approximated by a smooth curve, i.e. departures occur continuously at all times in the cycle. This simplification removes the cycle length from the formulation and as a consequence, their approach does not produce an optimal cycle length figure.
[29] This assumption is required to ensure equitable treatment of traffic arriving on each approach.

They provided a derivation of the optimal control policy using a graphical approach as well as a mathematical derivation using Pontryagin's maximum principle [2, 63, 64]. They demonstrated that that total intersection delay over the control period can be minimized by initially allocating the maximum possible green time to the approach with the larger saturation flow rate. At some point in time called the switching point, the allocation is reversed, and the approach with the lower saturation flow rate is then given the maximum possible green time. The switching point is determined so as to ensure that the queues are simultaneously dissolved [30]. This switching of the control variable from one extreme to another is called bang-bang control [63].

4.2.2 Extensions to methodology of Gazis and Potts

The optimization framework of Gazis and Potts [64] requires that queues are not eliminated before the end of the period of oversaturation. Michalopoulos and Stephanopoulos [65] resolved this problem by adding queue length constraints to the formulation[31]. This may result in a control policy with more than one switching point. Papageorgiou [66] extended the model to the case of an intersection with more than two traffic streams by removing the constraint that queues be eliminated simultaneously.

4.2.3 Other models

Chang and Lin [67] developed a discrete time model. A recursive expression for approach delay during each cycle is obtained in terms of approach delay in the previous cycle. The advantage of the discrete time approach is that the cycle length enters the formulation and can be optimized[32]. Standard optimal control methods are applied to obtain delay minimizing signal timings. As is the case with the method of Gazis and Potts [64], queue lengths must remain positive for the period of analysis. De Schutter [68] developed an alternative formulation that accommodates for fully dissipated queues. The control policy which minimizes either total intersection delay or the

[30] They demonstrated that under certain conditions, a control policy that eliminates both queues at the same time does not exist.

[31] The method of Gazis and Potts [64] assumes a constant departure rate, even when queues are not present. Thus departures will occur even when there are no vehicles to be serviced. This can result in the queue length (which is the difference of cumulative arrivals and departures) becoming negative. Using constraints on queue length as proposed by Michalopoulos and Stephanopoulos [65], a lower queue length constraint of zero can be used to prevent negative queues. The upper constraint can be used to account for limited link storage space. As is the case with the work of Gazis and Potts [64], a solution may not exist.

[32] Another advantage is that the switch-over point in the bang-bang optimal control policy is guaranteed to occur at the end of a cycle.

maximum queue length is obtained by transforming the problem into an extended linear complementary problem for which specialized heuristic solution algorithms have been developed.

4.3 Signalized networks

4.3.1 Extensions to the model of Gazis and Potts

Gazis [69] extended the formulation of Gazis and Potts [64] to a system of two oversaturated intersections with one-way streets. He ignored turning movements, assumed instantaneous travel between intersections and ignored the storage limit of the link between the intersections. Michalopoulos and Stephanopoulos [70] extended on this work, allowing for constant proportions of turning traffic, fixed travel time between intersections as well as queue length constraints. They demonstrated that the optimal control policy becomes considerably more complex with each additional intersection and queue length constraint.

4.3.2 Other analytical models

Singh and Tamura [71] considered a discrete time approach where the state of the system is considered at the beginning of each cycle. Their approach requires the cycle time to be specified and the link travel times must be some integer multiple of the cycle time. The optimization objective is a weighted sum of squares of queue lengths. An optimal control policy is obtained using a hierarchical optimization algorithm. Their formulation was later extended and made more realistic [72, 73]. A major drawback of their approach is that relative offsets are not considered.

Gazis [74] has suggested an alternative heuristic method which uses linear programming.

4.3.3 Modification to the bandwidth approach

Lieberman et al [75] found that progressively designed offsets can in fact amplify congestion in oversaturated conditions. Vehicles travelling in the green band are often required to stop upon reaching the tail of a queue of stationary vehicles. Bandwidth programs do allow for a time advance of through bands to allow for the clearance of stationary queues (see 3.3.2). The problem is that the value of the queue clearance time must be arbitrarily specified by the user.

Rathi [76] as well as Tsay and Lin [77], have developed bandwidth approaches which incorporate crude traffic models for determining the queue clearance time. Arterial signal timings obtained by such models may exhibit simultaneous green or even reverse progression (i.e. downstream signals turn green before the upstream signals) on account of the queue clearance times.

4.3.4 Modifications to TRANSYT-7F for oversaturated conditions

TRANSYT-7F release 8, introduced in 1998 provided the following enhancements to allow for realistic modelling of oversaturated conditions [49]:

- Explicit modelling of queue build-up and spillback
- Modelling multiple cycles so that time dependent effects can be captured
- Flow rates are allowed to change over time. This allows the build-up to the rush period to be modelled accurately.

These updates, in combination with the enhancements discussed in 3.4.3, have made TRANSYT-7F the "state of the art" optimization model for determining fixed-time traffic signal timings in all traffic conditions.

5 Chapter Five – Genetic Algorithms

5.1 Introduction

In this chapter, we discuss genetic algorithms which are the global optimization tools found to provide effective optimization of traffic signal timings when the impact on network performance is evaluated by a simulation model. We first give a broad outline of the genetic algorithm optimization process. We then discuss an early implementation called the Simple Genetic Algorithm [(78)]. We then justify why the genetic algorithm produces an effective and robust search. This is followed by an account of different methods of structuring the genetic algorithms search, alternative genetic operators and problem encodings that have been found to provide more efficient optimization. An overview of findings from the literature on the performance of genetic algorithms on "Noisy problems" where the objective function is estimated via a stochastic result is then given. Finally, we discuss the various implementations of genetic algorithms to solve the traffic signal timing problem, highlighting the studies that justify the choice of the genetic algorithm as an appropriate optimization tool.

5.2 What are genetic algorithms and how do they work? [(78, 79)]

Genetic algorithms are computational models based on the mechanisms of natural selection and evolutionary theory. A primary application of genetic algorithms has been in the field of function optimization. Genetic algorithms are capable of handing both continuous and combinatorial optimization problems.

Genetic algorithms perform a heuristic global optimization search using a form of guided random search. The search is performed using a population of individuals. Each individual represents a point in the search space. For the traffic signal timing problem, each individual represents a particular network signal timing plan. The set of decision variables is encoded into a form of genetic material. Associated with each individual is the computed objective function value. For delay minimization in traffic networks, the associated objective function value will be the delay produced by the particular signal timings. Optimization is performed by manipulating the population of individuals using the following steps:

Initialization: The individuals in the initial population are assigned to points in the search domain. Typically, each individual is assigned to a random point in the search domain (i.e. each point in the search domain has the same probability of being chosen).

Selection: Individuals in the population are selected for reproduction. The selection probability for each individual is usually a function of the objective function value. For maximization problems, individuals with a larger objective function value have a larger selection probability. For minimization problems, individuals with a smaller objective function value are favoured for selection.

Recombination/Crossover: Once individuals have been selected for reproduction, these "parents" are paired and one or more "children" are created using a crossover operator. Crossover creates children by combining or blending the genetic material of the two parents (i.e. the decision variable sets of the two parents are combined to form a new set for each child).

Mutation: The mutation operator performs random alterations to the genetic material of an individual. Mutation will alter one or more of the individual's decision variables with small probability. Mutation is typically applied to the children created by recombination.

Replacement: Typically, a fixed finite population size is applied and a replacement scheme is defined to determine which individuals from the parent and child populations will survive.

Initialization is performed and the remaining steps are repeated until a stopping criterion is met. One possible stopping condition is to stop the search after a pre-specified number of objective function evaluations have been performed. The individual with the best objective function value is taken as the optimal solution produced by the algorithm.

5.3 Simple Genetic Algorithm [78]

5.3.1 Introduction

A popular basic implementation of a genetic algorithm is given by Goldberg [78] which he refers to as the Simple Genetic Algorithm (SGA). We discuss the algorithm details under the following headings:

- Problem encoding
- Initialization
- Selection
- Crossover
- Mutation
- Replacement

5.3.2 Problem encoding

A genetic encoding of the decision variables is required for the construction of operators to act on the genetic material in an appropriate way. The SGA, along with many earlier genetic algorithm implementations, makes use of a binary coding. For the optimization of a problem in a single real valued variable $x \in [a,b]$, a binary string of length l can be used.

If $y = y_l y_{l-1} \ldots y_2 y_1, \quad y_k \in \{0,1\} \ \forall\, k \in \{1,2,\ldots l\}$,

denotes a particular point in the search space in the binary coding, then the corresponding real valued value can be computed by converting the binary value to a decimal value in the range $\{0,1,2,\ldots,2^l - 1\}$ and then converting this into a value in the range $[a,b]$ using a linear mapping i.e.

$$x = a + \frac{b-a}{2^l - 1} \sum_{k=1}^{l} y_k 2^k \tag{5-1}$$

The resolution of the binary coding can be increased by using a larger value of l. For a multi-parameter optimization problem, each decision variable x_j may have its own search domain $[a_j, b_j]$ and binary string z_j of length l_j. The vector of decision variables $\underline{x} = (x_1, x_2, ...)$ can be

represented by a single binary string of length $L = \sum_{j=1} l_j$ which is the concatenation of the binary strings $z_1, z_2, ...$

5.3.3 Initialization

Initialization is performed by independently setting the bit values in the binary string to either 0 or 1 with equal probability. Thus each point in the search space has equal probability of being included in the initial population. The individuals in the initial population constitute the first generation.

5.3.4 Selection

Selection in the SGA is defined with respect to maximization problems[33]. In addition, the range of the objective function must be positive.

Let N_{Pop} = number of individuals in the population and

f_i = objective function value for the *i*'th individual (also called the fitness).

Selection probabilities are given by:

$$\Pr(i\text{'th individual selected}) = \frac{f_i}{\sum_{j=1}^{N_{Pop}} f_j} \qquad (5\text{-}2)$$

That is, selection is proportional to fitness. The selection procedure is repeated N_{Pop} times with replacement to form a mating pool of N_{Pop} individuals.

5.3.5 Crossover

Individuals in the mating pool are paired at random to produce offspring. Genetic material is interchanged to form two children using 1-point crossover. A crossover point is chosen at random and the genetic material of the parents on the two sides of the crossover point is swapped to form two children i.e.

[33] In 5.5.4 we give details on how minimization problems can be treated.

Let $c = c_L c_{L-1} \ldots c_2 c_1$ and

$d = d_L d_{L-1} \ldots d_2 d_1$ denote the binary encodings of the two parents,

u = discrete uniform random variable $\in \{1,2,\ldots,L-1\}$

The binary encodings for the children c^* and d^* are given by $c^* = c_L \ldots c_{u-1} d_u \ldots d_1$ and $d^* = d_L \ldots d_{u-1} c_u \ldots c_1$.

Crossover is performed with probability p_c. If crossover is not performed then the children are identical to the parents

5.3.6 Mutation

Mutation is performed by flipping each bit in the children independently with probability p_m.

5.3.7 Replacement

The individuals in the population constitute a generation. After selection, crossover and mutation, a new population or generation of individuals is formed. These individuals completely replace the ones in the previous generation. This process is repeated until the stopping criterion is met.

5.4 Why do genetic algorithms work? [78, 79]

5.4.1 Introduction

The question naturally arises as to why an algorithm as outlined in 5.2 and 5.3 will result in an effective optimization search. We first give qualitative justifications for each component of the optimization procedure. This is followed by a more rigorous theory based explanation which highlights the intrinsic power of the genetic algorithm search.

5.4.2 Qualitative justifications of optimization steps

Selection: By favouring better solutions, the selection procedure guides the search towards high performance regions of the search space.

Crossover: Crossover performs a structured yet randomized information exchange, leading the algorithm towards new points in the search space. Taken in conjunction with selection, the algorithm is more likely to search points that are close to the better performing individuals. For the SGA, the crossover probability p_c can be used to control the rate of exploration. Setting $p_c = 1$ and always performing crossover may result in the loss of good genetic material in the before it has been sufficiently exploited. With $p_c < 1$, there is the possibility of the genetic material passing intact into the next generation.

Mutation: The benefits of mutation are two-fold:

- Mutation induces random alterations to the genetic material. If the alteration induced is desirable in that the objective function value is improved, then the individual will have a higher chance of passing on its genetic material thanks to the selection procedure. Thus mutation can be thought of as a hill-climbing component of the search. In fact, a search algorithm using mutation alone can be effective.
- Selection, crossover and replacement may result in the loss of potentially useful genetic material (e.g. for the binary encoding, a 0 may no longer be present at a particular bit position in any of the individuals in the population). Alternatively, the initial population may lack this genetic material. Mutation allows for the restoration or introduction of this genetic material in a new context where it may be found to improve the objective function value.

5.4.3 Genetic algorithm theory

Aside from these qualitative justifications, several theories have been proposed to explain the success of genetic algorithms. The most widely cited result is the schema theorem which requires the notions of schema and hyperplane sampling and leads to the idea of implicit parallelism.

Schema

Schema are solution templates. A schema is a string of the same length as the binary string, *L*, with either a 0, 1 or "*" at each position, where "*" is a "do not care" symbol. The number of positions with either 0 or 1 is the order of the schema. A binary string which matches the schema at all positions other than those with "*" is called a schemata.

For example, the schema *0*1 is of order 2 and is matched by the 4 binary strings (schemata) 0001, 0011, 1001 and 1011. A schema represents a subset of "similar" strings. The schemata fall on a hyperplane in the search space $\{0,1\}^L$. The objective function value averaged over the schemata $\{0000, 0011, 1001, 1011\}$ is the fitness of the schema *0*1. We have 3^L possible schema.

Schema can also be thought of as partial solutions to a problem. For example, suppose you have a problem with 2 decision variables where each decision variable is coded with 2 bits. The schema **01 corresponds to solutions where one of the decision variables is specified and the other is not. High-order schema correspond to detailed partial solutions. The partial solutions are also called building blocks.

Hyperplane sampling

Consider a population made up of the following $N_{Pop} = 3$ individuals: 0000, 0011 and 1111. The schema (hyperplane) *0*1 is represented by 2 individuals in this population. However, the schema 1*** and 0*** are also represented. In general, each binary string is a member of 2^L schema[34]. Thus, a population of individuals samples numerous hyperplanes.

Implicit parallelism

Information on many hyperplanes (partial solutions) is obtained when a single individual is evaluated. Through the action of genetic operators, the proportional representation of hyperplanes is altered. Thus the genetic algorithm search exhibits implicit parallelism in that many hyperplane competitions are being solved simultaneously. We expect high fitness hyperplanes to receive increasing number of samples as the genetic algorithm search progresses. Low-order schema (i.e. vague partial solutions) are sampled more frequently in the initial population. As the proportional representation of particular schema dominate, the representation of high-order schemata increase and the partial solutions are refined.

[34] For example, the binary string 0000 is a member of any schema with 0 or "*" at each bit position. There are $2^4 = 16$ schema satisfying this criteria.

Schema Theorem

The parallel nature of the genetic algorithm search is captured mathematically by the schema theorem (see Goldberg [78]). In summary we find that the power of genetic algorithm search, according to the schema theorem, lies in its ability to construct optimal solutions through a process of discovery by speculating on many combinations of the best partial solutions.

5.5 Improvements to the simple genetic algorithm

5.5.1 Introduction

Since the pioneering work of Holland [81] and DeJong [82] on genetic algorithms, many new operators, alternative algorithms and encodings have been, and continue to be introduced. Since the genetic algorithm is a heuristic search method, the modifications are usually evaluated empirically by comparing the performance of the search algorithm with, and without the modification, on a suite of test functions of varying attributes. Several independent replications of the algorithm are performed since genetic algorithms are stochastic search algorithms and researchers are interested in the mean performance of the algorithm. The performance of the algorithm on a particular test function is usually measured using one or more of the following criteria:

- Probability of the algorithm locating the global optimum given a fixed number of function evaluations.
- Mean number of function evaluations required to locate the global optimum.
- Quality of the best solution for a fixed number of function evaluations.

Other than the best solution, online and offline performance are sometimes considered (see Goldberg [78]).

Here, we restrict our attention to algorithm adjustments that:

- Have been conclusively proven to increase search efficiency,
- Have been adopted in applications outside the original publications and
- Have potential for implementation in the traffic signal optimization problem.

5.5.2 The elitist method

The elitist method guarantees a monotonic improvement in the best value of the objective function by ensuring that the best individual survives intact into the next generation. This can be accomplished by introducing the following mechanism into the SGA [78]:

If the best individual in generation *t* is not included in generation $t+1$, then include it as the $\left(N_{Pop}+1\right)$'th member in generation $t+1$.

Incorporating the elitist method with a more explorative crossover operator such as uniform crossover (see section 5.5.3) has been shown to improve optimization performance [83, 84, 85].

5.5.3 Uniform crossover [86, 87]

Uniform crossover is an alternative crossover operator. Individual bits in the binary strings of the two parents are swapped independently with probability 0.5. This process is repeated for all bits in the string, forming the two children. Uniform crossover, in practice, is preferred to 1-point crossover since:

- Uniform crossover has more explorative power.
- With multi-parameter problems, 1-point crossover is sensitive to the order in which decision variables are concatenated into a single string (e.g. variables that are far apart on the string are more likely to be separated by 1-point crossover). Uniform crossover, on the other hand, is indiscriminate to the order of bits in the string.
- Uniform crossover has been empirically shown to give improved optimization performance.

5.5.4 Ranking and tournament selection

The SGA selection procedure has several shortcomings [78]:

- Selection pressure is sensitive to the form of the objective function e.g. for maximization of the objective functions $f_1(x)=x^2$ and $f_2(x)=x^{10}$, $x\in[1,10]$, the selective pressure in favour of better solutions will be much larger for f_2 than f_1 and the genetic algorithm will converge faster when maximizing f_2.

- A transformation of the objective function value is required to convert a minimization problem into a maximization problem to which fitness proportional selection can be applied. The performance of the algorithm is sensitive to the form of the transformation[35].
- At the start of the genetic algorithm run, it is not uncommon for there to be an individual in the initial population with unusually high relative fitness. Using selection proportional to fitness may result in the predominant dissemination of genetic material from this super individual, leading to premature convergence.
- Later on in the run, when the population has mostly converged and fitness values are more homogenous, the selective pressure induced by the normal selection rule may be too small and the search may stagnate.

Several mechanisms to scale fitness values have been proposed to maintain more constant selective pressure, solving the last three problems [78, 88, 89]. Two selection schemes that solve all four problems are ranking and tournament selection.

Ranking [90, 91]

Individuals in the population are ranked from best to worst based on the objective function value. Selection probabilities are computed based on rank. This is best accomplished by a linear function where probability of selection decreases linearly with rank. The selection process is parameterized by the bias factor β where:

$$\beta = \frac{\Pr(\text{Top ranking individual selected})}{\Pr(\text{Median ranked individual selected})} \tag{5-3}$$

A linear function is appropriate for $1 \le \beta \le 2$.

[35] In optimization, a common approach for modifying a maximization algorithm to perform minimization is to run the maximization algorithm on the negative of the objective function. The transformed objective function has the same form as original objective function, just with a negative sign. However this transformation cannot be used for a positive objective function as the SGA selection procedure can only be applied to a positive objective function (see section 5.3.4). If $f(\underline{x})$ is the positive objective function to be minimized, one can maximize $[f(\underline{x})]^{-1}$ or $-f(\underline{x}) + C$ for an appropriately large value of C. There are many transformations that can be applied but the performance of the algorithm then becomes dependent on the choice of transformation.

Tournament selection [92, 93]

Tournament selection operates by selecting *m* individuals randomly from the population and choosing the best of the *m* individuals for further genetic processing. For $m = 2$, the selection procedure is called binary tournament selection and has been shown to give the same selection probabilities as linear ranking with $\beta = 2$. The value of β can be increased by increasing *m* and decreased by selecting the worst individual as the tournament winner with some positive probability.

Linear ranking and tournament selection have been shown to induce a more constant selective pressure and improve optimization performance over fitness proportional selection or selection using other scaling mechanisms [84, 90, 91, 94, 95, 96, 97]. Generally tournament selection is recommended as it does not require the additional overhead of sorting the population [92, 93].

5.5.5 Alternative algorithms

Here we discuss two alternative genetic algorithms that have been shown to give improved performance over genetic algorithms based on the SGA. We will refer to implementations of the SGA with modifications such as those given in sections 5.5.2, 5.5.3 and 5.5.4 as a Generational Genetic Algorithm (GGA).

Steady State Genetic Algorithm

Steady state genetic algorithm (SSGA) is a term coined by Syswerda [86]. GENITOR [91, 97], the first SSGA, searches from a population of points like all genetic algorithms, with the following differences:

- Unlike GGA's where reproduction creates an entire generation of offspring, reproduction in GENITOR involves the generation of a single offspring at a time from two parents.
- Offspring do not replace parents as is the case with GGA's. Instead, the offspring will replace the least fit member of the population provided the new individual has better fitness.
- Duplicates are not allowed. If any of the children produced are duplicates of any of the current members of the population in that the genetic material is identical, the offspring

will be rejected. Selection, recombination and mutation will be repeatedly applied until a unique individual is created.

- Since an ordered list of individuals by rank is required for the replacement mechanism, selection is performed using ranking.

The advantages of performing the search in this way are [91, 97]:

- For GGA's, the elitist method guarantees the survival of the best individual. With the SSGA, the replacement mechanism is population elitist in that all good individuals are protected from deletion.
- Crossover is always applied. Thus the search can be more explorative without endangering the survival of good members of the population.
- In the GGA, probabilistic crossover and mutation may allow individuals to survive intact into the next generation. Since selection favours high performance individuals, the algorithm is more likely to produce duplicates, resulting in a loss of genetic diversity and possible premature convergence. By always applying crossover and performing duplicate checking, SSGA's avoid this problem.
- Since reproduction is performed by introducing a single new offspring at a time, newly created high performance genetic material is immediately available for exploitation[36].

Empirically, SSGA's have been shown to consistently outperform GGA's [83, 91, 94, 97, 98, 99].

CHC

CHC [93, 100] stands for cross-generational elitist selection, heterogeneous recombination and cataclysmic mutation. CHC operates on a generational basis. Individuals in the population have equal selection probability (i.e. there is no selective pressure towards better individuals). The replacement mechanism operates by combining the parent and offspring population and selecting the *N* best individuals to form the next generation. CHC also includes several unique mechanisms to improve search capabilities:

- An incest prevention mechanism is implemented to prevent "similar" individuals from mating. The "dissimilarly" of two individuals is measured by the Hamming distance of

[36] In the GGA, we must wait an entire generation before new genetic material is available for further processing.

the bit strings which is the count of the number of differing bits. Only when the Hamming distance is above a certain threshold will individuals be allowed to mate and produce offspring. The value of this threshold is decreased as the population converges and individuals become more "similar".

- Reproduction is via crossover alone. A crossover operator called Half Uniform Crossover (HUX) is applied. HUX randomly swaps exactly half of the differing bits of the two parents, forming two children.
- Once the population has converged and no further reproductive opportunities are available, a process called cataclysmic mutation is applied. A new population is created of which the best individual so far is a member and all remaining individuals are heavily mutated copies of this best individual. These mutations are obtained by specifying a divergence rate *r* and flipping each bit in the best individual independently with probability *r*. The Hamming threshold is reset and the algorithm resumes in the usual manner.

CHC shares the following advantages with SSGA's [100]:

- An elitist type replacement strategy is utilized which protects good individuals from deletion.
- Crossover can always be applied thanks to the elitist mechanism and search can be more explorative.
- The incest prevention mechanism, HUX and cataclysmic mutation results in a very small probability of generating duplicate strings.

CHC also has the following advantages over other types of genetic algorithms [100, 101]:

- Incest prevention prohibits similar individuals from mating, thus delaying convergence. This allows for a more sustained search.
- CHC uses HUX which is a highly explorative crossover operator.
- The inclusion of incest prevention and HUX crossover prevent premature convergence and allow for reproduction to occur without the need for mutation to restore lost genetic material. Thus crossover which is the source of the power of genetic algorithm search can be applied without the disruptive effect of mutation.

- When convergence is achieved, mutation is applied at the outer loop of the algorithm to reintroduce diversity while preserving the progress made so far. Micro Genetic Algorithms [102] operate in a similar manner[37] and thus CHC inherits the advantages of Micro Genetic Algorithms in terms of being able to search with a smaller population size [102].
- CHC requires only two parameters to be specified: The population size N_{Pop}, and the divergence rate *r*.

CHC has been found to outperform GGA's [100, 101, 103] and GENITOR [93]. In addition, CHC has been found to require less parameter tuning than other genetic algorithms [93, 100].

5.5.6 Alternative problem encodings

Shortcomings of the binary coding

According to the schema theorem [78], genetic algorithms operate on the assumption that "similar" strings yield similar performance. In CHC, the "dissimilarity" between strings is measured by the Hamming distance. Thus for the ideas of hyperplane sampling and implicit parallelism [78], the identified source of power of the genetic algorithm search, to hold, we require "similar" strings to have a small Hamming distance. An extreme case where the binary representation fails in this regard is the Hamming cliffs problem [104] which we illustrate via an example:

Consider a problem a single decision variable problem with search integer domain $x \in \{0,1,\ldots,15\}$ for which a 4 bit binary string is appropriate. The strings 0111 and 1000 correspond to the integer points $x = 7$ and $x = 8$ respectively, which are adjacent in the problem space. However, these strings are the maximal Hamming distance apart. Suppose that the function optimum is at $x = 8$ and the function is smooth in that $x = 7$ gives a value close to the optimum. It is very unlikely that mutation and crossover acting on the string 0111 (a good individual), will be able to change the value of each bit and move to the global optimum.

[37] Micro Genetic Algorithms operate by repeatedly performing standard genetic algorithm optimization with a small population. After each iteration of the genetic algorithm search, the genetic algorithm is restarted by generating a new population and transferring a specified number of the best individuals of the converged population to the new population and generating the remaining individuals randomly [102].

Gray coding

A method for removing the Hamming cliffs problem is the use of an alternative binary coding called Gray coding which guarantees that adjacent points in the integer domain have a Hamming distance of one [(104)]. The standard binary coding and corresponding Gray coding for a 3 bit problem is given in Table 1.

Decimal value	Standard binary coding	Gray coding
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Table 1: Binary coding and Gray coding[38]

We present algorithms for conversion between the standard binary code and Gray code as given by Wright [(105)]:

For a particular decision variable coded in using l bits, let $y_l^b y_{l-1}^b \ldots y_2^b y_1^b$ denote the standard binary representation and $y_l^g y_{l-1}^g \ldots y_2^g y_1^g$ denote the representation in Gray code. Conversion from standard binary to Gray code can be performed using equation (5-4) where $\oplus$ denotes addition mod 2. Conversion from Gray to binary code is performed using equation (5-5). Alternative conversion algorithms in terms of binary operators are given by Michalewicz [(104)].

$$y_k^g = \begin{cases} y_l^b & for\ k = l \\ y_k^b \oplus y_{k+1}^b & for\ k < l \end{cases} \qquad (5\text{-}4)$$

[38] From the table we find that the Hamming distance between adjacent points under the standard binary coding ranges from one to three. In Gray code, the Hamming distance of adjacent points is always one.

$$b_k = g_k \oplus g_{k+1} \oplus \cdots \oplus g_l$$

(5-5)

Genetic algorithms using Gray coding have been empirically shown to perform as well or better than those using the standard binary coding [106, 107, 108]. On the other hand, Gray coding has been found to marginally retard performance on some test functions [107, 108]. There is no general consensus on which type of functions benefit from the use of Gray coding in optimization.

Real coding and operators

Binary representation is the traditional problem encoding method for genetic algorithms [78, 104]. The reasons for preference towards binary representation in the genetic algorithm literature, in terms of both theoretical and applied work are [78, 104]:

- It allows for the construction of simple genetic operators.
- Theoretical analysis is facilitated.
- According to the principle of minimal alphabets, the quantity of information (i.t.o. number of schema) processed by a population of individuals is maximized by the binary representation.

The last point is the primary theoretical result leading towards the predominance of the binary representation. Antonisse [109] and Goldberg [110] present alternative arguments, highlighting methodological shortcomings in the theory and argue in favour of non-binary representations. Reeves [111] gives other arguments in favour of binary alphabets.

Theoretical arguments aside, genetic algorithms using real or integer representations have become more popular over time. With these representations, encoding and problem spaces correspond [104]. Genetic operators thus manipulate genetic material directly in the problem space. Artefacts of the encoding space such as Hamming cliffs [104] and the impact of number of bits used in parameter coding on mutation size [108] are removed. The implementation of the genetic algorithm is also simplified as a binary representation is no longer required.

Genetic operators need to be modified for these representations. Of the many real crossover operators that have been developed, we will consider blend crossover (BLX-α) [112].

With BLX-α, crossover is applied separately to each decision variable. For the *i*'th decision variable, let

x_i and y_i denote the values of the decision variables in the two parents ($x_i, y_i \in \Re$),

$I_i = |x_i - y_i|$,

$l_i = \min(x_i, y_i) - \alpha I_i$,

$u_i = \max(x_i, y_i) + \alpha I_i$ and

$z_i =$ value of the *i*'th decision variable in an offspring of the two parents.

z_i is computed stochastically by simulation using

$$z_i \sim U(l_i, u_i),$$

(5-6)

where $U(a,b)$ denotes the continuous uniform distribution on the interval (a,b). That is, the value of *i*'th decision variable in an offspring is chosen randomly from a set of values in a close vicinity of the decision variables of the two parents. We illustrate this domain graphically in Figure 10.

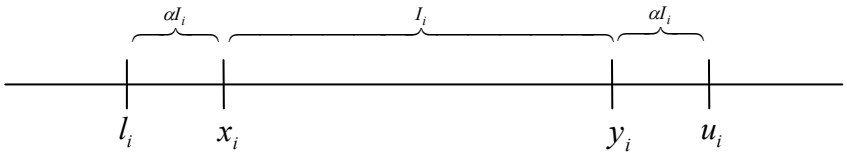


Figure 10: Graphical illustration of blend crossover

The upper and lower limits in equation (5-6) can be restricted for a problem with a limited search domain. Typical choices for α are $\alpha = 0$ and $\alpha = 0.5$ [(112)].

Mutation can be achieved by replacing the value of a decision variable with a uniformly distributed value in the allowable search domain [(104)].

Empirically, mixed results have been observed when comparing empirical performance of binary coding with real coding using real genetic operators [84, 104, 112, 105, 113].

5.5.7 Optimization in noisy environments

The majority of empirical work on genetic algorithms has been performed using test suites of mostly deterministic functions. The performance of genetic algorithm search on a function returning a stochastic result is of importance in this dissertation as the measure of effectiveness produced by a stochastic traffic simulation model is exactly such a function. We wish to search for the decision variables that maximize or minimize the mean of the objective function.

An estimate of the mean function value can be obtained by evaluating the objective function once (i.e. a single replication) [114]. The precision of this estimate can be improved by performing repeated replications and using the sample mean [114]. However, if the objective function value is obtained via a time consuming computer simulation, then there will be a trade-off: For a fixed amount of computing time (number of function evaluations), increasing the number of replications per fitness evaluation will decrease the extent of the search.

GGA's have been found to perform effective optimization using just a single replication [96, 103, 114, 115, 116]. Explanations for the robust performance of GGA's in noisy environments using a single replication for fitness evaluations have been given based on schema theory and hyperplane sampling [103, 114, 115]. However, for higher noise levels, performing additional replications can be beneficial [96, 117]. Aizawa and Wah [117] note that the optimal number of replications to perform increases as the total number of allowable function evaluations increases. Thus, there does not appear to be a unified consensus on whether additional replications can improve optimization performance.

When applying GGA's to problems with stochastic elements using a single replication, a particular individual may receive an unusually high fitness evaluation when in fact the solution is poor (i.e. true mean value of objective function far from optimal). The individual will be favoured by selection and may persist in the population for a long time and disseminate bad

genetic material. To resolve this problem, it is common practice to re-evaluate all individuals which survive unaltered from one generation to the next [84, 99, 113].

For SSGA's and CHC, such a re-evaluation mechanism is not feasible and consequently the GGA has been favoured for applications to noisy problems [84]. However, the studies demonstrating the superiority of these alternative genetic algorithms (see section 5.5.5) have included a 10-dimensional unimodal quartic function with normally distributed noise in the test function suite. The alternative algorithms have shown improved performance over GGA's for this function. In addition, Rana et al [103] demonstrated the superiority of CHC over a GGA on three noisy test functions.

Thus the superiority of SSGA's and CHC for optimization in noisy environments has not been convincingly demonstrated.

5.6 Genetic algorithms in traffic signal optimization

Foy et al [118] were the first to apply genetic algorithms to the traffic signal timing problem. Delay for a hypothetical four signal network under two phase control was minimized using the SGA. Delay calculations were performed by a simple microscopic traffic simulation model. The problem encoding included cycle length, green splits and offsets[39].

Memon and Bullen [119] compared the optimization efficiency Quasi-Newton search and the SGA on a five signal network. Delay computations were obtained from a macroscopic simulation model. Phase sequence optimization was not considered. They noted superior performance of the SGA for more complicated optimization scenarios.

Oda et al [120] compared the TRANSYT [40] hill-climber, Random Search as well as a SGA with 2-Point crossover[40] and elitist strategy on two large grid networks. Objective function evaluations were performed by the TRANSYT simulation model. The genetic algorithm outperformed the other two optimization methods on both networks.

[39] The encoding allowed for offset to be adjusted only in a very limited manner.
[40] In 2-point crossover, two crossover points are randomly chosen and the genetic material between the two crossover points is swapped to form offspring.

Park [121] provides a thorough application of genetic algorithms to optimize all timing variables besides the number and structure of signal stages. He used a SGA with uniform crossover, tournament selection and the elitist method. A variety of optimization criteria were considered and measures of effectiveness were computed by a stochastic traffic simulation model. The genetic algorithm was found to provide effective optimization for hypothetical two signal and four signal networks and outperformed the TRANSYT-7F [49] hill climbing procedure. For a simplified two signal network, the genetic algorithm found a solution with delay only 1% larger than that of a full enumerative search.

Lo and Chow [122] used a SGA with fitness scaling to compare optimal static and dynamic signal timing policies on a three signal arterial. Optimization of phasing was not considered. The delay of each signal timing policy was evaluated by a cell transmission model which is an alternative macroscopic modelling approach for traffic networks.

Rouphail et al [123] demonstrated the feasibility of optimization with genetic algorithms using CORSIM [50] to evaluate delay. Optimization of phasing was not considered in their study. The latest release of TRANSYT-7F, version 10.2 [49], as already mentioned in section 3.4.3, now supports optimization of all traffic timing variables besides the number and structure of stages using the genetic algorithm given by Park [121]. Measures of effectiveness can be computed using the TRANSYT-7F internal macroscopic traffic model or by calling the CORSIM microscopic simulation model externally. PASSER V, a traffic signal optimization program with a macroscopic traffic model very similar to TRANSYT-7F also allows for genetic algorithm optimization [124]. Approaches combining genetic algorithms with the TRANSYT-7F hill climbing optimization procedure have also been considered [125, 126].

Sun et al [127] applied a multi-objective genetic algorithm for generating the set of Pareto-optimal signal timing policies for minimum delay and stops at an isolated intersection. Delay and stops were computed via analytical formulae. Park et al [128] utilized genetic algorithms for determining the times to change signal timing plans in multi-dial systems. Genetic algorithms have also been applied to models for use with on-line optimization of traffic signals [60, 129, 130].

6 Chapter Six – Research Questions

6.1 Introduction

In this dissertation, we will be examining several untested modifications to the genetic algorithm when applied to traffic signal optimization. The problem will be considered for the case where a stochastic traffic simulation model is used to evaluate the quality of solutions in the search process.

In this chapter, we first give motivation for such a study. We then discuss the alternative operators, parameter values and different algorithms that were tested.

6.2 Motivation

6.2.1 Why do we need a better genetic algorithm?

In section 5.6, an overview of literature demonstrating the effectiveness of genetic algorithms as an effective optimization strategy for setting traffic signal timings was given. However, the majority of the research has considered small problems with only a few intersections. In certain cases, not all decision variables were optimized.

Rouphail et al [123] using the genetic algorithm given by Park [121] (see section 5.6) applied a genetic algorithm to a nine signal network in Chicago. Cycle length, green times and offsets were optimized, resulting in a total of 22 decision variables. Phase structure and sequence were pre-specified. The genetic algorithm converged to a very poor solution. A heuristic reduction in the size of the parameter space searched by the genetic algorithm was performed and a set of signal timings with 53% less delay than the genetic algorithm without the modification was obtained. CORSIM [50] was used for performing the objective function evaluations. The genetic algorithms was limited to 625 function evaluations and took over 7 hours to complete the optimization using desktop computing technology of the year 2000.

Based on this study on a large network, a search for more efficient genetic algorithms optimization strategies is in order.

6.2.2 Why use a stochastic traffic simulation model?

Stochastic microscopic traffic simulation models, CORSIM [50] being the most widely used, have been found to provide adequate representations of reality (see section A1.4.3). A deterministic macroscopic simulation model with faster execution speed such as the TRANSYT-7F [49] traffic simulation model would be preferred as a much more extended optimization search can be performed given the same amount of computing time. However, Park et al [131] performed comparisons between TRANSYT-7F and CORSIM under identical traffic situations and noted discrepancies between the computed performance measures, especially in oversaturated conditions. Similar discrepancies between CORSIM and TRANSYT-7F have been found in other studies [37, 52, 57, 132]. Wong [133] also noted poor agreement between field measurements and TRANSYT-7F model predictions.

Thus, a microscopic stochastic traffic simulation model called MSTRANS was developed for the purpose of this research. MSTRANS is discussed in section 7.2.

6.3 Research Questions

6.3.1 Introduction

We now detail the various parameter and implementation modifications of the genetic algorithm that were tested in this research. We highlight the unanswered questions about the changes to give justification for testing them as potential improvements to the optimization process.

6.3.2 Re-evaluation of fitness in GGA's

When GGA's are applied to functions with noisy evaluations, individuals that enter the next generation unaltered are re-evaluated (see section 5.5.7 for the arguments in favour of this approach). The disadvantage of this is that when the total number of function evaluations is limited, the re-evaluation of existing points will reduce the number of new points that can be investigated. Although the logic in favour of re-evaluating individuals may be sound, we have not found any empirical research demonstrating its effectiveness. We wish to investigate whether

re-evaluation in fact increases search efficiency. If so, we would expect it to be less useful at lower noise levels. Performance measures produced by microscopic traffic simulation models exhibit larger variability when flow is close to, or exceeds capacity [131, 134, 135]. Thus, lower noise levels correspond to undersaturated conditions where the re-evaluation mechanism may be less effective.

6.3.3 Common Random Numbers

The use of common random number's (CRN's) is a well known variance reduction technique for stochastic simulation experiments [136].

Here, we discuss CRN's briefly. When comparing alternative system configurations (e.g. in traffic signal optimization, we compare different signal timing policies) via stochastic simulation, differences in computed performance measures arise from two sources:

- Difference in system design (e.g. alternative signal timing policies have varying degrees of effectiveness).
- The stochastic nature of experiments.

The variability attributable to the second source can be removed by performing experiments under identical conditions (e.g. subjecting different signal timing policies to the same pattern of traffic flow). This is achieved by using CRN's where the same random numbers are used for generation of stochastic effects in all configurations. CRN's and their application to traffic signal optimization are discussed further in Appendix A2.

Rathi [137] found CRN's to provide substantial reductions in the variance of the difference between performance measures of alternative traffic signal timing policies. A facility for performing CORSIM runs with pre-specified random seeds was subsequently implemented [50].

When utilized in genetic algorithms for fitness evaluations, the reduction in the variability of the difference between individual fitness measurements allow for selection and replacement to operate with greater precision. The genetic algorithm implementations in traffic simulation, given in section 5.6, which utilize stochastic simulation models for objective function

evaluations performed independent replications. When executing genetic algorithms using CORSIM [50] for measuring fitness, TRANSYT-7F uses CRN's [49].

In spite of the compelling arguments in favour of utilizing CRN's in genetic algorithms, we have a few reservations:

- When performing only a single replication for fitness evaluations, a single arrival pattern and routing scheme will be applied to all individuals. Will a single set of experimental conditions be adequate when searching for a set of signal timings that optimizes mean performance?
- The explanations for the success of genetic algorithms in noisy environments in terms of the schema theorem and hyperplane sampling (see section 5.5.7) are no longer applicable when CRN's are used.
- The re-evaluation mechanism in GGA's can no longer be used[41]. The benefit of re-evaluating individuals is no longer available.
- We are unaware of any genetic algorithms research demonstrating the benefit of applying CRN's.

Thus, we will investigate the effect that CRN's has on the optimization performance of genetic algorithms when applied to the traffic signal timing problem.

6.3.4 Number of replications

In section 5.5.7, we noted several studies demonstrating robust performance of genetic algorithms on noisy functions, even when performing a single replication for fitness evaluations. Other studies found that improved performance could be achieved at higher noise levels, or when a large number of total function evaluations are available, by increasing the number of replications per fitness evaluation. The studies on traffic signal optimization listed in section 5.6 have made use of a single replication for fitness evaluations. When performing genetic algorithms optimization using CORSIM [50] as the evaluator, the TRANSYT-7F user manual [49] recommends performing additional replications in oversaturated conditions, but does not give any information on how many additional replications to perform.

[41] Re-evaluating an individual with the same random seed values will produce the same objective function value.

With respect to the application of genetic algorithms to the determination of optimal traffic signal timing policies, we will investigate:

- Whether performing more than one replication is beneficial.
- If so, are there any guidelines on how many replications to perform such as:
    - In oversaturated scenarios, there is a larger variability in measured performance values and the genetic algorithm may benefit from additional replications.
    - How does this vary with respect to the number of total function evaluations that are available?

6.3.5 Alternative problem encodings

All applications thus far on traffic signal optimization have used the standard binary representations with the standard genetic operators. We wish to test whether the following alterative problem encodings improve traffic signal optimization efficiency.

Gray coding

In section 5.5.6, we noted that Gray coding removes the Hamming cliffs problem and has potential to improving optimization.

Real coding

In section 5.5.6 we discussed the potential benefits of using a real parameter coding for genetic algorithms. Empirically, results have been inconclusive. We wish to test if the BLX-0.5 crossover and real mutation operators improve optimization efficiency.

6.3.6 Algorithm type

In section 5.5.5, we discuss SSGA's and CHC which are alternative methods for structuring the genetic algorithm search. Although they have demonstrated performance improvements over GGA's, even on noisy problems, they have not been widely applied in these problem domains as they lack the re-evaluation mechanism of GGA's. We wish to test if the alternative genetic algorithms offer improvements over the GGA for traffic signal optimization.

6.3.7 Optimal parameter tunings

Each genetic algorithm requires the specification of several parameters, most important being the population size and the parameters governing selection pressure. Parameter tuning of GGA's for traffic signal timing optimization have been performed to a limited extent [49, 121]. There are several studies into the determination of robust parameters for GGA's such as that of Grefenstette [89]. The alternative algorithms SSGA and CHC have not been as well studied. If they are found to be beneficial, we wish to determine the optimal algorithm parameters values for traffic signal optimization.

7 Chapter Seven – Methodology

7.1 Introduction

In this chapter we discuss the methodology employed to answer the research questions. We first discuss MSTRANS which is the evaluative tool used for the empirical work in this dissertation. We then describe the traffic networks against which the alternative optimization strategies are tested. Simulation run length and the choice of objective function to utilize in the optimization is then motivated. The coding scheme used for transforming traffic signal timing variables into genetic material amenable to manipulation by genetic operators is given next. Modifications to the real genetic operators to cope with this coding scheme are needed and these amendments are discussed. The implementation of the genetic algorithms is discussed and the methodology for comparing the efficiency of alternative search algorithms is given. Finally, the experimental design used for testing the genetic algorithm modifications is discussed.

7.2 MSTRANS

Evaluation of the effectiveness of signal timing policies was performed using a custom built stochastic traffic simulation called MSTRANS (Microscopic Stochastic Traffic Network Simulation Model). MSTRANS is written in the Delphi 7 programming language [138]. The compact disc accompanying this dissertation contains the entire code listing. Here we present a high level overview of the model. A thorough account of the functional details and a review of the literature justifying the model logic are given in Appendix A1.

MSTRANS simulates traffic operations on a road network under signalized control. The modelling methodology caters for undersaturated and oversaturated conditions. The model applies a fixed increment time step to advance the simulation. Vehicle status and kinematic properties are updated each second, along with the traffic signal indications. Vehicle entry headways, turning movements and certain driver behavioural decisions are generated stochastically using a pseudo random number generator.

The assumptions for a model run are specified via an Excel spreadsheet. The road network configuration is described by nodes and links (see section 3.2). Nodes at the boundary of the network denote entry/exit points from which vehicles are generated or removed from the network. Internal nodes represent traffic signals. Average hourly flow rates at the entry nodes must be specified as well as turning proportions for links with traffic signals at their end. A common free flow speed is applied to all links. Each link in the network is assigned the same number of lanes. Other assumptions regarding vehicle characteristics and driver behaviour must also be supplied.

At the start of a simulation run, the road network is empty. As the run progresses, new vehicles are introduced at the entry nodes. An initialization period is first completed before results are recorded. Once the required run length has been completed the run ends and performance statistics are output to a text file.

Modelling logic was based on findings from the literature on driver behaviour. In certain cases pragmatic rules were used instead. Traffic flow is assumed to be comprised entirely of passenger cars for which the vehicle and driver assumptions given in Appendix A1 are valid.
The logic encompasses items such as:

- Acceleration and deceleration characteristics of leading vehicles
- Behavioural regime of following vehicles based on microscopic car following models
- Lane changing
- Gap acceptance response model for left-turning vehicles receiving unprotected green
- Response delays
- Response to amber signals

A validation exercise was performed and the model has been found to give comparable results to CORSIM [50] (see section A1.4.3).

7.3 Test networks

The genetic algorithms were tested on a 9 signal arterial network and a 14 signal grid network. The network specifications are given in Appendix A3 and are based on real world data. These

networks were chosen as they are large networks and are representative of tough signal optimization problems. Improved optimization techniques are more easily identified on these difficult problems.

Undersaturated and oversaturated scenarios were considered. The flow rates given in Appendix A3 were used for the undersaturated scenarios. The flow rates for the arterial network and grid network were increased by 50% and 60% respectively for the oversaturated scenarios.

Thus, the genetic algorithms were tested in four scenarios:

- Arterial Undersaturated
- Arterial Oversaturated
- Grid Undersaturated
- Grid Oversaturated

In oversaturated scenarios, small alterations of the decision variables (i.e. the signal timing policy) can lead to substantial changes in network performance. Thus optimization is generally more difficult in oversaturated conditions as one must be relatively precise to move close to an optimal solution. The grid network problem is the more challenging network as we have a larger number of decision variables to optimize.

7.4 Simulation run length

A 3 minute initialization period was utilized followed by a 15 minute analysis period. For an undersaturated scenario, the initialization period serves as the transient phase of the simulation. We are interested in the steady state properties of the system and we assume that this has been attained approximately after the initialization period[42]. In oversaturated scenarios, there is typically a build-up of flow prior to the period of oversaturation. The initialization period allows for this build-up, as vehicles will be present in the system when the analysis period begins. A 15 minute analysis period was used based on the recommendations in the U.S. Highway Capacity Manual [(139)].

[42] A steady state may not exist if the system is pseudo-congested (see section 7.5).

7.5 Optimization objective

A delay minimization strategy was selected as it has been shown to achieve a satisfactory compromise in terms of minimizing delay and at the same time reducing stops, fuel consumption, emissions and increasing throughput in both undersaturated and oversaturated conditions [140, 141, 142].

We used control delay (see section 1.2.2) which is the difference between the actual and the uninterrupted travel time. The control delay for all vehicles completing their journey through the network after the initialization time is averaged to obtain a measure we will refer to as network delay.

A problem with this approach for computing delay is that the delay for vehicles still present on the network at the end of the simulation run will not be considered. This becomes a problem in oversaturated conditions where there may be many queued vehicles that have not completed their journey through the network and will not make any contribution to the network delay figure. This is also a problem in undersaturated conditions where the ability to provide the capacity to satisfy demand exists, but sufficient capacity is unavailable due to non-optimal signal timings. These are known as pseudo-congested conditions [124].

Park et al [61] recommend the use of queue delay which is the total time accumulated by vehicles waiting in queues [50] as a surrogate for delay as it includes the delay experienced by vehicles that have not exited the network. However this measure is still unsatisfactory as the delay still to be experienced by the residual queues is not taken into account.

To resolve this difference an automatic feature to extend the simulation period to allow sufficient time for all vehicles to clear the network was implemented in MSTRANS. No new vehicles are allowed to enter the network during this period[43]. The simulation is stopped once all vehicles have exited the network. We will refer to the delay computed by this extension of the simulation period as extended network delay.

[43] Aside from those waiting at the first-in-first-out queues at the source nodes (see section A1.3.7).

The genetic algorithms were applied to the minimization of mean extended network delay. This modified version of delay relieves certain shortcomings of the traditional delay measures as we've discussed at the cost of an increased run-time of the traffic simulation model.

7.6 Genetic algorithm problem encoding

7.6.1 Introduction

In this section, we provide description of the different phasing possibilities and the coding scheme used for transforming genetic material into traffic signal timing decision variables. This genetic encoding is similar to the fraction-based coding scheme given by Park [121]. The proposed encoding is an enhancement over those used thus far as it allows for the number and structure of the phases to be optimized.

7.6.2 Phasing

All signals in the test networks control conflicting flow from two perpendicular two-way streets. Since signals are identical in this respect, we need only consider a single intersection for the derivation of the phasing schemes. In Figure 11 below, we illustrate the traffic movements from the four approaches, *N*, *E*, *S* and *W* at a signalized intersection.

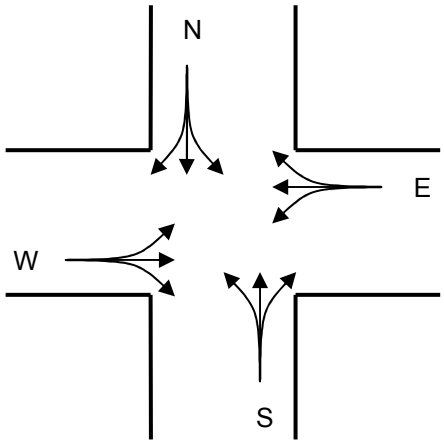


Figure 11: Traffic movements at a single intersection.

We assume that green indications are given to vehicles on the *N* and *S* approaches (N/S green phase), followed by green indications for vehicles on the *E* and *W* approaches (E/W green phase). This completes the cycle and the sequence of signal indications is repeated.

For the *W* approach, we can display green indications for all movements for the entire duration of the E/W green phase. Alternatively, we can have green indications for only part of the E/W green phase. If this is the case, then the remainder of the phase can be used to allow for protected left turn movements from the *E* approach.

Let ρ = duration of E/W green phase (s),

g_W^1 = duration of stage with green indications for all traffic movements on approach *W* (s) and

g_W^2 = duration of stage allowing for protected left turn movements for left turning vehicles on approach *E* (s).

We can visualize the staging possibilities diagrammatically in Figure 12 below:

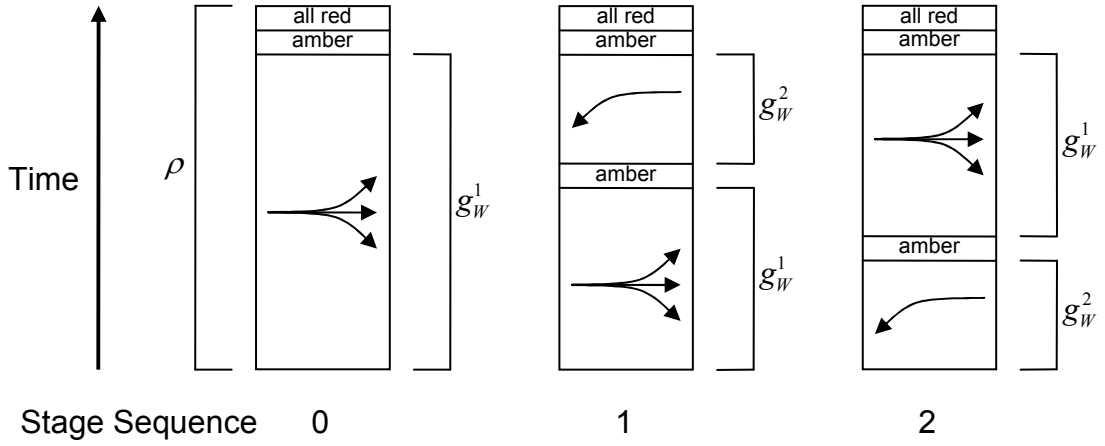


Figure 12: Staging of green for traffic movements on approach *W*.

Let δ_W = sequence of green stages for traffic movements on approach *W*.

The three possible stage sequences are:

- $\delta_W = 0$, i.e. no protected stage for opposing left turning movements
- $\delta_W = 1$, i.e. protected stage for opposing left turning movements at end of green stage (lagging stage).
- $\delta_E = 2$, i.e. protected stage for opposing left turning movements at beginning of green stage (leading stage).

Similarly, for the *E* approach, let

g_E^1 = duration of stage with green indications for all traffic movements on approach *E* (s),

g_E^2 = duration of stage allowing for protected left turn movements for left turning vehicles on approach *W* (s) and

δ_E = sequence of green stages for traffic movements on approach *E*.

We have similar staging possibilities for the *E* approach as illustrated in Figure 13.

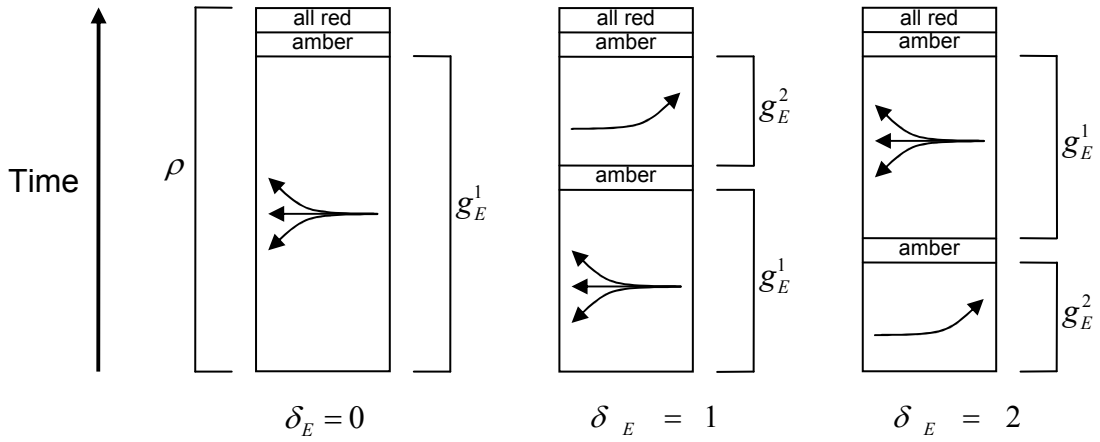
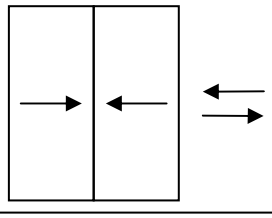
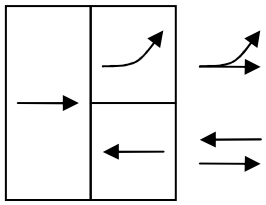
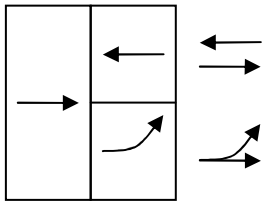
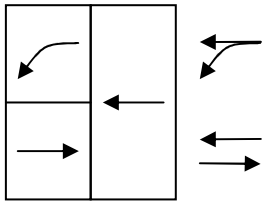
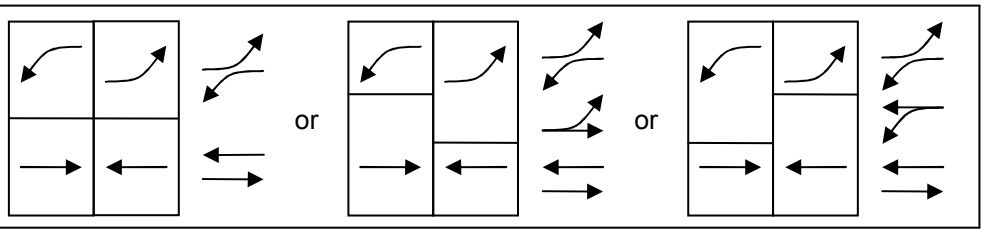
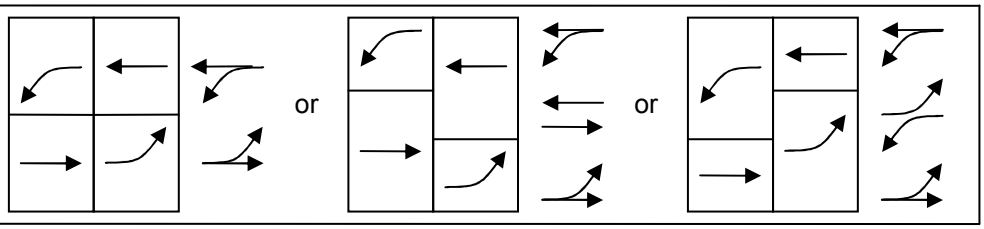
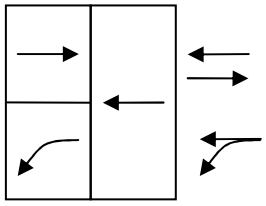


Figure 13: Staging of green sequences for traffic movements on approach *E*.

Taking the cross product of the stage sequences on each *E* and *W* approaches, we obtain 9 phasing possibilities for the E/W green phase. These are illustrated diagrammatically in Figure 14.

$\delta_W = 0$	$\delta_E = 0$	
$\delta_W = 0$	$\delta_E = 1$	
$\delta_W = 0$	$\delta_E = 2$	
$\delta_W = 1$	$\delta_E = 0$	
$\delta_W = 1$	$\delta_E = 1$	
$\delta_W = 1$	$\delta_E = 2$	
$\delta_W = 2$	$\delta_E = 0$	

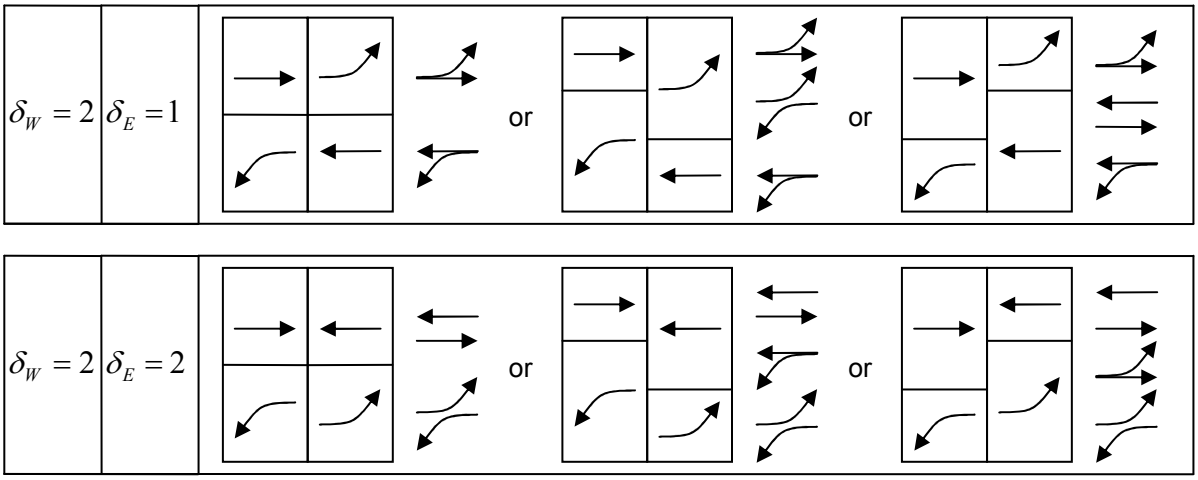


Figure 14: Nine phasing possibilities during E/W green phase[44].

In the figure above, we see that when two stages are applied to each approach and the lengths of the first stages is not equal for both approaches, then we have a middle phase called the overlap phase.

Similarly, the sequence of green indications applied in the N/S green phase can also be allotted in 9 possible ways. Taking the cross-product, we obtain 81 phasing possibilities at each traffic signal.

7.6.3 Genetic encoding

Let

N_s = number of traffic signals in the network under study,

i = particular traffic signal $\in \{1,2,\ldots,N_S\}$,

j = green phase $\in \{N/S,\ E/W\}$,

k = approach at signal $\in \{N,\ E,\ S,\ W\}$,

[44] The amber and all-red are ignored in the Figure. The straight arrows denote green indications for both through and turning movements (right turning movements and unprotected left turning movements). The left curving arrows denote protected turn movements for left turning vehicles

$G(k)$ = green phase corresponding to a particular approach

$$= \begin{Bmatrix} N/S \ \ for\ k \in \{N, S\} \\ E/W \ \ for\ k \in \{E, W\} \end{Bmatrix},$$

(7-1)

A = duration of amber interval (s),

R = duration of all-red interval (s),

$C_{\min}$ = minimum cycle length to search (s),

$C_{\max}$ = maximum cycle length to search (s),

$g_{\min,stage}$ = minimum duration of any green stage (s) and

$g_{\min,phase}$ = minimum duration of green phase including signal change period (s)

$= 2(g_{\min,stage} + A) + R$ (See footnote[45])

(7-2)

Now, for a particular individual in the genetic search, let

C = cycle length (s),

ϕ_i = offset of traffic signal i with respect to beginning of the N/S green phase relative to the start of the simulation (s),

$\rho_{i,j}$ = duration of green phase j at traffic signal i including amber and all-red transition period (s),

$\delta_{i,k}$ = sequence of green stages for approach k at traffic signal $i, (\delta_{i,k} \in \{0,1,2\})$,

$g_{i,k}^1$ = duration of stage with green indications for all traffic movements on approach k at traffic signal i (s) and

$g_{i,k}^2$ = duration of stage allowing for protected left turn movements on approach opposing approach k at traffic signal i (s).

[45] We have to allow for the possibility of two green stages during each green phase.

For a particular decision variable, let

x = value of decision variable in the problem space,

l = number of bits in the binary encoding for representing x and

$y = y_l y_{l-1} \ldots y_1$ = binary value mapping to x.

$D(x)$ = decoded value of x expressed as a fraction in the range $[0,1]$

$$= \frac{1}{2^l - 1} \sum_{m=0}^{l} y_m 2^m \qquad \text{(See equation (5-1)).}$$

(7-3)

We call $D(x)$ the normalized value of the decision variable x.

We use two bits for the encoding of each $\delta_{i,k}$. Let $y_{i,k,1}$ and $y_{i,k,2}$ denote the values of these two bits.

The binary genetic material of a particular individual is decoded into a set of traffic signal timing policies using the following formulae:

$$C = C_{\min} + \lfloor (C_{\max} - C_{\min}) D(C) \rfloor \qquad \text{(See equation (5-1))}$$

(7-4)

$$\phi_i = \begin{cases} 0 & for \quad i = 1 \\ \lfloor C D(\phi_i) \rfloor & for \quad i \in \{2,3,\ldots,N_S\} \end{cases} \qquad \text{(See footnote[46])}$$

(7-5)

$$\rho_{i,N/S} = g_{\min,phase} + \lfloor (C - 2g_{\min,phase}) D(\rho_{i,N/S}) \rfloor \qquad \text{(See footnote[47])}$$

(7-6)

$$\rho_{i,E/W} = C - \rho_{i,N/S} \qquad \text{(See footnote[48])}$$

(7-7)

[46] The offset of the first signal is arbitrarily set to 0. The offsets of all other signals are fractions of the cycle length and are relative to the start of the N/S green phase of the first signal.

[47] The N/S green phase time is a fraction of the cycle time, taking into account the minimum phase duration.

[48] The remainder of the cycle time is allotted to the E/W green phase

$$\delta_{i,k} = \begin{cases} 0 & for \quad \left(y_{i,k,1} = 0\right) \\ 1 & for \quad \left(y_{i,k,1} = 1,\ y_{i,k,2} = 0\right) \\ 2 & for \quad \left(y_{i,k,1} = 1,\ y_{i,k,2} = 1\right) \end{cases} \qquad \text{(See footnote[49])}$$

(7-8)

$$g_{i,k}^{1} = \begin{cases} \rho_{i,G(k)} - A - R & for \quad \delta_{i,k} = 0 \\ g_{\min,stage} + \left\lfloor \left(\rho_{i,G(k)} - g_{\min,phase}\right) D\left(g_{i,k}^{1}\right) \right\rfloor & for \quad \delta_{i,k} \neq 0 \end{cases} \qquad \text{(See footnote[50])}$$

(7-9)

$$g_{i,k}^{2} = \begin{cases} 0 & for \quad \delta_{i,k} = 0 \\ \rho_{i,G(k)} - g_{i,k}^{1} - 2A - R & for \quad \delta_{i,k} \neq 0 \end{cases} \qquad \text{(See Figure 12)}$$

(7-10)

Using this problem encoding, we have 90 decision variables and a binary string of 315 bits for the arterial problems (see the formulae in Appendix A4). For the grid network problems, the corresponding values are 140 and 490.

Cycle lengths at integer values between 50 and 120 seconds were searched (i.e. $C_{\min} = 50$ and $C_{\max} = 120$). A minimum green stage time of 5 seconds was implemented (i.e. $g_{\min,stage} = 5$). An amber period of 3 seconds and all-red interval of 2 seconds was assumed for all phase transitions (i.e. $A = 3$ and $R = 2$)

7.7 Modifications to real genetic operators

7.7.1 Introduction

Modifications to the blend crossover operator and real mutation operator given in section 5.5.6 were required in order for them to be utilized with the proposed encoding scheme for traffic signal timing policies.

[49] One bit denotes whether one or two stages are used for the green indications for the approach. If there are two stages, then the other bit denotes the order of these stages. This encoding for signal staging is in-line with the principle of meaningful building blocks [(78)].

[50] The duration of the green stage for all traffic movements on the approach is a fraction of the green phase time subject to the minimum green stage duration.

7.7.2 Blend crossover

The blend crossover operator discussed in 5.5.6 can be applied to decision variables with unlimited search domain or where the search domain is restricted to an interval of the form $[a,b]$. For the fraction-based encoding scheme proposed in section 7.6.3, the search domains for the decision variables $\{\phi_i\}$, $\{\rho_{i,N/S}\}$ and $\{g^1_{i,k}\}$ may differ for each individual e.g. the maximum allowable values of the $\{\phi_i\}$ is *C* and the value of *C* may differ in two individuals selected for recombination. When the allowable search domain differs for each individual, then blend crossover will not allow for a meaningful blending of the decision variables. In certain situations, it may be impossible to generate and a valid offspring e.g. consider Figure 10 in section 5.5.6 and suppose that the upper limit on the search domain for the *i*'th decision variable in the offspring is less than l_i.

To resolve this problem, we first converted the real valued decision variable to a fractional value in the range $[0,1]$ in each parent. We can then apply blend crossover to these fractional values and obtain the fractional value of the decision variable in the offspring. We then convert the fractional value of the decision variable in the offspring to the corresponding value in the search domain. That is, the blend crossover is applied to the normalized values, $\{D(x_i)\}$ (see section 7.6.3).

We cannot apply blend crossover to $\{\delta_{i,k}\}$, the decision variables specifying the number, structure and sequence of stages. We applied the standard binary crossover operators to these decision variables. Thus when we state that real crossover is applied in a genetic algorithm, we mean that blend crossover is applied to all decision variables besides the variables specific to phasing.

7.7.3 Real mutation

For real mutation, the mutation operator discussed in 5.5.6 is applied to the each decision variable *C*, $\{\phi_i\}$, $\{\rho_{i,N/S}\}$ and $\{g^1_{i,k}\}$ with probability p_m. Since the allowable domain of certain decision variables depend on the value of other decision variables, we must be careful about the order in which the decision variables are mutated. We removed this complication by applying

mutation to the normalized values of the decision variables, $\{D(x_i)\}$ i.e. for each real decision variable x_i, replace $D(x_i)$ with a realization from a $U(0,1)$ random-variate with probability p_m. This same procedure can be used for initializing the initial population, where we set $D(x_i) =$ realization from a $U(0,1)$ random-variate for each real decision variable x_i.

The bits coding the phasing decision variables undergo standard bit-flip mutation with p_m. Thus the real mutation applied is a combination of real mutation with the standard binary mutation.

When both real crossover and real mutation are applied, we no longer require a binary encoding for the real decision variables.

7.8 Genetic algorithms and parameter values

Three different genetic algorithms were applied: GGA, SSGA and CHC.

The GGA is a SGA with binary tournament selection, elitist method and uniform crossover. These modifications were used in the GGA by Park [121] and are generally acknowledged methods for improving search efficiency (see sections 5.5.2, 5.5.3 and 5.5.4). The SSGA is based on GENITOR [91, 97] and uses uniform crossover. The CHC algorithm was implemented according to the pseudo code given by Eshelman [100] and uses HUX crossover. The algorithms were implemented using Delphi 7 [138] and the complete code listing is given on the compact disc provided with this dissertation. The stochastic elements of the genetic algorithm search were simulated using the ISAAC pseudo-random number generator [143] (see section A1.3.13).

The performance of a genetic algorithm is known to be sensitive to the algorithm parameters. In order for us to meaningfully test modifications to the algorithm and for useful comparisons between the algorithms to be made, we require the genetic algorithms to be executed with parameter values appropriate for the particular problem. Parameter values for each algorithm were carefully chosen using one or more of the following methods:

- Performing simple test runs to find appropriate parameter values.

- Selecting parameter values based on studies into robust genetic algorithm parameter values, such as the study by Grefenstette [89].
- Recommendations by Park [121] and the TRANSYT-7F user manual [49] on optimal parameter values specifically for traffic signal optimization problems.

The following parameter values were used, unless specified otherwise:

GGA

$N_{Pop} = 30$

$p_c = 0.6$

$p_m = 0.01$

SSGA

$N_{Pop} = 50$

$\beta = 1.4$

$p_m = 0.01$

CHC

$N_{Pop} = 50$

$r = 0.35$

7.9 Comparisons of genetic algorithms

7.9.1 Introduction

We discuss the methodology for comparing the different genetic algorithms under the following headings:

- Multiple runs
- Number of function evaluations
- Obtaining an unbiased measure of solution quality
- Constructing sample statistics
- Statistical tests

7.9.2 Multiple runs

Genetic algorithms are stochastic search algorithms. A comparison of the optimization performance of two algorithms on the same function using a single run of each would thus be inappropriate. The studies into genetic algorithms optimization of traffic signals discussed in sections 5.6 and 6.2.1 have based findings on a single optimization run owing to computational resource limitations. Due to the run-time efficiency of the MSTRANS model and increases in modern computing power, it was possible for us to perform a total of 20 independent executions of each search algorithm and compare mean performance. Averaging the results of several optimization results produces more stable results and allows for statistical comparisons to be made.

When CRN's and a single replication were utilized in fitness evaluations, the same set of common random number seeds was applied to all individuals in a particular run of a particular genetic algorithm. When CRN's and more than one replication were performed, different sets of random number seeds were used in each replication of the objective function, but for each replication, the same set of random number seeds was applied to all individuals. Between runs of the genetic algorithm, totally different random number seeds were used for implementing the CRN's variance reduction to ensure that the runs were independent. Runs of different genetic algorithms were also independent.

7.9.3 Number of function evaluations

Trial runs were performed with the GGA on the test networks and the convergence after 10000 function evaluations was thought to be adequate given the computational resource limitations[51]. Thus, it was decided to execute each search algorithm for a total of 10000 objective function evaluations. Results were output at intervals of 1000 function evaluations so that comparisons at different computational resource levels could be made.

[51] This number of function evaluations is much larger than that used in the study by Rouphail et al [123] (see section 6.2.1) due to the larger complexity of the test networks and the consideration of additional signal timing variables in the optimization.

7.9.4 Obtaining an unbiased measure of solution quality

When independent replications are performed, the extended network delay value produced for the best performing individual is biased, since selection and replacement in the genetic algorithm favour solutions with "lucky evaluations" (see section 5.5.7). When CRN's are utilized, the extended network delay of the best individual is computed using the arrival patterns, routing schemes and driver decisions induced by the particular random seeds. The decision variables of the best individual will be tuned to work well for this particular realization and the sample measure of extended network delay will underestimate mean extended network delay.

Thus, to obtain an unbiased measure, independent replications of the signal timing policy of the best individual are performed to obtain a true measure of the quality of the solution. MSTRANS was executed 20 and 40 times to obtain independent samples of extended network delay of the best solution for the undersaturated and oversaturated scenarios respectively[52]. These replications were not included when computing the total number of function evaluations as they were performed for evaluative purposes.

7.9.5 Construction of sample statistics

For a particular genetic algorithm, X, applied to a particular scenario, let

$\mu_{X,F}$ = population mean of extended network delay of best individual produced by the genetic algorithm on average after F function evaluations,

N = number of independent runs of the genetic algorithm
= 20,

n = number of independent replications of the best individual
$= \begin{cases} 20 \ \textit{for the undersaturated scenarios} \\ 40 \ \textit{for the oversaturated scenarios} \end{cases}$,

$X_{i,F,j}$ =extended network delay on the j'th independent replication of the best individual after F function evaluations on the i'th independent run of the genetic algorithm, where

$i \in \{1,2,\ldots,N\}$, $F \in \{0,1000,2000,\ldots,10000\}$ and $j \in \{1,2,\ldots,n\}$

[52] In oversaturated conditions, extended network delay will have a larger variance than in undersaturated conditions, so it was decided to perform more replications in the oversaturated scenarios.

Then $X_{i,F} = \frac{1}{n}\sum_{j=1}^{n} X_{i,F,j}$ is an unbiased estimate of the mean extended network delay of the best solution after F function evaluations on the i'th run of the genetic algorithm.

Finally, $X_F = \frac{1}{N}\sum_{i=1}^{N} X_{i,F}$ gives an unbiased estimate of $\mu_{X,F}$.

Genetic algorithms X and Y are compared at F function evaluations using the samples values of X_F and Y_F. To make the comparisons more meaningful, we quote the relative improvement/degradation of genetic algorithm Y over genetic algorithm X after F function evaluations, $\frac{X_F - Y_F}{X_F}$. Positive values indicate an improvement and negative values signify a reduction in performance.

7.9.6 Statistical tests

In experiments with genetic algorithms on noisy functions, there are two sources of variability in the sample performance measures:

- Genetic algorithms are stochastic search algorithms and exhibit different search trajectories on each independent execution of the search process.
- The objective function is noisy and a different value is obtained for each independent replication of the same individual.

When performing comparisons of genetic algorithms, we wish to test whether the mean objective function value of the best individual of alternative genetic algorithms differ on average over all possible search paths. That is, for two alternative algorithms genetic algorithms X and Y we wish to compare $\mu_{X,F}$ and $\mu_{Y,F}$.

When performing empirical comparisons of genetic algorithms on noisy functions, researchers construct a noisy test function by adding a normally distributed noise term to a deterministic test function. When comparisons of the performance of different genetic algorithms are made, the

noise in the objective function evaluations can be explicitly removed by evaluating the objective function without the noise term. This removes the second source of variability and simplifies the analysis. Repeated independent runs of the genetic algorithms are performed and conventional two sample tests can be used to test whether differences in the sample performance measures are statistically significant.

For the consideration of a problem such as the traffic signal timing problem where objective function evaluations are via time consuming computer simulation, such a luxury is not available. Thus, for the purpose of this research, a statistical model addressing both sources of variability was developed. Experiments were performed so that appropriate statistical distributions that describe the variability could be determined. A translated gamma distribution was found to describe the variation in the quality of the best solution over repeated executions of the same search algorithm. For a specified signal timing policy, extended network delay was found to be normally distributed with variance a linear function of the mean where the linear function has a positive slope. Thus, signal timing policies with a larger mean extended network delay have a larger variance in extended network delay. This auxiliary finding is useful as it demonstrates that delay minimization strategies produce the most reliable timing policies in that the variability of delay is simultaneously minimized. The likelihood ratio test [144] was then used to construct a statistical test. A detailed account of the experiments performed and the formulation of this statistical model is given Appendix A6

A test of the null hypothesis, $H_0 : \mu_{X,F} = \mu_{Y,F}$ versus the alternative hypothesis, $H_1 : \mu_{X,F} \neq \mu_{Y,F}$ is performed separately at $F = 5000$ and $F = 10000$ function evaluations. We note that the tests are not independent since they are based on sample data from the same set of N independent replications of each genetic algorithm. Unless specified otherwise, the statistical comparisons were made at the 5% significance level. We compare multiple algorithms on 4 traffic network scenarios and utilize the same output in multiple comparisons in some cases. Thus the overall significance level will be greater than 5% and the hypothesis tests are not always independent.

7.10 Experimental design

In this research, we are hoping to more obtain improved optimization of traffic signals by varying the following parameters (which we call factors in this section) of the genetic algorithm search procedure:

- Re-evaluation of fitness in GGA's
- The use of CRN's
- Number of replications for fitness evaluation
- Standard binary code versus Gray code
- Binary operators versus real operators
- Algorithm type (GGA/SSGA/CHC)
- Other search parameters (e.g. Population size)

The modifications are evaluated using a sequential procedure. An alteration that is found to improve optimization is retained when evaluating the effectiveness of other factors. Pairwise comparisons are made against the best algorithm found at each stage. This can lead to misleading results if there is strong interaction between the factors. A factorial design [(145)] where factors are simultaneously varied would be more appropriate but is beyond the scope of this research. However in many cases we did vary more than one factor at a time (e.g. Algorithm type is included as a factor in most comparisons). Also the base value of the factors we call "Other search parameters" were carefully set (see section 7.8). When comparing binary and real operators or coding, we will show preference to the real operators/codings even when there is no evidence of any difference in optimization performance as the real number versions provide other benefits (see section 5.5.6).

8 Chapter Eight – Results

8.1 Introduction

In this chapter, we present the results of the empirical experiments with alternative genetic algorithms, genetic operators and search parameter values. The experiments conducted to answer each of the research questions in section 6.3 are presented separately along with the results. In order to reduce on the amount of output displayed, we present results at only $F = 5000$ and $F = 10000$ function evaluations in most situations. More detailed output is given in Appendix A7.

8.2 Re-evaluation of fitness in GGA's

The effectiveness of the re-evaluation mechanism in GGA's was assessed by applying the GGA with and without the re-evaluation mechanism. Experiments were performed using 1, 2, 4 and 8 independent replications of each individual for fitness evaluations to evaluate whether the re-evaluation mechanism still provides benefits when additional replications are performed and the noise level is reduced. Comparisons were made by contrasting the GGA with and without re-evaluation separately for the same number of independent replications for fitness evaluations. The results given in Table 2 are for the relative improvement/degradation in extended network delay of the runs with the re-evaluation mechanism relative to those without the re-evaluation mechanism.

Function Evaluations	Relative Improvement/Degradation				Significant Difference Test - p value			
	Number of Replications				Number of Replications			
	1	2	4	8	1	2	4	8
Arterial Undersaturated								
5000	-3.2%	4.2%	-2.1%	-9.8%	0.46009	0.23299	0.64621	0.19096
10000	0.3%	3.9%	-3.8%	-4.5%	0.73374	0.09763	0.22934	0.31708
Grid Undersaturated								
5000	3.0%	0.8%	-5.4%	-3.7%	0.20407	0.81553	0.09149	0.57261
10000	1.6%	1.5%	-2.5%	-4.1%	0.48429	0.64488	0.23426	0.34885
Arterial Oversaturated								
5000	-1.2%	1.9%	4.8%	-7.6%	0.75613	0.47125	0.17272	0.10112
10000	5.1%	-0.3%	3.9%	-7.8%	0.07868	0.88855	0.21889	0.04469
Grid Oversaturated								
5000	3.4%	3.2%	-2.2%	-6.7%	0.29408	0.22788	0.59575	0.30324
10000	6.9%	2.6%	1.9%	-6.8%	0.04234	0.37544	0.51398	0.20545

Table 2: Assessment of re-evaluation mechanism in GGA

We find that the re-evaluation mechanism offers no significant improvement in the undersaturated scenarios. In the oversaturated grid network scenario, an observed 6.9% reduction in delay at $F = 10000$ is just significant when a single replication is performed. For this same value of F, a 5.1% reduction in delay for the oversaturated arterial scenario with a single replication is close to significant. We find performance degradations with the re-evaluation mechanisms when fitness evaluations are performed using 8 replications. However the degradation is only just significant in the oversaturated arterial scenario and insignificant in the other scenarios.

Thus, considering the larger number of tests performed and borderline significant results, we do not find convincing evidence in favour of the re-evaluation mechanism based on the experiments performed. Further investigation is required to test whether re-evaluations are:

- Beneficial in oversaturated networks when performing a single replication for fitness evaluation
- Disadvantageous when performing multiple replications for fitness evaluation.

We do not pursue these points further and perform the comparisons in the next section using GGA's without re-evaluation.

8.3 Common Random Numbers

The efficacy of CRN's in improving optimization performance was evaluated by comparing the results of genetic algorithms runs with independent replications versus those with CRN's for fitness evaluation. The GGA, SSGA and CHC were executed using 1, 2, 4 and 8 replications of each individual for fitness evaluations. The results from 8.2 were utilized for the GGA with independent replications[53]. In each case the same type of algorithm with the same number of replications was compared with and without the use of CRN's. The results for the GGA, SSGA and CHC are presented in Table 3, Table 4 and Table 5 respectively. The results are quoted in terms of improvement/degradation of CRN's relative to independent replications.

Function Evaluations	% Improvement/Degradation				Significant Difference Test - p value			
	Number of Replications				Number of Replications			
	1	2	4	8	1	2	4	8
Arterial Undersaturated								
5000	11.2%	10.9%	0.6%	-6.9%	0.00003	0.00045	0.89404	0.28625
10000	6.1%	10.2%	0.6%	1.0%	0.01177	0.00250	0.92373	0.80931
Grid Undersaturated								
5000	9.0%	4.4%	0.0%	7.1%	0.00204	0.17541	0.98588	0.25980
10000	9.0%	9.0%	1.9%	4.3%	0.00082	0.00038	0.31477	0.22666
Arterial Oversaturated								
5000	10.1%	4.2%	6.2%	0.2%	0.00490	0.13526	0.03824	0.95131
10000	8.8%	4.9%	8.1%	-2.7%	0.01443	0.05265	0.00139	0.39180
Grid Oversaturated								
5000	11.7%	10.7%	0.2%	-1.9%	0.00012	0.00141	0.96338	0.77222
10000	15.0%	12.4%	6.8%	2.2%	0.00031	0.00062	0.02991	0.67077

Table 3: Effectiveness of CRN's in GGA

[53] The results for the GGA without re-evaluation were taken in all cases.

Function Evaluations	Relative Improvement/Degradation				Significant Difference Test - p value			
	Number of Replications				Number of Replications			
	1	2	4	8	1	2	4	8
Arterial Undersaturated								
5000	5.8%	3.6%	5.6%	2.3%	0.06125	0.10044	0.12769	0.67232
10000	1.8%	3.3%	8.5%	6.1%	0.66837	0.12404	0.00029	0.13376
Grid Undersaturated								
5000	7.5%	3.6%	1.3%	0.6%	0.00323	0.05849	0.68549	0.90140
10000	4.6%	3.2%	3.7%	3.5%	0.05431	0.05713	0.04194	0.08403
Arterial Oversaturated								
5000	5.6%	4.2%	3.7%	2.9%	0.04584	0.11465	0.13780	0.42154
10000	2.1%	4.4%	2.7%	0.8%	0.43804	0.04186	0.24092	0.69978
Grid Oversaturated								
5000	10.4%	8.8%	2.0%	1.7%	0.00154	0.00446	0.57604	0.73357
10000	7.9%	9.7%	6.3%	-1.3%	0.00996	0.00330	0.01456	0.74770

Table 4: Effectiveness of CRN's in SGGA

Function Evaluations	Relative Improvement/Degradation				Significant Difference Test - p value			
	Number of Replications				Number of Replications			
	1	2	4	8	1	2	4	8
Arterial Undersaturated								
5000	7.1%	10.3%	7.1%	-3.7%	0.01251	0.00031	0.04211	0.42026
10000	3.8%	6.0%	5.7%	1.3%	0.13381	0.01274	0.00006	0.62771
Grid Undersaturated								
5000	6.0%	3.9%	-3.1%	0.9%	0.00043	0.02179	0.32756	0.80072
10000	2.1%	6.2%	2.2%	7.0%	0.16559	0.00067	0.17252	0.01038
Arterial Oversaturated								
5000	8.8%	6.2%	3.8%	-6.6%	0.00973	0.00470	0.04895	0.12802
10000	6.5%	5.2%	4.6%	0.9%	0.02189	0.03613	0.02786	0.74747
Grid Oversaturated								
5000	10.0%	4.6%	-1.4%	7.1%	0.00188	0.10995	0.61680	0.18565
10000	7.1%	8.9%	-0.8%	3.0%	0.01774	0.00128	0.73583	0.36467

Table 5: Effectiveness of CRN's in CHC

For the case of a single replication for fitness evaluations, the genetic algorithms with CRN's exhibit improved sample performance measures. These improvements are statistically significant in most cases. The improvements at 10000 function evaluations are generally smaller than those at 5000 function evaluations. Thus we see that although the genetic algorithms with CRN's initially outperform the genetic algorithms with independent replications, the genetic algorithms

with independent replications do appear to "catch-up" as additional function evaluations are allowed. For the case of 2 replications for fitness evaluations, the genetic algorithm with CRN's also, in most cases, demonstrate improved performance. For 4 and 8 replications per fitness evaluation, we see degradations in performance in some cases although none of these observed reductions in performance are significant. Thus, for fitness evaluation using 1 or 2 replications, the use of CRN's for fitness evaluations benefits all algorithm types. There is less evidence of improvement when performing more than 2 replications for fitness evaluation.

8.4 Number of replications

The optimization runs discussed in section 8.3 allow us to evaluate the benefit of performing additional replications. Fitness evaluation using more than one replication reduces the noise level of the objective function at the expense of decreasing the extent of the genetic algorithm search[54]. We evaluated the relative improvement/degradation in performance of 2, 4 and 8 replications for fitness evaluations over that of performing a single replication. We considered the results where CRN's were used in the fitness evaluations[55]. Results are presented separately for each algorithm type in Table 6, Table 7 and Table 8.

[54] The computational resource level which is the total number of function evaluations remains fixed.
[55] We did not compare the impact of additional replication on performance when independent replications are performed since CRN's were shown to do no worse than independent evaluations in 8.3.

Function Evaluations	Relative Improvement/Degradation			Significant Difference Test p value		
	Number of Replications			Number of Replications		
	2	4	8	2	4	8
Arterial Undersaturated						
5000	-3.9%	-24.9%	-65.5%	0.29211	0.00001	0.00000
10000	2.7%	-8.7%	-20.8%	0.17593	0.01496	0.00001
Grid Undersaturated						
5000	-9.5%	-31.2%	-71.7%	0.00364	0.00000	0.00000
10000	-2.2%	-13.6%	-36.1%	0.23407	0.00006	0.00000
Arterial Oversaturated						
5000	-7.4%	-21.9%	-50.1%	0.02470	0.00003	0.00000
10000	-0.5%	-7.9%	-22.0%	0.90199	0.02291	0.00004
Grid Oversaturated						
5000	-5.5%	-30.9%	-92.6%	0.03586	0.00000	0.00000
10000	-1.9%	-9.8%	-38.1%	0.51744	0.00670	0.00000

Table 6: Impact of additional replications on GGA (Using CRN's)

Function Evaluations	Relative Improvement/Degradation			Significant Difference Test p value		
	Number of Replications			Number of Replications		
	2	4	8	2	4	8
Arterial Undersaturated						
5000	-3.9%	-17.1%	-68.7%	0.19571	0.00056	0.00000
10000	1.0%	1.6%	-18.4%	0.63858	0.39160	0.00167
Grid Undersaturated						
5000	-10.0%	-33.9%	-77.8%	0.00107	0.00000	0.00000
10000	-1.7%	-10.9%	-32.6%	0.40202	0.00109	0.00000
Arterial Oversaturated						
5000	-2.1%	-14.9%	-42.5%	0.49219	0.00040	0.00000
10000	3.4%	-1.0%	-11.5%	0.17589	0.73972	0.00197
Grid Oversaturated						
5000	-6.7%	-35.2%	-104.2%	0.01978	0.00000	0.00000
10000	1.6%	-9.1%	-35.9%	0.51112	0.00819	0.00000

Table 7: Impact of additional replications on SSGA

Function Evaluations	Relative Improvement/Degradation			Significant Difference Test p value		
	Number of Replications			Number of Replications		
	2	4	8	2	4	8
Arterial Undersaturated						
5000	-5.4%	-41.8%	-137.6%	0.12770	0.00000	0.00000
10000	2.8%	-6.6%	-46.9%	0.11462	0.01400	0.00000
Grid Undersaturated						
5000	-18.9%	-62.3%	-146.6%	0.00000	0.00000	0.00000
10000	-1.4%	-23.8%	-56.7%	0.38041	0.00000	0.00000
Arterial Oversaturated						
5000	-9.9%	-33.2%	-110.1%	0.00764	0.00001	0.00000
10000	-1.3%	-12.6%	-47.8%	0.63697	0.00106	0.00000
Grid Oversaturated						
5000	-15.7%	-67.8%	-152.1%	0.00025	0.00000	0.00000
10000	-0.8%	-25.4%	-68.0%	0.78911	0.00003	0.00000

Table 8: Impact of additional replications on CHC

When performing 2 replications we note statistically insignificant differences in performance for $F = 10000$. For 2 replications, degradations in performance are observed at $F = 5000$ (i.e. when there is a small amount of computing resources available) and the degradations are significant in most cases. The performance declines are larger for 4 replications and largest for 8 replications. In most cases, the degradations, especially with 8 replications, are statistically highly significant. Thus a single replication with CRN's for fitness evaluations is optimal, even in oversaturated conditions. In all executions of the genetic algorithms that follow, we make use of a single replication with CRN's.

8.5 Alternative problem encodings

8.5.1 Gray coding

To evaluate the benefit of the use of Gray code over the standard binary code, the performance of GGA, SSGA and CHC were compared using the different codings. The results from section 8.3 for the case of a single replication with CRN's for fitness evaluations were utilized for the standard binary code. The relative improvements/degradations of Gray code over the standard binary code are given in Table 9.

Function Evaluations	Relative Improvement/Degradation			Significant Difference Test p value		
	Algorithm Type			Algorithm Type		
	GGA	SSGA	CHC	GGA	SSGA	CHC
Arterial Undersaturated						
5000	1.4%	-0.2%	-1.2%	0.55610	0.83361	0.89737
10000	2.1%	0.7%	-0.8%	0.46306	0.81333	0.91851
Grid Undersaturated						
5000	6.2%	-0.8%	2.5%	0.00257	0.73437	0.16527
10000	4.9%	1.1%	-1.0%	0.01730	0.48421	0.58197
Arterial Oversaturated						
5000	2.5%	0.0%	-0.8%	0.32089	0.96491	0.76926
10000	3.5%	0.2%	-2.4%	0.22737	0.91831	0.32477
Grid Oversaturated						
5000	10.5%	6.9%	6.1%	0.00134	0.00911	0.02602
10000	9.9%	8.2%	2.6%	0.00057	0.00054	0.27011

Table 9: Impact of Gray code on genetic algorithms

Gray coding significantly improves optimization performance of all algorithms on the difficult oversaturated grid network problem (except at 10000 function evaluations using CHC where the performance improvement is statistically insignificant). The enhancement of the GGA on the undersaturated grid network using Gray code is also statistically significant. Generally, the sample performance measures exhibited the largest improvements for the GGA. Small, but statistically insignificant degradations in performance are observed in certain instances when applying Gray coding on the SSGA and CHC. Thus, Gray coding does appear to improve optimization performance in certain problems while offering equivalent performance on others.

8.5.2 Real Crossover

The effectiveness of the blend crossover operator over the standard binary crossover operator was evaluated by comparing the optimization performance of the genetic algorithms with each recombination operator. Even though real crossover is applied, mutation is via the standard binary mutation operator. The mutation operator is retained so that any observed performance differentials can be attributed to the difference in crossover operator. Mutation was performed on the Gray coded variables. The bases for comparison were the results from section 8.5.1 where the

standard binary crossover operators were applied to the Gray coded variables[56]. The results for the three algorithm types are given in Table 10.

Function Evaluations	Relative Improvement/Degradation			Significant Difference Test – p value		
	Algorithm Type			Algorithm Type		
	GGA	SSGA	CHC	GGA	SSGA	CHC
Arterial Undersaturated						
5000	1.1%	2.2%	5.5%	0.57051	0.25777	0.02003
10000	1.5%	2.4%	4.0%	0.48732	0.16396	0.08470
Grid Undersaturated						
5000	-4.7%	3.6%	3.7%	0.01131	0.01402	0.01035
10000	-2.6%	4.2%	5.0%	0.16537	0.00150	0.00006
Arterial Oversaturated						
5000	-0.2%	3.0%	3.7%	0.94080	0.15410	0.07349
10000	-2.0%	3.2%	3.9%	0.40657	0.17093	0.03179
Grid Oversaturated						
5000	4.8%	8.3%	10.7%	0.01215	0.00004	0.00000
10000	3.7%	9.3%	12.5%	0.02745	0.00015	0.00000

Table 10: Impact of real crossover on genetic algorithm performance

Highly significant improvements using real crossover are obtained for the oversaturated grid network. CHC benefits the most from real crossover. For this algorithm, significant improvements in performance in all instances except two are observed. The performance of the SSGA is enhanced in all cases with statistically significant improvements in performance on the grid network problems. The impact of real crossover on the GGA is mixed. Statistically significant improvements are noted for the oversaturated grid scenario and a statistically significant decline is observed at $F = 5000$ on the undersaturated grid scenario. On the whole, blend crossover does appear to improve the optimization.

8.5.3 Real mutation

We also tested the effectiveness of real mutation as opposed to binary mutation. The results from section 8.5.2 with real crossover and binary mutation were compared with those from optimization runs with both real crossover and real mutation. The exception was the GGA on the

[56] The Gray coding is used as the basis of comparison as it has been shown to be superior to the standard binary coding in section 8.5.1.

undersaturated grid scenario where we used the GGA with Gray coding and the standard binary operators from section 8.5.1 as the base for comparison[57]. We do not require Gray coding for the runs with both real crossover and real mutation as they do not require a binary encoding (except for the decision variables determining phasing – see section 7.6). We did not evaluate real mutation in the CHC genetic algorithm. Mutation in CHC occurs in the outer loop of the algorithm when a restart of the algorithm is performed via cataclysmic mutation. From the executions of the CHC algorithm in section 8.5.2, the restart mechanism was never invoked in any of the $N = 20$ replications of the algorithm on the grid scenarios. For the arterial scenarios, the restart mechanism was initiated once in 20 replications of each scenario at the very end of the run. Thus, with a limit of a total of 10000 function evaluations, we cannot measure the impact of real mutation on CHC. The results for the GGA and SSGA are given in Table 11.

Function Evaluations	Relative Improvement/Degradation		Significant Difference Test - p value	
	Algorithm Type		Algorithm Type	
	GGA	SSGA	GGA	SSGA
Arterial Undersaturated				
5000	0.5%	-0.5%	0.90831	0.96191
10000	0.8%	-2.4%	0.71792	0.21470
Grid Undersaturated				
5000	0.2%	0.6%	0.90921	0.43949
10000	0.5%	-1.7%	0.77031	0.15983
Arterial Oversaturated				
5000	-1.4%	0.9%	0.55425	0.65327
10000	1.7%	1.3%	0.45783	0.53365
Grid Oversaturated				
5000	0.0%	-0.6%	0.96712	0.62173
10000	0.6%	-2.3%	0.69562	0.20144

Table 11: Evaluation of real mutation on performance of GGA and SSGA

We find no significant differences in performance. The binary mutation rate was chosen based on prior research and calibration runs (see section 7.8) whereas the real mutation rate was set equal to the binary mutation rate. Thus we can conclude that real mutation does no worse than binary mutation and has potential for improving performance. Combining real crossover and real

[57] This GGA has demonstrated the best performance on this problem so far relative to all GGA alternatives.

mutation, we now find equivalent performance for the GGA on the undersaturated grid scenario to that with binary operators and Gray codes. Based on the findings in this section and section 8.5.2, we note improved performance with removing the binary encoding and using a real parameter encoding for all variables except those determining phasing decisions (see Appendix A4). An additional benefit is that the optimization performance is no longer dependent on the number of bits used for representing the decision variables (see section 5.5.6).

8.5.4 Real CHC

With real encodings we no longer require a binary representation of the decision variables for the GGA and SSGA. However, with CHC, the incest prevention mechanisms permits recombination based on the Hamming distance between parent strings. Thus the binary representation is still required. With the success observed with real crossover and mutation, a modified version of CHC which we call Real CHC was evaluated. The distance metric for incest prevention in Real CHC is in terms of the distance between parameters in the real parameters space (In fact the distance measure requires the combination of real and binary distances as binary variables are still needed for coding signal phasing). Details on the distance metric and the modifications to the formulae for computing the mating thresholds are given in Appendix A5. The proposed benefits of Real CHC are:

- Binary representation no longer required for real parameter decision variables
- Improved search efficiency as incest prevention algorithm operates in the real parameter space

A Real CHC algorithm with real crossover and mutation was compared to the Standard CHC algorithm in section 8.5.2 with real crossover and binary mutation[58]. The results are presented in Table 12.

[58] The differences in mutation operators are unlikely to affect the results as mutation was rarely applied in the standard CHC as discussed in section 8.5.3.

Function Evaluations	Relative Improvement/Degradation	Significant Difference Test - p value
Arterial Undersaturated		
5000	-0.2%	0.97321
10000	1.3%	0.49886
Grid Undersaturated		
5000	0.2%	0.96881
10000	0.5%	0.74755
Arterial Oversaturated		
5000	0.4%	0.85413
10000	2.5%	0.17442
Grid Oversaturated		
5000	0.4%	0.61517
10000	1.0%	0.38208

Table 12: Evaluation of Real CHC

Statistically insignificant improvements in performance are observed in most cases. From these results, we conclude that the Real CHC allows for the corresponding binary representation of real decision variables to be discarded without any measurable reductions in performance.

8.6 Algorithm Type

A comparison of the three alternative methods of structuring genetic algorithm search was performed. The performance of GGA and SSGA were evaluated relative to the Real CHC. In all cases real crossover and real mutation were applied as they have been shown to improve all three algorithms in section 8.5. The results are given in Table 13.

Function Evaluations	Improvement/Degradation Relative to Real CHC		Significant Difference Test - p value	
	Algorithm Type		Algorithm Type	
	GGA	SSGA	GGA	SSGA
Arterial Undersaturated				
5000	-8.0%	-2.5%	0.00018	0.07838
10000	-6.9%	-4.7%	0.00077	0.01341
Grid Undersaturated				
5000	-12.1%	-1.2%	0.00000	0.54663
10000	-10.1%	-2.7%	0.00000	0.03694
Arterial Oversaturated				
5000	-10.9%	-3.9%	0.00062	0.12497
10000	-10.5%	-6.9%	0.00065	0.00727
Grid Oversaturated				
5000	-14.2%	-3.3%	0.00000	0.01189
10000	-12.4%	-4.4%	0.00000	0.00545

Table 13: Comparison of GGA and SSGA with Real CHC

We find that Real CHC outperforms the GGA in all situations. The differences are statistically highly significant in all cases. The observed degradations in performance with GGA are between 6.9% and 14.2%. The performance of SSGA and Real CHC are closer. Real CHC outperforms the SSGA at 10000 function evaluations in all cases. The observed deteriorations in performance with SSGA are between 2.7% and 6.9% at 10000 function evaluations. Statistically insignificant degradations in performance are observed with the SSGA at 5000 function evaluations except for the oversaturated grid scenario where the degradation is significant. Thus Real CHC appears to be the unanimous winner from the three algorithms tested.

8.7 Optimal parameter tunings

Different genetic algorithm parameter values were tested for the Real CHC search algorithm. The parameter α of the blend crossover operator and the population size *N* were investigated. In section 8.5.2, we found that the blend crossover offers considerable improvements over the standard binary crossover operators. The performance may be further improved by tuning the value of α. Genetic algorithms are also known to be sensitive to the choice of the population size. The divergence rate *r* was not considered as the restart mechanism was rarely applied.

A factorial experiment [145] where both parameters are simultaneously varied is beyond the scope of this study and was not performed. Instead a sensitivy analysis was conducted where different values of a single parameter were tested while keeping the value of the other at the base condition.

8.7.1 Blend crossover parameter

The search performances with $\alpha = 0, \alpha = 0.25, \alpha = 0.75$ and $\alpha = 1$ were compared against the runs with $\alpha = 0.5$ in section 8.5.4. The results are given in Table 14.

Function Evaluations	% Improvement/Degradation				Significant Difference Test - p value			
	BLX-α				BLX-α			
	0	0.25	0.75	1	0	0.25	0.75	1
Arterial Undersaturated								
5000	-38.0%	-11.3%	-6.7%	-18.6%	0.00000	0.00001	0.00025	0.00000
10000	-29.3%	-11.0%	-2.1%	-6.8%	0.00000	0.00002	0.24509	0.00169
Grid Undersaturated								
5000	-36.1%	-14.7%	-17.5%	-28.6%	0.00000	0.00000	0.00000	0.00000
10000	-26.5%	-10.8%	-5.8%	-16.6%	0.00000	0.00000	0.00011	0.00000
Arterial Oversaturated								
5000	-32.8%	-6.9%	-10.5%	-18.8%	0.00000	0.00734	0.00050	0.00001
10000	-30.5%	-10.0%	-4.6%	-14.7%	0.00000	0.00077	0.03104	0.00006
Grid Oversaturated								
5000	-47.9%	-15.1%	-16.2%	-31.9%	0.00000	0.00000	0.00000	0.00000
10000	-39.0%	-14.6%	-4.9%	-19.1%	0.00000	0.00000	0.00023	0.00000

Table 14: Impact of blend crossover parameter on performance of Real CHC (relative to $\alpha = 0.5$)

From the table we see that the search performance is highly sensitive to the choice of α. A more exploitive search ($\alpha = 0$ or $\alpha = 0.25$) significantly retards search performance in all instances. The search may be converging too quickly and the neighbourhood of good solutions are not investigated sufficiently. An excessively explorative search ($\alpha = 0.75$ or $\alpha = 1$) is not advantageous either. A plausible explanation of this outcome is that good solutions are not being effectively capitalized upon. We find that $\alpha = 0.5$ produces an effective balance between exploration and exploitation. Larger degradations in performance are obtained as we move further away from the optimal choice of $\alpha = 0.5$.

8.7.2 Population size

Optimization runs were performed using larger and smaller population sizes. Optimization performance using population sizes of $N_{Pop} = 30$ and $N_{Pop} = 70$ are compared to that with $N_{Pop} = 50$ in the base scenario in 8.5.4. The results are given in Table 15.

Function Evaluations	Relative Improvement/Degradation		Significant Difference Test - p value	
	Population Size		Population Size	
	30	70	30	70
Arterial Undersaturated				
5000	-0.9%	-5.1%	0.62597	0.00352
10000	-1.3%	-1.0%	0.58109	0.56426
Grid Undersaturated				
5000	-1.7%	-5.2%	0.15575	0.00057
10000	-0.6%	0.4%	0.68466	0.44424
Arterial Oversaturated				
5000	-0.8%	-4.2%	0.70061	0.07046
10000	-1.9%	-0.8%	0.35594	0.67882
Grid Oversaturated				
5000	-2.7%	-4.8%	0.05312	0.00007
10000	-1.2%	-0.2%	0.40356	0.85287

Table 15: Impact of population size of performance of Real CHC

We find that the search performance is less sensitive to the choice of the population size. A smaller population $(N_{Pop} = 30)$ produces statistically insignificant reductions in performance. For the oversaturated grid scenario performance degradation at $F = 5000$ is almost significant. A larger population $(N_{Pop} = 70)$ results in significant reductions in performance early on in the search for three of the four scenarios. However, at 10000 function evaluations, the performance is not significantly any worse. Thus the Real CHC algorithm is robust to the choice of population size, although a population size of $N_{Pop} = 50$ is the marginally better choice.

We find no improvement over the base parameter values for both the blend crossover parameter and the population size.

8.8 Overall improvement

We now evaluate the total improvement in performance when the modifications are combined. The base genetic algorithm that we started with is the GGA with standard binary encoding, binary crossover and binary mutation operators with independent replications for fitness evaluations with the re-evaluation mechanism and a single replication for fitness evaluation. This corresponds to the genetic algorithm used by Park [121] (see section 5.6). After investigating several alternatives, the best performing algorithm is the Real CHC algorithm with BLX-0.5 crossover, real mutation and the use of CRN's in fitness evaluations and a single replication.

The relative improvement of this algorithm over the standard one is given for all scenarios and number of function evaluations in Table 16.

Function Evaluations	Scenario			
	Arterial Undersaturated	Grid Undersaturated	Arterial Oversaturated	Grid Oversaturated
1000	9%	2%	5%	20%
2000	16%	14%	11%	29%
3000	21%	17%	17%	29%
4000	23%	21%	20%	32%
5000	20%	22%	21%	32%
6000	19%	20%	19%	32%
7000	19%	19%	19%	31%
8000	17%	20%	17%	31%
9000	13%	19%	16%	32%
10000	13%	21%	16%	30%

Table 16: Relative improvement of enhanced genetic algorithm over the standard genetic algorithm

We find that the improved algorithm offers delay reductions between 20% and 32% at 5000 function evaluations. The delay reductions at 10000 function evaluations are slightly smaller, lying between 13% and 30%. The largest improvements are obtained for the difficult oversaturated grid network problem. Plots of the mean best extended network delay figures (i.e. X_F using the notation in section 7.9) are given for each network scenario in Figures 15 to 18. The performance of the standard algorithm is given by the solid line and the improved algorithm

is depicted with the broken line. From these figures we see that in all scenarios, the improved algorithm within 2000-3000 function evaluations matches the performance of the standard algorithm with 10000 function evaluations.

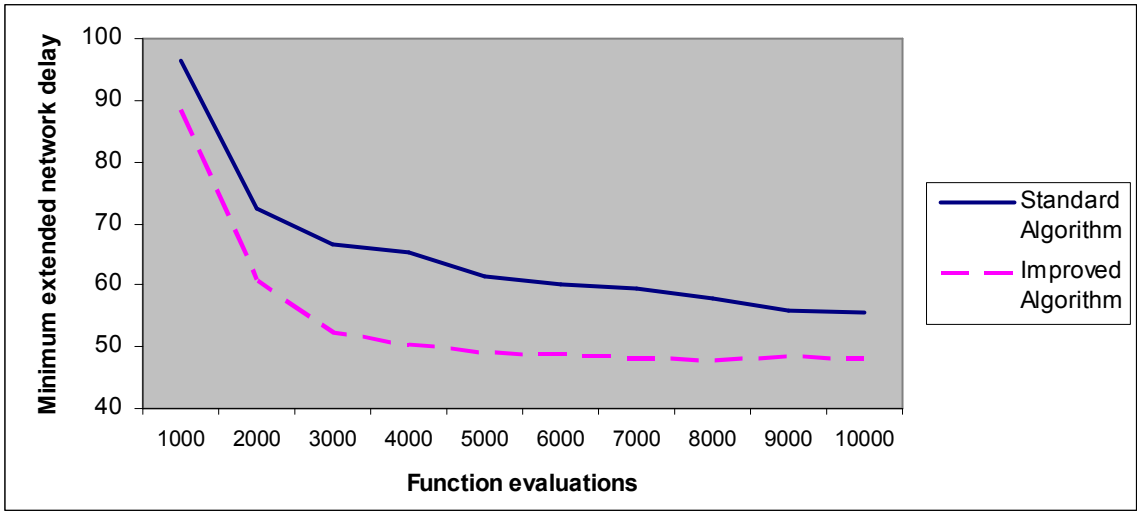


Figure 15: Comparison of algorithm performance – Arterial Undersaturated scenario

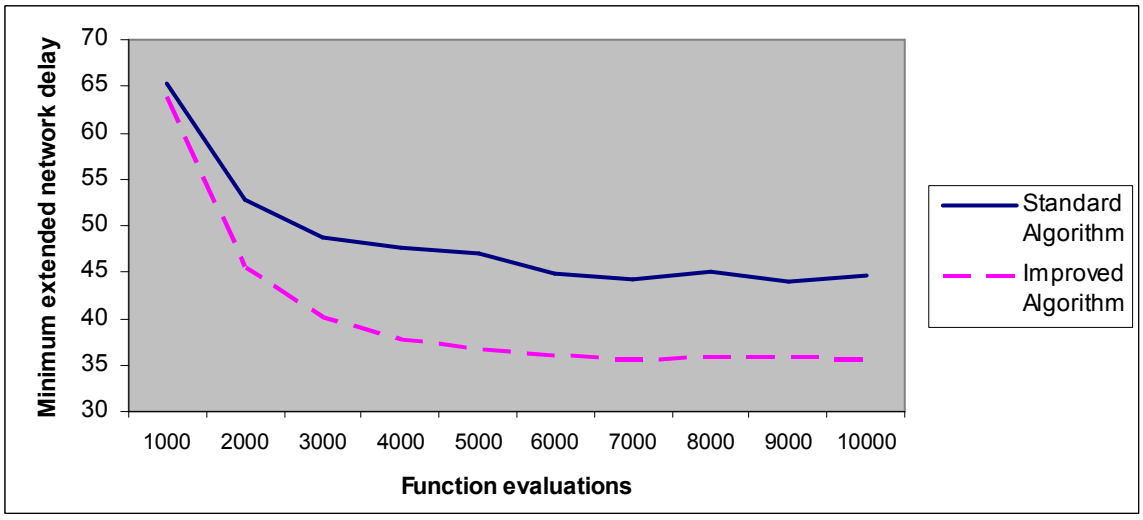


Figure 16: Comparison of algorithm performance – Grid Undersaturated scenario

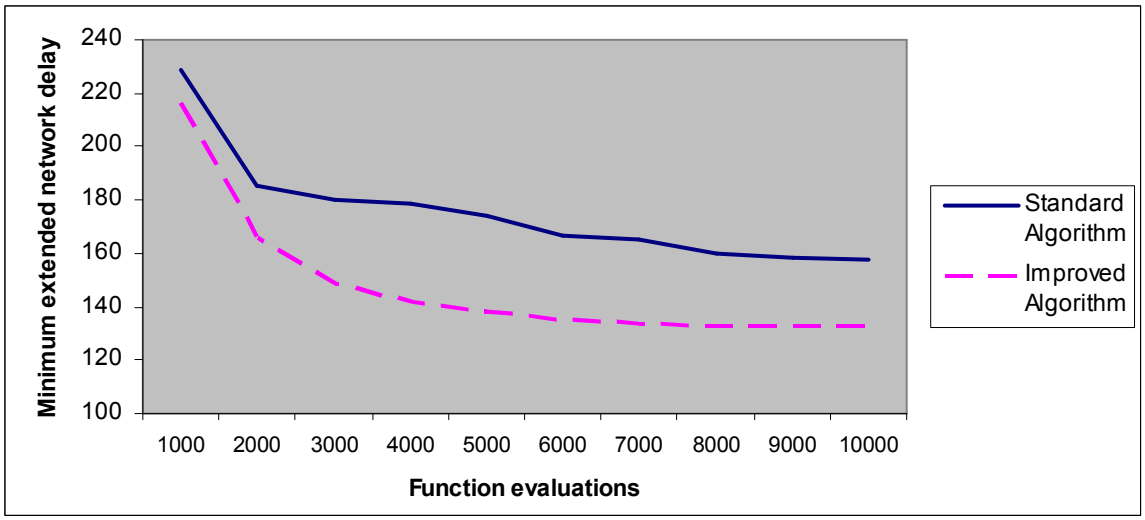


Figure 17: Comparison of algorithm performance – Arterial Oversaturated scenario

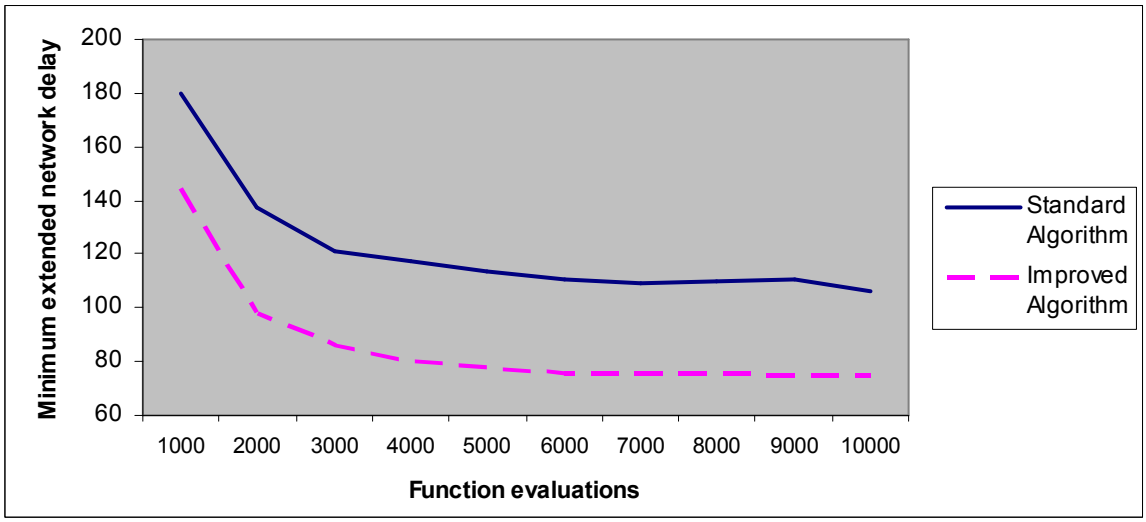


Figure 18: Comparison of algorithm performance – Grid Oversaturated scenario

9 Chapter Nine – Conclusions and Recommendations

This dissertation has explored improvements in the optimization of fixed time traffic signals via genetic algorithms.

We have proposed an alternative measurement of delay which we call extended network delay that is applicable to both undersaturated and oversaturated conditions. The disadvantage of this delay measure is that it requires an extension of the simulation process and thus requires additional computation time to evaluate any signal timing plan. We also suggest an enhanced problem encoding that allows the number, structure and sequence of signal stages to be optimized. The genetic algorithm approaches previously require the number and structure of stages to be pre-specified.

A review of the optimization literature was performed to identify potential improvements to the genetic algorithm. Based on this we identified and considered several modifications to the genetic algorithm search, encompassing alternative algorithms, genetic operators, problem encodings as well as tunings of the parameters governing the search process. A high fidelity microscopic traffic simulation model was developed for the evaluation of alternative signal timing. The model has been found to produce results very similar to a well validated simulation model in commercial use. The model summarizes the assessment into a single objective function value, extended network delay, which is used as the minimization objective of the genetic algorithm search. This same model was used for generating statistics on which the evaluation of alternative search policies was based. The algorithm modifications were thoroughly tested on a test bed of four traffic networks covering arterial and grid network structures in both undersaturated and oversaturated conditions. The efficiency of search was measured by the quality of the best solution produced after a pre-specified number of objective function evaluations. The stochastic nature of both the search and evaluation processes was addressed using a rigorously validated statistical model. This allowed for testing of whether observed sample statistics indicated statistically significant differences in mean search performance, allowing us to reach definitive conclusions.

Our investigations into various genetic search algorithms on the test networks revealed the following:

- The GGA re-evaluation mechanism does not provide any statistically significant benefit in undersaturated or oversaturated scenarios.
- Comparing solution quality of individuals in a genetic population with the use of CRN's considerably improves the optimization over that with independent replications. The benefits with CRN's make the re-evaluation mechanism in GGA's redundant.
- Although the measures of solution quality from the microscopic traffic simulator are stochastic, the greatest optimization efficiency is achieved by basing selection and replacement on a single replication of the objective function value of each individual. This is true in both undersaturated and oversaturated scenarios, in contrast to the suggestions in the TRANSYT-7F user manual [(49)].
- Optimization is more effective when the binary genetic operators operate on the Gray coded variables.
- Even better than the Gray coding is the use of real crossover and mutation operators that operate on the decision variables in the problem space. Specifically blend crossover and real mutation outperform their binary counterparts. The standard binary operators are still required for operating on the decision variables regarding the number, structure and sequence of stages as these are encoded using binary genetic material.
- We propose a modification to the CHC algorithm we call Real CHC which extends the idea of searching in the real parameter space.
- Optimization is most effective with the Real CHC, followed by the SSGA and then the GGA.
- The Real CHC algorithm is robust to the choice of population size. Optimization performance is quite sensitive to the blend crossover parameter α and a value of $\alpha = 0.5$ was found to be optimal.
- Combining these search enhancements, we see substantial improvements in optimization efficiency.

Several of the findings are applicable to genetic algorithm function optimization in general:

- The re-evaluation mechanism in GGA's is not found to aid optimization of noisy functions.
- The use of CRN's for fitness evaluations is recommended as it considerably improves optimization over the use of independent replications.
- The robustness of genetic algorithms in noisy environments with a single replication for fitness evaluation corroborates the findings by other researchers.
- Gray codes improve search, supporting the general consensus on their efficacy over the standard binary coding.
- Real genetic operators outperform their binary counterparts.
- We demonstrate that binary and real genetic coding and operators can be successfully combined.
- We introduce Real CHC which is a modification of CHC where the incest prevention mechanism is based on distance measurements in the problem space.
- The superiority of the SSGA over the GGA and CHC over the SSGA are once again demonstrated, endorsing the findings in the literature.
- The superiority of CHC and the SSGA over the GGA on a noisy problem is verified.
- Blend crossover is most effective with crossover parameter $\alpha = 0.5$.
- The robustness of CHC to choice of population size is confirmed.
- We propose a statistical model which may be applied to the comparison of genetic algorithms on other noisy problems. Certain assumptions may need to be modified for the particular application, but the general framework can be retained.

Some auxiliary findings and developments arising from this study are:

- Generalizations of blend crossover and real mutation for application to the fraction-based signal timing encoding scheme are introduced. These generalizations allow for the real genetic operators to be applied to optimization problems where the search domains of decision variables vary from individual to individual or the domains of certain decision variables are a function of other decision variables.
- The experiments used to set the assumption of the statistical model for comparing genetic algorithm performance has revealed that mean and variance of delay are positively correlated. Thus traffic signal timing policies that minimize mean extended

delay are also the most reliable signal timing schemes in that variance of extended delay is also minimized.

- The source code for the traffic simulation model MSTRANS used for the empirical dissertation is provided and is open to modification and application on other road traffic research.

Our recommendations to the developers of commercial traffic signal optimization software are:

- Utilize extended network delay as the performance measure when delay minimization is the intended.
- Adoption of the proposed problem encoding scheme to allow the number of structure of stages to be optimized.
- To implement the Real CHC algorithm for optimization with blend crossover parameter $\alpha = 0.5$, population size $N_{Pop} = 50$ and divergence rate $r = 0.35$.
- Apply a single replication with CRN's for fitness evaluation in both undersaturated and oversaturated traffic network scenarios.
- To incorporate the optimization algorithm and traffic simulation model in a single program. Currently, TRANSYT-7F [49] performs the genetic algorithm optimization, calling the CORSIM [50] traffic simulation routine externally to evaluate candidate solutions. This involves considerable overhead for evaluating each individual. TRANSYT-7F must first produce an input file for CORSIM specifying the signal timing policy along with all other network specific information. CORSIM is executed as a DLL and the network is read in and the required initializations are performed, followed by the execution of the simulation model. CORSIM then produces an output file from which the required performance measure is extracted by TRANSYT-7F. The MSTRANS model that we have developed performs the optimization and evaluation processes within a single application. A comparison of the execution times was made by optimizing the undersaturated arterial network scenario using both TRANSYT-7F and MSTRANS. The GGA with the parameter values in section 7.8, standard binary crossover and mutation operators with CRN's for fitness evaluations was executed for 900 function evaluations in both programs. We used network delay and not extended network delay as the optimization objective to ensure that the network simulation periods were identical in

both models. The required run-times on a 2.8GHz hyper-thread Pentium 4 PC were 35 minutes for TRANSYT-7F and 4 minutes for MSTRANS. Thus execution of MSTRANS was over 8 times faster. Part of the increased execution speed may be attributable to the fact that the MSTRANS simulation model is less detailed than that of CORSIM. Nevertheless, incorporating the optimization and simulation processes into a single application may substantially speed up the execution and allow for a more extended search given identical computational resources.

Suggestions for further research into improved genetic algorithm optimization of traffic signals are:

- Given the success observed with real genetic operators, other real genetic algorithms may offer further benefits. Leung and Wang [146] have developed a genetic algorithm that uses ideas from orthogonal design for the generation of the initial population and the crossover operator. Their genetic algorithm has demonstrated improved performance to a standard genetic algorithm as well as to other heuristic optimization algorithms such as evolutionary strategies, simulated annealing and particle swarms. This algorithm may further enhance traffic signal optimization.
- Testing the improved optimization strategies in a multi-objective optimization approach. Genetic algorithms have been found to be effective multi-objective optimization tools [147]. In section 5.6, we mention a study by Sun et al [127] where a multi-objective genetic algorithm was used to generate the set of Pareto-optimal signal timing policies for minimum delay and stops at an isolated intersection. The approach can be extended to handle a network of intersections by using a stochastic traffic simulation like CORSIM [50] or MSTRANS as the evaluator. Multi-objective optimization is more difficult than univariate optimization and a larger number of function evaluations are required for a successful optimization. The modifications to improve univariate genetic algorithm optimization may also be beneficial if utilized in their multi-objective counterparts.
- Extending the enhanced problem encoding to allow for optimization of phasing for more complicated junctions layouts.

- Investigation of the feasibility of applying genetic algorithms for determining whether low volume intersections in the network can be operated using half cycle lengths.

Suggestions for further genetic algorithm research include:

- Further evaluation of Real CHC which we have not convincingly demonstrated as an enhancement over the standard CHC algorithm.
- Extensions to the statistical model. The model is applicable to the comparison of performance of genetic algorithms after a fixed number of function evaluations. If we assume a parametric form for the evolution of mean best performance in terms of number of function evaluations and estimate the parameters of this curve, this would allow for more powerful inferences to be made.
- The statistical model can also be improved by using an extreme value distribution [(148)] to describe the variability in search trajectories. The objective function value of the best individual in a randomized population must, according to statistical theory, follow an extreme value distribution[59]. Thus an extreme value distribution should be appropriate for describing the variability in the initial population and may still be appropriate in describing the variability in search trajectory after the application of genetic operators.

[59] Approximately, for large population sizes [(148)].

A1 Appendix One – Microscopic Stochastic Traffic Network Simulator

A1.1 Introduction

In this appendix, we discuss the implementation and operational characteristics of the computer simulation model MSTRANS (Microscopic Stochastic Traffic Network Simulation Model) used for performing the empirical work in this dissertation. An overview of the model is given in section 7.2.

We first describe the algorithmic structure of the model. Next, we discuss each modelling aspects in detail. Finally, we present the experiments performed to test the validity of the model. The entire code listing for MSTRANS is contained on the compact disc accompanying this dissertation.

A1.2 Algorithmic structure

Execution of the MSTRANS model occurs sequentially and can be separated into three components:

- Initialization
- Simulation
- Finalization

A1.2.1 Initialization

Initialization involves performing the following tasks:

- Read model assumptions and road network structure from the assumption spreadsheet into memory. Certain checks are performed along the way to ensure data validity.
- Pre-processing is performed to compute the value of other items required for the simulation run. Additional data checks are also performed.
- Seed random number generator.
- Create output files.

- Set simulation clock set to zero.
- Set network is set to an empty state.
- Clear the value of output variables.

A1.2.2 Simulation

Simulation is performed by repeating the steps below sequentially until the simulation clock exceeds the sum of the initialization and run times[60]:

- Update the indications of all traffic signals.
- Generate new vehicles at the entry nodes.
- Insert new vehicles into the network.
- Update acceleration decision and vehicle status or all vehicles in the network (1st pass).
- Update position and velocity of all vehicles in the network using equations of motion for constant acceleration (2nd pass).
- Perform additional processing for vehicles that have changed links (3rd pass). Remove vehicles that have exited the network. Accumulate output statistics for the vehicles that have exited the network if the simulation clock exceeds the initialization time.
- Accumulate additional output statistics if the simulation clock exceeds the initialization time.
- Advance the simulation clock.

A1.2.3 Finalization

At the end of the run, the performance measures are output to a text file.

A1.3 MSTRANS components

Here we comprehensively discuss the individual components of the MSTRANS model formulation. Wherever possible, we make reference to findings from the research to justify the modelling logic employed. Formulae relating to the motion of vehicles will be presented and we will assume that the reader is familiar with the equations of motion for objects undergoing constant acceleration. The items are discussed under the following headings:

- Simulation time step

[60] The run may be extended for the computation of extended delay (see section 7.5).

- Free flow acceleration
- Stopping
- Car following
- Lane changing
- Routing
- Vehicle generation
- Turning movements
- Start-up delay
- Response to amber
- Behaviour in oversaturated conditions
- Computation of measures of effectiveness
- Random number generation

A1.3.1 Simulation time step

A larger time step reduces the execution time of the model at the cost of increasing the coarseness of the modelling. The CORSIM simulation model which has been extensively validated uses a one second time step [50]. Thus, it was decided to use the same step size for the runs in MSTRANS.

A1.3.2 Free flow acceleration

Leading vehicles

Free flow behaviour applies to lead vehicles. These are vehicles that are first in their lane and not constrained by the movements of other vehicles.

Linear acceleration model

The simplest acceleration model is the constant acceleration model where acceleration is constant and independent of speed [9, 149]. However this model does not accurately match observed behaviour [149, 150]. The most commonly used acceleration model is one where acceleration decreases linearly with speed [9]:

$$A = \frac{dV}{dt} = \begin{cases} \alpha - \beta V \ \ for \ \ V \le \dfrac{\alpha}{\beta} \\ 0 \qquad for \ \ V > \dfrac{\alpha}{\beta} \end{cases},$$

(A1-1)

where A = acceleration (ft/s^2)

V = speed (ft/s)

t = time (s)

α (ft/s^2) and β (s^{-1}) are the parameters of the model $(\alpha, \beta \ge 0)$

This is called the linear acceleration model [9]. Solving the first order differential equation (A1-1), we obtain the speed of a vehicle that is initially at rest at $t = 0$ as a function of time [9]:

$$V = \frac{\alpha}{\beta}(1 - e^{-\beta t})$$

(A1-2)

Letting $t \to \infty$, we find that the maximum speed $\frac{\alpha}{\beta}$ is attained asymptotically. Differentiating equation (A1-2) with respect to time, we obtain the acceleration of a vehicle starting from rest as a function of time:

$$A = \alpha e^{-\beta t}$$

(A1-3)

Integrating equation (A1-2), we obtain an equation for distance travelled, X, as a function of time:

$$X = \frac{\alpha}{\beta}(t - \frac{1 - e^{-\beta t}}{\beta})$$

(A1-4)

The acceleration, velocity and distance curves produced by the linear acceleration model are illustrated in Figure 19.

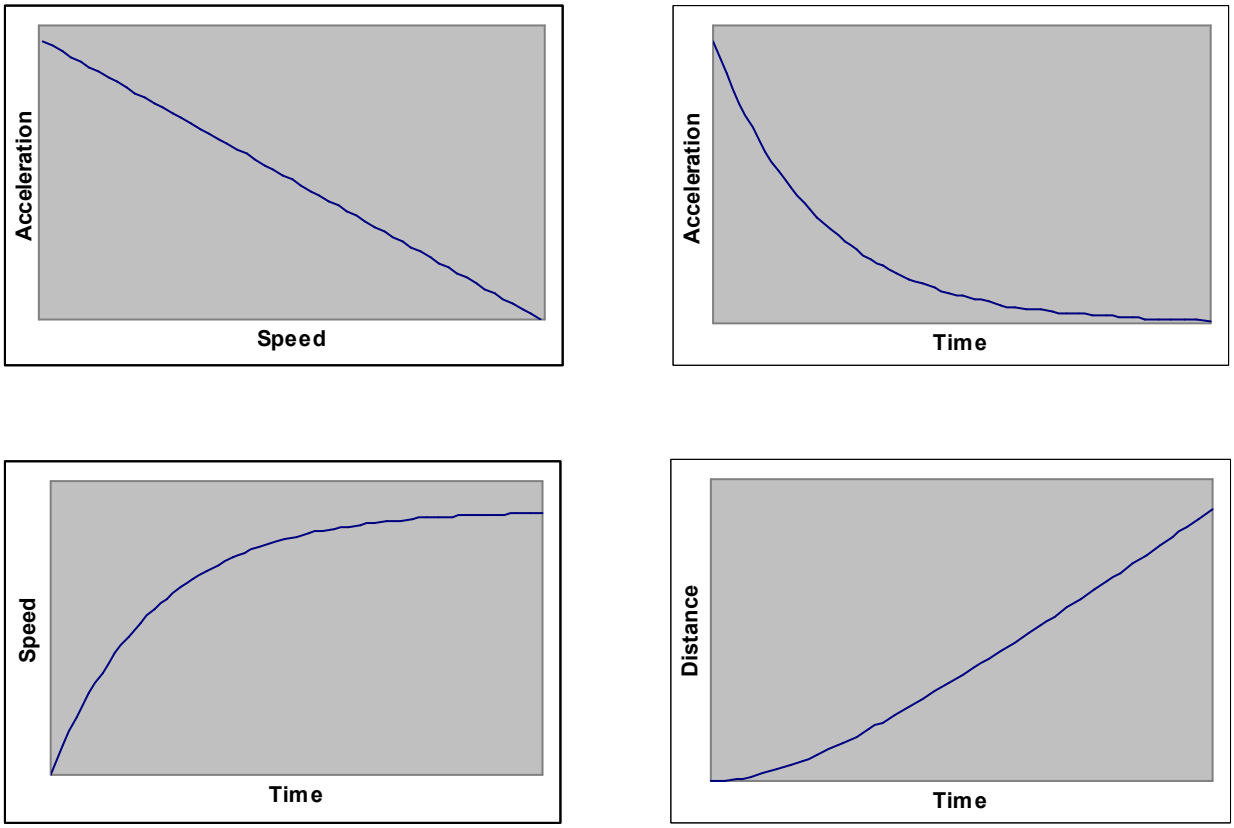


Figure 19: Vehicle kinematics according to the linear acceleration model

The linear acceleration model has been found to accurately describe observed maximum vehicle acceleration rates as well as driver preferred acceleration rates [151, 152]. It satisfies the common sense notion that motorists apply high level of power initially to overcome inertia when at rest and reduce acceleration gradually so as to make a smooth transition to zero acceleration at the desired speed of travel. The smoothness of the acceleration profile can be measured by jerk which is the slope of the acceleration versus time curve [152].

Polynomial acceleration model

Despite the wide acceptance of the linear acceleration model, two criticisms remain:

- Maximum velocity is attained asymptotically (see equation (A1-2) and Figure 19), a situation which is thought to be unrealistic [149].

- The jerk at the start of acceleration (see Figure 19), is conceptually incorrect [153]. In reality, drivers increase acceleration gradually up to the maximum acceleration rate. This has been confirmed by the observation of S-shaped speed versus time curves [153].

These shortcomings are resolved by the polynomial acceleration model developed by Akcelik and Biggs [153], which describes acceleration as a polynomial function of time. The general shape of the curve is illustrated in Figure 20.

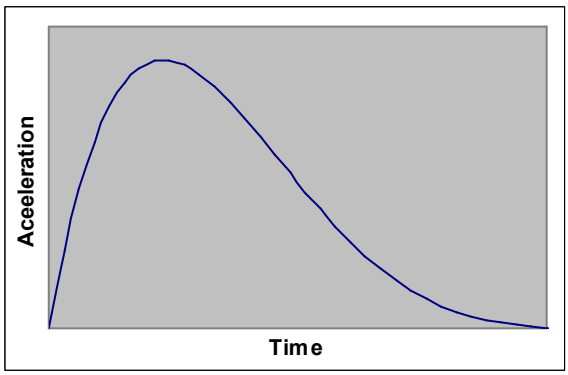


Figure 20: Polynomial acceleration model

The polynomial acceleration model requires considerable overhead when implemented in a computer simulation model. Furthermore, the superiority of the model fit to real data over the linear acceleration model has not been demonstrated[61]. The polynomial acceleration model has been empirically compared to the constant acceleration model and only when site specific data is available has the polynomial acceleration model demonstrated closer replications of real world observations. As a consequence, the linear acceleration model was adopted for modelling free flow acceleration in MSTRANS, despite the shortcomings mentioned.

[61] In the paper by Akcelik and Biggs [153], the polynomial model is compared to a linear acceleration model. However, the "linear acceleration model" considered in their paper is one where accerlation decreases linearly with time. This is not the same as one where acceleration decreases linearly with speed. A linear decrease in acceleration with speed corresponds to an exponential decrease in speed with time (see equation (A1-3) and Figure 19). This point has been overlooked by some researcher's who have acknowledged the superiority of the polynomial acceleration model.

Implementation on linear acceleration model in MSTRANS

For modelling of free-flow acceleration in MSTRANS we refer to parameter calibrations of the linear acceleration model for driver preferred acceleration rates. From now on, our references to acceleration apply to preferred acceleration rates. The most comprehensive study by Bonneson [151] found that $\alpha = 6.63$ ft/s^2 (2.02 m/s^2) described average acceleration characteristics at several sights. This value is close to the findings in other studies [9, 152]. Since the maximum velocity is given by $\frac{\alpha}{\beta}$, we can solve for β using:

$$\beta = \frac{\alpha}{f}, \tag{A1-5}$$

where f = desired free flow speed (ft/s).

A free flow speed of f = 54 ft/s (59.25 km/h) was assumed for all test networks resulting in β = 0.1228 s^{-1}.

A discrete approximation of this model is used in that the instantaneous acceleration rate as computed by equation (A1-1) is applied for the full length of the time step. The speed and distance profiles resulting from this approximation were computed for a vehicle accelerating from rest and were found to closely follow the continuous curves in Figure 19.

A1.3.3 Stopping

Constant deceleration

A lead vehicle approaching a red signal must decelerate to a stop. Empirical evidence indicates that drivers tend to decelerate to a stop at an approximately constant rate [154, 155]. The absolute value of the constant deceleration rate applied has been found to be larger for higher initial speeds.

Implementation in MSTRANS

The constant deceleration rates used are given in the table below. The values are the mean constant deceleration rates obtained in a study by Haas et al [155]:

Initial Speed (ft/s)	Initial Speed (km/h)	Constant Deceleration Rate (ft/s^2)	Constant Deceleration Rate (m/s^2)
29-36	32-39	3.22	0.98
37-43	40-47	3.81	1.16
44-50	48-55	5.05	1.54
>50	>55	5.74	1.75

Table 17: Constant deceleration rates

The deceleration rate for vehicles travelling slower than 29 ft/s (32 km/h) was taken to be 3.22 ft/s^2 (0.98 m/s^2). We now present the algorithm used for stopping a vehicle:

From kinematics, it can be shown that the required constant deceleration rate d (>0) required for a vehicle travelling at velocity V to come to rest in a distance X can be computed as:

$$d = \frac{V^2}{2X} \qquad \text{(A1-6)}$$

The deceleration rate required for the vehicle to come to rest at the stop line is computed for a leading vehicle facing a red signal using the equation above. Only when d equals or exceeds the desired constant deceleration rate in Table 17 will the vehicle begin decelerating at rate d.

A1.3.4 Car-following

The free flow behaviour discussed in sections A1.3.2 and A1.3.3 can be used to describe the motion of the lead vehicle in a particular lane. The acceleration behaviour of vehicles following the leader can be modelled using car-following theory [156]. Car-following theory assumes that the behaviour of the following vehicle is based entirely on its relationship with the vehicle immediately ahead of it (the leading vehicle) [156]. For the purpose of introducing the car following models, we need some notation.

Let $X_L(t)$ = position of lead vehicle at time *t* in terms of distance from the upstream node[62] (ft)

$X_F(t)$ = position of following vehicle at time *t* in terms of distance from the upstream node (ft)

$\Delta X(t)$ = distance between leader and follower at time *t* (ft)

$= X_L(t) - X_F(t)$

$V_L(t)$ = speed of leader at time *t* (ft/s)

$V_F(t)$ = speed of follower at time *t* (ft/s)

$\Delta V(t)$ = relative speed difference between leader and follower at time *t* (ft/s)

$= V_L(t) - V_F(t)$

$a_F(t)$ = acceleration of follower at time *t* (ft/s^2)

τ = driver reaction time

In car-following, we wish to determine $a_F(t)$, the acceleration rate of the follower.

Stimulus response model

The earliest and most well known car-following model is the stimulus response model [156, 157] which has the following general form:

$$a_F(t) = \begin{cases} \alpha^+ [V_F(t)]^{m^+} \dfrac{\Delta V(t-\tau)}{[\Delta X(t-\tau)]^{l^+}} & if \ \ \Delta V(t-\tau) \geq 0 \\ \alpha^- [V_F(t)]^{m^-} \dfrac{\Delta V(t-\tau)}{[\Delta X(t-\tau)]^{l^-}} & if \ \ \Delta V(t-\tau) < 0 \end{cases}, \tag{A1-7}$$

where $\alpha^+, l^+, m^+, \alpha^-, l^-$ and m^- are the model parameters (usually non-negative)

According to equation (A1-7), the follower will accelerate if the leader is travelling faster than the follower and decelerate if the leader is travelling slower than the follower, allowing for a reaction time for the follower to perceive this difference. The extent of the

[62] See section 3.2 for the definitions of upstream and downstream

acceleration/deceleration depends on the distance between the two vehicles and the speed of the follower.

The stimulus response model has been implemented in several traffic simulation models (e.g. TRANS [154], TEXAS [150] and MITSIM [158]).

Despite being so well established, the stimulus response model has become less popular recently due to contradictory findings as to appropriate parameter values [156, 157]. The calibrated parameter values vary drastically for different data sets, even on repeated runs with the same driver [159], casting doubt as to the general applicability of the model. Thus it was decided not to implement this car following model in MSTRANS.

Collision avoidance models

Another family of car-following models are the collision avoidance models [156, 157]. These models make the assumption that the following driver selects his acceleration such that he can bring his vehicle to a safe stop should the leader undergo emergency braking. One such model is the Gipps car-following model [160]. This model was implemented in MSTRANS for the following reasons:

- The Gipps model has been used in some of the newly developed traffic simulation programs [156, 157].
- The model has been calibrated with test track data in recent studies by Ranjitkar et al [157] and Brockfeld et al [161]. In these studies, the parameter values for several other car following models were also calibrated using the same data sets. The collision constraint models provided the best overall fit to the observed speed and headway
- In the studies mentioned above, the parameters of each model were calibrated separately for several data sets. It was found that the collision constraint models required the smallest adjustments to the parameter values between data sets. Thus the collision constraint models are the most robust in terms of general applicability.

Gipps model formulation [160]

We know provide the formulation of the Gipps model[63]. We require the following additional notation:

Let $X_L^S(t)$ = stopped position of the leader should the leader undergo emergency braking at time t (ft)

$X_F^S(t)$ = stopped position of the follower allowing for a reaction time before braking in response to the leader's deceleration (ft)

dl = emergency deceleration rate of the leader > 0 (ft/s^2)

df = emergency deceleration rate of the follower > 0 (ft/s^2)

L = effective length of leading vehicle (ft). This includes a buffer into which the following vehicle is not willing to intrude, even at rest.

Now, from kinematics, it can be shown that the distance travelled by a vehicle, X, travelling at speed V and decelerating at a constant rate of d (>0) before coming to a stop is given by:

$$X = \frac{V^2}{2d} \qquad \text{(A1-8)}$$

Thus, $X_L^S(t) = X_L(t) + \frac{V_L^2(t)}{2(dl)}$

The follow will not react until time $t+\tau$ at which point the velocity of the follower will be $V_F(t) + a_F(t)\tau$. The distance travelled by the follower during the reaction time is $V_F(t)\tau + \frac{1}{2}a_F(t)\tau^2$.

We assume that the follower will require an additional $\frac{\tau}{2}$ seconds before decelerating, during which time the vehicle will be travelling at velocity $V_F(t) + a_F(t)\tau$. The distance travelled during this additional time is $[V_F(t) + a_F(t)\tau]\frac{\tau}{2}$.

[63] We use slightly different notation from that used by Gipps [160] and a formulation that is easier to program in a computer simulation model.

Thus $X_F{}^S(t) = X_F(t) + V_F(t)\tau + \frac{1}{2}a_F(t)\tau^2 + [V_F(t) + a_F(t)\tau]\frac{\tau}{2} + \frac{[V_F(t) + a_F(t)\tau]^2}{2(df)}$

In order for a collision not to occur, we require $X_L{}^S(t) - X_F{}^S(t) \geq L$

Substituting the expressions for $X_L{}^S(t)$ and $X_F{}^S(t)$ into the inequality above, we obtain the following inequality:

$[a_F(t)]^2 + b[a_F(t)] + c \geq 0$,

where $b = 2\left[\frac{V_F(t)}{\tau} + (df)\right]$

$$c = \frac{-2(df)}{\tau^2}\left[X_L(t) - X_F(t) - L + \frac{1}{2}\left\{\frac{V_L{}^2(t)}{(dl)} - \frac{V_F{}^2(t)}{(df)}\right\} - \frac{3}{2}V_F(t)\tau\right]$$

(A1-9)

From the quadratic formula, we can conclude that the largest acceleration the follower can apply while satisfying the collision constraint is given by:

$$a_F(t) = \frac{-b + \sqrt{b^2 - 4c}}{2}$$

(A1-10)

According to the Gipps model, the actual acceleration rate applied by the follower is the smaller of:

- The acceleration rate as computed by (A1-10)
- The free flow acceleration rate

That is, the follower will never accelerate so rapidly so as to exceed free flow acceleration. The shape of the free flow acceleration equation proposed by Gipps is illustrated in Figure 21.

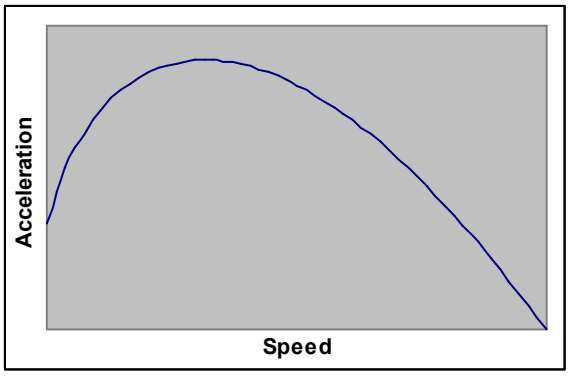


Figure 21: Gipps model free flow acceleration curve

Implementation in MSTRANS

The Gipps car-following model [160] was implemented in MSTRANS with a slight modification. It was decided to use the free flow acceleration constraint as computed by the linear acceleration model (see section A1.3.2) instead of that given by Gipps [160] (see Figure 21) for the following reasons:

- No details on the data and methods used to construct the curve in Figure 21 are given in the paper by Gipps [160].
- Using a free flow acceleration constraint for following vehicles that differed from the free flow acceleration applied to lead vehicles would be inconsistent.

The parameter values obtained in the calibration study by Ranjitkar et al [157] were used[64]:

τ = 1.13 s

dl = 11.65 ft/s^2 (3.55 m/s^2)

df = 13.02 ft/s^2 (3.97 m/s^2)

L = 24.61 ft (7.50m)[65]

The Gipps car-following acceleration rate is re-computed for each following vehicle at each time step. The computed rate is applied uniformly for the duration of the simulation time step.

[64] Similar parameter values have been obtained in other calibration study by Ranjitkar et al [162].
[65] This value is confirmed in an independent study on the distance between queued vehicles by Bonneson [151].

In certain cases, $b^2 - 4c < 0$ and the Gipps model will not produce a follower acceleration rate (see equation (A1-10)). This is indicative of a situation in which the follower cannot safely stop behind the leader. In this case, the deceleration rate applied is taken to be $\frac{V_F(t)}{\Delta t}$ where Δt is the simulation step size. This deceleration rate will result in the follower reaching a complete stop at the end of the simulation time step.

A1.3.5 Lane Changing

Detailed lane changing behaviour was not implemented in MSTRANS. Pragmatic rules are used to govern lane channelization and lane changing.

Lane channelization

The rightmost lane is reserved exclusively for right turns. Similarly, the leftmost lane is allows only left turns. The lanes between are reserved for through vehicles. Thus a minimum of three lanes must be specified. Using this approach, a through vehicle will never be restricted by a vehicle ahead that wishes to turn. For the test networks considered in this dissertation (see Appendix A3), two through lanes were assumed for each link resulting in a total of four lanes for each direction of travel.

Lane changing logic

As a consequence of the lane channelization above, we need to ensure that a vehicle is in the appropriate lane. Lane changes occur instantaneously when required. We now enumerate the various lane changing possibilities:

- Consider a vehicle that is currently on link A in one of the through lanes and will be travelling through to link B. Let us consider the possibilities:
    - *The driver wishes to execute a through movement at the end of link B:*
      In this case, the driver does not need to change lanes. He will remain in the same lane as he moves to link B.

- *The driver wishes to execute a right or left turn at the end of link B:*
  In this case, the driver will instantaneously move to either the rightmost or leftmost lane when he enters link B depending on whether he wishes to execute a right or left turn. The driver may cross several lanes instantaneously.

- Consider a vehicle that is currently on link A in the rightmost lane and will be turning right into link B. In MSTRANS, right turners turn from the rightmost link on the approach link to the right most lane on the receiving link (Turning movements are discussed in more detail in A1.3.8). Let us consider the possibilities:
  - *The driver wishes to execute a right turn at the end of link B:*
    In this case, the driver does not need to change lanes. He will turn from the rightmost lane on link A into the rightmost lane on link B where he needs to be in order to execute the next right turn[66].
  - *The driver wishes to execute a through movement at the end of link B:*
    In this case, the driver will instantaneously move to the through lane with the most unoccupied space once the right turn is completed. The driver may cross more than one lane instantaneously.
  - *The driver wishes to execute a left turn at the end of link B:*
    Once the right turn is completed, the driver will move to the leftmost lane, crossing the through lanes in between instantaneously.

- The situation for a vehicle travelling on link A, making a left turn into link B is analogous to the case of a right turn into link B.

The potential lane changer will only perform the lane change if there is no vehicle present in the zone parallel to him in the target lane. If this is not the case, the driver will come to a stop and will instantaneously change lanes only when a space becomes available.

[66] Although the model caters for such situations, the routing scheme employed does not allow for two consecutive right turns (see section A1.3.6).

A1.3.6 Routing

The route taken by a vehicle is assigned stochastically based on the specified through and turning probabilities. The sequence of through or turning movements the vehicle makes along the network constitute the route through the network. There is the possibility of the vehicle making a series of unrealistic turning movements e.g. a vehicle may make four consecutive left turns and "go around the block". To prevent these situations from occurring, the following rules are used to determine a valid route through the network:

- A vehicle that has at any point travelled north bound may not perform any turn movement that results in the vehicle travelling south[67].
- A vehicle that has at any point travelled south bound may not perform any turn movement that results in the vehicle travelling north.

Similar rules apply for the eastern and western directions of travelled. Constraining routes to satisfy these rules will prevent any nonsensical travel paths[68]. A list of valid routes is generated for each source node during the model initialization phase. A consequence of this procedure is that the expected proportion of through, left and right turning vehicles at the end of each link will differ from those specified in the network assumptions.

A1.3.7 Vehicle generation

In section 2.3.1 we noted that vehicle headways at the entry nodes in light traffic can be described by an exponential distribution. At higher flow levels, Luttinen (163) and Al-Ghamdi (164) have found the Gamma distribution to give a better fit to observed headways. In both studies, the parameter governing the shape of the distribution has been found to depend on flow. Luttinen's study (163) is based on more comprehensive data, but estimated parameter values for the Gamma distribution are not given. He does provide information regarding the location and density at the mode of the empirical distribution as well as the sample coefficient of variation. Each of these items is given for different flow levels. We computed estimates of the Gamma distribution

[67] Vehicles may only travel in one of the four perpendicular directions: north, south, east and west. All street are two-way.
[68] CORSIM uses an alternative approach where turning proportions at the end of a link can differ based on the entry movement (50). However the procedure in approach outlines in A1.3.6 was preferred as it requires no additional inputs.

parameters separately for each data item and our findings were not consistent. Al-Ghamdi's [164] was conducted in Saudi Arabia where driver behaviour has been shown to be drastically different from other countries [164]. Since we were unable to parameterize the Gamma headway distribution for higher flow levels, we decided to use the exponential distribution for generating headways at all flow levels.

The lane assigned to a vehicle entering the network is determined based on the movement performed at the end of the first link it is to be introduced on. A right turning vehicle is assigned to rightmost lane. A left turning vehicle is assigned to the leftmost lane. A through vehicle is assigned to the through lane with the fewest number of vehicles.

The new vehicle will be introduced travelling at the free flow speed. If the lane which the vehicle has been assigned to is full, then it will not be possible to introduce the vehicle. If the last vehicle in the lane is stationary and very close to the source node, then it would be unrealistic to introduce the new vehicle travelling at the free flow speed.

To solve this problem, a new vehicle is only introduced if it's computed deceleration in response to the last vehicle in the lane, as obtained by the car following equation (see section A1.3.4) is smaller in absolute value than $\frac{df}{2}$. If this is not the case, then the vehicle will be stored in a first-in-first-out queue at the entry node and an attempt to introduce the new vehicle will be made in each of the subsequent time steps.

A1.3.8 Turning movements

Turning arcs

In MSTRANS, right turning vehicles turn from the centre of the rightmost lane on the approach link to the centre of the rightmost lane on the receiving link. The vehicle is assumed to follow a circular turning arc. A similar situation is assumed for left turning vehicles. The left and right turning arcs for northbound traffic are illustrated in Figure 22.

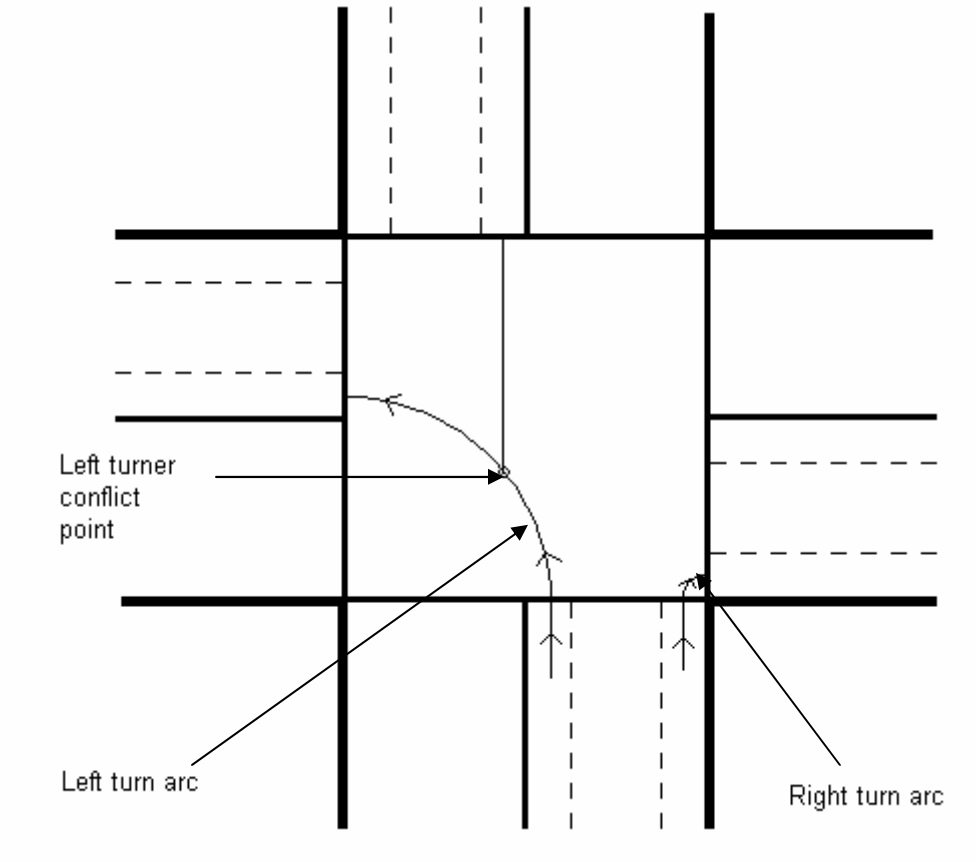


Figure 22: Turning arc's for right hand rule of the road

The start and end points of the turn corresponds to the extremes of the illustrated quarter circular arcs.

The right and left turns have radii $\frac{l}{2}$ and $\left(m+\frac{1}{2}\right)l$ respectively where

l = lane width (ft) and

m = number of lanes.

The length of the right and left turn arc's can be computed as $\frac{\pi\, l}{4}$ and $\frac{(2m+1)\pi\, l}{4}$ respectively.

A standard lane width of 12 ft (3.66 m) was assumed in all runs [139, 151].

Turning speeds

Vehicles cannot perform turn movements at the free flow speed. Maximum turning speeds are related to the turning radius and side friction by the equation [154]

$$V_T = \sqrt{Fgr},$$

(A1-11)

where V_T = maximum turning speed (ft/s)

F = coefficient of friction

r = turning radius (ft)

g = acceleration of gravity (= 32.2 ft/s^2 = 9.8 m/s^2)

However, rather than computing the maximum turning speeds from the equation above, the maximum turning speeds were computed by performing a calibration for saturation flow (see section A1.4.2). Based on the calibration, maximum turning speeds of 24 ft/s (26.33 km/h) and 31 ft/s (34.02 km/h) were assumed for right and left-turn movements respectively.

Slowing down for turns

A leading vehicle in free flow will need to slow down to the required maximum turning speed. From kinematics, we find that the constant deceleration rate d required by a vehicle a distance X from the stop line travelling at velocity V, to slow down to speed V_T $(V > V_T)$ is given by:

$$d = \frac{V^2 - {V_T}^2}{2X}$$

(A1-12)

For a turning vehicle approaching the stop line travelling faster than the maximum turning speed, the required deceleration rate is computed at each time step using the equation above. Only when the required deceleration rate exceeds the desired deceleration rate in Table 17[69], will the vehicle begin to slow down using the required deceleration rate d.

[69] Although there is some evidence that indicates that drivers apply smaller deceleration rates for slowdowns than for complete stops, it was decided to use the same deceleration rates for stopping and slowing down in MSTRANS.

Left turn gap acceptance

For the right hand rule of the road, vehicles turning left at a signalized intersection must cross over the path of through vehicles arriving from the opposing direction. If both streams receive green simultaneously, then the through movements from the opposing direction receive preference. The left turning vehicles may only proceed when a "gap" is available in the opposing traffic stream [9]. A left turner must decide whether to complete his turn based on the size of the "gap".

The most common model for gap acceptance is based on the time hypothesis where it is assumed that drivers evaluate gaps based on the expected time of arrival of the next opposing through vehicle at the conflict point (i.e. the time gap) [9]. However, studies have demonstrated that drivers have difficulty in perceiving the actual time to arrival [165]. In a study by Davis and Swenson [165], the distance of the opposing vehicle from the conflict point was found to be the only significant variable in predicting the gap acceptance probability. The logistic model that they fit to observed behaviour to distance gaps acceptance is presented below:

$$\pi = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} ,$$

(A1-13)

Where π = probability of accepting a gap,

x = size of the gap (ft),

β_0 = -8.28 and

β_1 = 0.055 are the estimated model parameters.

A left turner starts searching for a gap once it crosses the stop line and enters the intersection. If a gap is rejected, the vehicle will stop at the conflict point illustrated in Figure 22 and evaluate subsequent gaps. Left turner's that have entered the intersection and are still waiting at the end of the unprotected green interval will complete their turns during the amber and all-red interval.

A1.3.9 Start-up delay

On the onset of green, the lead vehicle in a stopped queue will experience a delay before moving (see section 2.3.2). A start-up delay of 2 seconds was implemented in MSTRANS, a value

confirmed in several studies [27, 151, 166, 167]. Queued drivers after the first also incur a starting response time. The average response time of these drivers has been estimated to be 1 second [151, 167] and this value was implemented in MSTRANS[70].

A1.3.10 Response to amber

A driver that is close to the stop line must decide whether to stop or rush through an amber signal. The decision to stop or not has been found to be independent of the length of the amber interval [9]. One would expect vehicles that are close to the stop line or travelling at a high speed to pass through the intersection on amber. These two factors can be combined by modelling the probability of stopping as a function of the deceleration rate required to stop at the stop line [9]. Table 18 presents stopping probabilities obtained in a study by Williams [168].

[70] The shorter response time of the second and subsequent drivers has been attributed to their ability to anticipate time to initiate motion by seeing the signal change as well as perceiving the movement of vehicles ahead [151].

Required Deceleration Rate (ft/s^2)	Required Deceleration Rate (m/s^2)	Probability of Stopping
0	0.00	1.00
1	0.30	1.00
2	0.61	1.00
3	0.91	1.00
4	1.22	0.97
5	1.52	0.94
6	1.83	0.89
7	2.13	0.81
8	2.44	0.71
9	2.74	0.58
10	3.05	0.45
11	3.35	0.34
12	3.66	0.25
13	3.96	0.18
14	4.27	0.13
15	4.57	0.10
16	4.88	0.07
17	5.18	0.06
18	5.49	0.05
19	5.79	0.04
20	6.10	0.03
21	6.40	0.03
22	6.71	0.03
23	7.01	0.02
24	7.32	0.02
>24	>7.32	0

Table 18: Amber stopping probabilities[71]

A1.3.11 Behaviour in oversaturated conditions

Spillback, intersection blockage and reduced saturation flow may occur in oversaturated conditions (see section 4.1). Here we discuss the program logic utilized to simulate these phenomena.

[71] Williams [(168)] does not provide details on the model that he fit to his data. He does however provide a graph plotting the probability of stopping as a function of the required deceleration rate. Selected values were read from the graph and a logistic model was fit, yielding the figures in Table 18.

Vehicles modelled in MSTRANS may not enter an intersection if there is insufficient storage space in the desired lane on the target link. The space available must equal or exceed the effective vehicle length *L*. Thus spillback is modelled and intersection blockage will never occur.

The lead vehicle in a particular lane will treat the last vehicle on the next link in the lane it will enter the next link on as its leader and will apply an acceleration rate that is the smaller of the car-following acceleration rate and the free flow acceleration rate. This will allow for a reduction of the saturation flow rate in oversaturated conditions.

A1.3.12 Computation of measures of effectiveness

Average values for delay, stops, speed and occupancy are computed. The averages are determined based on all vehicles that complete their journey through the network after the initialization time. These measures are computed separately for each link as well as at a network-wide level. Average delay is used for the empirical work and we will discuss the calculation of this performance measure only.

Delay is defined as control delay (see section 1.2.2) which is the difference between the actual and the uninterrupted travel time. The uninterrupted travel time for a particular vehicle can be computed as:

$$T_U = \frac{D}{f},$$

(A1-14)

where T_U = uninterrupted travel time (s)

D = distance traversed on path through the network (ft)

f = free flow speed (ft/s)

However, to account for the fact that vehicles are required to slow down in order to perform turning movements, T_U is computed in the pre-processing stage of MSTRANS by repeating the algorithm below for each possible route through the network:

- Set indications of all signals in all directions and for all movements to green
- Generate single vehicle to travel on route

- Set T_U = actual travel time for this vehicle

The delay for all vehicles can be averaged to compute network delay or extended delay (see section 7.5).

A1.3.13 Random number generation

Stochastic components of the model were simulated using the ISAAC pseudo-random number generator [143]. This particular generator was chosen as it has been ranked 1st in a comparison with 11 other pseudo-random number generators via empirical tests of pseudo-random number generator quality [95]. A facility for performing replications of the traffic simulation model using pre-specified seed values for the random number generator was implemented. This was required for performing evaluations using the Common Random Numbers variance reduction technique for comparison of alternative signal timing policies (see section 6.3.3 as well as Appendix A2). Separate random number streams for generating arrival patterns and routing schemes, and driver stochastic decisions were required for the implementation of this facility. Thus two seeds are required for performing each run of the model.

A1.4 Validation of MSTRANS

The validity of the MSTRANS model as a realistic and bug-free computer model for evaluating traffic signal timing plans was evaluated using:

- Animation
- Examination of queue discharge
- Comparisons with CORSIM

A1.4.1 Animation

The CORSIM model includes an animation facility called TRAFVU which allows for all features of a single network simulation run to be animated [50]. For the purpose of producing these animations, the CORSIM model produces several output files containing information regarding the state of all vehicles and signals at each second of the simulation. The CORSIM manual includes a document describing the format of these output files. MSTRANS was programmed to produce these same output files and this it was possible to animate and visualize

the MSTRANS model behaviour in TRAFVU. Programming errors could be easily identified by implementing and animating simple test networks.

In addition, TRAFVU allows for two networks to be animated simultaneously on a single screen. With this feature, we could compare identical networks where the results of one were generated by CORSIM and the others by MSTRANS and inconsistencies between the models could be identified and resolved.

A1.4.2 Examination of queue discharge

The queue discharge process was considered to be a critically important component of the model. Delay at an intersection is strongly dependent on the number of vehicles which can be discharged during the green period. The queue discharge process for through movements is a function of the free flow acceleration and car-following model (see sections A1.3.2 and A1.3.4). For turning movements, it is also dependent on the maximum turning speed.

Through movements

The discharge headway which is the time headway between vehicles as they cross the stop line was recorded for a queue of vehicles travelling through to the next link, producing Table 19.

Vehicle	Time of crossing the stop line after onset of green (seconds)	Discharge Headway (seconds)
1	2.1	
2	6.3	4.2
3	9.1	2.8
4	11.5	2.4
5	13.9	2.4
6	16.2	2.3
7	18.4	2.2
8	20.6	2.2
9	22.7	2.1
10	24.9	2.2
11	27	2.1
12	29.1	2.1
13	31.1	2
14	33.2	2.1
15	35.2	2
16	37.3	2.1

Table 19: Discharge headways for a standing queue[72]

The discharge headways remain relatively constant after the discharge of the 8th queued vehicle. This is consistent with empirical findings [151, 166]. The average discharge headway of the 9th - 16th queued vehicles is 2.1 seconds. The implied saturation flow rate is $\frac{3600}{2.1} = 1714$ vehicles/hr. A saturation flow rate of 1800 vehicles/hour is recommended for "ideal" conditions [139, 151, 166]. It would be possible to reproduce a saturation flow rate of 1800 vehicles/hr by changing the parameters of the car following and free flow acceleration models but the parameters were not altered. It was felt that a slight mismatch between the implied and "ideal" saturation flow would not impact the research findings.

[72] MSTRANS uses a one second time step and events may only be calculated to a minimum resolution of one second. To obtain more accurate headway estimates, the headways were computed using linear interpolation i.e. By assuming that distance is traversed linearly over any simulation time step. From the equations of motion, we know that this assumption is only approximately correct for a constant acceleration rate during each time step.

Turning movements

The saturation flow for turning movements will be lower than that of through movements due to the restriction on turning speeds. The Highway Capacity Manual [139] recommends a 15% reduction in saturation flow for right turning movements. This reduction was obtained by limiting the right-turning speed to 24 ft/s (26.33 km/h). A reduction of 5% is recommended for protected left-turning movements, but empirical studies have found the reduction to be between 7% and 18%. Several maximum left-turning speeds were tested and it was decided to use a maximum left-turning speed of 31 ft/s (34.02 km/h). This produced a saturation flow of 1511 vehicles/hr for the case of four lanes for each direction of travel. This corresponds to a 12% reduction compared to the through the saturation flow. The turning speeds assumed are in line with observed turning speeds [171].

A1.4.3 Comparisons with CORSIM

The CORSIM model is widely used and has been extensively tested and validated [172, 173, 174]. It was thus decided to complete the validation exercise by comparing the performance measures produced by MSTRANS and CORSIM under identical conditions. Since the models were developed independently we would inevitably find statistically significant differences between the model outputs given a sufficiently large number of independent replications. Rather than testing for these differences, we instead attempt to quantify the magnitude of the difference. Delay is considered as it is the only performance measure used for the empirical work in this dissertation. An undersaturated and an oversaturated scenario were considered

Undersaturated conditions

The nine signal arterial network described in Appendix A3 was coded in both MSTRANS and CORSIM. A cycle time of 70 seconds was used and the offset of all signals was set to zero. Each signal was coded with 2 phases and 35 seconds of green time was given to arterial movements (North/South) and 25 seconds for cross street movements (East/West). Vehicle flow levels were reduced by 50 % to ensure that all movements were undersaturated (i.e. no pseudo-congestion). An amber time of 3 seconds and all-red time of 2 seconds was assumed for the end of each phase. A free flow speed of 54 ft/s (59.25 km/h) and identical lane channelization was coded in both models.

The parameters values given in section A1.3 of this Appendix were used in MSTRANS. The default parameters in CORSIM were retained with the exception of the discharge headway. The mean discharge headway[73] in CORSIM was changed from the default value of 1.8 seconds to 2.1 seconds to match MSTRANS. The exponential distribution was used for entry headway generation in both programs. An initialization time of 3 minutes followed by 15 minutes of simulation time was assumed. 100 Independent replications of the network were performed in each program. The results for network delay are summarized in Table 20.

	MSTRANS	CORSIM
Sample Mean (sec/veh)	33.58	36.54
Sample Standard Deviation (sec/veh)	0.90	1.03
95 % Confidence interval for mean of network delay (sec/veh)	(33.41, 33.76)	(36.33, 36.74)
95 % Confidence interval for standard deviation of network delay (sec/veh)	(0.79, 1.05)	(0.82, 1.44)

Table 20: MSTRANS and CORSIM network delay in undersaturated conditions[74].

Table 20 demonstrates a reasonable agreement between the models in terms of mean and variance of network delay. It was possible to obtain greater conformity between the models by using the same assumptions (e.g. the effective vehicle length L assumed is quite different), but we retained the MSTRANS parameters as the assumptions have been determined based on objective research.

The models were further compared by examining output at a link level. The sample mean and sample standard deviation of delay for the 56 links in the network are illustrated via scatter plots

[73] Discharge headways are generated stochastically in CORSIM [(50)].
[74] The confidence interval for the mean is constructed by appealing to the central limit theorem. The confidence interval for the variance is obtained under the assumption that network delay has a normal distribution. This was verified by an examination of the histogram of the network delay figures.

in Figure 23 and Figure 24. The 45° lines correspond to points where the two models are in perfect agreement. Again we find satisfactory agreement between the two models.

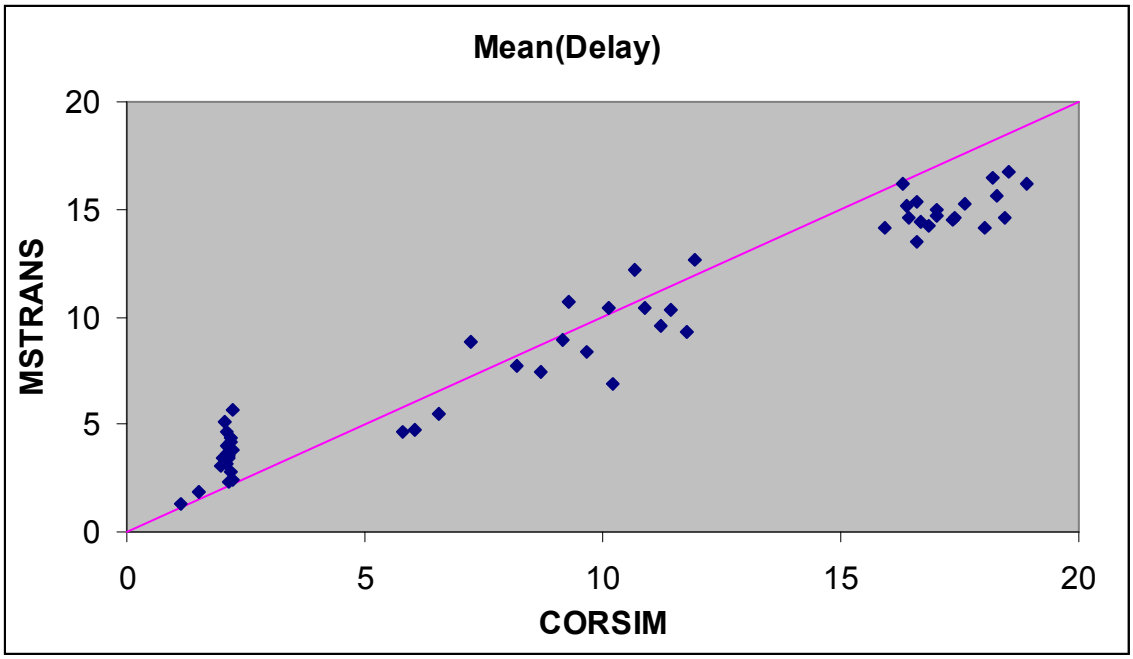


Figure 23: Comparison of CORSIM and MSTRANS – sample mean of link delay

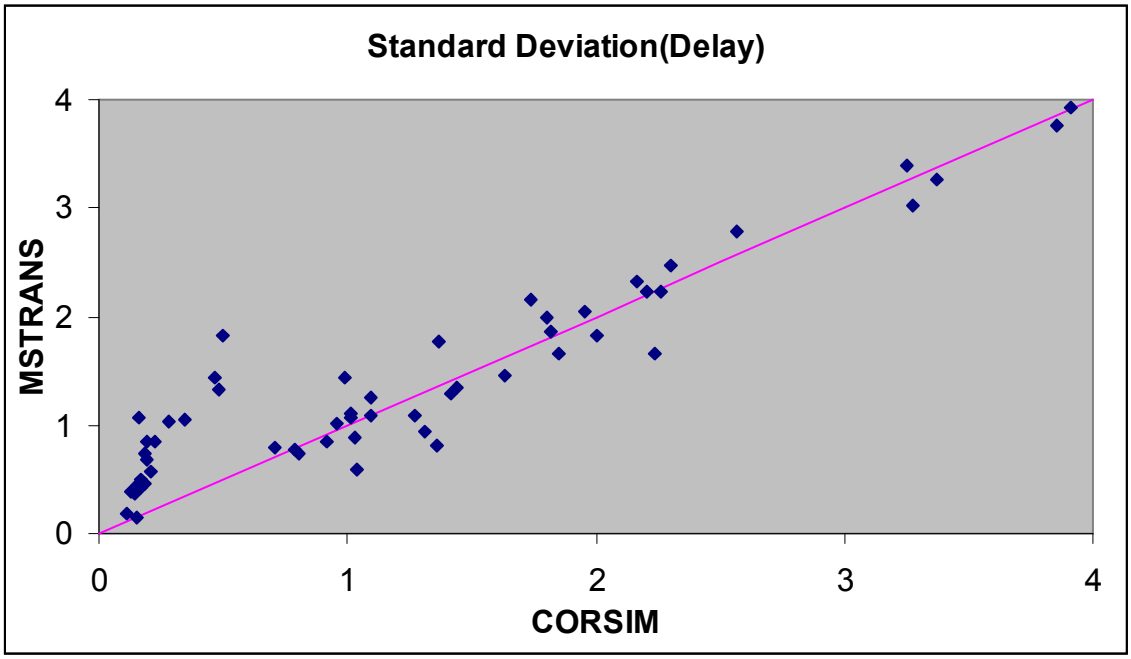


Figure 24: Comparison of CORSIM and MSTRANS – sample standard deviation of link delay

Oversaturated conditions

A second test was performed to validate MSTRANS for oversaturated conditions using the same test network. Vehicle flows were restored back to there full level and the East/West and North/South green splits were reversed to produce pseudo-congestion on the arterial. Based on 100 replications, MSTRANS and CORSIM produced substantially different estimates of network delay (the sample means were 71.71 sec/veh and 95.54 sec/veh respectively). The large difference was found to be due to different methodologies for computing delay for vehicles still present on the network at the end of the simulation run. The facility to compute extended network delay was enabled in MSTRANS (see section 7.5). From considerations of the length of the extension period from the 100 MSTRANS replications, it was determined that an extension period of 20 minutes was sufficient to allow all vehicles to exit the network in all circumstances. This extension period was implemented in CORSIM and allowed for the computation of extended network delay in CORSIM (no new vehicle arrivals occur during the extension period). The results for extended network delay are given in Table 21.

	MSTRANS	CORSIM
Sample Mean (sec/veh)	119.32	115.27
Sample Standard Deviation (sec/veh)	16.97	7.37
95 % Confidence interval for mean of network delay (sec/veh)	(115.95, 122.69)	(113.80, 116.73)
95 % Confidence interval for standard deviation of network delay (sec/veh)	(14.90, 19.71)	(6.47, 8.56)

Table 21: MSTRANS and CORSIM extended network delay in oversaturated conditions.

From Table 21, we find reasonable agreement between the models with respect to the mean of extended network delay. MSTRANS produces a much larger variance of extended network delay. The following causes for the differences were identified:

- Examinations of the animations of the network produced by each program identified more frequent spillback in MSTRANS than CORSIM. When spillback occurs, delay for the affected link is considerably increased. Thus the opportunity for spillback will increase the variability of delay. The causes for the increased likelihood for spillback in MSTRANS were identified:
    - CORSIM uses more sophisticated lane changing logic. Thus vehicles make better use of the storage space on oversaturated links and consequently spillback is less likely to occur.
    - The default effective vehicle lengths in CORSIM are much smaller that that used in MSTRANS (17 ft and 19 ft for the two different passenger car types in CORSIM as compared with 24.61 ft in MSTRANS). Thus links in CORSIM can store more vehicles.
- When the option for exponential headway generation is selected in CORSIM, the entry headways distribution is conditional in that the actual number of vehicles generated at each source node is exactly equal to the expected number based on the flow [(50)]. The entry headway distribution in MSTRANS allows for fluctuations in total flow between

replications whereas CORSIM does not. This contributed to the smaller variance of network delay produced by CORSIM.

Summary of comparisons

The MSTRANS model produces reasonable estimates of network delay when compared with CORSIM in both undersaturated and oversaturated conditions. Where differences exist, these differences can be explained in terms of differences in assumption values or methodology. MSTRANS produces larger variability in network delay than CORSIM in oversaturated conditions.

A2 Appendix Two – Common Random Numbers (136)

A2.1 Introduction

In this appendix, we briefly discuss Common Random Numbers (CRN's), in the context of the traffic signal optimization problem.

A2.2 Variability in stochastic experiment output

The performance measures produced by stochastic traffic simulation models like CORSIM or MSTRANS vary due to the stochastic nature of the models. Arrival patterns, route allocations and certain driver decisions are generated stochastically. The realized values are generated based on the assumed statistical distributions and differ between replications.

A2.3 CRN's for comparing alternatives

When comparisons between two or more alternative signal timing configurations are made, differences between the observed performance values are a consequence of the different signal timings as well as the fluctuations in experimental conditions (e.g. each timing plan may have been subjected to drastically different arrival patterns).

If we apply similar experimental conditions for all system configurations (i.e. identical arrival pattern and routing schemes), then we can attribute the computed differences in measured performances to the difference in the signal timing policies[75].

A2.4 Variance reduction

We can express this variance reduction mathematically for the case of comparing two alternative signal timing plans.

Let X_i and Y_i be the performance measures for the first and second signal timing policies on the *i*'th replication.

[75] Experimental conditions will not be perfectly identical as the driver's stochastic decisions are a function of the signal settings (see sections A1.3.8 and A1.3.10).

Let us assume that *n* replications are performed where:

$\{X_i\}_{i=1}^{n}$ are independently and identically distributed with mean μ_X,

$\{Y_i\}_{i=1}^{n}$ are also independently and identically distributed with mean μ_Y and

the tuples $\{(X_i, Y_i)\}_{i=1}^{n}$ are independent.

Let $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and

$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$.

We can measure the mean difference between the timing plans, $\mu_z = \mu_x - \mu_y$ using the unbiased estimate $\overline{Z} = \overline{X} - \overline{Y}$

$$= \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} Z_i ,$$

where $Z_i = X_i - Y_i$.

Now $Var(\overline{Z}) = \frac{1}{n} Var(Z)$

$$= \frac{1}{n}[Var(X) + Var(Y) - 2Cov(X,Y)]$$

(A2-1)

Using the same arrival patterns and routing schemes on the *i*'th replication for both signal timing policies will induce a positive correlation between the random variables X_i and Y_i. We do so for each replication and this will result in $Cov(X,Y) > 0$, reducing the variance of $\overline{Z}$ according to equation (A2-1).

Identical experimental conditions can be obtained by using the same random number seeds in the pseudo-random number generators for generating these outcomes. The sequence of random numbers produced by the generators will be identical for the same seed values. Provided that we ensure that the numbers are used for generating exactly the same items for each signal timing scenario, seeding the generators with identical seeds is sufficient.

A3 Appendix Three – Test Networks

A3.1 Introduction

In this appendix, we give the full specifications of the two test networks used for the empirical work in this dissertation.

The 9 signal arterial network was constructed by combing two US data sets. The signal spacing was taken from a New Orleans arterial network [33] and the traffic flows were from an arterial network in Iowa [175].

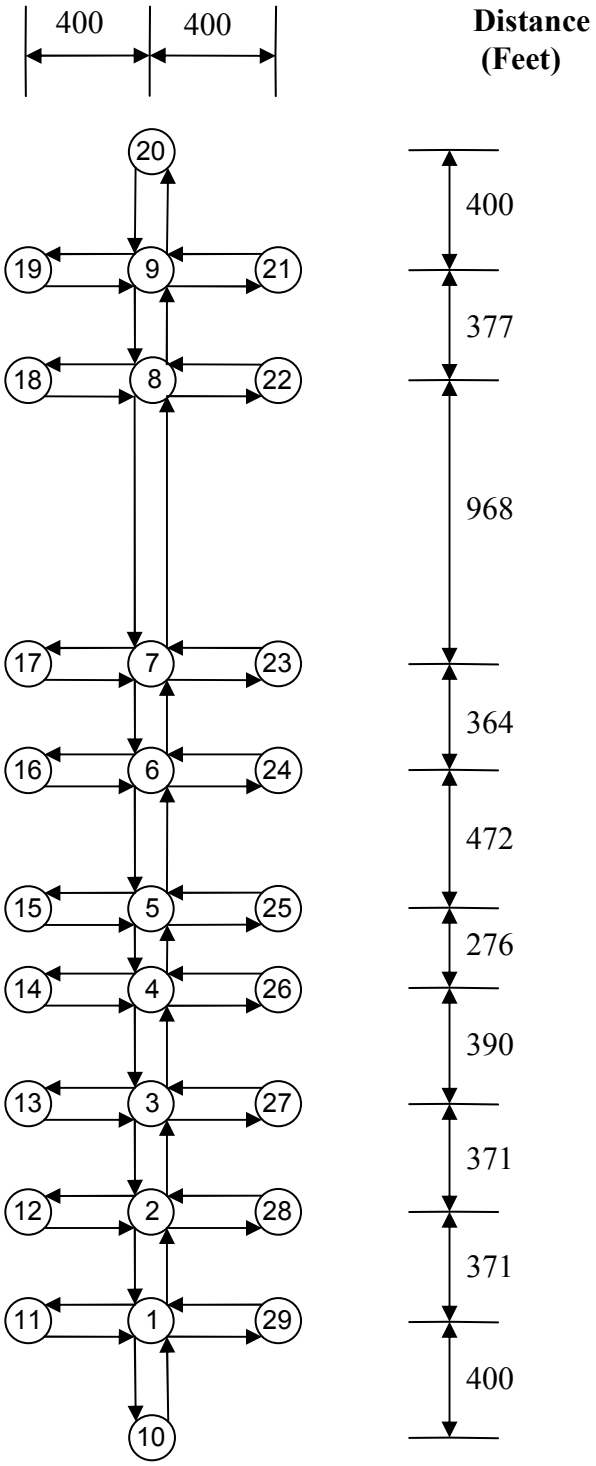
The 14 signal grid network was also based on a US data set [35]. Six of the one-way streets in the original data were converted into two-way streets as the MSTRANS model logic was not general enough to handle one-way streets. Distances on the East-West axis were not given, so reasonable values were assumed.

For each network, we provide the following descriptive items:

- A network diagram illustrating the positions of the links and numbered nodes.
- Average flow rates at the source nodes
- Average turning proportions at the end of each link.

A3.2 Arterial Network

A3.2.1 Network diagram



A3.2.2 Flow rates

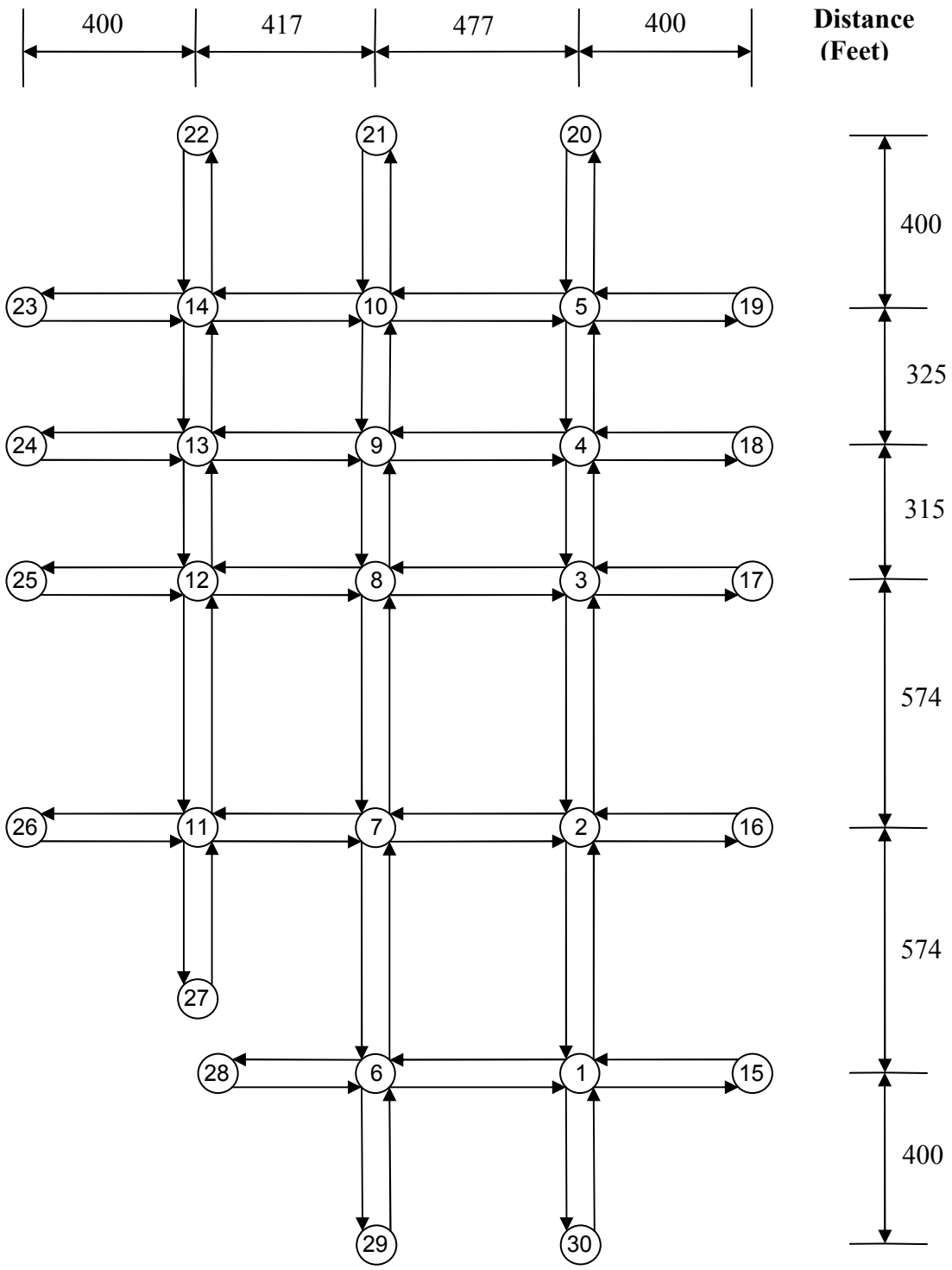
Source Node Number	Vehicles Per Hour
10	483
11	178
12	499
13	351
14	271
15	789
16	230
17	642
18	151
19	479
20	1197
21	712
22	117
23	687
24	590
25	1082
26	163
27	352
28	508
29	297

A3.2.3 Turning proportions

Upstream Node Number	Downstream Node Number	Percentage of Left Turning Traffic	Percentage of Through Traffic	Percentage of Right Turning Traffic
10	1	4%	87%	9%
11	1	45%	25%	30%
12	2	18%	49%	33%
13	3	47%	42%	11%
14	4	57%	22%	21%
15	5	29%	53%	18%
16	6	18%	29%	53%
17	7	25%	31%	44%
18	8	30%	23%	47%
19	9	25%	57%	18%
20	9	13%	82%	5%
21	9	35%	46%	19%
22	8	34%	38%	28%
23	7	24%	60%	16%
24	6	45%	8%	47%
25	5	26%	47%	27%
26	4	29%	39%	32%
27	3	23%	52%	25%
28	2	21%	51%	28%
29	1	20%	15%	65%
1	2	22%	71%	7%
2	3	6%	88%	6%
3	4	5%	92%	3%
4	5	14%	59%	27%
5	6	4%	84%	12%
6	7	22%	67%	11%
7	8	3%	95%	2%
8	9	10%	89%	1%
9	8	2%	96%	2%
8	7	12%	81%	7%
7	6	19%	80%	1%
6	5	16%	67%	17%
5	4	5%	84%	11%
4	3	12%	73%	15%
3	2	20%	68%	12%
2	1	18%	72%	10%

A3.3 Grid Network

A3.3.1 Network diagram



A3.3.2 Flow rates

Source Node Number	Vehicles Per Hour
15	304
16	94
17	297
18	207
19	952
20	780
21	108
22	861
23	1153
24	148
25	219
26	163
27	769
28	131
29	180
30	180

A3.3.3 Turning proportions

Upstream Node Number	Downstream Node Number	Percentage of Left Turning Traffic	Percentage of Through Traffic	Percentage of Right Turning Traffic
17	15	0%	100%	0%
18	16	17%	68%	15%
19	17	18%	64%	18%
20	18	26%	49%	25%
21	19	14%	72%	14%
22	20	0%	100%	0%
23	21	0%	95%	5%
24	22	17%	80%	3%
25	23	6%	89%	5%
26	24	22%	67%	11%
27	25	16%	81%	3%
28	26	0%	21%	79%
29	28	17%	72%	11%
30	27	0%	97%	3%
31	29	0%	82%	18%

32	30	0%	82%	18%
33	6	25%	20%	55%
34	7	10%	28%	62%
35	8	17%	83%	0%
36	9	17%	53%	30%
37	1	25%	20%	55%
38	2	10%	28%	62%
39	3	17%	83%	0%
40	4	17%	53%	30%
41	11	6%	80%	14%
42	12	18%	70%	12%
43	13	0%	86%	14%
44	5	0%	89%	11%
45	4	0%	93%	7%
46	3	0%	74%	26%
47	2	0%	100%	0%
48	10	18%	67%	15%
49	9	19%	81%	0%
50	8	0%	31%	69%
51	7	0%	93%	7%
52	14	10%	90%	0%
53	13	13%	87%	0%
54	12	25%	75%	0%
55	6	0%	100%	0%
56	11	4%	93%	3%
57	7	0%	100%	0%
58	12	11%	43%	46%
59	8	0%	97%	3%
60	13	10%	78%	12%
61	9	0%	97%	3%
62	14	5%	86%	9%
63	10	0%	100%	0%
64	1	0%	91%	9%
65	2	23%	6%	71%
66	7	27%	54%	19%
67	3	11%	70%	19%
68	8	17%	79%	4%
69	4	16%	84%	0%
70	9	22%	59%	19%
71	5	1%	99%	0%
72	10	0%	100%	0%

A4 Appendix Four – Formulae for evaluating magnitude of optimization problem

A4.1 Introduction

In this appendix, we provide derivation of formulae for computing the number of decision variables as well as the number of bits in binary coding when applying the genetic encoding given in section 7.6 for traffic signal optimization.

A4.2 Number of decision variables

A signal timing policy must specify the common cycle time for the entire network, *C*. For each of the N_S traffic signals besides the first, we specify an offset ϕ_i. For each signal we also specify the duration of the N/S green phase $\rho_{i,N/S}$. For each approach at each signal, we must specify the sequence of stages, $\delta_{i,k}$ and the duration of the stage with green indications for all traffic movements $g_{i,k}^1$. The other variables determining the characteristics of the traffic signal timing policy can be determined from these decision variables (see section 7.6.3). Thus the total number of decision variables N_d is given by:

$$\begin{aligned} N_d &= 1 + N_s(1+1+4\times 2) - 1 \\ &= 10N_s \end{aligned} \qquad \text{(A4-1)}$$

A4.3 Bit string length

Let l_C = number of bits for encoding the cycle time,

l_ϕ = number of bits for encoding an offset,

l_ρ = number of bits for encoding length of N/S green phase,

l_δ = number of bits for encoding a stage sequence,

l_g = number of bits for encoding duration of stage with green indications for all traffic movements.

From sections 7.6.2 and 7.6.3 we have $l_\delta = 2$. For the other decision variables, the number of bits used for representing each type are:

$l_C = 6$

$l_\phi = 6$

$l_\rho = 5$

$l_g = 4$

The total length of binary string in the binary encoding N_b is given by:

$$\begin{aligned} N_b &= l_C + N_s\left[l_\phi + l_\rho + 4\left(l_\delta + l_g\right)\right] - l_\phi \\ &= 35N_s \end{aligned} \qquad \text{(A4-2)}$$

A5 Appendix Five – Real CHC

A5.1 Introduction

In section 8.5.4 we consider a modified version of the CHC algorithm that we call Real CHC. In this appendix we discuss the amendments that were made to the standard CHC algorithm to arrive at Real CHC.

With real genetic operators, we no longer require a binary representation of the decision variables for the GGA and SSGA. However, with CHC, the incest prevention mechanisms permits recombination based on the Hamming distance between parent strings. Thus the binary representation is still required. We can omit the binary representation if we can obtain a distance measure between individuals using the real valued decision variables. With Real CHC, we overlook the binary representations for the real decision variables and measure distances using the L_1 norm. We need to account for distance between real valued decision variables as well as the variables specifying phasing which use a binary encoding. We first elaborate the calculation of distance between two individuals in the population. We then describe how the mating distance threshold is set initially. We then explain how decrements to the threshold are performed and finally, we give the formula for resetting the distance threshold after cataclysmic mutation. We describe the method in which these quantities are dealt with in the standard CHC algorithm as given by Eshelman [100] in order to make the extension to a real parameter distance intuitive.

A5.1.1 Distance calculation

In the standard CHC algorithm, distance between individuals is measured by the Hamming distance of the binary string encodings. Thus each bit is given an equal weighting. With a representation that combines real and binary parameters, we modify the distance metric by requiring that each decision variable be given an equal weighting. We compute the distance separately for each decision variable and standardize the distance to a value in the interval $[0,1]$, where 0 and 1 corresponding to the minimum and maximum distances. The distance between the individuals is the sum of the distances on each parameter.

For a real decision variable, let x and y denote the values of the decision variable in the problem space in the first and second individual. Let $D(x)$ and $D(y)$ denote the respective normalized values of these decision variables (see section 7.6.3). The distance is given by:

$$|D(x) - D(y)| \tag{A5-1}$$

The use of the normalized values of the decision variables ensures that:

- Distances in the range $[0,1]$ are attainable even when x and y have different domains (see section 7.7.2)
- Distance calculation will be consistent with the manner in which real crossover operates as real crossover operates on the normalized values (see section 7.7.2) e.g. $|D(x) - D(y)| = 0 \Rightarrow D(x) = D(y)$ and blend crossover will not be able to perform any exploration on this decision variable. There will be no contribution to the total distance from this decision variable and thus the two individuals will be less likely to reproduce according to the incest prevention mechanism.

For an l bit binary encoded decision variable, let $x_l x_{l-1}, \cdots, x_1$ and $y_l y_{l-1}, \cdots, y_1$ denote the binary coded values of the decision variable in the two individuals. The distance is given by:

$$\frac{1}{l}\sum_{i=1}^{l}|x_i - y_i| \tag{A5-2}$$

That is, the distance is taken as the Hamming distance expresses as a fraction of the number of bits used for representing that parameter. This ensures that the distance will be in the range $[0,1]$.

A5.1.2 Initial value of mating threshold

In the standard CHC algorithm, the distance mating threshold is initially set to $\frac{N_b}{2}$ where N_b is the length of the binary string encoding all decision variables. This is equal to the expected Hamming distance between two individuals in the initial population. We use this same criteria for setting the initial threshold in Real CHC i.e. we set the initial threshold to the expected distance between two individuals in the initial population.

For real decision variables, the expected distance for a single decision variable is

$$E\left[|X-Y| \mid X \sim U(0,1) \text{ and } Y \sim U(0,1) \text{ independently}\right]$$

$$= \int_0^1 \int_0^1 |x-y| dy dx$$

$$= \int_0^1 \left[\int_0^x (x-y) dy + \int_x^1 (y-x) dy \right] dx$$

$$= \int_0^1 \left[\left(xy - \frac{y^2}{2} \right) \Big|_{y=0}^{y=x} + \left(\frac{y^2}{2} - xy \right) \Big|_{y=x}^{y=1} \right] dx$$

$$= \int_0^1 \left(x^2 - x + \frac{1}{2} \right) dx$$

$$= \frac{1}{3} - \frac{1}{2} + \frac{1}{2}$$

$$= \frac{1}{3}$$

(A5-3)

Thus the threshold is initially set to:

$$\frac{N_{Real}}{3} + \frac{N_{Binary}}{2},$$

(A5-4)

where

N_{Real} = the number of real decision variables and

N_{Binary} = the number of decision variables that utilize a binary encoding.

A5.1.3 Decrements to mating threshold

In the standard CHC algorithm, the mating distance threshold is decreased by one whenever no individuals can be paired for mating or all offspring are discarded by the replacement mechanism. This is the minimum decrement that can be performed since Hamming distance is a non-negative integer value. With the distance measurement we have proposed for Real CHC, the distance threshold decrement can be set arbitrarily. To ensure consistent comparisons between

the standard CHC and Real CHC in section 8.5.4, we set the threshold decrement in the Real CHC so as to ensure that the same number of steps are required to reduce the threshold to 0 in both algorithms.

A5.1.4 Value of mating threshold after cataclysmic mutation

CHC reintroduces diversity into the population by performing cataclysmic mutation. A new population is created using the best individual. The best individual is retained and all remaining individuals are constructed by flipping bits in the best individual with probability *r*, the divergence rate. The mating distance threshold is then reset to $2r(1-r)N_b$.

Now for two individuals obtained via cataclysmic mutation,

Pr(Any bit differs) = Pr(Bit mutated in one individual and not the other)

$= 2r(1-r)$.

Thus, the mating distance threshold is reset to the average Hamming distance between two individuals (where neither of which is the best individual) immediately after cataclysmic mutation. The same logic will be applied to the Real CHC i.e. we set the threshold to the expected distance between two mutated individuals.

We first derive the following results for $X \sim U(0,1)$ and a constant $\kappa \in [0,1]$:

$$
\begin{aligned}
E[|X-\kappa|] &= \int_0^1 |x-\kappa| dx \\
&= \int_0^{\kappa} (\kappa - x) dx + \int_{\kappa}^1 (x-\kappa) dx \\
&= \left(\kappa x - \frac{x^2}{2} \right) \Big|_{x=0}^{x=\kappa} + \left(\frac{x^2}{2} - \kappa x \right) \Big|_{x=\kappa}^{x=1} \\
&= \kappa^2 - \kappa + \frac{1}{2}
\end{aligned}
\qquad \text{(A5-5)}
$$

Now when performing cataclysmic mutation with Real CHC, the real decision variables undergo real mutation with probability *r*. Bit values encoding binary genes are flipped with probability *r*.

Let κ_i denote the normalized value of the *i*'th real decision variable in the best individual. The expected distance contribution from the *i*'th real decision for two mutated individuals is:

Pr(Decision variable mutated in both individuals)$\times E\left[\left|X-Y\right| \mid X \sim U(0,1), Y \sim U(0,1)\right]$ +

Pr(Decision variable mutated in one individual but not the other) $\times E\left[\left|X-\kappa_i\right| \mid X \sim U(0,1)\right]$

$= \frac{r^2}{3} + 2r(1-r)\left(\kappa_i{}^2 - \kappa_i + \frac{1}{2}\right)$ (See equations (A5-3) and (A5-5))

Summing the expected distances over all real and binary encoded decision variables, the distance threshold is reset to:

$$\frac{N_{Real}r^2}{3} + 2r(1-r)\left[\sum_{i=1}^{N_{Real}}\left(\kappa_i^2 - \kappa_i + \frac{1}{2}\right) + N_{Binary}\right] \quad \text{(A5-6)}$$

A6 Appendix Six – Statistical model for comparing genetic algorithm performance

A6.1 Introduction

In this appendix, we provide the derivation and formulation of the statistical model for testing for significant differences in mean search performance of different genetic algorithms. The need for this model is motivated and the sources of variability to be addressed are given in section 7.9.6. We first elaborate on the experiments used to determining the assumptions of the model. We then present the model and discuss how it can be applied.

A6.2 Setting assumptions of statistical model

A6.2.1 Variability in search trajectories

In order to assess the manner in which the performance of the best individual varies between runs of a genetic algorithm, 100 independent runs of a particular genetic algorithm were performed. The specifications of the algorithm are given in Table 22.

Algorithm Type	SSGA
Population size, selection bias factor and probability of mutation	See section 7.8
Number of replications of objective function for fitness evaluations	1
Use CRN's	Yes
Representation	Standard binary
Crossover and mutation operators	Standard binary operators

Table 22: Specifications of genetic algorithm for evaluating variability in search trajectories

This genetic algorithm was applied to the undersaturated arterial test network scenario and the signal timing policy of the best individual after 2000 function evaluations was saved in each case.

Let

$X_{i,j}$ = extended network delay on *j*'th independent replication of the best individual after 2000 function evaluations on the *i*'th replication of the genetic algorithm in Table 22 (seconds/vehicle) and

θ_i = population mean of extended network delay for the best individual after 2000 function evaluations on the *i*'th replication of the genetic algorithm in Table 22 (seconds/vehicle).

For each *i*, we would like to obtain a precise measure of θ_i so that we can assess the distribution of the $\{\theta_i\}$. 20 Independent replications of MSTRANS were performed using the signal timing plans of the best solution for each *i*. The number of additional replications to perform to ensure that the sample mean of extended network delay was within 2 seconds of θ_i with 95% confidence was estimates as $n_i^* - 20$, where

$$n_i^* = \min\left\{ j \geq 20 \;\middle|\; t_{n-1,0.975}\sqrt{\frac{S_i^2}{j}} \leq 1 \right\}, \tag{A6-1}$$

$t_{n-1,1-\alpha} = 100(1-\alpha)$ Percentile of the t-distribution with $n-1$ degrees of freedom,

$\overline{X_i} = \frac{1}{20}\sum_{j=1}^{20} X_{i,j}$ and

$S_i^2 = \frac{1}{19}\sum_{j=1}^{20}\left(X_{i,j} - \overline{X_i}\right)^2$ (See Law and Kelton [136]).

The additional replications were performed and the sample mean and sample variance of extended network delay based on a total of n_i^* replications are given for each *i* in Table 23.

i	Sample mean	Sample variance
1	53.61	22.83
2	57.50	13.23
3	58.19	46.35
4	67.10	104.56
5	55.28	48.96
6	57.18	63.25
7	54.89	38.57
8	63.17	50.79
9	66.55	42.61
10	54.01	51.70
11	58.88	45.82
12	60.58	152.62
13	70.99	118.57
14	59.63	13.59
15	52.72	29.00
16	56.87	59.68
17	55.33	41.31
18	57.81	23.21
19	51.75	31.62
20	66.53	102.30
21	53.65	18.86
22	53.05	13.92
23	57.18	80.21
24	54.61	36.14
25	49.44	43.01
26	55.72	24.45
27	53.15	33.89
28	51.76	4.53
29	57.46	80.77
30	78.75	146.76
31	51.37	37.15
32	60.61	81.29
33	62.00	104.36
34	53.41	38.73
35	56.62	87.37

i	Sample mean	Sample variance
51	51.78	21.84
52	57.74	91.68
53	61.54	48.67
54	51.62	24.83
55	58.70	102.72
56	56.51	54.66
57	51.04	19.31
58	70.22	145.56
59	55.42	24.28
60	54.21	12.16
61	53.79	38.33
62	59.24	76.39
63	54.64	30.78
64	56.02	62.10
65	52.04	12.65
66	55.80	50.59
67	51.95	25.67
68	55.50	19.57
69	62.03	76.89
70	62.14	14.54
71	66.81	146.31
72	52.97	17.50
73	53.32	44.66
74	55.64	23.19
75	61.72	76.09
76	54.21	3.62
77	59.77	70.05
78	57.30	50.37
79	54.40	18.88
80	50.73	21.87
81	50.68	16.99
82	63.53	29.10
83	50.72	36.34
84	64.67	60.88
85	53.51	48.23

36	59.90	22.20	86	50.97	35.38
37	55.83	17.69	87	60.35	17.87
38	54.59	60.94	88	49.61	27.19
39	57.89	57.75	89	52.40	55.14
40	56.15	57.16	90	53.78	17.93
41	55.77	32.41	91	58.11	42.40
42	55.18	25.78	92	52.75	9.59
43	53.65	68.03	93	55.37	42.57
44	64.51	81.32	94	52.68	18.87
45	60.11	62.74	95	52.03	18.84
46	50.31	10.00	96	54.18	58.58
47	57.71	51.57	97	49.31	10.77
48	59.83	70.25	98	57.46	43.15
49	55.80	65.89	99	53.01	44.65
50	52.84	22.41	100	60.04	39.91

Table 23: Sample mean and variance of extended network delay over 100 independent replications of genetic algorithm

The sample means can be taken as accurate approximations of the θ_i. A histogram of these sample means is given in Figure 25.

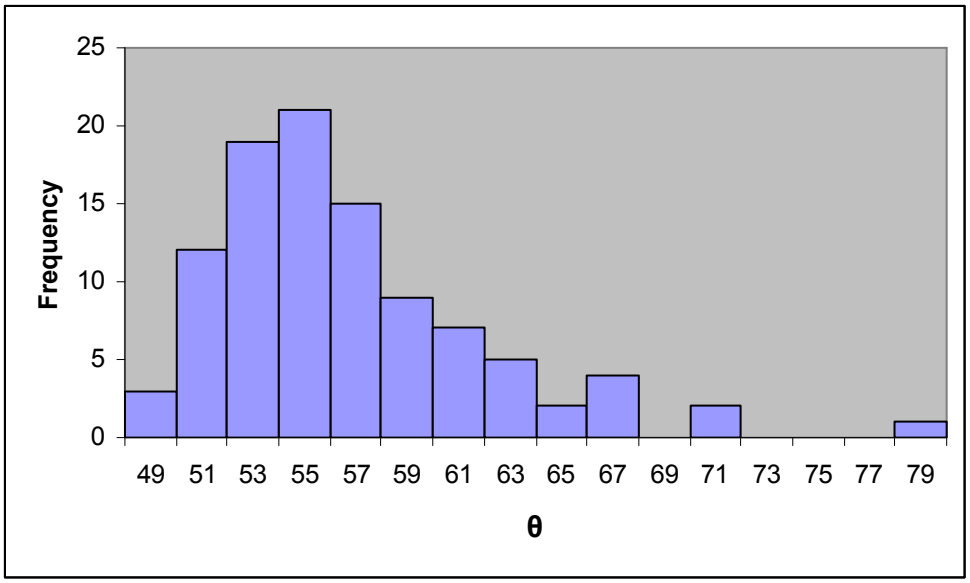


Figure 25: Histogram of θ_i

Several standard continuous distributions including the Normal, Log-Normal and Gamma were applied and none gave an adequate description to the θ_i. Based on the findings in A6.2.2, it was discovered that a translated distribution was required. The Translated Gamma distribution which has probability density function

$$\pi(\theta) = \frac{\left(\frac{\alpha}{\mu-\tau}\right)^2}{\Gamma(\alpha)}(\theta-\tau)^{\alpha-1}e^{-\frac{\alpha}{\mu-\tau}(\theta-\tau)}, \theta > \tau, \alpha > 0, \mu > \tau, \quad \text{(A6-2)}$$

was found to give a satisfactory fit to the θ_i when evaluated via a chi-squared goodness of fit test. The parameterization given in equation (A6-2) elicits the mean of the distribution as an explicit parameter.

A6.2.2 Variability in the objective function

Examining the sample variance of the different solutions in Table 23, we see that the variance of extended network delay differs for different signal timing policies. A scatter plot of the sample mean and sample variance values is given in Figure 26.

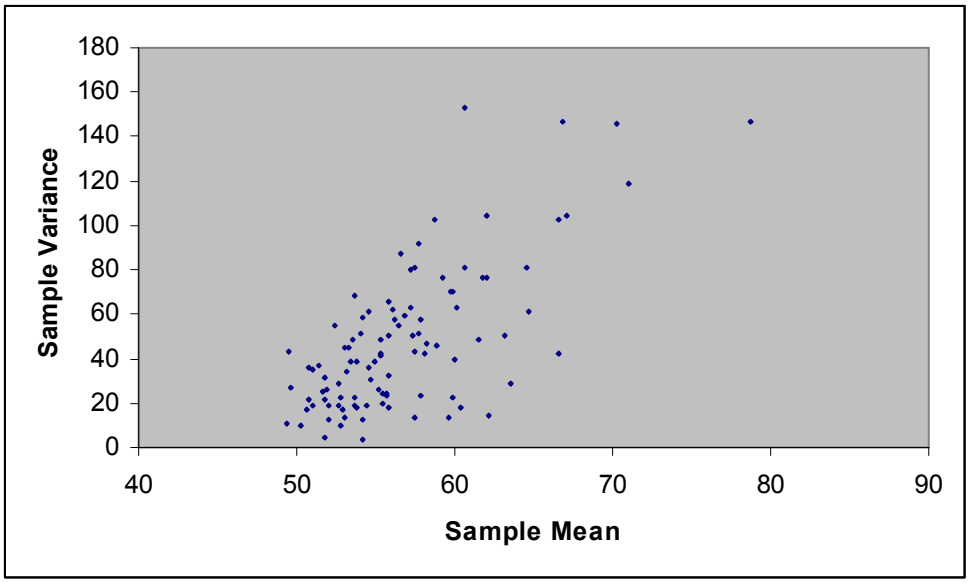


Figure 26: Scatter plot of sample mean and sample variance values in Table 23

We note an approximate linear trend with sample variance increasing linearly with the sample mean. A linear trend was also noted in the same mean and same variance of extended network delay in the combined data generated for comparing the genetic algorithms in all four test network scenarios. Thus, a linear model for variance was proposed i.e.

$Var(X_{i,j}) = a\theta_i + b$.

(A6-3)

To ensure variance to always be non-negative, we require $\theta_i \geq -\frac{b}{a}$. Clearly, from Figure 26, we have $a > 0$ and $b < 0$. Thus, the distribution describing θ_i as described in section A6.2.1 must have support on the positive real numbers, with minimum value $-\frac{b}{a} > 0$. Thus a translated distribution such as the Translated Gamma is appropriate.

A6.2.3 Normally distributed objective function

To assess whether extended network delay follows a normal distribution, the signal timing policy of the best individual after 2000 function evaluations on the first iteration of the genetic algorithm in A6.2.1 was selected. 100 Independent observations of extended network delay were

made for this set of signal timings. A histogram of these observations is given in Figure 27. A normal distribution was found to give an adequate description of extended network delay based on the chi-squared goodness of fit test.

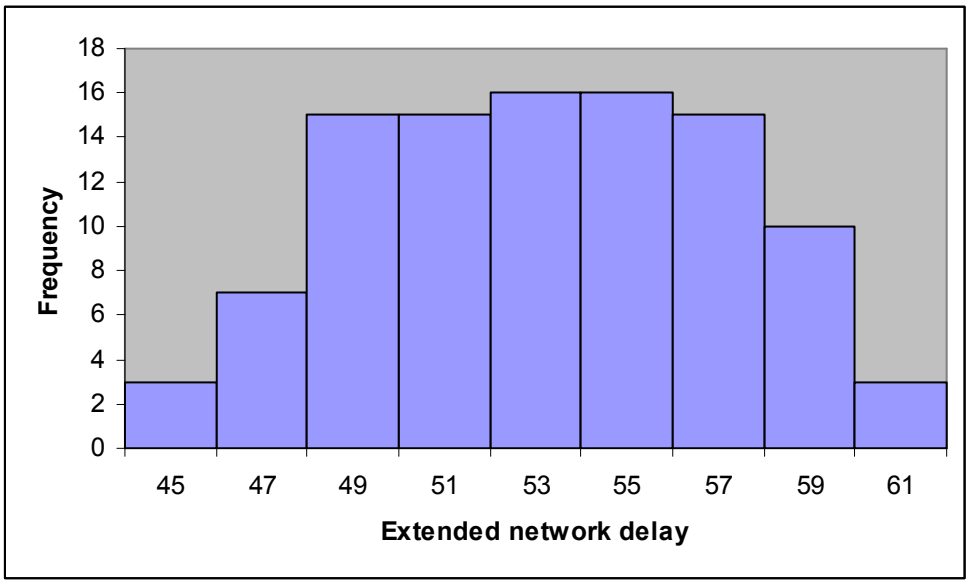


Figure 27: Histogram of extended network delay

A6.3 Statistical model

A6.3.1 Model describing distribution of observed performance measures

For each genetic algorithm applied, we perform repeated independent runs of the genetic algorithm. We also record the best individual generated at intervals of 1000 function evaluations and perform independent replications of MSTRANS for evaluating the quality of this individual. For the purpose of defining the statistical model, we use the notation introduced in section 7.9.5.

Based on the findings in section A6.2, the following underlying statistical model is proposed for describing the observed performance measure of the best individual after F objective function evaluations of genetic algorithm X on a particular network scenario:

Genetic algorithm run	**Replication of best individual**
1	$X_{1,F,1},\quad X_{1,F,2},\quad \cdots\cdots\quad X_{1,F,n}\quad ,\ X_{1,F,j} \mid \theta_1 \sim N(\theta_1, a\theta_1 + b)$
$\vdots$	
i	$X_{i,F,1},\quad X_{i,F,2},\quad \cdots\cdots\quad X_{i,F,n}\quad ,\ X_{i,F,j} \mid \theta_i \sim N(\theta_i, a\theta_i + b)$
$\vdots$	
N	$X_{N,F,1},\quad X_{N,F,2},\quad \cdots\cdots\quad X_{N,F,n}\quad ,\ X_{N,F,j} \mid \theta_N \sim N(\theta_N, a\theta_N + b)$

$$\theta_i \sim \Gamma(\alpha_{X,F}, \mu_{X,F}, \tau_{X,F})$$ [76]

Figure 28: Statistical model describing distribution of observed performance measures

For the sake of clarity, we now omit the F subscript in all derivations that follow, keeping in mind that we are concerned with the output of the best individual after F generations.

A6.3.2 Estimation of model parameters

Assuming that estimates of a and b are available, the parameters of the translated gamma distribution in Figure 28 can be estimated via maximum likelihood. We omit the X subscript from the parameters of the translated gamma distribution. Let

$x_{i,j}$ = realized or observed values of the $X_{i,j}$,

$\underline{x_i}$ = $(x_{i,1}, x_{i,2}, \cdots, x_{i,n})^T$,

X = $(\underline{x_1}, \underline{x_2}, \cdots, \underline{x_N})$ and

$L_i(\underline{x_i}, \alpha, \mu, \tau)$ = likelihood of observing the sample $\underline{x_i}$

$$= \int_{\forall\theta} \left[\prod_{j=1}^{n} f(x_{i,j} \mid \theta) \right] \pi(\theta) d\theta,$$

[76] $\Gamma(\alpha, \mu, \tau)$ denotes the Translated Gamma distribution as defined in equation (A6-2).

$l_i\left(\underline{x_i},\alpha,\mu,\tau\right)$ = log-likelihood of sample $\underline{x_i}$

$= \log L_i\left(\underline{x_i},\alpha,\mu,\tau\right)$ and

$l\left(X,\alpha,\mu,\tau\right)$ = total log-likelihood of the sample X

$= \sum_{i=1}^{N} l_i\left(\underline{x_i},\alpha,\mu,\tau\right)$

where

$f\left(x_{i,j}\mid\theta\right) = \frac{1}{\sqrt{2\pi\left(a\theta+b\right)}} e^{-\frac{1}{2\left(a\theta+b\right)}\left(x_{i,j}-\theta\right)^2}$ and

$\pi\left(\theta\right) = \frac{\left(\frac{\alpha}{\mu-\tau}\right)^2}{\Gamma\left(\alpha\right)}\left(\theta-\tau\right)^{\alpha-1} e^{-\frac{\alpha}{\mu-\tau}\left(\theta-\tau\right)}$.

Now $L_i\left(\underline{x_i},\alpha,\mu,\tau\right) = \frac{\left(\frac{1}{2\pi}\right)^{\frac{n}{2}}\left(\frac{\alpha}{\mu-\tau}\right)^{\alpha}}{\Gamma\left(\alpha\right)} \int_{\tau}^{\infty}\left(\frac{1}{a\theta+b}\right)^{\frac{n}{2}}\left(\theta-\tau\right)^{\alpha-1} e^{-\left[\frac{1}{2\left(a\theta+b\right)}\sum_{j=1}^{n}\left(x_{i,j}-\theta\right)^2+\left(\frac{\alpha}{\mu-\tau}\right)\left(\theta-\tau\right)\right]} d\theta$

Using the change of variable $z=\theta-\tau$ in the integral, we get

$L_i\left(\underline{x_i},\alpha,\mu,\tau\right) = \frac{\left(\frac{1}{2\pi}\right)^{\frac{n}{2}}\left(\frac{\alpha}{\mu-\tau}\right)^{\alpha}}{\Gamma\left(\alpha\right)} I\left(\alpha,\mu,\tau,\sum_{j=1}^{n}x_{i,j},\sum_{j=1}^{n}{x_{i,j}}^2\right)$

where

$I\left(\alpha,\mu,\tau,c,d\right) = \int_{0}^{\infty} z^{\alpha-1}\left(az+a\tau+b\right)^{-\frac{n}{2}} e^{-\left[\frac{1}{2\left(az+a\tau+b\right)}\left\{d-2\left(z+\tau\right)c+n\left(z+\tau\right)^2\right\}+\frac{\alpha}{\mu-\tau}z\right]} dz$.

The integral $I\left(\alpha,\mu,\tau,c,d\right)$ can be evaluated numerically.

Thus, $l_i\left(\underline{x_i},\alpha,\mu,\tau\right) = \alpha\left[\log\alpha-\log\left(\mu-\tau\right)\right]-\frac{n}{2}\log\left(2\pi\right)-\log\Gamma\left(\alpha\right)-\log I\left(\alpha,\mu,\tau,\sum_{j=1}^{n}x_{i,j},\sum_{j=1}^{n}{x_{i,j}}^2\right)$

and

$$l(X,\alpha,\mu,\tau) = N\left\{\alpha[\log\alpha - \log(\mu-\tau)] - \frac{n}{2}\log(2\pi) - \log\Gamma(\alpha)\right\} - \sum_{i=1}^{N}\log I\left(\alpha,\mu,\tau,\sum_{j=1}^{n}x_{i,j},\sum_{j=1}^{n}x_{i,j}\right)$$

(A6-4)

The parameters of the model in Figure 28 can be estimated by maximizing the log likelihood in equation (A6-4) numerically.

A6.3.3 Testing for significant differences in performance

The proposed model can be used for testing for significant differences in mean performance of the best individual after *F* function evaluations of two alternative genetic algorithms. We assume that the alternative genetic algorithms, *X* and *Y* follow the model Figure 28. The mean performance of the two genetic algorithms can be tested for statistically significant difference after *F* function evaluations by testing the null hypotheses $H_0 : \mu_X = \mu_Y = \mu$ versus the alternative hypothesis $H_1 : \mu_X \neq \mu_Y$.

We construct a test using the theory of the likelihood ratio test [144]. Let

$$\chi^2 = 2\left\{\max_{\alpha_X,\mu_X,\tau_X} l(X,\alpha_X,\mu_X,\tau_X) + \max_{\alpha_Y,\mu_Y,\tau_Y} l(Y,\alpha_Y,\mu_Y,\tau_Y) - \max_{\alpha_X,\alpha_Y,\mu,\tau_X,\tau_Y}\left[l(X,\alpha_X,\mu,\tau_X) + l(Y,\alpha_Y,\mu,\tau_Y)\right]\right\}$$

(A6-5)

Under H_0, the test statistic χ^2 will approximately follow a χ_1^2 distribution. Large values of χ^2 indicate a deviation from H_0. We use the tail probability of the χ_1^2 distribution for evaluating the p-value.

A6.4 Estimation of *a* and *b*

In section A6.2.2, we noted that the sample variance of extended network delay of different signal timing policies varies approximately as a linear function of the sample mean of extended network delay. The model in equation (A6-3) and Figure 28 describing the population variance of extended network delay as a linear function of the population mean of extended network delay

with slope *a* and intercept *b* was thus proposed. We require estimates of these parameters for applying the statistical model in section A6.3[77].

Many runs of alternative genetic algorithms were performed in the empirical work (see chapter 8) and the sample mean and sample variance of extended network delay of the best individuals at different number of function evaluations were computed. We can use this output to estimate *a* and *b* for each test network scenario.

The model in section A6.3 can be used to estimate *a* and *b* via maximum likelihood by including the parameters *a* and *b* as unknowns in the likelihood equation (A6-4) and maximizing the combined likelihood of the entire sample of alternative genetic algorithms with output at differing total function evaluations. However, this would be a huge optimization task. Three separate parameters (the parameters of the Translated Gamma distribution) would be required for each genetic algorithm. Separate sets of parameters are also required at each interval of output (output is given at 1000, 2000, …, 9000 and 10000 function evaluations). All these parameters in conjunction with *a* and *b* would need to be simultaneously optimized.

We employed a simpler approach for the estimation. Let

m = number of paired values of sample mean and sample variance of extended network delay, from all genetic algorithms and all output intervals

n = number of independent replications of best individual performed at each output interval for each genetic algorithm,

x_i = sample mean of extended network delay of the *i*'th sample based on the *n* independent replications,

y_i = sample variance of extended network delay of the *i*'th sample based on the *n* independent replications,

θ_i = population mean of extended network delay of the *i*'th sample and

$\underline{y}$ = $(y_1, y_2, \cdots, y_m)$

[77] The linear variance model applies to a particular traffic network scenario. Since we have four test network scenarios, a separate estimation is required for each network setup.

A simple linear regression of $\{y_i\}$ on $\{x_i\}$ would offer a straightforward approach for estimating *a* and *b*. However, we need to account for differing levels of variability in the response variable i.e. $Var(y_i)$ is not constant. We demonstrate this below:

Since extended network delay has a normal distribution, we have:

$$y_i \sim \frac{(a\theta_i + b)}{n-1}\chi^2_{n-1} \quad \text{(see footnote[78])}.$$

(A6-6)

For *n* large, we can approximate the Chi-Squared distribution in equation (A6-6) with a normal distribution with the same moment's i.e.

$$y_i \sim N\left(a\theta_i + b, \frac{2(a\theta_i + b)^2}{n-1}\right).$$

(A6-7)

Thus, sample values with larger θ_i will exhibit larger variability in the response y_i. These observations should have less influence in the estimation of *a* and *b*. Approximating θ_i with x_i, we have

$$y_i \sim N\left(ax_i + b, \frac{2(ax_i + b)^2}{n-1}\right).$$

(A6-8)

The likelihood of the entire sample is given by

$$L(\underline{y}, a, b) = \prod_{i=1}^{m}\left\{\sqrt{\frac{n-1}{4\pi}}\frac{1}{ax_i + b}e^{-\frac{n-1}{4(ax_i+b)^2}[y_i-(ax_i+b)]^2}\right\}.$$

(A6-9)

The log-likelihood is thus

$$\begin{aligned} l(\underline{y}, a, b) &= \log L(\underline{y}, a, b) \\ &= \sum_{i=1}^{m}\left\{k - \log(ax_i + b) - \frac{n-1}{4(ax_i + b)^2}[y_i - (ax_i + b)]^2\right\}, \end{aligned}$$

---

[78] If $x_1, x_2, \ldots, x_n \sim N(\mu, \sigma^2)$ independently with $\bar{x} = \sum_{i=1}^{n} x_i$, then $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \sim \frac{\sigma^2}{n-1}\chi^2_{n-1}$

(A6-10)

where k denotes a constant independent of a and b. We can maximize the log-likelihood in equation (A6-10) numerically to obtain maximum likelihood estimates of a and b for each test network

Since tests for statistically significant differences in performance were only performed at 5000 and 10000 total function evaluations, the sample output of the best individual at 5000, 6000, 7000, 8000, 9000 and 10000 total function evaluations of each algorithm was used in the estimation of a and b. Furthermore, we only used the output from the runs detailed in section 8.2 and 8.3 for the estimation. The estimated values of a and b for each test network are given in Table 24. In Figures 29-32, we plot the sample values used in the estimation along with the estimated linear variance model for each test network.

Test network	***a***	***b***
Arterial Undersaturated	2.49865	-88.88926
Arterial Oversaturated	1.98873	-115.96436
Grid Undersaturated	1.04285	-35.65330
Grid Oversaturated	1.38381	-60.51166

Table 24: Estimated values of a and b for each test network

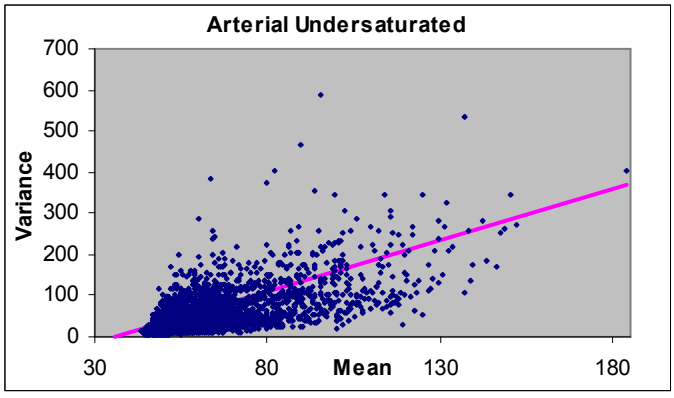


Figure 29: Fit of linear variance model – Arterial Undersaturated scenario

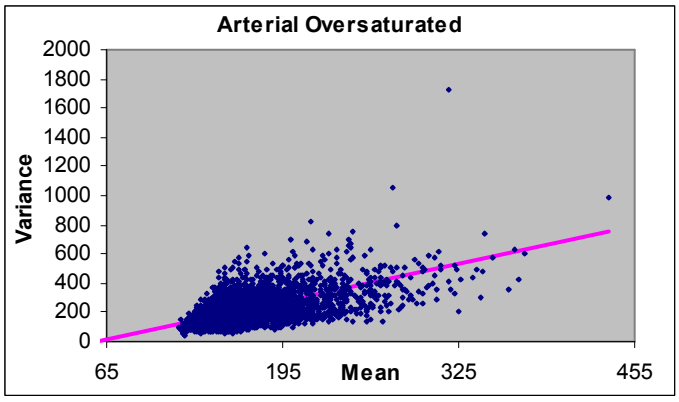


Figure 30: Fit of linear variance model – Arterial Oversaturated scenario

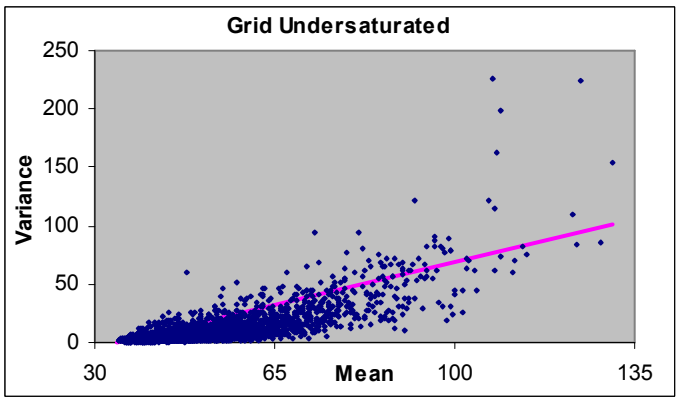


Figure 31: Fit of linear variance model – Grid Undersaturated scenario

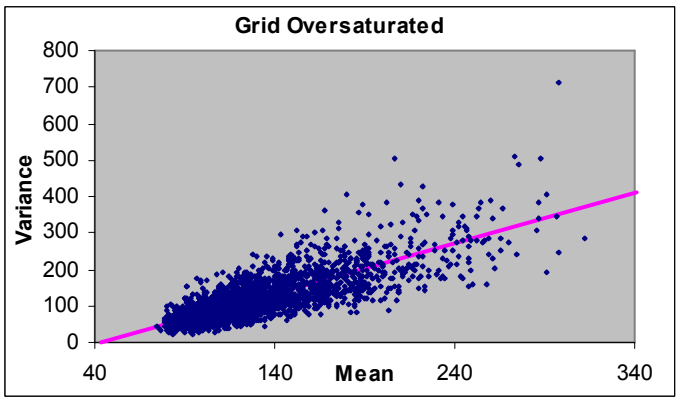


Figure 32: Fit of linear variance model – Grid Oversaturated scenario

A7 Appendix Seven – Experimental Output

A7.1 Introduction

In this appendix, we present more detailed output for the various genetic algorithms runs discussed in chapter 8. For each genetic algorithm X, the sample values of X_F are given for $F \in \{0, 1000, 2000, \ldots, 10000\}$. The same section headings as those in chapter 8 are applied so that the output can be easily identified.

A7.2 Re-evaluation of fitness in GGA's

Arterial Undersaturated

Function Evaluations	Without re-evaluation				With re-evaluation			
	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	447.44	400.51	431.66	396.78	413.45	400.61	429.46	407.13
1000	77.22	119.78	165.28	253.02	96.44	120.30	172.07	242.76
2000	66.32	80.33	102.36	172.90	72.39	83.60	116.80	181.66
3000	63.70	71.17	85.73	126.99	66.47	70.93	97.40	142.78
4000	59.75	67.39	76.47	105.21	65.38	63.74	82.13	112.12
5000	59.34	64.12	70.48	89.32	61.25	61.43	71.95	98.11
6000	58.87	62.29	65.01	81.41	59.95	60.17	69.41	89.59
7000	58.45	60.43	61.65	76.51	59.46	57.08	66.93	84.86
8000	57.34	59.92	60.66	72.89	57.92	57.20	65.47	79.19
9000	55.88	58.78	59.82	69.67	55.79	55.90	61.12	74.14
10000	55.59	59.10	58.75	68.46	55.43	56.79	61.01	71.55

Grid Undersaturated

Function Evaluations	Without re-evaluation				With re-evaluation			
	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	355.95	354.85	349.80	388.16	394.83	390.25	415.92	386.16
1000	66.32	94.42	149.89	229.54	65.37	93.80	158.45	229.17
2000	53.79	65.74	89.38	162.28	52.76	68.70	100.07	152.22
3000	50.00	57.24	71.41	123.39	48.64	56.43	78.33	123.89
4000	49.26	52.43	63.08	93.04	47.63	52.19	66.74	100.61
5000	48.33	50.36	57.68	81.27	46.88	49.95	60.78	84.24
6000	46.88	48.66	54.59	73.47	44.91	48.32	57.27	75.09
7000	46.05	47.71	51.68	69.04	44.27	48.27	55.18	71.36
8000	45.80	46.99	50.04	65.72	45.01	45.85	52.24	67.63
9000	45.75	46.56	48.88	62.13	44.05	45.56	50.94	63.26
10000	45.48	46.47	47.91	58.86	44.73	45.78	49.11	61.28

Arterial Oversaturated

Function Evaluations	Without re-evaluation				With re-evaluation			
	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	841.21	793.88	790.77	853.52	753.25	826.88	847.31	814.80
1000	211.94	268.05	381.96	533.43	228.59	274.81	409.06	509.36
2000	186.48	209.54	261.71	413.19	184.99	211.86	276.82	388.91
3000	182.21	188.61	227.55	313.63	180.04	186.90	231.92	321.25
4000	175.22	181.08	211.50	262.91	178.79	175.68	205.82	284.07
5000	172.36	173.71	201.56	233.20	174.44	170.42	191.97	250.95
6000	171.24	171.17	194.03	213.23	166.90	168.01	184.15	231.44
7000	170.09	165.38	188.96	202.74	165.33	168.72	176.93	221.56
8000	168.41	163.54	184.47	194.50	159.98	162.80	174.30	208.58
9000	166.97	162.70	181.02	187.36	158.64	160.60	170.06	200.56
10000	166.40	160.34	178.22	180.23	157.94	160.83	171.19	194.25

Grid Oversaturated

Function Evaluations	Without re-evaluation				With re-evaluation			
	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	784.14	703.22	764.53	797.65	704.36	720.05	779.98	714.87
1000	174.48	238.05	351.96	497.38	180.17	231.84	346.44	468.05
2000	141.93	171.20	223.70	359.18	137.48	158.54	235.30	347.59
3000	130.85	141.99	180.69	282.84	121.29	136.97	184.91	282.61
4000	123.41	126.95	153.71	237.69	117.70	125.38	159.65	243.48
5000	117.92	123.00	136.69	197.01	113.85	119.08	139.76	210.26
6000	116.40	119.28	126.46	177.23	111.00	116.92	132.01	187.98
7000	116.58	118.08	123.03	161.79	109.44	112.94	123.87	169.41
8000	115.66	116.82	119.79	152.71	109.76	114.82	118.55	161.31
9000	114.74	115.34	115.27	144.63	110.45	111.92	115.95	152.98
10000	114.37	113.10	114.55	137.37	106.46	110.12	112.33	146.66

A7.3 Common Random Numbers

The results presented in this section are for the execution of the genetic algorithm with CRN's for fitness evaluations. The results with independent replications for fitness evaluations are given in section A7.2.

A7.3.1 GGA

Arterial Undersaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	403.26	432.99	469.00	455.08
1000	77.45	112.52	186.26	266.05
2000	61.01	75.77	119.24	204.56
3000	56.66	65.30	92.01	139.62
4000	54.43	59.91	77.06	111.94
5000	52.70	57.16	70.04	95.51
6000	52.26	55.05	65.65	87.49
7000	52.32	53.52	63.26	80.07
8000	53.02	53.83	61.48	74.93
9000	51.91	53.45	59.51	70.98
10000	52.22	53.05	58.41	67.78

Grid Undersaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	366.80	364.33	378.62	370.74
1000	63.98	99.58	157.91	223.76
2000	52.14	64.38	93.04	143.01
3000	47.05	55.45	74.47	109.00
4000	44.69	51.00	63.03	86.22
5000	43.96	48.14	57.69	75.48
6000	42.90	45.59	54.13	69.25
7000	42.17	45.22	51.32	64.11
8000	41.77	43.59	49.72	59.83
9000	41.64	42.62	47.79	58.56
10000	41.39	42.29	47.00	56.33

Arterial Oversaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	867.08	928.18	764.42	814.71
1000	207.71	257.59	387.53	535.34
2000	175.29	201.43	275.47	381.82
3000	164.54	184.29	226.79	298.06
4000	159.96	174.70	202.15	261.23
5000	155.02	166.47	188.97	232.65
6000	154.27	162.98	179.84	218.46
7000	153.38	157.09	175.68	207.39
8000	152.97	154.46	169.21	198.80
9000	153.50	153.37	164.82	190.58
10000	151.71	152.45	163.76	185.07

Grid Oversaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	753.70	815.08	763.99	779.38
1000	158.33	236.77	376.73	481.75
2000	118.65	149.10	254.40	377.45
3000	110.03	126.46	191.83	281.49
4000	105.95	115.71	157.07	232.34
5000	104.17	109.85	136.40	200.68
6000	101.96	106.66	124.03	180.17
7000	99.58	103.92	118.08	165.42
8000	98.71	102.76	110.98	152.71
9000	97.58	100.80	108.96	141.78
10000	97.26	99.11	106.78	134.30

A7.3.2 SSGA

Arterial Undersaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	370.98	397.01	350.92	387.07
1000	68.63	110.09	159.02	287.13
2000	55.75	68.50	101.99	189.47
3000	52.59	58.58	77.29	129.37
4000	51.25	55.57	64.94	99.01
5000	51.14	53.11	59.88	86.29
6000	51.73	51.12	56.20	77.50
7000	50.63	51.23	54.69	70.91
8000	51.09	50.69	52.48	65.52
9000	50.36	50.31	50.84	63.26
10000	50.68	50.14	49.88	60.02

Grid Undersaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	342.80	303.76	322.82	333.78
1000	54.68	81.48	126.69	220.85
2000	44.60	56.02	78.92	133.70
3000	40.69	48.52	62.22	98.86
4000	39.17	44.42	55.42	80.02
5000	38.45	42.29	51.47	68.38
6000	38.12	41.00	48.42	61.57
7000	38.01	39.95	46.11	57.54
8000	37.75	39.33	44.37	54.94
9000	37.91	38.68	43.46	52.58
10000	37.88	38.53	42.02	50.23

Arterial Oversaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	803.00	811.36	754.83	698.50
1000	188.07	253.94	393.70	535.03
2000	163.24	185.89	245.62	372.41
3000	157.29	168.26	203.20	287.27
4000	151.20	160.26	181.72	250.29
5000	149.80	152.88	172.04	213.43
6000	150.33	149.26	161.51	196.30
7000	149.27	146.74	158.55	183.78
8000	150.20	145.20	154.50	178.00
9000	148.81	142.67	152.38	172.22
10000	148.71	143.60	150.23	165.86

Grid Oversaturated

Function Evaluations	Num Replications			
	1	2	4	8
0	674.14	722.24	675.84	679.27
1000	145.23	207.26	322.90	503.04
2000	110.82	133.29	208.18	371.03
3000	99.97	115.74	160.88	273.49
4000	95.32	104.87	138.62	226.85
5000	93.42	99.69	126.31	190.74
6000	92.77	97.13	118.50	169.86
7000	91.99	94.26	111.75	153.28
8000	90.99	92.88	106.04	138.65
9000	90.94	91.09	102.95	130.93
10000	91.37	89.94	99.69	124.15

A7.3.3 CHC

Arterial Undersaturated **Grid Undersaturated**

Function Evaluations	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	368.13	384.46	382.03	378.58	353.73	314.80	358.04	340.41
1000	89.88	142.44	197.34	290.68	73.05	122.50	172.02	257.94
2000	60.90	85.24	137.53	210.89	51.04	72.67	106.69	178.01
3000	55.59	66.59	102.50	173.05	44.22	57.30	84.17	131.76
4000	53.15	59.55	86.51	146.84	40.91	51.06	71.66	110.09
5000	51.26	54.01	72.70	121.82	39.16	46.56	63.54	96.58
6000	50.35	52.53	65.34	109.24	37.74	44.24	57.05	87.76
7000	50.15	51.08	61.00	96.93	37.40	41.62	54.17	72.85
8000	49.16	50.07	56.45	86.63	37.45	39.89	51.05	67.90
9000	49.37	49.38	54.70	77.33	37.24	38.69	47.72	64.65
10000	50.28	48.89	53.60	73.88	37.24	37.76	46.08	58.36

Arterial Oversaturated **Grid Oversaturated**

Function Evaluations	Num Replications				Num Replications			
	1	2	4	8	1	2	4	8
0	774.02	792.16	742.64	784.23	725.23	693.31	746.56	620.85
1000	226.65	314.36	447.04	580.21	188.65	278.77	419.62	511.41
2000	169.37	230.25	314.57	476.67	123.37	173.21	292.11	388.47
3000	152.92	189.24	255.15	392.27	106.37	137.73	228.33	333.08
4000	147.58	169.93	218.20	326.13	98.87	117.67	186.20	273.92
5000	143.39	157.59	190.96	301.29	92.96	107.53	156.03	234.31
6000	140.96	152.30	182.67	278.51	90.18	100.62	141.90	202.48
7000	139.94	147.92	173.17	251.02	89.04	97.14	132.59	184.37
8000	139.20	144.33	165.01	230.03	89.24	92.40	123.24	168.69
9000	138.94	143.09	162.00	218.02	88.46	90.38	117.38	163.34
10000	138.27	140.10	155.76	204.29	88.35	89.05	110.77	148.39

A7.4 Number of replications

See output in section A7.3.

A7.5 Alternative problem encodings

A7.5.1 Gray coding

The results given in this section are for the genetic algorithms with Gray coding. The results using the standard binary encoding are given in section A7.3.

Arterial Undersaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	387.54	334.30	346.47
1000	77.37	69.87	97.47
2000	59.68	56.58	64.93
3000	57.69	52.72	56.68
4000	54.69	51.46	52.37
5000	53.90	51.22	51.87
6000	52.51	50.52	51.37
7000	53.15	50.12	50.74
8000	53.06	50.95	51.09
9000	53.09	49.70	51.25
10000	52.56	50.30	50.70

Grid Undersaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	328.75	346.11	332.85
1000	65.30	55.80	71.55
2000	50.83	44.90	48.42
3000	46.03	41.15	42.11
4000	42.93	39.67	39.17
5000	41.25	38.77	38.17
6000	40.80	38.08	37.81
7000	40.40	37.90	37.60
8000	40.01	37.87	37.74
9000	39.79	37.83	37.85
10000	39.35	37.46	37.59

Arterial Oversaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	894.93	790.34	698.23
1000	200.98	192.23	229.31
2000	169.61	162.22	170.48
3000	159.28	156.02	155.67
4000	153.58	152.56	150.50
5000	151.21	149.74	144.48
6000	148.12	150.87	143.34
7000	147.54	148.87	143.89
8000	147.14	149.45	142.56
9000	146.08	148.84	142.73
10000	146.32	148.40	141.65

Grid Oversaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	787.32	712.36	695.38
1000	160.41	138.04	170.81
2000	112.03	104.76	113.57
3000	99.54	94.54	95.04
4000	95.23	89.99	90.01
5000	93.20	86.96	87.31
6000	91.68	86.77	87.21
7000	90.19	85.32	86.23
8000	89.10	84.91	86.01
9000	88.41	84.58	86.63
10000	87.63	83.89	86.06

A7.5.2 Real Crossover

The results given in this section are for the genetic algorithms with real crossover and binary mutation on the Gray coded variables. The results for the standard binary crossover on Gray coded variables are given in section A7.5.1.

Arterial Undersaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	434.72	362.80	429.81
1000	79.37	69.63	78.87
2000	61.56	55.71	56.44
3000	56.53	50.43	51.38
4000	54.80	50.80	49.35
5000	53.30	50.08	49.01
6000	53.48	49.86	48.98
7000	52.44	49.80	48.60
8000	52.24	49.53	48.34
9000	51.66	49.12	48.72
10000	51.79	49.11	48.67

Grid Undersaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	387.41	313.34	336.09
1000	60.74	54.88	62.45
2000	51.07	43.43	45.77
3000	46.47	39.88	39.89
4000	44.22	38.01	37.39
5000	43.19	37.36	36.75
6000	42.39	36.61	36.20
7000	41.73	36.33	36.19
8000	41.58	36.23	36.02
9000	40.81	35.94	36.16
10000	40.39	35.89	35.72

Arterial Oversaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	817.81	690.82	779.78
1000	194.28	183.67	195.72
2000	167.74	157.33	160.02
3000	158.20	149.37	144.36
4000	154.72	146.26	140.79
5000	151.51	145.18	139.07
6000	151.09	144.65	137.86
7000	151.53	143.90	138.76
8000	150.42	145.03	136.94
9000	149.80	143.97	136.93
10000	149.23	143.70	136.10

Grid Oversaturated

Function Evaluations	Algorithm Type		
	GGA	SSGA	CHC
0	738.02	733.60	668.16
1000	148.25	121.07	139.04
2000	111.26	92.68	94.62
3000	97.11	85.86	83.83
4000	91.25	82.10	79.34
5000	88.70	79.78	78.01
6000	87.24	79.39	76.53
7000	86.01	77.39	76.11
8000	85.30	77.67	76.17
9000	85.11	76.67	75.92
10000	84.35	76.09	75.33

A7.5.3 Real mutation

Here we give the results from the genetic algorithms runs with both real crossover and real mutation. The performance results without real mutation are given in A7.5.1 and A7.5.2.

Arterial Undersaturated

Function Evaluations	Algorithm Type	
	GGA	SSGA
0	382.03	335.68
1000	68.92	65.68
2000	58.18	53.84
3000	55.85	51.41
4000	54.05	50.27
5000	53.01	50.30
6000	53.08	50.35
7000	51.58	50.31
8000	51.90	50.53
9000	51.40	49.81
10000	51.37	50.29

Grid Undersaturated

Function Evaluations	Algorithm Type	
	GGA	SSGA
0	336.31	312.55
1000	60.69	53.66
2000	48.55	42.43
3000	44.45	39.08
4000	42.31	37.66
5000	41.15	37.12
6000	40.24	37.11
7000	39.88	36.85
8000	39.51	36.61
9000	39.35	36.60
10000	39.13	36.50

Arterial Oversaturated

Function Evaluations	Algorithm Type	
	GGA	SSGA
0	744.77	780.99
1000	199.87	179.17
2000	168.52	156.48
3000	160.77	150.09
4000	155.27	145.81
5000	153.61	143.86
6000	152.09	142.65
7000	148.17	142.87
8000	148.18	143.03
9000	148.00	141.99
10000	146.63	141.77

Grid Oversaturated

Function Evaluations	Algorithm Type	
	GGA	SSGA
0	689.61	644.98
1000	134.41	115.52
2000	107.44	91.79
3000	95.65	84.26
4000	90.74	81.20
5000	88.70	80.23
6000	87.82	79.46
7000	85.48	78.74
8000	84.50	78.53
9000	84.03	77.90
10000	83.85	77.87

A7.5.4 Real CHC

The results given in this section are for the Real CHC with real crossover and real mutation. The results for the standard CHC with real crossover and binary mutation are given in section A7.5.2.

Arterial Undersaturated

Function Evaluations	
0	353.67
1000	88.18
2000	60.61
3000	52.29
4000	50.50
5000	49.08
6000	48.85
7000	48.07
8000	47.85
9000	48.37
10000	48.04

Grid Undersaturated

Function Evaluations	
0	333.49
1000	63.78
2000	45.47
3000	40.15
4000	37.75
5000	36.69
6000	35.96
7000	35.69
8000	35.88
9000	35.74
10000	35.55

Arterial Oversaturated

Function Evaluations	
0	675.84
1000	216.34
2000	165.55
3000	148.88
4000	142.17
5000	138.48
6000	135.56
7000	133.57
8000	133.08
9000	133.19
10000	132.65

Grid Oversaturated

Function Evaluations	
0	613.90
1000	144.48
2000	98.00
3000	85.82
4000	80.01
5000	77.68
6000	75.62
7000	75.61
8000	75.65
9000	74.77
10000	74.58

A7.6 Algorithm Type

See output in sections A7.5.3 and A7.5.4.

A7.7 Optimal parameter tunings

A7.7.1 Blend crossover parameter

The results for $\alpha = 0.5$ are given in section A7.5.4.

Arterial Undersaturated **Grid Undersaturated**

Function Evaluations	BLX-α				BLX-α			
	0	**0.25**	**0.75**	**1**	**0**	**0.25**	**0.75**	**1**
0	344.70	347.29	383.30	361.48	288.37	299.62	289.55	309.26
1000	73.02	72.10	99.22	105.87	55.38	57.61	75.01	76.55
2000	69.03	58.57	72.12	79.77	50.52	44.66	53.97	59.42
3000	68.14	55.59	61.77	65.05	50.18	42.60	47.52	51.93
4000	68.19	54.57	56.72	60.24	50.26	42.04	44.56	48.54
5000	67.75	54.63	52.38	58.21	49.95	42.07	43.10	47.20
6000	65.32	54.55	50.98	55.36	47.89	42.06	40.89	45.47
7000	63.90	54.71	51.00	54.16	46.69	41.62	39.64	44.13
8000	62.92	54.32	50.57	51.77	46.38	40.46	38.77	44.22
9000	63.03	53.35	50.28	51.97	46.22	39.76	38.03	42.09
10000	62.12	53.33	49.06	51.33	44.99	39.41	37.61	41.46

Arterial Oversaturated **Grid Oversaturated**

Function Evaluations	BLX-α				BLX-α			
	0	**0.25**	**0.75**	**1**	**0**	**0.25**	**0.75**	**1**
0	788.15	667.17	700.72	673.14	617.38	625.08	635.02	600.03
1000	193.09	183.56	247.05	258.89	122.71	115.93	182.59	209.64
2000	186.53	155.68	186.65	197.86	117.27	94.32	124.82	142.15
3000	186.07	151.79	173.29	179.57	116.00	91.48	106.91	119.11
4000	185.46	151.17	158.69	167.98	116.39	89.79	95.39	109.61
5000	183.84	148.06	153.02	164.52	114.86	89.39	90.24	102.44
6000	181.71	148.32	147.69	160.76	114.63	89.85	84.42	99.41
7000	177.34	148.08	146.76	159.31	113.34	89.37	82.51	95.50
8000	175.49	147.04	144.19	157.57	109.49	89.34	81.05	92.47
9000	174.07	146.63	140.64	154.08	105.85	87.45	79.56	90.08
10000	173.13	145.96	138.80	152.16	103.67	85.46	78.22	88.79

A7.7.2 Population size

The results for a population size of $N_{Pop} = 50$ are given in section A7.5.4.

Arterial Undersaturated

Function Evaluations	Population Size	
	30	70
0	440.69	294.74
1000	65.58	102.67
2000	53.04	71.92
3000	50.34	60.23
4000	49.83	54.39
5000	49.53	51.56
6000	49.72	50.25
7000	48.93	49.80
8000	48.47	49.10
9000	49.02	48.82
10000	48.67	48.54

Grid Undersaturated

Function Evaluations	Population Size	
	30	70
0	357.32	276.93
1000	52.86	76.61
2000	40.59	55.05
3000	38.01	45.98
4000	37.76	41.28
5000	37.30	38.58
6000	37.38	37.19
7000	37.06	36.40
8000	36.12	35.91
9000	35.83	35.63
10000	35.78	35.43

Arterial Oversaturated

Function Evaluations	Population Size	
	30	70
0	765.91	665.82
1000	173.88	244.12
2000	151.43	187.03
3000	144.53	163.76
4000	141.14	150.17
5000	139.60	144.29
6000	139.22	139.52
7000	137.50	138.45
8000	136.29	135.29
9000	135.01	136.07
10000	135.19	133.77

Grid Oversaturated

Function Evaluations	Population Size	
	30	70
0	688.40	607.74
1000	111.29	182.20
2000	86.72	115.00
3000	81.45	95.85
4000	81.27	86.22
5000	79.81	81.42
6000	79.32	78.41
7000	78.68	77.31
8000	77.99	75.81
9000	75.96	75.24
10000	75.43	74.75

References

1. Wohl, M. and Martin, B.V. *Traffic System Analysis for Engineers and Planners*, McGraw-Hill Series in Transportation, McGraw-Hill, 1967.
2. Editor Gazis, D.C. *Traffic Science*, John Wiley and Sons, 1974.
3. Yagar, S. and Dion F. Distributed Approach to Real-Time Control of Complex Signalized Networks, *Transportation Research Record 1554*, 1996, pp. 1-8.
4. Lan, C-J., Messer, C.J., Chaudhary, N.A. and Chang, E.C.P. Compromise approach to optimize traffic signal coordination problems during undersaturated conditions, *Transportation Research Record 1360*, 1992, p.p. 112-120.
5. Akcelik, R. *Traffic Signals: Capacity and Timing Analysis*, Australian Road Research Board, Research Report No. 123, 1981.
6. Heydecker, B.G. and Dudgeon, I.W. Calculation of signal settings to minimize delay at a junction, *Proceedings of the 10th International Symposium on Transportation and Traffic Theory*, Elsevier, New York, 1987, p.p.159-178.
7. Adams, W.F. Road traffic considered as a random series, *Journal of the Institute of Civil Engineers*, vol. 4, 1936, p.p. 121-130.
8. Messer, C.J. and Bonneson, J.A. *NCHRP Web Document 12: Capacity Analysis of Interchange Ramp Terminals*, Transportation Research Board, National Research Council, Washington D.C., 1997.
9. van As, S.C. and Joubert, H.S. *Traffic Flow Theory*, N.T.C. Chair in Transportation Engineering, University of Pretoria, 1989.
10. Allsop, R.E. Delay-minimizing setting for fixed-time traffic signals at a single road junction, *Journal of the Institute of Mathematics and its Applications*, vol. 8, 1971, p.p. 164-185.
11. Yagar, S. Minimizing delay at a signalized intersection for time-invariant demand rates, *Transportation Research*, vol. 9, 1975, p.p. 129-141
12. Gross, D. and Harris, C.M. *Fundamentals of Queuing Theory*, New York: Wiley, 1985.
13. Webster, F.V. *Traffic Signal Settings*, Road Research Technical Paper No. 39, London: H.M.S.O., 1958.
14. Hutchinson, T.P. Delay at a fixed time traffic signal – II: Numerical comparisons of some theoretical expressions, *Transportation Science*, vol. 6, 1972, p.p. 286-305.

15. Clayton, A.J.H. Road traffic calculations, *Journal of the Institute of Civil Engineers*, vol. 16, 1941, p.p. 247-284, p.p. 588-594.

16. Improta, G. and Cantarella, G.E. Control system design for an individual signalized junction, *Transportation Research B*, vol. 18B, 1984, p.p. 147-167.

17. Gallivan, S. and Heydecker, B. Optimising the control performance of traffic signals at a single junction, *Transportation Research B*, vol. 22B, 1988, p.p. 357-370.

18. Silock, J.P. and Sang, A. SIGSIGN: a phase-based optimisation program for individual signal-controlled junctions, *Traffic Engineering and Control*, vol. 31, 1990, p.p. 291-298.

19. Allsop, R.E. Estimating the traffic capacity of a signalized road junction, *Transportation Research*, vol. 6, 1972, p.p. 245-255.

20. Yagar, S. Capacity of a signalized road junction: Critique and extensions, *Transportation Research*, vol. 8, 1974, p.p. 137-147.

21. Ohno, K. and Mine, H. Optimal traffic signal settings – I: Criterion for undersaturation of a signalized intersection and optimal signal setting, *Transportation Research*, vol. 7, 1973, p.p. 243-267.

22. Ohno, K. and Mine, H. Optimal traffic signal settings – II: A refinement of Webster's methods, *Transportation Research*, vol. 7, 1973, p.p. 269-292.

23. Reljic, S. TRAFSIG: A computer program for signal settings at an isolated, under-or-oversaturated fixed-time controlled intersection, *Traffic Engineering and Control*, vol. 29, 1988, p.p. 562-566.

24. Akcelik, R. SIDRA 4.0 software status, *Traffic Engineering and Control*, vol. 32, 1991, p.p. 585-589.

25. http://www.mctrans.ce.ufl.edu

26. Bleyl, R.L. A practical computer program for designing traffic-signal-system timing plans, *Highway Research Record*, no. 211, 1967, p.p. 19-33.

27. Webster, F.V. and Cobbe, B.M. *Traffic Signals*, Road Research Technical Paper No. 39, London: H.M.S.O., 1966.

28. Morgan, J.T. and Little, J.D.C. Synchronizing traffic signals for maximal bandwidth, *Operations Research*, vol. 12, 1964, p.p. 896-912

29. Little, J.D.C. The synchronization of traffic signals by mixed-integer linear programming, *Operations Research*, vol. 14, 1966, p.p. 568-594.

30. Little, J.D.C., Kelson, M.D. and Gartner, N.H. MAXBAND: A program for setting signals on arteries and triangular networks, *Transportation Research Record 795*, 1981, p.p. 40-46.

31. Messer, C.J., Whitson, R.H., Dudek, C.L. and Romano, E.J. A variable-sequence multiphase progression optimization program, H*ighway Research Record*, no. 445, 1973, p.p. 24-33.

32. Chang, E.C.P., Messer, C.J., and Cohen, S.L. Directional weighting for maximal bandwidth arterial signal optimization program, *Transportation Research Record 1057*, 1986, p.p. 10-19.

33. Gartner, N.H., Assmann, S.F., Lasaga, F. and Hou, D.L. MULTIBAND – A variable-bandwidth arterial progression scheme, *Transportation Research Record 1287*, 1990, p.p. 212-222.

34. Chang, E.C.P., Cohen, S.L., Liu, C., Chaudhary, N.A. and Messer, C.J. MAXBAND-86: Program for optimizing left-turn phase sequence in multiarterial closed networks. *Transportation Research Record 1181*, 1988, p.p. 61-67.

35. Stamatiadis, C. and Gartner, N.H. MULTIBAND-96: A program for variable bandwidth progression optimization of multiarterial traffic networks, *Transportation Research Record 1554*, 1996, p.p. 9-17.

36. Gartner, N.H. Constraining relations among offsets in synchronized signal networks, *Transportation Science*, vol. 6, 1972, p.p. 88-93.

37. Chaudhary, N.A. and Messer, C.J. PASSER IV: A program for optimizing signal timings in grid networks, *Transportation Research Record 1421*, 1993, p.p. 82-93.

38. Sripathi, H.K., Gartner, N.H. and Stamatiadis, C. Uniform and variable bandwidth arterial progression schemes, *Transportation Research Record 1494*, 1995, p.p. 135-145.

39. Pillai, R. S., Rathi, A.K. and Cohen, S.L. A Restricted Branch-and-Bound Approach for Generating Maximum Bandwidth Signal Timing Plans for Traffic Networks, *Transportation Research B*, vol. 32, 1998, p.p. 517-529.

40. Robertson, D.I. *TRANSYT: A traffic network study tool*, Road Research Laboratory Report LR253, 1969.

41. Yagar, S. An analysis and investigation of philosophical contrasts in traffic models, *Proceedings of the Seventh International Symposium in Transportation and Traffic Theory*, Kyoto 1977. 69-93.

42. Hillier, J.A. Appendix to Glasgow's experiment in area traffic control, *Traffic Engineering and Control*, vol. 7, 1966, p.p. 569-571.

43. Gartner, N.H., Little, J.D.C. and Gabbay, H. Simultaneous optimization of offsets, splits and cycle time, *Transportation Research Record 596*, 1976, p.p. 6-15.

44. Huddart, H.W. and Turner, E.D. Traffic signal progressions – G.L.C. Combination method, *Traffic Engineering and Control*, vol. 11, 1969, p.p. 320-322 and p.p. 327.

45. Gartner, N.H. and Little, J.D.C. Generalized Combination method for area traffic control, *Transportation Research Record 531*, 1975, p.p. 58-69.

46. Improta, G. and Sforza, A. Optimal Offsets for Traffic Signal Systems in Urban Networks, *Transportation Research B*, vol. 16B, 1982, p.p. 143-161.

47. MacGowan, C.J. and Lum, H.S. SIGOP or TRANSYT?, *Traffic Engineering and Control*, vol. 45, 1975, p.p. 46-50.

48. SYNCHRO 5.0 User's Guide, Trafficware, Albany California, 1999.

49. Traffic Network Study Tool, TRANSYT-7F, United States Version 10, McTrans Center, University of Florida, 2005.

50. Traffic Software Integrated System (TSIS) Version 5.1 User Guide, Federal Highway Administration Office of Operations Research, Development and Technology, 2003.

51. Chang, E.C.P., Messer, C.J. and Marsden, B.G. Reduced-delay optimization and other enhancements to the PASSER II-84 program, T*ransportation Research Record 1005*, 1986, p.p. 80-89.

52. Cohen, S.L. Concurrent use of MAXBAND and TRANSYT signal timing programs for arterial signal optimization, *Transportation Research Record 906*, 1983, p.p. 81-84.

53. Rogness, R.O. and Messer, C.J. Heuristic programming approach to arterial signal timing, *Transportation Research Record 906*, 1983, p.p. 67-75.

54. Skabardonis, A. and May, A.D. Comparative analysis of computer models for arterial signal timing, *Transportation Research Record 1021*, 1985, p.p. 45-52.

55. Cohen, S.L. and Mekemson, J.R. Optimization of left-turn phase sequence of signalized arterials, *Transportation Research Record 1021*, 1985, p.p. 53-58.

56. Cohen, S.L. and Liu, C.C. The bandwidth-constrained TRANSYT signal-optimization program, *Transportation Research Record 1057*, 1986, p.p. 1-9.

57. Liu, C.C. Bandwidth-Constrained Delay Optimization for Signal Systems, *ITE Journal*, vol. 58, 1988, p.p. 21-26.

58. Hadi, M.A. and Wallace, C.E. Progression-based optimization model in TRANSYT-7F, *Transportation Research Record 1360*, 1992, p.p. 74-81.

59. Wallace, C.E. and Courage, K.G. Arterial progression – new design approach, *Transportation Research Record 881*, 1982, p.p. 53-59.

60. Abu-Lebdeh, G. and Benekohal, R.F. Development of traffic control and queue management procedures for oversaturated arterials, *Transportation Research Record 1603*, 1997, p.p. 119-127.

61. Park, B., Messer C.J. and Urbanik II, T. Traffic Signal Optimization Program for Oversaturated Conditions: Genetic Algorithm Approach, *Transportation Research Record 1683*, 1999, p.p. 133-142.

62. Organisation for Economic Co-Operation and Development, *Traffic Control in Saturated Conditions*, Paris, 1981.

63. Elgerd, O.I. *Control Systems Theory*, McGraw-Hill, 1967.

64. Gazis, D.C. and Potts, R.B. The oversaturated intersection, *Proceedings of the 2nd International Symposium on the Theory of Road Traffic Flow*, Paris 1965, p.p. 221-237.

65. Michalopoulos, P.G. and Stephanopoulos, G. Oversaturated signal systems with queue length constraints – I: Single intersection, *Transportation Research*, vol. 11, 1977, p.p. 413-421.

66. Papageorgiou, M. *Application of Automatic Control Concepts to Traffic Flow Modelling and Control*, Springer-Verlag, Berlin, 1983.

67. Chang, T-H and Lin, J-T. Optimal signal timing for an oversaturated intersection, *Transportation Research B*, vol. 34B, 2000, p.p. 471-491.

68. De Schutter, B. Optimal traffic light control for a single intersection, *Proceedings of the 1999 American Control Conference*, San Diego 1999, p.p. 2195-2199.

69. Gazis, D.C. Optimum control of a system of oversaturated intersections, *Operations Research*, vol. 12, 1964, p.p. 815-831.

70. Michalopoulos, P.G. and Stephanopoulos, G. Oversaturated signal systems with queue length constraints – II: Systems of intersections, *Transportation Research*, vol. 11, 1977, p.p. 423-428.

71. Singh, M.G. and Tamura, H. Modelling and hierarchical optimization for oversaturated urban road traffic networks, *International journal of Control*, vol. 20, 1974, p.p. 913-934.

72. Lim, J.H., Hwang, S.H., Suh, I.H. and Bien, Z. Hierarchical optimal control of urban traffic networks, *International journal of Control*, vol. 33, 1981, p.p. 727-737.

73. Park, E.S., Lim, J.H., Suh, I.H. and Bien, Z. Hierarchical optimal control of urban traffic networks, *International Journal of Control*, vol. 40, 1984, p.p. 813-829.

74. Gazis, D.C. Modelling and optimal control of congested transportation systems, *Networks*, vol. 4, 1974, p.p. 113-124.

75. Lieberman, E.B., Rathi, A.K., King, G.F. and Schwartz, S.I. Congestion-based control scheme for closely spaced, high traffic density networks, *Transportation Research Record 1057*, 1986, p.p. 49-57.

76. Rathi, A.K. A control scheme for high traffic density sectors, *Transportation Research B*, vol. 22B, 1988, p.p. 81-101.

77. Tsay, H-S. and Lin, L-T. New algorithm for solving the maximum progression bandwidth, *Transportation Research Record 1194*, 1988, p.p. 15-30.

78. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts, 1989.

79. Whitley, D. A genetic algorithm tutorial, *Statistics and Computing*, vol. 4, 1994, p.p. 65-85.

80. Fogel, D.B. and Atmar, J.W. Comparing Genetic Operators with Gaussian Mutation in Simulated Evolutionary Processes Using Linear Systems, *Biological Cybernetics*, vol. 63, 1990, p.p. 111-114.

81. Holland, J. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

82. DeJong, K.A. *Analysis of the behaviour of a class of genetic adaptive systems*, PhD thesis, University of Michigan, 1975.

83. Gordon, V.S. and Whitley, D. Serial and Parallel Genetic Algorithms as Function Optimizers, *Proceedings of the Fifth International Conference on Genetic Algorithms,* Morgan Kaufmann, San Francisco, California, 1993, p.p. 177-183.

84. Davis, L. *Handbook of Genetic Algorithms*, van Nostrand Reinhold, New York, 1991.

85. Schaffer, J.D., Eshelman, L.J. and Offutt, D. Spurious Correlations and Premature Convergence in Genetic Algorithms, *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 102-112.

86. Syswerda, G. Uniform Crossover in Genetic Algorithms, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 2-9.

87. Eshelman, L.J., Caruana, R.A. and Schaffer, J.D. Biases in the Crossover Landscape, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 10-19.

88. Michalewicz, Z. and Janikow, C.Z. Genetic Algorithms for Numerical Optimizations, *Statistics and Computing*, vol. 1, 1991, p.p. 75-91.

89. Grefenstette, J. Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-16, 1986, p.p. 122-128.

90. Baker, J.E. Adaptive Selection Methods for Genetic Algorithms, *Proceedings of the First International Conference on Genetic Algorithms and their Applications,* Lawrence Erlbaum, Mahwah, New Jersey, 1985, p.p. 101-111.

91. Whitley, D. The GENITOR Algorithm and Selection Pressure: Why Rank Based Allocation of Reproductive Trials is Best, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 116-121.

92. Goldberg, D.E. and Deb, K. Comparative Analysis of Selection Schemes Used in Genetic Algorithms, *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 69-93.

93. Whitley, D. Beveridge, R., Graves, C. and Mathias, K. Test Driving Three 1995 Genetic Algorithms: New Test Functions and Geometric Matching, *Journal of Heuristics*, vol. 1, 1995, p.p. 77-104.

94. Bäck, T. and Hoffmeister, F. Extended Selection Mechanisms in Genetic Algorithms, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 92-99.

95. Meysenburg, M.M. and Foster, J.A. The Quality of Pseudo-Random Number Generators and Simple Genetic Algorithm Performance, *Proceedings of the Seventh International*

Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, California, 1997, p.p. 276-282.

96. Nissen, V. and Propach, J. On the Robustness of Population-Based Versus Point-Based Optimization in the Presence of Noise, vol. 2 1998, p.p. 107-119.

97. Whitley, D. and Kauth, J. GENITOR: A Different Genetic Algorithm, *Proceedings of the Rocky Mountain Conference on Artificial Intelligence,* Denver, Colorado, 1988, p.p. 118-130.

98. Whitley, D. and Hanson, T. Optimizing Neural Networks Using Faster More Accurate Genetic Search, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 391-397.

99. Davis, L. Bit Climbing, Representational Bias, and Test Suite Design, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 18-23.

100. Eshelman, L.J. The CHC Adaptive Search Algorithm: How to Have Safe Search when Engaging in Nontraditional Genetic Recombination, *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 265-283.

101. Eshelman, L.J. and Schaffer, J.D. Preventing Premature Convergence in Genetic Algorithms by Preventing Incest, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 115-122.

102. Goldberg, D.E. Sizing of Populations for Serial and Parallel Genetic Algorithms, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 70-79.

103. Rana, S. Whitley, D. and Cogswell, R. Searching in the Presence of Noise, *The 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, Springer, Berlin, Germany, 1996, p.p. 198-207.

104. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, 1996.

105. Wright, A.H. Genetic Algorithms for Real Parameter Optimization, *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1991, p.p. 205-218.

106. Caruana, R.A. and Schaffer, J.D. Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms, *Proceedings of the Fifth International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, California, 1988, p.p. 153-161.

107. Mathias, K.E. and Whitley, D. Transforming the Search Space with Gray Coding, *IEEE International Conference on Evolutionary Computation*, IEEE, 1994, p.p. 513-518.

108. Hinterding, R. Gielewski, H. and Peachey, T.C. The Nature of Mutation in Genetic Algorithms, *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco, California, 1995, p.p. 65-72.

109. Antonisse, J. A New Interpretation of Schema Notation that Overturns the Binary Encoding Constraint, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 86-91.

110. Goldberg, D.E. The theory of virtual alphabets, *Parallel Problem Solving from Nature*, Springer-Verlag, New York, 1991, p.p. 13-22.

111. Reeves, C.R. Using Genetic Algorithms with Small Populations, *Proceedings of the Fifth International Conference on Genetic Algorithms,* Morgan Kaufmann, San Francisco, California, 1993, p.p. 92-99.

112. Eshelman, L.J. and Schaffer, J.D. Real-Coded Genetic Algorithms and Interval-Schemata, *Foundations of Genetic Algorithms-2*, Morgan Kauffman, San Mateo, California, 1993, p.p. 187-202.

113. Hunter, A. Crossing Over Genetic Algorithms: The Sugal Generalized GA, *Journal of Heuristics*, vol. 4, 1998, p.p. 179-192.

114. Grefenstette, J.J. and Fitzpatrick, J.M. Genetic Search with Approximate Function Evaluations, *Proceedings of the First International Conference on Genetic Algorithms and their Applications,* Lawrence Erlbaum, Mahwah, New Jersey, 1985, p.p. 112-120.

115. Fitzpatrick, J.M. and Grefenstette, J.J. Genetic Algorithms in Noisy Environments, *Machine Learning*, vol. 3, 1988, p.p. 101-120.

116. Chi, P-C. Genetic Search with Proportion Estimations, *Proceedings of the Third International Conference in Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1989, p.p. 92-97.

117. Aizawa, A.N. and Wah, B.W. Scheduling of Genetic Algorithms in a Noisy Environment, *Evolutionary Computation*, vol. 2, 1994, p.p. 97-122.

118. Foy, M.D. Benekohal, R.F. and Goldberg, D.E. Signal Timing Determination Using Genetic Algorithms, *Transportation Research Record 1365*, 1992, p.p. 108-115.

119. Memon, G.Q. and Bullen, A.G.R. Multivariate Optimization Strategies for Real-Time Traffic Control Signals, *Transportation Research Record 1554*, 1996, p.p. 36-42.

120. Oda, T. Otokita, T. Tsugui, T. Kohno, M. and Mashiyama, Y. Optimization of Signal Control Parameters Using a Genetic Algorithm, *Third annual World Congress on Intelligent Transportation Systems*, Orlando, Florida, 1996.

121. Park, B. *Development of Genetic Algorithm Based Signal Optimization Program for Oversaturated Intersections*, PhD Thesis, Texas A&M University, 1998.

122. Lo, H.K. and Chow, A.H.F. Control Strategies for Oversaturated Traffic, *Journal Of Transportation Engineering*, vol. 130, 2004, p.p. 466-478.

123. Rouphail, N.M., Park, B. and Sacks, J. *Direct Signal Timing Optimization: Strategy Development and Results*, National Institute of Statistical Sciences, Technical Report 109, www.niss.org, 2000.

124. Chaudhary, N.A., Kovvali, V.G. and Mahabubul Alam, S.M. *Guidelines for Selecting Signal Timing Software*, Texas Transportation Institute, Report FHWA/TX-03/0-4020-P2, College Station, Texas, 2002.

125. Hadi, M.A. and Wallace, C.E. Hybrid Genetic Algorithm to Optimize Signal Phasing and Timing, *Transportation Research Record 1421*, 1993, p.p. 104-112.

126. Ceylan, H. Developing Combined Genetic Algorithm-Hill Climbing Optimization Method for Area Traffic Control, *Journal of Transportation Engineering*, vol. 132, 2006, p.p. 663-671.

127. Sun, D., Benekohal, R.F. and Waller, S.T. Multi-objective Traffic Signal Timing Optimization Using Non-dominated Sorting Genetic Algorithm, *IEEE Intelligent Vehicles Symposium*, Picastaway, N.J., 2003. p.p. 198-203.

128. Park, B., Santara, P., Yun, I. and Lee, D, Optimization of Time-of-Day Breakpoints for Better Traffic Signal Control, *Transportation Research Record 1867*, 2004, p.p. 217-223.

129. Abu-Lebdeh, G. and Benekohal, R.F. Genetic Algorithms for Traffic Signal Control and Queue Management of Oversaturated Two-Way Arterials, *Transportation Research Record 1727*, 2000, p.p. 61-67.

130. Girianna, M. and Benekohal, R.F. Dynamic Signal Coordination for Networks with Oversaturated Intersections, *Transportation Research Record 1811*, 2002, p.p. 122-130.

131. Park, B., Rouphail, N.M., Hochandel, J.P. and Sacks, J. Evaluating Reliability of TRANSYT-7F Optimization Schemes, *Journal of Transportation Engineering*, vol. 127, 2001, p.p. 319-326.

132. Park, B., Messer, C.J. and Urbanik II, T. Initial Evaluations of New TRANSYT-7F Version 8.1 Program, *Transportation Research Record 1683*, 1999, p.p. 127-132.

133. Wong, S-Y. TRANSYT-7F or PASSER II, Which Is Better-A Comparison Through Field Studies, *Transportation Research Record 1324*, 1991, p.p. 83-97.

134. Chang, G-L and Kanaan, A. Variability Assessment for TRAF-NETSIM, *Journal of Transportation Engineering*, vol. 116, 1990, p.p. 636-657.

135. Tian, Z.Z., Urbanik II, T., Engelbrecht, R. and Balke, K. Variations in Capacity and Delay Estimates from Microscopic Traffic Simulation Models, *Transportation Research Record 1802*, 2002, p.p. 23-31.

136. Law, A.M. and Kelton, D.W. *Simulation Modeling and Analysis*, McGraw-Hill, New York, 1991.

137. Rathi, A.K. The Use of Common Random Numbers to Reduce the Variance in Network Simulation of Traffic, *Transportation Research B*, vol. 26B, 1992, p.p. 357-363.

138. www.borland.com

139. *Special Report 209: Highway Capacity Manual*, Transportation Research Board, National Research Council, Washington D.C., 1994.

140. Jovanis, P.P. and May, A.D. Alternative Objectives in Arterial-Traffic Management, *Transportation Research Record 682*, 1978, p.p. 1-8.

141. Berg, W.D and Do, C-U. Evaluation of Network Traffic Performance Measures by Use of Computer Simulation Models, *Transportation Research Record 819*, 1981, p.p. 43-49.

142. Park, B., Messer, C.J. and Urbanik II, T. Enhanced Genetic Algorithm for Signal-Timing Optimization of Oversaturated Intersections, *Transportation Research Record 1727*, 2000, p.p. 32-41.

143. http://www.burtleburtle.net/bob/rand/isaac.html

144. Geisser, S. *Modes of Parametric Statistical Inference*, John Wiley and Sons, 2006.

145. Box, G.E.P. and Draper, N.R. *Empirical Model-Building and Response Surfaces*, John Wiley and Sons, 1987.

146. Leung, Y.W. and Wang, Y. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization, *IEEE Transactions on Evolutionary Computation*, vol. 5, 2001, p.p. 41-53.

147. Deb, K. *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, 2001.

148. Coles, S. *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London, 2001.

149. Bennett, C.R. and Dunn, R.C.M. Driver Deceleration Behaviour on a Freeway in New Zealand, *Transportation Research Record 1510*, 1995, p.p. 70-75.

150. Rioux, T.W. and Lee, C.E. Microscopic Traffic Simulation Package for Isolated Intersections, *Transportation Research Record 644*, 1977, p.p. 45-51.

151. Bonneson, J.A. Modelling Queued Driver Behaviour at Signalized Junctions, *Transportation Research Record 1365*, 1992, p.p. 99-107.

152. Long, G. Acceleration Characteristics of Starting Vehicles, *Transportation Research Record 1737*, 2000, p.p. 58-70.

153. Akcelik, R. and Biggs, D.C. Acceleration Profile Models for Vehicles in Road Traffic, *Transportation Science*, Vol. 21, 1987, p.p. 36-54.

154. Gerlough, D.L. and Wagner, F.A. *NCHRP Report 3: Improved Criteria for Traffic Signals at Individual Intersections*, Highway Research Board, National Research Council, Washington D.C., 1964.

155. Haas, R., Inman, V., Dixson, A. and Warren, D. Use of Intelligent Transportation System Data to Determine Driver Deceleration and Acceleration Behaviour, *Transportation Research Record 1899*, 2004, p.p. 3-10.

156. Brackstone, M. and McDonald, B. Car-following: a historical review, *Transportation Research Part F*, Vol. 2, 1999, p.p. 181-196.

157. Ranjitkar, P., Nakatsuji, T. and Asano, M. Performance Evaluation of Microscopic Traffic Flow Models with Test Track Data, *Transportation Research Record 1876*, 2004, p.p. 90-100.

158. http://mit.edu/its/mitsimlab.html

159. Wolshon, B. and Hatipkarasulu, Y. Results of Car Following Analyses Using Global Positioning System, *Journal of Transportation Engineering*, Vol. 126, 2000, p.p. 324-331.

160. Gipps, P.G. A Behavioural Car-Following Model For Computer Simulation, *Transportation Research B*, Vol. 15, 1981, p.p. 105-111.

161. Brockfeld, E. Kuhne, R.D. and Wagner, P. Calibration and Validation of Microscopic Traffic Flow Models, *Transportation Research Record 1876*, 2004, p.p. 62-70.

162. Ranjitkar, P., Nakatsuji, T. and Kawamura A. Experimental Analysis of Car-Following Dynamics and Traffic Stability, *Transportation Research Record 1934*, 2005, p.p. 22-32.

163. Luttinen, R.T. Statistical Properties of Vehicle Time Headways, *Transportation Research Record 1365*, 1992, p.p. 92-98.

164. Al-Ghamdi, A.S. Analysis of Time Headways on Urban Roads: Case Study from Riyadh, *Journal of Transportation Engineering*, Vol. 127, 2001, p.p. 289-294.

165. Davis, G.A. and Swenson, T. Field Study of Gap Acceptance by Left-Turning Drivers, *Transportation Research Record 1899*, 2004, p.p. 71-75.

166. Li, H. and Prevedouros, P.D. Detailed Observations of Saturation Headways and Start-Up Lost Times, *Transportation Research Record 1802*, 2002, p.p. 44-53.

167. Long, G. Start-Up Delays of Queued Vehicles, *Transportation Research Record 1934*, 2005, p.p. 125-131.

168. Williams, W.L. Driver Behaviour During the Yellow Interval, *Transportation Research Record 644*, 1977, p.p. 75-78.

169. Kimber, R.M. and Semmens, M.C. An experiment to investigate saturation flows at traffic signal junctions, *Traffic Engineering and Control*, Vol. 23, 1982, p.p. 110-117.

170. Niittymaki, J. and Pursula, M. Saturation Flows at Signal-Group-Controlled Traffic Signals, *Transportation Research Record 1572*, 1997, p.p. 24-32.

171. Fitzpatrick, K. and Schneider IV, W.H. Turn Speeds and Crashes with Right-Turn Lanes, *FHWA Report FHWA/TX-05/0-4365-4,* 2005.

172. Chundury, S. and Wolshon B. Evaluation of CORSIM Car-Following Model by Using Global Positioning System Field Data, *Transportation Research Record 1710*, 2000, p.p. 114-121.

173. McGhee, C.C. and Arnold, E.D. Review and Evaluation of Methods for Analyzing Capacity at Signalized Intersections, *Transportation Research Record 1572*, 1997, p.p. 160-166.

174. Sacks, J., Rouphail, N.M., Park, B. and Thakuriah, P. Statistically-based Validation of Computer Models in Traffic Operations and Management, National Institute of Statistical Sciences, Technical Report 112, www.niss.org, 2000.

175. Yang, X.K. Comparisons Among Computer Packages in Providing Timing Plans for Iowa Arterial in Lawrence, Kansas, *Journal of Transportation Engineering*, vol. 127, 2001, p.p. 311-318.