DMU's Interdisciplinary research Group in Intelligent Transport Systems, (DIGITS)
Faculty of Computing, Engineering and Media

Multi-objective Optimization in Traffic Signal Control

Author:
Phuong Thi Mai Nguyen

Supervisor:
Prof. Yingjie Yang
Dr. Benjamin Passow
Dr. Lipika Deka

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

August 2019

Abstract

Traffic Signal Control systems are one of the most popular Intelligent Transport Systems and they are widely used around the world to regulate traffic flow. Recently, complex optimization techniques have been applied to traffic signal control systems to improve their performance. Traffic simulators are one of the most popular tools to evaluate the performance of a potential solution in traffic signal optimization. For that reason, researchers commonly optimize traffic signal timing by using simulation-based approaches. Although evaluating solutions using microscopic traffic simulators has several advantages, the simulation is very time-consuming.

Multi-objective Evolutionary Algorithms (MOEAs) are in many ways superior to traditional search methods. They have been widely utilized in traffic signal optimization problems. However, running MOEAs on traffic optimization problems using microscopic traffic simulators to estimate the effectiveness of solutions is time-consuming. Thus, MOEAs which can produce good solutions at a reasonable processing time, especially at an early stage, is required. Anytime behaviour of an algorithm indicates its ability to provide as good a solution as possible at any time during its execution. Therefore, optimization approaches which have good anytime behaviour are desirable in evaluation traffic signal optimization. Moreover, small population sizes are inevitable for scenarios where processing capabilities are limited but require quick response times. In this work, two novel optimization algorithms are introduced that improve anytime behaviour and can work effectively with various population sizes.

NS-LS is a hybrid of Non-dominated Sorting Genetic Algorithm II (NSGA-II) and a local search which has the ability to predict a potential search direction. NS-LS is able to produce good solutions at any running time, therefore having good anytime behaviour. Utilizing a local search can help to accelerate the convergence rate, however, computational cost is not considered in NS-LS. A surrogate-assisted approach based on local search (SA-LS) which is an enhancement of NS-LS is also introduced. SA-LS uses a surrogate model constructed using solutions which already have been evaluated by a traffic simulator in previous generations.

NS-LS and SA-LS are evaluated on the well-known Benchmark test functions: ZDT1 and ZDT2, and two real-world traffic scenarios: Andrea Costa and Pasubio. The proposed algorithms are also compared to NSGA-II and Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D). The results show that NS-LS and SA-LS can effectively optimize traffic signal timings of the studied scenarios. The results also confirm that NS-LS and SA-LS have good anytime behaviour and can work well with different population sizes. Furthermore, SA-LS also showed to produce mostly superior results as compared to NS-LS, NSGA-II, and MOEA/D.

Acknowledgements

I would like to express my sincere gratitude to my supervisory team Prof. Yingjie Yang, Dr. Benjamin N. Passow and Dr. Lipika Deka who provided unstinting support with their insights, expertise, and valuable comments. Without their encouragement and support, this thesis would not have been completed on a limited time frame. Especially, I would like to expand deepest thank to my dedicated supervisor Dr. Benjamin N.Passow who share his pearls of wisdom during this research, devoted his time and made valuable comments for better insight. Also, inspiration and encouragement play important role in keeping me moving forward.

I gratefully thank the Ministry of Education and Training of Vietnam for funding me a four-year scholarship for my study in the UK. Without this financial sponsorship, I would not be able to come to study in the UK.

My sincere thanks also go to the De Montfort University Interdisciplinary research Group in Intelligent Transport Systems (DIGITS) for the financial support to participate the WCCI 2016 conference in Vancouver and the International student workshop 2016 in Wroclaw, Poland. I also would like to thank all member of DIGITs for offering assistance to my study.

Last but not least, I would like to thank my parents and my sister for always encouraging me throughout this journey. Especially, I owe thanks to a very special person, my husband, for his love, support, and understanding during my pursuit of Ph.D. I greatly appreciate his belief in me that gave me extra strength to get things done.

Contents

List of Figures

List of Tables

Abbreviations

ITS	Intelligent Transportation System
TSC	Traffic Signal Control
MOOP	Multi-objective Optimization Problem
MOEA	Multi-objective Optimization Evolutionary Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Algorithm
DE	Differential Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm Based on Decomposition
NS-LS	Multi-objective optimization algorithm based on local search
SA-LS	Surrogate-assisted optimization algorithm based on fuzzy distance and local search
SUMO	Simulation of Urban Mobility
MSE	Mean Square Error
RPROP	Resilient Back-propagation Learning Algorithm
FNN	Feedforward Neural Networks
ANN	Artifical Neural Networks
MLP	Multilayer Feedfoward Perceptrons
TraCI	Traffic Control Interface
O/D	Origin/Destination
ZDT1	Zitzler-Deb-Thiele's function N.1
ZDT2	Zitzler-Deb-Thiele's function N.2
SBX	Simulated Binary Crossover
PLM	Polynomial Mutation

Symbols

HV	Hypervolume
S	Schott metric
MS	Maximum Spread
$C(A, B)$	The set coverage (C-metric) of algorithms A and B
tl_i	The time loss of $t^{(th)}$ vehicle
$\bar{TL}$	Average time lost
$\bar{F}$	Average traffic flow
N_{veh}	Total number of vehicles
N_e	Total number of detectors
N	Population size of the evolutionary algorithm
$maxEval$	Maxinum number of evaluations using a traffic simulator
p_c	Crossover probability
p_m	Mutation probability of a chromosome
P_{mv}	Mutation probability of a variable in a chromosome
C	Cycle length
C_{max}	Maximum cyle length
C_{min}	Minimum cyle length
g_i	Green duration of $i^{(th)}$ phase
g_i^{min}	Minimum green duration of $i^{(th)}$ phase
g_i^{max}	Maximum green duration of $i^{(th)}$ phase
$<_c$	The crowded tournament selection operator
$nb_{R_i^{(t)}}$	Neighbour of solution $R_i^{(t)}$
$P^{(t)}$	The population of the evolutionary search at i^{th} generation
$Q^{(t)}$	The offspring population created from $P^{(t)}$ at i^{th} generation
$R^{(t)}$	The population merged by $P^{(t)}$ and $Q^{(t)}$at i^{th} generation

L	A database consisting all solutions evaluated by SUMO
L_{temp}	A database consisting solutions evaluated by SUMO in the current generation
SUP_1	A set of solutions of a sub-population which belong to the first non-dominated front
SUP_2	A set of solutions of a sub-population which belong to the second non-dominated front
F_i	$i^{(th)}$ non-dominated front
E	Error function for a learning algorithm
E_c	The Cross-validation error function
err_{cur}	Average approximation error of the surrogate using solutions in L_{temp}
δ	An error threshold
$\bar{HV}$	The average hypervolume
S_{HV}	Standard deviation of hypervolume

Chapter 1

Introduction

1.1 Motivation

Transportation plays an important role in society as it contributes to economic growth, social development, and improvement to human lifestyle. However, the transport sector is facing several challenges, especially in urban areas. First, congestion has become a serious issue which can lead to an increase in fuel consumption, air pollution and accordingly can cause detrimental impacts on economic growth. The second challenge is to reduce the number of fatalities and serious injuries from road accidents and collision, especially in low- and middle-income countries. According to the global status report on road safety 2018 of the World Health Organization, approximately 1.35 million people die each year due to road traffic accidents, WHO (2018). Third, reducing traffic exhaust emissions is an urgent mission since the transportation industry is a key player in global warming. To solve these mentioned problems, a number of methods can be applied such as constructing new roads, expanding existing transport systems, optimizing the performance of existing transportation systems and making transport policies. Depending on the situation and characteristics of each area, suitable and efficient methods would be chosen. However, for urban cities where there is no available space for building new transport roads, constructing more roads or expanding transport systems is often infeasible. Therefore, upgrading and optimizing an existing transport system to make it become smarter has become an attracting trend in transportation research. Intelligent Transport System (ITS) has been proposed and deployed in many cities around the world to improve the performance of the transport sector, Chen et al. (2014), Chen and

Chang (2014), Djalalov (2013), Hamza-Lup et al. (2008), Sanchez-Medina et al. (2010), Zhang et al. (2011).

Intelligent Transportation System (ITS) combines information and communication technologies into the transportation system's infrastructure to improve performance, efficiency, and safety. The purpose of ITS is to take advantages of advanced technologies to address transportation problems, for example, safety, traffic congestion, transport efficiency, and environmental protection by creating more intelligent roads. Over the past decade, ITS has greatly improved transportation conditions and access capacity of road networks Chen and Chang (2014), Kouvelas et al. (2011), Yan et al. (2013), reduced traffic congestion Adacher (2012), Sabar et al. (2017), Shen et al. (2013) and exhaust emissions Armas et al. (2017), Passow et al. (2012), Sanchez-Medina et al. (2010) in many urban areas over the world.

Traffic signal control system is a cost-effective tool for urban traffic management and has become an important research area in ITS. It controls the traffic at road intersections, determines which flows are allowed to pass through and which flows have to stop. Its final purpose is to make sure that every traffic users including vehicles, pedestrians, and bicyclist move through the intersection safely and efficiently. The correct and efficient operation of traffic signal control of the overall traffic network is therefore critical to the performance of the urban transport network and is considered to be an essential element of ITS.

The role of traffic signal optimization is to significantly improve traffic network performance by optimizing objectives such as reducing delay and number of stops and increasing network throughput or average speed within the traffic network. Setting traffic signals in a signal-controlled street network involves the determination of cycle time, splits of green (and red) time, and offsets. Traffic light signal optimization might optimize a part of or all these values.

Traffic signal timing optimization methods fall within two main categories: mathematical programming method and simulation-based approach, Chen and Chang (2014). The former scheme utilizes mathematical formulations to capture the characteristics of traffic flow models which will be utilized to optimize objectives in traffic management. However, the calculations of these mathematical models are often very complicated and hard to meet real-time requirements, Zhao et al. (2012). Furthermore, the interrelationship

between the traffic flows of complex intersections, such as queue spillback or blockage between through and turning lanes, cannot be adequately captured by mathematical programming formulations, Chen and Chang (2014). Moreover, not every optimization problem can be expressed by mathematical formulas. On the other hand, the simulation-based approaches aim at capturing the complex interactions between traffic characteristics. For that reason, more recently, researchers tend to optimize traffic signal timing by using simulation-based approaches, Chen and Chang (2014), Papatzikou and Stathopoulos (2015), Poole and Kotsialos (2016).

Multi-objective Evolutionary Algorithms (MOEAs) are widely used to solve the multi-objective optimisation problem in transportation, Caraffini et al. (2013), Goodyer et al. (2013), Witheridge et al. (2014), Zheng et al. (2015). However, when applying MOEAs to optimise a transportation problem, traffic simulation always needs to be called when a solution is evaluated. Moreover, MOEAs need to evaluate solutions many times in the optimisation process to obtain optimal solutions. Time to run multiple simulations requires much processing time. For example, it takes 25 seconds to run one simulation of the Andrea Costa traffic scenario Bieker et al. (2015) using a PC with Intel(R) Core(TM) i5-6500 CPU 3.2GHz. If the population size is 60 and there are 20 generations in the evolutionary process, the number of simulations needed in the optimization algorithm is 1200. Therefore, the time to run simulations is about 8.3 hours. The computation time will rapidly rise as the scale of the traffic network increases, such as in road network size and number of vehicles. In order to address this problem, a few research methods have utilized powerful and expensive hardware to reduce computation time. However, such approaches are expensive and not always feasible. As a result, optimisation approaches which have the ability to provide good solutions, which produce high fitness values and satisfy all constraints, at a reasonable processing time, especially at an early stage, are desired. Nevertheless, the optimization literature mostly focuses on the quality of solutions reached by an algorithm at the end of the optimization process. However, such studies might not work efficiently in optimization problems where function evaluations are limited by time or cost. In these situations, in order to evaluate the efficiency of an optimisation algorithm, an indicator, which can measure the ability of that algorithm to produce good solutions at any time during its operation, is needed. Anytime behaviour of an algorithm is its ability to provide as good a solution as possible at any

time during its execution and continuously improves the quality of the results as computation time increases, Dubois-Lacoste et al. (2015), Lopez-Ibanez and Stutzle (2014). Anytime behaviour may be described in terms of the curve of hypervolume over time. Hypervolume, introduced by Zitzler and Thiele Zitzler and Thiele (1998), measures the volume of the objective space which is dominated by a non-dominated set. Therefore, if one non-dominated set has a higher hypervolume, it will be closer to the Pareto-optimal front. The hypervolume indicator is used to compare anytime behaviour between two multi-objective optimization algorithms. As optimizing traffic signal control is time-consuming and the time to run the optimization process is limited and scenario specific, anytime behaviour of the system is a preferred indicator for system performance.

In transportation optimization problems, small population sizes can be important for scenarios where limited processing capabilities meet demand for quick response time. Such scenarios are typical for local and distributed signal controllers which offer very limited processing power while requiring optimised signal timings within a few cycles or minutes. Therefore, optimization algorithms with the ability to work effectively in small population sizes are preferable.

A combination of a local search and a global evolutionary algorithm may accelerate the convergence speed of the search. Furthermore, Espinoza et al. (2003) indicates that local search also helps to reduce the population size of the optimization algorithm. Therefore, with selective use of a local search, anytime behaviour of an evolutionary algorithm can be improved and the efficiency of a traffic signal optimization model can be increased.

Surrogate or approximation models are computational models used to estimate objective values of candidate solutions at a cheaper cost compared to original objective function. Surrogates are used to reduce the number of evaluations using original objective function while remaining a reasonable good quality of results obtained. Surrogate may reduce the number of traffic simulator-based evaluations in a generation of the evolutionary search. Therefore, with a limited budget of the maximum number of evaluations using the traffic simulator, the number of generations may be increased. Consequently, surrogate-assisted MOEAs are very promising to improve anytime behavior of traffic signal optimization algorithms.

For all the afore-mentioned reasons, this study proposes a multi-objective optimization

algorithm based on local search (NS-LS) and a surrogate-assisted multi-objective optimization algorithm based on fuzzy distance and local search (SA-LS) for improving anytime behaviour in traffic signal timing. Furthermore, these algorithms can work effectively when the population size is small. The performance of the proposed algorithms will be compared with NSGA-II and MOEA/D with different sizes of the population, demonstrating their improved effectiveness.

1.2 Propositions

In this demanding field of intelligent transport systems, the following research propositions have been set and studied:

Proposition 1: *A local search method can be used to improve anytime behaviour of multi-objective optimization algorithms in traffic signal optimization problems.*

A novel local search algorithm looking for neighbours which potentially have good fitness values is introduced in Chapter 4. The proposed local search method can predict potential search directions before searching for better solutions. Therefore, the chance to find a superior neighbour at early stages would be increased. Consequently, anytime behaviour of the search algorithm may be improved. The experiments are conducted in Chapter 6 and the results are shown in Chapter 7.

Proposition 2: *A method based on an approximation model can be designed to evaluate candidate solutions in traffic signal optimization problems.*

A novel surrogate model is proposed in Chapter 4 based on an Artificial Neural Network. By using solutions evaluated by the traffic simulator in previous generations, this surrogate can learn the relationship between the input which is the duration of phases of a traffic signal system and the output that are values of traffic parameters such as flow and delay. The surrogate is continuously updated during the optimization process to increase the accuracy of the approximation result. This surrogate is partially used with a traffic simulator to evaluate objective values of candidate solutions in every generation of the evolutionary search.

Proposition 3: *A local search method can be combined with an approximation model to enhance anytime behaviour of evolutionary search in traffic signal optimization problems, especially in small population sizes.*

A novel surrogate-assisted evolutionary algorithm is introduced in Chapter 5 for traffic signal optimization problems. An approximation model is used to reduce the number of traffic simulator-based evaluations while a local search can accelerate the convergence rate of the evolutionary search. Therefore, using the same number of evaluations conducted by a traffic simulator, the number of iterations in the optimization process of the proposed algorithm will be increased. An appropriate management model is also proposed to use the surrogate effectively and properly. Experiments are carried out in Chapter 5 to evaluate the performance of the combination of a local search with an approximation model in traffic signal optimization problems in terms of anytime behaviour improvement. The results of the experiments are shown in Chapter 6.

1.3 Aims and objectives

The main aim of this research is to evaluate the ability of combining a surrogate-assisted evolutionary algorithm and a local search method in improving anytime behaviour of a traffic signal optimization system, especially when the population size of the evolutionary process is small. This research also intents to assess the possibility of using an approximation model to evaluate candidate solutions in traffic signal optimization problems. Furthermore, another subsidiary aim of this research is to investigate the ability of local search methods in increasing anytime behaviour of multi-objective optimization algorithms in traffic signal optimization problems.

The objectives of this study are:

1. To provide a comprehensive literature review of traffic signal optimization based on multi-objective evolutionary algorithms and traffic microscopic simulators.

2. To extend the knowledge of optimizing traffic signal control using surrogate-assisted evolutionary algorithms and local search.

3. To construct an optimization model for traffic signal control based on a local search method to improve anytime behaviour and this model can work effectively in small population sizes.

4. To develop a surrogate-assisted evolutionary algorithm for optimizing multiple objectives in traffic signal control. This methodology utilizes a surrogate to decrease the number of traffic simulator-based evolutions. A local search is also used to accelerate the convergence rate of the evolutionary search.

5. To assess and compare the performance of the proposed models on traffic scenarios.

1.4 Major Contributions of the Thesis

Major contributions of the thesis are summarized as follows:

1. A local search methodology for superior neighbours in local areas is introduced. This local search has the ability to predict potential search directions, therefore, the chance to find out a better neighbour from an early stage can be increased.

2. A multi-objective evolutionary algorithm based on local search is proposed for improving anytime behaviour in traffic signal timing. The local search is performed inside the iteration process of the evolutionary algorithm to quickly find superior solutions. This helps to increase the convergence rate of the evolutionary search.

3. A surrogate model is constructed to evaluate the fitness value of candidate solutions in the optimization process. This surrogate is able to learn the relationship between the phase duration of the signal timing setting and the traffic parameters needed such as flow and time lost. Solutions which are already evaluated using the traffic simulator in the previous generations are utilized to train the surrogate model. The model is also updated during the optimization process to improve the approximation accuracy.

4. A surrogate-assisted multi-objective evolutionary optimization algorithm for traffic light signal control in urban intersections is introduced. This algorithm utilizes the surrogate model to estimate the fitness value of candidate solutions. Both traffic simulator and the surrogate are used together in the fitness evaluation process

to prevent the evolutionary search from obtaining false optima. Moreover, the local search is also used in the iterations of the evolutionary search to accelerate the convergence rate. A hybrid of the local search and the surrogate improve the anytime behaviour of the evolutionary algorithm in traffic signal optimization problems.

5. A fitness evaluation scheme is proposed to effectively choose a model between the surrogate and the traffic simulator SUMO to estimate fitness values of solutions. This scheme is used to guarantee that the surrogate is used effectively. This scheme is based on the closeness of the solution to the solutions already evaluated by the traffic simulator in the database which is used to build the surrogate and the MSE of approximation error of the surrogate.

1.5 Thesis structure

The thesis is organized as follows:

Chapter 2 provides a background of traffic signal control systems, road traffic simulators as well as optimization algorithms which have been applied in transportation problems. Fundamental definitions of traffic signal control systems are introduced in the first part of this chapter. Basis introduction to road traffic simulators and Simulation of Urban Mobility (SUMO) software are present in the next section. Afterward, definition and basic concepts as well as the general framework of Multi-objective Evolutionary Algorithms (MOEAs) are explained. Definition of surrogate-assisted evolutionary algorithms and techniques for constructing a surrogate are introduced in the last part of this chapter.

Chapter 3 contains a comprehensive literature review. Although many computational intelligent methods have been applied to optimize traffic signal problems, this chapter mainly focuses on multi-objective traffic signal optimization using MOEAs and local search-based MOEAs. Evaluating the objective value of a candidate solution using traffic simulators is also reviewed. Advantages and drawbacks of optimizing a traffic signal optimization problem using traffic simulator-based MOEAs are shown and the gap in the previous researches of traffic signal optimization using MOEAs is outlined. Studies on traffic signal optimization using surrogate-assisted MOEAs are also in this chapter.

Chapter 4 introduces the algorithms proposed in this study. Firstly, the motivation and the flow of the local search strategy are provided. Afterwards, this chapter presents NS-LS which is a multi-objective optimization algorithm for improving anytime behaviour in traffic signal timing. The overview, flow, framework of NS-LS, as well as discussion of the design of the evolutionary search of NS-LS are explained. The process to construct the surrogate including choosing the model, the training algorithm, the error function, tuning hyperparameters, and updating the surrogate are also offered. The surrogate is used together with the traffic simulator to estimate the fitness value of candidate solutions and fitness evaluation scheme which is a strategy to effectively use the surrogate is also proposed in this chapter. Afterwards, SA-LS - a surrogate assisted multi-objective traffic signal optimization algorithm based on fuzzy distance and local search is introduced, including an overview of SA-LS and a discussion of the SA-LS's search flow.

Chapter 5 discusses the experimental setup used to evaluate the performance of the proposed algorithms in this thesis. Two benchmark test functions and two traffic scenarios are introduced in the first part. Procedure to connect a traffic scenario and an optimization model as well as methods to extract optimization objective value from SUMO output are presented in the next sections. Performance indicators used in this thesis are also discussed. At the end of this chapter, the details of the three experiments are introduced to evaluate the performance of the algorithms.

Chapter 6 illustrates the experimental results. The performance of proposed algorithms is evaluated and compared against NSGA-II and MOEA/D using several performance indicators introduced in Chapter 6. The optimization results of the algorithms in three experiments are presented to examines the propositions.

Chapter 7 concludes the thesis and it contains conclusions, recommendations, and future work. The propositions introduced in the introduction chapter are reconfirmed in this chapter. Overall summary of the major contributions of research is also provided.

Chapter 2

Background

2.1 Introduction

Before evaluating hypotheses formulated in Chapter 1, general knowledge about relevant components is reviewed. Therefore, this chapter provides a background of traffic signal control systems, road traffic simulators, and optimization algorithms applied in transportation problems. This chapter is organized as follows: Section 2.2 introduces the fundamental definitions of traffic signal control systems while basic introduction to road traffic simulators as well as Simulation of Urban Mobility (SUMO) software are presented in Section 2.3. Multi-objective Optimization Algorithms (MOEAs) definitions, basic concepts, and the general framework are explained in Section 2.4. The difference between surrogate-assisted evolutionary algorithms and evolutionary algorithms is illustrated in Section 2.5. Techniques for constructing a surrogate model are also shown in this section. Finally, Section 2.6 concludes this chapter.

2.2 Traffic Signal Control Systems

2.2.1 Introduction to Traffic Signal Control Systems

Transportation is a critical and non-separable part of any society as it links various regions and helps people move easily between different destinations. Advances in transportation have made possible changes the way in which societies are organized and the

way of living. Hence, transportation has a high influence on the development of civilisations. The rapid increase in population has enabled the number of registered vehicles to grow quickly. The number of vehicles is increasing and transport characteristics are growing more complex such as different types of drivers, pedestrians, bicyclists, vehicles, and road infrastructure. Traffic demand is rapidly increasing and continues to exceed the transport capacity. To better meet traffic demand, it is essential to build new transport infrastructures or to upgrade existing road systems. Traffic demand in urban cities are normally much higher than that of rural areas but space for constructing new roads or expanding existing transport infrastructure in big cities is no longer enough. Consequently, traffic congestion in urban areas has become prevalent and continues to have detrimental consequences on both society and economy of the region and country. According to a report of CE Delft, which is an independent organization specialized in developing solutions for environmental problems; the external cost of road traffic, which is the cost imposed by side effects of transport such as congestion, noise level, and air pollution, in the European Union accounts for 1 to 2 % of GDP ,van Essen et al. (2011). Furthermore, the transportation system is currently facing several challenges and there is a need to decrease travel time and delays, improving passenger safety and reducing traffic exhaust emissions. Therefore, Intelligent Transportation Systems (ITSs) have been proposed and developed in many cities around the world to improve the performance of the transport sector. Over the past decade, ITSs have greatly improved transportation conditions and capacity of road networks, reduced traffic congestion and exhausted emissions in many urban areas over the world, DOrey and Ferreira (2014), Hess et al. (2015), Quddus et al. (2019), Sheng-hai et al. (2011).

Traffic Signal Control (TSC) Systems is one of the most popular ITSs and it is widely used around the world to regulate traffic flow. TSC systems play an important role in transportation network management and they are one of the most effective traffic control methods for safe and efficient travel in urban areas. Traffic signal control systems are placed at road intersections to control conflicting traffic movements and determines which approaches are allowed to travel through and which traffic streams have to stop. Its final purpose is to guarantee that every traffic user, including vehicles, pedestrians, and bicyclists move through the intersection safely and efficiently. TSC systems are also meant to reduce traffic congestion and emissions. However, inefficient operation of the traffic movement control system at intersections is one of the main reasons leading to

traffic congestions. The efficiency of a TSC system is directly related to the effectiveness of the employed control methodology. It is estimated that 50-80 % of traffic issues happen at intersections and their surroundings, 1/3 travel time and 80-90 % waiting time is consumed at red phases of signalized intersections, Ben et al. (2010). Therefore, a proper and efficient traffic signal control systems is essential to the performance of the whole transport system. Basically, most signal control approaches aim to increase traffic flow and to reduce delay or to prevent traffic congestion, Chen and Chang (2014), Sanchez-Medina et al. (2010), Shen et al. (2013).

2.2.2 Fundamental Definitions of Traffic Signal Control Systems

A traffic signal control system is a signaling device placed at intersections, junctions, crossroads or pedestrian crossing to regulate traffic movements. In the UK and many other countries, a TSC system commonly consists of three lights: a red, indicating that incoming vehicles have to stop, a green light meaning that the vehicles are allowed to travel through the intersection if it is safe. The green arrow pointing right or left means the vehicles are allowed to make a protected turn. An amber warning light, coming after a green light, indicating that the traffic light is about turn red and the vehicles have to stop if possible. When the red and amber lights are shown at the same time, the vehicles have to completely stop. For pedestrians, there are only two lights: a red light, which means pedestrians have to stop, and a green light, indicating that pedestrians can cross the road.

The TSC deployed at an intersection implements traffic signal timing to control vehicles, bicyclists, pedestrians, and other traffic participants safely passing through the intersection. Traffic signal timing includes deciding the sequence of movements and allocating green time to each group of movements at a signalized intersection. Pedestrians, cyclist and other users also should be taken into account when designing signal timings. An example of movements in a two-phase signal system of a four-legged intersection is illustrated in Figure 2.1. A diagram of signal timing is demonstrated in Figure 2.2. Some fundamental definitions in signal timing are described as follows, Kittelson & Associates (2008), Papageorgiou et al. (2003):

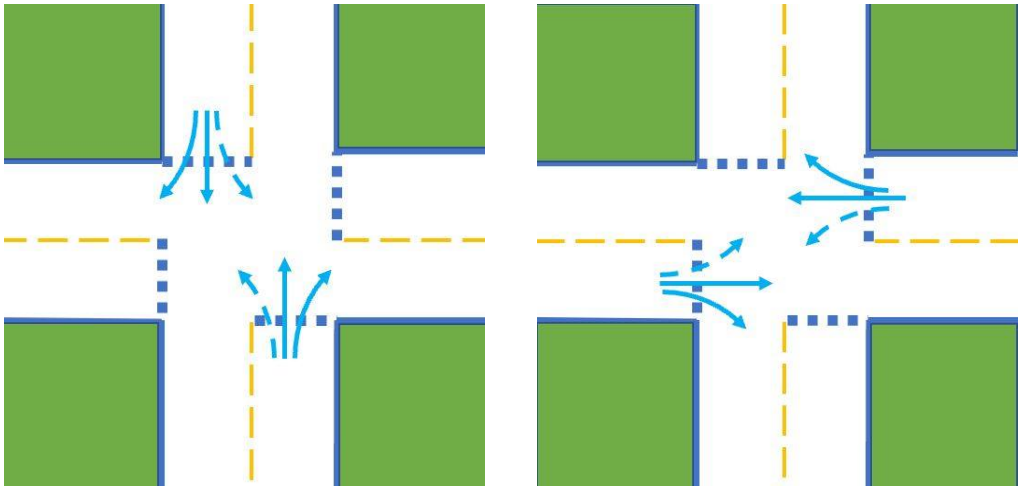


FIGURE 2.1: Movements in a two-phase system.

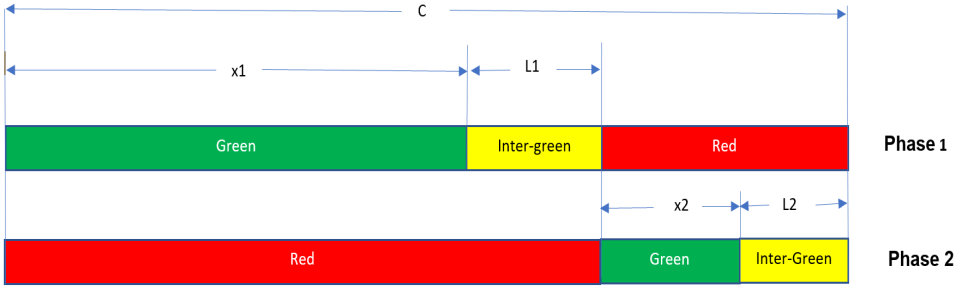


FIGURE 2.2: A diagram of two-phase signal system (C is signal cycle length, x1 and x2 are green durations of phase one and phase two, L1 and L2 are inter-green durations).

- *A signal cycle* is a complete sequence of all traffic movements at an intersection. A signal cycle length is defined as the total time required to accomplish one signal cycle and it is determined by the sum of green times of all stages, yellow change intervals and all-red clearance intervals.

- *A phase* is a portion of a signal cycle assigned to one set of movements and it is defined as the green, yellow or all-red clearance intervals.

- *Offset* is the difference between two green initiation times for two successive intersections. Offset helps vehicles moving through successive intersections without being stopped.

- *Green splits* are a portion of total available green time in the cycle allocated to each phase at an intersection.

- *Inter-green* time consists of both the yellow indication and the all-red indication(if applicable) in one cycle and it is necessary when changing states to avoid collision between traffic movements.

A proper and effective traffic signal timing can have a number of benefits: (1) vehicles can pass the intersection safely; (2) increase the number of vehicles served at the intersection - or increase the capacity of signalized intersections; (3) reduce congestion and delay; (4) allow pedestrians and side street traffic to travel through the intersection with appropriate levels of accessibility.

2.2.3 Overview of Traffic Signal Control Systems

The most important role of traffic control is to regulate traffic flow, improve congestion, and reduce emissions. Information technology and computer technology are two of dependencies of traffic control progress and development, Wang et al. (2018). Recent improvements in traffic control methods can provide flexible control strategies, Chow (2010).

As mentioned in Board et al. (2010), a lot of traffic signal control systems have been proposed and developed, but less than half of them have been deployed in the real world traffic to use. According to Wang et al. (2018), signal control strategies employed for road signalized intersections may be classified as follows:

- *Fixed-time or pre-timed signal control methods* use pre-determined traffic signal control parameters such as the sequence of operation, split and offset, is suitable for regular and relatively stable traffic flows. Pre-time strategies are obtained off-line by utilizing appropriate optimization methods based on historical data.

- *Traffic-responsive or real-time signal control methods* automatically regulate the signal timing based on current traffic conditions which were studied from real-time traffic data. These data are collected from equipment such as inductive loops or sensors, which are installed along the roads. Therefore, various traffic signal control parameters can be dynamically changed depending on recent traffic conditions. Real-time TSC provides an effective management method for urban traffic networks which are highly complex, uncertain and dynamic.

Signal control strategies can be classified by the number of intersections involved as shown as follows:

- *Isolated strategies* which are applicable to a single intersection without consideration of any adjacent intersections and signal timings at this intersection do not significantly affect other neighbouring intersections. In this instance, each intersection will have signal settings that are the most suitable for only that particular intersection.
- *Coordinated strategies* which consider several adjacent intersections or a traffic area. Coordinated strategies allow vehicles to move through successive intersections without encountering a red signal. Accordingly, the green time of one junction always starts later than its predecessor by the amount of time the vehicle needed to travel between two intersections. This travel time is determined by congestion-free conditions.

Traffic signal control is an dependency of the development of modern control theory, artificial intelligence theory, traffic information technology, and traffic engineering technology. Rapidly development of Artificial Intelligence (AI) theory and methods, which include agents, neural networks, fuzzy logic, and group intelligence, also impact the traffic control strategies, Papageorgiou et al. (2003).

TRANSYT is a well-known fixed-time coordinated traffic signal control system, Robertson (1986). It contains a traffic model and is fed with initial signal settings including initial values of splits, cycle length, and offsets as well as of the minimum value of green duration for each signal stage and the pre-defined staging of each intersection. It can produce fixed-time signal plans for different hours of a day. The optimization model determines the corresponding output, which is the performance metrics, from given input of decision variables. In TRANSYT, the hill-climbing algorithm is utilized to look for the optimum. Split Cycle and Offset Optimization Technique (SCOOT) is considered to be the traffic-responsive version of TRANSYT. In both TRANSYT and SCOOT, the major objective is to minimize the sum of the average queues in the area. SCOOT collects real-time measurements (instead of historical data) from vehicle detectors and runs repeatedly a network model to examine the effect of incremental changes of cycle length, offsets, and splits. The parameters are adjusted through an iterative process

of gradient optimization. SCOOT has been deployed in many cities in the UK and overseas, Robertson and Bretherton (1991).

Leicester, Leicestershire and Rutland traffic are controlled by a Area Traffic Control Centre. In this centre, day by day traffic is managed and controlled using intelligent transport system. Currently, the systems is used to manage over 800 sets of traffic signals. Timings of traffic signal are adjusted to aid the flow of traffic. SCOOT and traffic cameras are two main data source for the system, Council (2019).

2.2.4 Performance Measures of Traffic Signal Control Systems

Several measures have been used in evaluating the quality of traffic signal control systems. These measures are all related to the experience of drivers travelling through a signalized intersection. The most popular indicators are delay and queue length.

A. Delay

Delay is the most important indicator of effectiveness evaluation at a signalized intersection. It is directly related to the amount of lost travel time, fuel consumption and the discomfort of car occupants. Delay at an intersection is measured as the extra time spent by the vehicle to pass the intersection compared to the time required to travel through the intersection without any stoppage. The total delay time of a vehicle at an intersection can be divided into acceleration delay, deceleration delay, and stopped time delay. The time loss that the vehicle takes to slow down and stop when the red signal is on, or in case there is a queue of vehicles passing through the intersection at the beginning of the green phase is the deceleration delay. The stopped delay is identified as the time a vehicle stops in the queue waiting to travel through the intersection. It is calculated as the time period from the vehicle is fully stopped until when the vehicle starts to accelerate. Acceleration delay begins when the vehicle starts to accelerate at the beginning of the green phase and ends when the vehicle gets the normal speed, which is the moving speed without any obstruction.

The accuracy of delay prediction is very important, however, it is a complex task to calculate delay because of its un-uniform arrival rate. Delay can be estimated by measurement in real traffic networks, simulation, and analytical models. Delay measurement using analytical models are simple and convenient, as a result, they have been widely

used to estimate delay at a signalized intersection. There are a number of delay models, which have been introduced to estimate average delay that a vehicle has to take at an intersection, for example, HCM 2000 delay model Board (2000) and Webster's delay model, Webster (1958). However, these models are based on some assumptions, for example, vehicles arrive at the traffic light according to a Poisson process, to simplify the complex flow conditions to a quantifiable model to approximate delay, Mathew (2014). Consequently, delay calculated using such models may not be accurate as the models are based on the theoretical concept only Mathew (2014) and the actual traffic is highly dynamic and its characteristics cannot be adequately captured by mathematical formulations, Chen and Chang (2014).

B. Queue length

Queue length is a crucial indicator, which can be used to determine whether to stop discharging vehicles from an adjacent upstream intersection, Mathew (2014). Over the years, many studies have been conducted to determine the average queue length of traffic signals. Generally, queue length estimation approaches can be divided into two types, Liu et al. (2009). The first type is based on cumulative traffic input-output, Sharma et al. (2007), Webster (1958). This type of model can only be used when the queue length is smaller than the distance between the intersection stop line and the detector installed on the road. The second type of queuing model is based on the behaviour of traffic shockwaves, Ban et al. (2011), Liu et al. (2009), Stephanopoulos et al. (1979). Shockwave theory can describe complex queueing processes but it has limitations, such as, these queuing models assume that the arrival rate of vehicles is known, which is not always satisfied, especially in congested situations.

C. Other Metrics

There are other metrics for assessing the performance of traffic signal control systems such as exhaust emissions, safety, and pedestrian level of service. In recent years, air pollution produced by vehicles is receiving increasing attention by researchers and policy makers. Tong et al. (2000) concludes that transient driving modes, for example, deceleration and acceleration, produce more emissions than the steady-speed driving modes. As a result, air pollution is often more serious at signalized intersections. Thus vehicle emissions has been considered as a metric when assessing the impacts of proposed traffic signal control systems. It is the fact that traffic safety at signalized intersections

significantly contributes to road safety in urban areas. Several strategies and tools have been developed for safety assessment in urban traffic networks, HSM (2010), Pirdavani et al. (2010). Pedestrian level of service in a signalized intersection measures its degree of pedestrian accommodation. This measure directly relates to delay experience, safety, and comfort of pedestrian crossing an intersection, and it reflects the pedestrian friendliness of an signalized intersection. A review on pedestrian level of service can be found in Kadali and Vedagiri (2016).

2.3 Traffic simulation

2.3.1 Introduction

In recent years, the rapid growth of ITS applications is generating an increasing demand for tools to support in designing and assessing the performance of proposed strategies. Traffic simulators are cost-effective tools to achieve these objectives. There are several reasons which make traffic simulators play an important role in traffic research area: (1) It is expensive and difficult to test and evaluate most proposed traffic strategies in real-world traffic networks; (2) For some studies, it is extremely difficult to establish expected traffic parameters in order to set up the experimental environment in real-world traffic networks as in simulation models; (3) Traffic simulators are a powerful tool which allows users to determine the correctness and efficiency of a proposed strategy before it is actually constructed. Therefore, the overall cost of constructing a specific strategy would be reduced significantly. Users also can use traffic simulators to compare the consequences' of a number of alternative strategies and improvement plans. Consequently, traffic simulators are one of the widely used methods in research of modelling and planning as well as the development of traffic networks and systems, Kotusevski and Hawick (2009).

Currently, there are several traffic simulation software, such as SUMO, VISSIM, MATSim, AIMSUN, and Paramics. According to the level of detail which transport simulators can represent, they are divided into three categories: microscopic, mesoscopic, and macroscopic simulators. Macroscopic simulators describe the traffic at a high level of aggregation without considering its parts. They are mainly used in traffic flow analysis. The dynamics of every single vehicle are modelled by microscopic traffic models based

```
<nodes>

    <node id = “0” x= “0.0” y= “0.0” type = “traffic_light”/>

    <node id = “1” x= “-500” y= “0.0” type = “priority”/>
    <node id = “2” x= “+500” y= “0.0” type = “priority”/>
    <node id = “3” x= “0.0” y= “-500” type = “priority”/>
    <node id = “4” x= “0.0” y= “+500” type = “priority”/>

    <node id = “m1” x= “-250” y= “0.0” type = “priority”/>
    <node id = “m2” x= “+250” y= “0.0” type = “priority”/>
    <node id = “m3” x= “0.0” y= “-250” type = “priority”/>
    <node id = “m4” x= “0.0” y= “+250” type = “priority”/>
</nodes>
```

FIGURE 2.3: The structure of the node file of a traffic scenario simulated by SUMO, Krajzewicz et al. (2019).

```
<edge id = “<ID>” from = “<FROM_NODE_ID>” to = “<TO_NODE_ID>” priority = “<PRIORITY>”>
    <lane  id = “<ID>_0” index = “<0>” speed = “<SPEED>” length = “<LENGTH>” shape = “0.00, 495.05 248.50, 495.05”/>
    <lane  id = “<ID>_1” index = “<1>” speed = “<SPEED>” length = “<LENGTH>” shape = “0.00, 498.35, 2.00 248.50, 498.35, 3.00”/>
</edge>
```

FIGURE 2.4: The structure of the edge file of a traffic scenario simulated by SUMO, Krajzewicz et al. (2019).

```
<tlLogic id = “<ID>” type = “<ALGORITHM_ID>” programID = “<PROGRAM_ID>” offset = “<TIME_OFFSET>”>
    <phase duration = “<DURATION#1>” state = “<STATE#1>”/>
    <phase duration = “<DURATION#2>” state = “<STATE#2>”/>
    ... further states...
    <phase duration = “<DURATION#n>” state = “<STATE#n>”/>
</tlLogic>
```

FIGURE 2.5: The structure of the traffic light file of a traffic scenario simulated by SUMO, Krajzewicz et al. (2019).

on the interactions between the vehicles and their neighbourhood in detail. Mesoscopic traffic models have an intermediate level of detail, for instance, describing the individual vehicle without their interactions. Microscopic traffic simulation has proven to be a useful tool to support the evaluation process of ITS's deployment, B D Venter and Barcelo (2001). Comparative studies of traffic simulators can be found at Pell et al. (2017) and Mustapha et al. (2016).

```
netconvert --node-files = hello.nod.xml -- edge-files = hello.edg.xml --output-file = hello.net.xml
```

FIGURE 2.6: The Netconvert command to generate a traffic network file of a scenario simulated by SUMO, Krajzewicz et al. (2019).

```
<routes>
    <vType accel = "1.0" decal = "5.0" id = "Car" length = "2.0" maxSpeed = "100.0" sigma = "0.0"/>
    <route id = "route0" edges = "1to2 out"/>
    <vehicle depart = "1" id = "veh0" route = "route0" type = "Car"/>
</routes>
```

FIGURE 2.7: The structure of the route file of a traffic scenario simulated by SUMO, Krajzewicz et al. (2019).

2.3.2 Simulation of Urban Mobility (SUMO)

Simulation of Urban Mobility (SUMO) is a well-known and widely used microscopic traffic simulators Kotusevski and Hawick (2009). SUMO is a microscopic traffic simulation package which is highly portable, open-source and created to handle large road networks. The development of SUMO started in the year 2000 and it is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Centre to provide the traffic research community a tool to implement and assess their own studies. SUMO is multi-modal which means that not only car movements are modelled, but also public transports, such as bus and train networks, can be included in the simulation. Due to SUMO's high portability, it may be used on different operating systems.

There are two main components to construct a traffic simulation using SUMO which are road network representation and traffic demand. The road networks represent real-world traffic network as directed graphs, where intersections and roads are represented by nodes and edges, respectively, and they are described in XML files. The nodes are declared in the node file. Figure 2.3 illustrates an example of a node file. The edges contains certain attributes such as the position, shape, and speed limit Krajzewicz et al. (2012) as shown in Figure 2.4. A SUMO network also can contain traffic lights, roundabouts and other transport components. An example of the traffic light file is provided in Figure 2.5.

```
<configuration>
    <input>
        <net-file value = "hello.net.xml"/>
        <route-files value = "hello.rou.xml"/>
    </input>
    <time>
        <begin value = "0"/>
        <end value = "10000"/>
    </time>
</configuration>
```

FIGURE 2.8: The structure of the configuration file of a traffic scenario simulated by SUMO, Krajzewicz et al. (2019).

All the information about road network are described in the net.xml file. SUMO road networks can be either generated from XML files or converted from other input data. "Netconvert" is a road network importer which is used to import road networks from other traffic simulators as Vissim, MATsim, or VISUM and produces road network that can be used by other tools in SUMO, Krajzewicz et al. (2019). Figure 2.6 describes the Netconvert command. SUMO can also read other common formats such as OpenStreetMap. The existing road network file can be edited using NETEDIT tool, Krajzewicz et al. (2019).

The second major component in SUMO scenarios is traffic demand defining routes of vehicles. The structure of a route file is provided in Figure 2.7. Routes can be generated either by using existing origin/destination matrices (O/D matrices) and convert them into route descriptions or specifying them manually. The first approach is applied mostly within the traffic science when dealing with large real-world scenarios. The second one is used when the researchers would like to have their own wishes about the traffic movements of the scenarios, Krajzewicz et al. (2012). SUMO also can import routes from other simulations. Additional information such as traffic light timing data can be integrated into the traffic simulation through additional files.

After creating network and route files, a configuration file is generated to glue every files together and the simulation scenario can be visualized in the SUMO-GUI. The structure of the configuration file of a traffic scenario simulated by SUMO is shown in Figure 2.8.

A large number of measurements can be generated for each simulation run in SUMO. The output can be unaggregated vehicle-based information such as positions and speed

for every simulation step or aggregated information of vehicles in their journeys. SUMO also provides information about simulated detectors, traffic lights, and values for lanes or edges. Besides common traffic measures, other metrics such as noise emission, pollutant emission, and a fuel consumption are also included in SUMO, Behrisch et al. (2011).

2.4 Multi-objective evolutionary algorithms

2.4.1 Definition of Multi-objective Optimization Problems and Basic Concepts

Optimization refers to maximizing or minimizing some functions to find a set of feasible solutions corresponding to optimal values of a single of multiple objectives. An optimization problem might consist of a single objective or multiple objectives. Single-objective optimization problem involves only one objective function while multi-objective optimization problems include several objective functions. The goal of optimizing a single-objective problem is to find the best solution which gives the minimum or maximum value of the problem depending on the requirement of the objective function. But for multi-objective optimization problems (MOOPs), there is often more than one optimal solution and it is complex to choose the best solution. Therefore, the decision maker has to choose one of the achieved solutions based on higher-level information. In the real world, optimization problems normally consist of multiple conflicting objectives with a number of constraints and multiple optimal solutions, namely Pareto solutions. Finding suitable trade-off solutions which provide acceptable performance over all objectives are the main aim of MOOPs.

MOOPs have a number of objectives needed to be either minimized or maximized simultaneously while satisfying the constraints. Deb (2008) states the overall form of a MOOP as follows:

$$\begin{aligned} \text{Minimize/maximize} \quad & f^m(x) && m \in [1, M]; \\ \text{subject to} \quad & g^j(x) = 0, && j \in [1, J]; \\ & h^k(x) \leq 0, && k = 1, 2, ..., K; \\ & x_L^{(i)} \leq x^{(i)} \leq x_U^{(i)} && i \in [1, n]. \end{aligned} \tag{2.1}$$

where J and K are the numbers of equality and inequality constraints, respectively, which are needed to be fulfilled. There are M objective functions in this optimization problem. Objectives in MOOPs can be continuous or discrete and linear or non-linear. x is the decision vector including n decision variables $x^{(i)}, i \in [1, n]$ while $x_L^{(i)}$ and $x_U^{(i)}$ are the lower and upper bounds for each decision variable $x^{(i)}$, respectively. These decision variables x_i can be continuous or discrete. A feasible solution is a solution satisfying all constraints and variable bound.

Here are the fundamental concepts in MOOPs, which are defined as follows, Deb (2008):

Decision variable space or *decision space* of a problem is its feasible space with all possible numerical amount that can be allocated to decision variables x_i of MOOPs.

Objective space is the space including all possible values produced by the objective functions of a MOOP.

Domination: most MOOPs use the concept of domination to compare two solutions. For two decision solutions $x^{(u)}$ and $x^{(v)}$, $x^{(u)}$ dominates $x^{(v)}$ (or mathematically denoted by $x^{(u)} \preceq x^{(v)}$) if and only if $x^{(u)}$ is strictly better than $x^{(v)}$ in at least one objective and better or equal to $x^{(v)}$ in all objectives. Domination definition can be described mathematically as:

$$x^{(u)} \preceq x^{(v)} \quad \text{if and only if} \quad x_i^{(u)} \leq x_i^{(v)} \wedge \exists i \in [1, n] : x_i^{(u)} < x_i^{(v)}, \forall i \in [1, n]. \tag{2.2}$$

Strong dominance: $x^{(u)}$ strongly dominates $x^{(v)}$ (or $x^{(u)} \prec x^{(v)}$) if $x^{(u)}$ is strictly better than $x^{(v)}$ in all objectives.

$$x^{(u)} \prec x^{(v)} \quad \text{if and only if} \quad \forall i \in [1, n] : x_i^{(u)} < x_i^{(v)} \tag{2.3}$$

Weak dominance: $x^{(u)}$ weakly dominates $x^{(v)}$ if $x^{(u)}$ is better or equal to $x^{(v)}$ in all objectives.

Non-dominated set: the non-dominated set $Q^{'}$ of a given set of solutions Q is a set including solutions that are not dominated by any solution in Q.

Pareto optimal solution: in the decision space X, a solution $x^{(i)}$ is named Pareto optimal if and only if there exists no solution $x^{(j)}$ that $x^{(j)}$ dominates $x^{(i)}$.

Pareto-optimal set: if P is the entire feasible search space, the non-dominated set $Q^{'}$ of set Q is then called the Pareto-optimal set. The Pareto-optimal set $P^{'}$ of a given MOOP $f(x)$ is defined as:

$$Q^{'} = \{x \in X | \nexists x^{'} \in X : f(x) \preceq f(x^{'})\} \tag{2.4}$$

Pareto front: the corresponding objective vectors of Pareto-optimal set are referred to as the Pareto-front. The Pareto front $PF^{'}$ of a given MOOP $f(x)$ and a Pareto-optimal set $P^{'}$ is defined as:

$$PF^{'} = \{\overrightarrow{u} = f(x) = (f_1(x), f_2(x), ..., f_n(x) | x \in P^{'}\} \tag{2.5}$$

In MOOPs, the task is to find a set of well acceptable solutions which are as close as possible to Pareto-optimal set. There are many real-life problems for which it is quite hard for the decision-maker to correctly and completely formulate them. Furthermore, all efficient solutions cannot be found out within an acceptable time in these problems. Therefore, decision-makers tend to use approximated solutions in such situations, Sanghamitra Bandyopadhyay (2013).

2.4.2 General Framework of Multi-objective Evolutionary Algorithms

Multi-objective Evolutionary Algorithms (MOEAs) imitate principles of nature's evolutionary process including reproduction, mutation, recombination, and selection to find multiple well acceptable solutions. Several characteristics of MOEAs are desirable for MOOPs and it has been used to solve MOOPs for more than one decade, Zitzler et al. (2004). One of the critical differences between classical search methodologies and MOEAs is that MOEAs use a set of potential solution candidates, namely population, in each iteration, instead of a single solution. This population is then transformed by the selection and variation principles. The first principle, selection, imitates the competition for reproduction among living beings in nature. The other one, variation, mimics

Algorithm 1 Principal steps of a MOEA framework

1: *Randomly initialize a population*
2: **While** *termination conditions are not satisfied*
3: *Step 1: Mating selection*
4: *Step 2: Offspring generation*
5: *Step 3: Environmental selection*
6: *Step 4: Check the termination conditions*
7: **Return** *Non-dominated set of solutions*

the natural ability to create "new" living beings using recombination and mutation. Although their working mechanisms are simple, MOEAs are proven to be robust, general, and powerful search approaches.

The fundamental principle of MOEAs is that it applies the principle of survival of the fittest to produce the next generation of solutions. The fittest individuals have a greater chance of survival than weaker ones. MOEAs randomly initialize a population of solutions, which are also called individuals, and then iteratively undergo four main steps which help to gradually increase the quality of the population and direct the solutions toward the Pareto front. Each iteration is also called a generation, and in most studies, a pre-defined maximum number of iterations is used as the termination condition of the loop. The basic principal steps of an MOEA framework are illustrated in Algorithm 1 and explained in the following sections, Cheshmehgaz et al. (2015).

Mating selection aims at choosing promising solutions for reproduction. The mating selection consists of two stages: fitness assignment and sampling. A fitness assignment strategy is needed in MOEAs to give a fitness value or a rank to solutions in the population based on their objective functions and constraints. It makes solutions become comparable to other solutions. In general, fitness assignment strategies can be classified into Pareto-based fitness assignment strategies, criterion-based, and aggregation-based, Konak et al. (2006). In the fitness assignment stage, each individual in the current population is evaluated using objective functions and then is assigned a fitness value, reflecting its quality. After that, a so-called mating pool is created in the sampling stage using mate-selection strategies. Roulette wheel selection, introduced by Holland (1992), and binary tournament selections, proposed by Goldberg (1989), are two common mate-selection strategies. Roulette wheel selection method selects an individual proportional to its probability which directly depends on its fitness. The implementation of this wheel selection follows a roulette-wheel mechanism. The percentage fitness values of solutions

can be used to configure the roulette wheel. Consequently, the fittest solution has the largest proportion in the wheel. Thereafter, the wheel is spun N times, where N is the population size. At each spin, the solution pointed by the pointer is selected. In binary tournament selection strategy, two solutions are randomly selected from the population and the solution with better fitness value is picked out and placed in the mating pool. Thereafter, two other random solutions are pick again and the better solution is selected to be filled in the pool. This procedure is ended when the mating pool is filled.

Recombination and mutation operators are applied to individuals in the mating pool to generate offspring. The recombination operator combines parts of parents pairs to create a pre-defined number of children using a crossover probability. By contrast, mutation operator changes one or more variables in an solution based on a pre-defined mutation rate. The mutation operator is used to preserve the diversity of the population from one generation to the next generation, Deb (2008). Mutation also helps the search to overcome local minimum by preventing solutions in the population from becoming too similar to each other.

After the process of generating offspring has been completed, the environmental selection is used to decide which solutions in the population and newly created children are selected to form a new population for the next generation. Thereafter, termination conditions are checked and the procedure is terminated if the conditions are satisfied. If not, the loop is continued with the new population created in the previous step.

Jones et al. (2002) showed that 90 % of multi-objective optimization approaches try to approximate the optimal Pareto front for the problem, and 70 % of all meta-heuristics approaches were based on evolutionary approaches. Comprehensive reviews of MOEAs can be found in Zhou et al. (2011) and Cheshmehgaz et al. (2015).

There are two fundamental goals of MOEAs, which are finding a set of solutions as close as possible to the Pareto-optimal front and finding a set of solutions as diverse as possible, Deb (2008). The first goal, convergence speed, is mainly related to mating selection strategies, in particular to fitness assignment methods. The second goal, diversity of solutions in the non-dominated front, is related to selection schemes. If the density of individuals surrounding a particular solution is large, the probability that the solution is selected decreases, Zitzler et al. (2004). Furthermore, the elitism mechanism, which is utilized to preserve the fittest solutions selected from the population and newly create

offspring, so they do not get removed from the population, might have negative affects on the diversity of the population in MOEAs.

2.5 Surrogate-assisted evolutionary algorithms

2.5.1 Evolutionary algorithms vs. surrogates-assisted evolutionary algorithms

Evolutionary Algorithms (EAs) have been very successful for solving optimization problems with multiple objectives in both academia and industry. In general, evolutionary algorithms defeat traditional optimization algorithms for many problems especially discontinuous, not well-defined, multi-model, and noisy problems, Jin (2005). However, EAs have encountered challenges when applying to real-world applications. One major challenge is that it needs a large number of objective evaluations to find good solutions. In many real-world applications, it is difficult or computationally expensive to perform large number of fitness evaluations. It is not uncommon that a single simulation process, which is utilized to evaluate the fitness value of an individual, take minutes, hours or even days to compute. Consequently, many simulation scenarios only allow for a fairly limited number of evaluations using the real fitness function. Such type of problems is called expensive optimization problem. In such situations, approximation models, also known as surrogate or meta-models, have been adopted to predict the fitness values of solutions. Surrogates are computational models used to estimate the fitness values of solutions at a cheaper cost compared to the original fitness function. The main aim of using meta-models is to decrease the total number of evaluations conducted by original fitness functions while remaining a reasonable good quality of the results achieved.

Surrogate-assisted evolutionary algorithms were proposed to decrease the computational cost of fitness evaluation in optimizing expensive problems, Jin (2011). Using an approximation model to estimate fitness values greatly reduces computational cost since the costs required to construct the approximation model and to use it are much lower than those in the standard EA which directly evaluate all individuals using costly objective functions. Surrogate-assisted evolutionary computation has been applied successfully in many real-world applications. Surrogates can be applied to most operations of evolutionary algorithms, for example, fitness evaluation, population initialization, crossover,

and mutation. The approximation model and the original fitness function should be used together as it is difficult to achieve an approximation with very high accuracy. Surrogate models should be combined with the real objective function to help the evolutionary search avoid obtaining a false minimum introduced by the surrogate model. Model management or evolution control is a strategy to use surrogate models properly and efficiently.

Multi-objective optimization of transportation is expensive as a traffic simulation needs to be run every time a solution's fitness is evaluated. Therefore, surrogate-assisted evolutionary algorithms are promising for solving multi-objective optimization problems in transportation.

2.5.2 Strategies for managing surrogates

2.5.2.1 Model management: its roles and classification

It is very difficult to achieve an approximation with very high accuracy due to the lack of available input data. It is emphasized in Jin (2005) that if only the surrogate model is utilized to estimate the fitness values, the evolutionary search will likely converge to a false optimum. Consequently, it is very important that the approximation model should be combined with the real objective function. In most cases, the real objective function is available although its computation is costly. Hence, the original fitness function should be used effectively to reduce the computational cost. This is known as model management or evolution control. According to Jin (2005), model management can be divided into three main groups:

(1) No evolution control: the surrogate is assumed to be highly accurate and it completely replaces the original objective function in the evolutionary computation.

(2) Fixed evolution control: a fixed number of solutions whose fitness values are calculated by the approximate model and the others are evaluated using the original objective function. The evolution control consists of three approaches:

A. Individual-based: in each generation, some of the individuals are evaluated by the real objective function and the others use the surrogate for fitness calculation.

B. Generation-based: fitness values of solutions in some generations are estimated by surrogates while in the other iterations, the original fitness function is used.

C. Population-based: there are more than one sub-population taking part in the evolution process. Each sub-population use its own surrogates to approximate fitness values.

(3) Adaptive evolution control: in this type of model management, the frequency of using the approximation model should be determined by its fidelity and it can be adaptive during the optimization process. The more accurate the surrogate is, the more frequently the surrogate is utilized to predict the fitness value of solutions.

2.5.2.2 Criteria for choosing individuals for re-evaluation

One of the most important questions when using surrogates is which solutions should be selected to be estimated by the surrogate and which solutions will be re-evaluated using the original objective function. This selection is strongly related to another question: how to adjust the number of the solutions to be evaluated using an original objective function. The main aim is to minimize the number of fitness evaluations using the original objective function while retaining the accuracy of the optimization process as the algorithm can still converge to the global optimum. Distribution of the available samples is determined by the re-evaluation selection scheme. Hence, choosing accurately and properly the individuals to be estimated using the original fitness function will help the surrogate learn the underlying relationship between input and output of the samples more accurately with less samples. As a result, the number of solutions to be re-estimated will be reduced and the the approximation error will be decreased more quickly. Thus, time to run the optimization process will be decreased. There are a number of strategies for choosing individuals for re-valuation, which are described as follows:

Random strategy: individuals are selected randomly to be evaluated by original objective function Fonseca et al. (2012).

Best strategy: The most straightforward method for selecting solutions for re-evaluation is to evaluate solutions whose potentially produce a good fitness value and

the more accurate the approximation model, the more individuals should be evaluated using surrogates, L. Graening (2005).

Uncertain strategy: choose individuals, which have large degrees of uncertainty in approximation, to be evaluated using original fitness functions, Branke and Schmidt (2005), Emmerich et al. (2002). There are two reasons explained for choosing uncertain solutions to be re-evaluated. Firstly, a high level of uncertainty in approximating fitness values suggests that the objective space around these individuals has not been adequately modelled. Consequently, there might be a high opportunity of finding a better solution in this part of the landscape. Second, re-estimation of most uncertain individuals may be an effective way to improve the approximation accuracy of the adaptive surrogate.

Representative strategy: the solutions are classified into several clusters. Representative solutions in each cluster, such as the individual nearest to the center of cluster or the best solution in each cluster L. Graening (2005), are selected to be evaluated using the original objective function.

2.5.3 Techniques for constructing surrogates

To construct a surrogate, a set of samples is needed. The approximation accuracy of surrogates depends on the number of available samples given in the search space and the selection of the appropriate approximation model utilized to represent the original objective functions. There are a variety of approximation models including polynomials (often known as Response Surface Methodology) Goel et al. (2007), Husain and Kim (2010), Liu et al. (2008), Support Vector Machines Basudhar et al. (2012), Bourinet (2016), Rosales-Perez et al. (2015), Kriging model Liu et al. (2014), Pan and Das (2015), Zhou et al. (2007), and Artificial Neural Networks Bhattacharjee et al. (2016), Jin et al. (2015), Sun et al. (2013). An overview of techniques used for constructing surrogates in multi-objective evolutionary optimization can be found in Santana-Quintero et al. (2010). A number of studies have been conducted to compare the performance of these approximation models Diaz-Manriquez et al. (2016), Jin et al. (2001), Jin (2005). There are no clear conclusions about which model is definitely superior, to the others. When choosing an approximation model, more than one criterion should be considered, such as approximation accuracy, efficiency, computational cost, and complexity. It is difficult to give specific rules on selecting approximation model. It is suggested that firstly, a simple

meta-model should be used for a given problem. If the accuracy is not satisfactory, a more complex approximation model should be considered. If the number of available samples is limited and the design space is highly-dimensional, a neural network model is recommended, Jin (2005). In transportation optimization problems using a traffic simulator to evaluate candidate solutions, the number of available samples using to construct a surrogate model is usually kept small as running a large number of simulations is time-consuming. Therefore, artificial neural networks are promising to construct a approximation model.

2.5.4 Artificial Neural Networks

Artificial Neural Networks (ANNs) are used to learn the relationship between inputs and the corresponding outputs. ANNs have shown to be effective tools for function approximation. Multilayer feed-forward perceptron networks have been widely applied to approximation problems.

A. Multilayer feed-forward perceptrons (MLPs) are a class of multilayered feed-forward artificial neural network. There are at least three layers of nodes in an MLP which are one input layer, one output layer, and one or more hidden layers. Each connection between nodes has a weight, which is randomly initialized at the beginning and is adapted during the training process. Each neuron takes the weighted sum of signals coming from the previous layer. There are several critical points needed to be determined when building an ANN: the structure of the ANN including the number of layers and the number of nodes in each of these layers, the selection of input and output, and the training algorithm, Santana-Quintero et al. (2010). The output of a neuron is:

$$y = f(\sum_{i=1}^{m} w_i x_i + b) \tag{2.6}$$

where x_i is the $i^{(th)}$ input and y is the output of the neuron, w_i is the weight of the connection between the i^{th} input and the neuron. The activation function f is a nonlinear function and one of the most commonly used activation function is the sigmoid function.

B. Overfitting and underfitting are common problems in machine learning, which can lead to an inefficient performance of approximation models. Overfitting happens when

the model learns the noise instead of the signal, which is the actual pattern expected to learn by the model from the samples. Consequently, the approximation model will work unusually well on the training data but very poorly on unseen samples. In contrast, an underfitting model refers to a model which can not model the training data as well as generalize the unseen data. To limit overfitting and underfitting, a resampling technique is recommended to estimate model accuracy. The most popular resampling technique is k-fold cross validation. It is commonly applied in machine learning as it is easy to implement, and its results generally have lower bias than other methods, Rodriguez et al. (2010).

C. Bias and Variance are sources of prediction errors for any machine learning algorithms. Bias, which are assumptions made by a model, is used to simplify the target function and make it easier to learn. Low bias means fewer assumptions and high bias means more assumptions about the form of the target function. High bias can lead to the missing the relevant relations between the inputs and the outputs, which causes underfitting phenomenon. Bias error is an error caused by erroneous assumptions. On the other hand, variance is the amount of difference between the outputs of the approximation model using different training data sets. Variance suggests the degree of dependence of the approximation model to the training data sets. If the variance is low, the estimate of the model does not change significantly when using different training dataset. High variance means the estimation result is sensitive to the change of training data. Ideally, the variance should not be high, meaning that the approximation model can learn the hidden underlying relationship between the inputs and the corresponding outputs. The objective of supervised learning algorithms is to obtain low variance and low bias. However, there is a trade-off between these two concerns as decreasing bias will increase the variance and vice versa, Geman et al. (1992).

D. Fine-tunning hyperparameters: one of the difficulties when working with neural networks is selecting an optimal architecture for a specific problem. Hyperparameters are parameters which determine the overall architecture of a neural network and they are usually determined before starting the training process. Some examples of hyperparameters are number of hidden layers, the learning rate, and the number of neurons in each hidden layer.

Hyperparameter optimization or tuning is to find an optimal set of hyperparameters for

a learning algorithm. Grid search is a technique used to optimize hyperparameters. Grid search is an exhausted search, which tests all possible combinations of hyperparameters. Grid search is simple and straightforward to implement Pontes et al. (2016). Although computational time required by grid search technique may be longer than that of other techniques, however, grid search can be easily parallelized as each combination of these hyperparameter values are independent. Therefore, currently grid search is one of the most widely used methods for hyperparameter optimization, Pontes et al. (2016), Zhang et al. (2009).

Tamura and Tateishi (1997) showed that a feed-forward neural network consisting of 2 hidden layers is superior to a feed-forward neural network containing one hidden layer for learning the pattern of the training data set. Furthermore, a neural network with two hidden layers would be capable of approximating any non-linear function and there is no need to use a neural network with more than two hidden layers, Heaton (2008). The sizes of hidden layers are also critical to the decision of the overall architecture of a neural network. If there are too few neurons in the hidden layers, under-fitting might be happened. Otherwise, using too many neurons might lead to over-fitting. A survey of methods to define the number of hidden neurons in a neural network is introduced in Sheela and Deepa (2013).

Surrogate-assisted evolutionary algorithms have been applied in optimising expensive problems. Surrogate models are utilized to decrease the fitness evaluation cost of candidate solutions. Therefore, surrogate-assisted evolutionary algorithms may work well on traffic signal optimization problems. A multilayer feedforward neural network is probably an efficient approximation model which can be used to predict the fitness value of solutions in the evolutionary process of traffic signal optimization problems. Its hyperparameters are fine-tuned by the grid search and k-fold cross-validation techniques to avoid the over-fitting and under-fitting, Fushiki (2011), Rodriguez et al. (2010).

2.6 Conclusion

Traffic light control systems for signalized intersections are one of the most important components in urban traffic control systems. Its effectiveness directly affects the safety of the participants and efficient operation of the traffic control system. Consequently,

their performance optimization is one of the main aims when designing a traffic signal control system. Several hypotheses related to traffic signal optimization are formulated in Chapter 1. Before evaluating their quality, general knowledge about relevant components is required. Therefore, this chapter provided a general background of traffic signal control systems, traffic simulators and computational intelligence techniques used to optimize the performance of traffic signal control systems, such as multi-objective evolutionary algorithms and surrogate-assisted evolutionary algorithms.

Chapter 3

Literature Review

3.1 Multi-objective Traffic Signal Optimization

3.1.1 Introduction

Traffic signal control is critical to urban traffic management as its performance directly affects the efficiency of the traffic system. Recently, optimization approaches have been utilized in traffic control models to increase the performance of traffic signal control systems. The main aim of a traffic signal optimisation is to significantly improve the performance of the traffic intersection by minimising the delay, queue length, the number of stops, emissions and maximising the traffic flow and average speed in the network. Setting traffic signal timing in a signal-controlled street network involves the determination of cycle time, splits of green time, and offsets. Traffic signal optimization might optimize a part of or all these values based on observed traffic parameters, such as flow and queue length. A single or multiple objectives might be involved in traffic signal optimization models.

Compared to conventional search methods, for example, random search and hill-climbing approaches, Multi-Objective Evolutionary Algorithms (MOEAs) are more robust and speedy Guangwei et al. (2007), as a result, MOEAs have been widely used to solve the multi-objective optimization problems. A number of MOEAs also have been widely deployed in traffic signal optimization such as Non-dominated Sorting Algorithm II Nguyen et al. (2016), Shen et al. (2013,?), Yan et al. (2013), Genetic Algorithm Abushehab et al.

(2014), Ben et al. (2010), Chen and Chang (2014), Tung et al. (2014), and Particle Swarm Optimization, Abushehab et al. (2014), Dong et al. (2010), Kai et al. (2014).

The main aim of a local search method is to find out a local optimum. The local search method performs a sequence of changes from an initial solution in the neighbourhood, which helps to increase the quality of the solutions in term of objective values, until the local optimum is found, Mladenovic and Hansen (1997). There are several studies optimizing urban traffic signal control problems using local search based MOEAs, Gao et al. (2016), Sabar et al. (2017).

3.1.2 Traffic Signal Optimization using MOEAs

In recent years, many studies using computational intelligence technologies have been introduce to optimize the performance of traffic light signal control systems. Computational intelligence methods for urban traffic signal control were reviewed in Zhao et al. (2012) and Araghi et al. (2015). MOEAs are well-known optimization techniques and they have been applied in traffic signal optimization problems. Table 3.1 shows a number of studies on traffic signal optimization and their corresponding evolutionary algorithms selected to optimize the objectives. As we can see from Table 3.1 that among MOEAs, Non-dominated Sorting Algorithm II (NSGA-II) and Genetic Algorithm (GA) are the most two popular algorithms.

In Zhou et al. (2008), a bi-level optimization model based on GA was proposed to increase the traffic quality and reduce emissions at the intersection. Qun (2009) introduced a signal control model for urban intersection based on GA. Ben et al. (2010) used GA to optimize several objectives concurrently by aggregating of different objectives. The objective function is calculated by the sum of different objectives. The proportion of each objective in this formula is indicated by its weighting factor, which represents the importance of that objective. Chin et al. (2011) implemented a traffic signal timing management system for coordinated intersections based on GA. Chen and Chang (2014) introduced a traffic light signal optimization approach for heavy mixed traffic flows of arterials using a GA-based approach and a gauzy branch-and-bound method. Link length and vehicle size are both considered in formulating traffic evolution, queue formation, allowing preventing queue spill-back.

TABLE 3.1: Evolutionary algorithms in traffic signal control systems.

No.	Evolutionary Algorithms	References
1	Genetic Algorithm	Guangwei et al. (2007), Zhou et al. (2008), Qun (2009), Ben et al. (2010), Shen et al. (2011), Chin et al. (2011), Passow et al. (2012), Abushehab et al. (2014), Chen and Chang (2014), Tung et al. (2014)
2	Non-dominated Sorting Algorithm II	Sun et al. (2003), Feng and Xiaoguang (2008), Yan et al. (2013), Shen et al. (2013), Nguyen et al. (2016), Armas et al. (2017), Mihaita et al. (2018)
3	Particle Swarm Optimization	Chen and Xu (2006), Dong et al. (2010), Kai et al. (2014), Abushehab et al. (2014)
4	Differential Algorithm	Zhang et al. (2009), Kai et al. (2014)
5	Memetic Algorithm	Sabar et al. (2017)
6	Harmony Search	Gao et al. (2016)

Sun et al. (2003) considered the ability of NSGA-II in optimizing traffic signal timing and the result demonstrated that NSGA-II is a very promising algorithm. In addition, the authors shows that NSGA-II can find an approximated optimal set with a good spread and high convergence speed. Feng and Xiaoguang (2008) combined NSGA-II and a cell transmission model to construct an urban intersection traffic signal control system. The proposed algorithm is compared with three other signal timing algorithms which are Webster, Synchro, and TRANSYT and the results showed that the proposed methodology has smaller mean delay than the other algorithms. Yan et al. (2013) managed the traffic flow at an isolated intersection under over-saturated conditions using NSGA-II. A hybrid of NSGA-II and local search was introduced in Nguyen et al. (2016) to improve anytime behaviour of traffic signal optimization systems.

A population-based stochastic optimization technique, namely Particle Swarm Algorithm (PSO), is inspired by the social behaviour of fish schooling or bird flocking. PSO has a number of similarities with genetic algorithms. PSO has some advantages compared with other MOEAs, for examples, PSO has fewer parameters to adjust and is easier to implement than other MOEAs. PSOs have been successfully implemented in a number of research areas. Some researchers have proposed an enhancement of PSO for the traffic light signal timing optimization problems. Chen and Xu (2006) examined the

ability of PSO in different traffic demands by a combination of PSO and fuzzy logic while Dong et al. (2010) constructed a new multi-objective optimization by a combination of PSO and the Simulated Annealing algorithm (SA) which is a random optimization algorithm. Kai et al. (2014) proposed a collaborative strategy using PSO and Differential Algorithm (DE). The study's results have showed that the proposed strategy is superior to PSO in terms of the average delay time.

Differential Algorithm (DE), another MOEA, has been also applied to solve the optimal configuration of a traffic signal control system. A real-time traffic flow control was implemented in Zhang et al. (2009) using multi-objective discrete differential evolution algorithm. Based on the experimental results, the authors concluded that the proposed method had better performance.

An optimization framework was implemented in Armas et al. (2017) for a large scale traffic network using an evolutionary algorithm and clustering technique. A number of specialized mutation operators were defined. Coordinated signals with similar cycle length are searched using the mutation operators. Different mutation rates were also utilized in this approach to accelerate the convergence rate of the evolutionary search. A multi-objective optimization method for urban intersections using evolutionary algorithm was proposed in Mihaita et al. (2018) for an intersection under reconstruction in Nancy. Moreover, an integrated framework of the optimization approach and a 3D mesoscopic traffic simulator was also introduced in this method.

Among MOEAs, NSGA-II is an effective algorithm and its performance has been indicated in Sun et al. (2003). This study shows that NSGA-II provides better performance than other multi-objective evolutionary algorithms. That is the reason why it was chosen to be the optimization algorithm in a number of studies.

3.1.3 Multi-objective Traffic Signal Optimization using Local Search based MOEAs

Population-based computational intelligence algorithms, such as NSGA-II and GA, present a higher performance than the traditional methods. However, despite good

results provided by population-based computational intelligence algorithms, it is well-known that population-based-algorithms frequently suffer slow convergence. Consequently, they might not be suitable for some real-time problems, Neri and Cotta (2012), Ong, Lim, Zhu and Wong (2006). To address this issue, Sabar et al. (2017) implemented an adaptive memetic algorithm for traffic signal optimization problems. A local search was introduced in the study to effectively explore the local search space around solutions. The main search algorithm uses GA to guide the search to move towards the Pareto-optimal front. The major role of the local search method is to speed up the convergence rate of the GA. Consequently, higher quality solutions will be obtained. The local search introduced in this work is based on the general rules of the simple descent method. From a given solution, the local search keeps searching for improvements until the termination conditions are satisfied. The current solution is modified to create a new one using a systematic neighbourhood operator at each generation of the local search. If the new solution is found to be better than the current solution in term of fitness value, it replaces the current solution. Otherwise, the local search moves to the next area and keep searching. The experimental result showed that this algorithm is superior to GA and fixed-time signal control method.

A combination of three local search operators with different structures was introduced in Gao et al. (2016) to increase the performance of the global search approach, which was based on the discrete harmony search (DHS) algorithm. A neighbourhood structure was divided into three categories and three corresponding local search operators had been constructed. The first local search operator deals with only one single intersection while the second operator is for coordinated intersections and the third one is for a sub-region of the whole traffic network. A local search strategy, integrating these three operators, was defined to find neighbouring solutions and help to improve the performance of the DHS. The proposed algorithm was compared to the DHS without local search operators on urban traffic signal problems. The results of the study indicate a better performance of the local search based DHS.

Another meta-heuristic algorithm with a combination of local-search operators was also introduced by Gao et al. (2017) to minimize delay of both pedestrians and vehicles. Artificial bee colony algorithm with a combination of local-search operators was utilized to solve the traffic light signal optimization problem. Eight real-life database cases have

been used to evaluate the performance of the proposed algorithm. The results verified that it outperforms NSGA-II for solving traffic signal optimization problems.

In conclusion, hybridizing a local search and a global evolutionary algorithm may accelerate the convergence speed of the search. Furthermore, Espinoza et al. (2003) shows that local search also help reduce the population size of the optimization algorithm. Therefore, a combination of an evolutionary algorithm and a selective use of local-search can improve the efficiency of a traffic signal optimization system.

3.2 Objectives in Traffic Signal Optimization

A number of objectives has been optimized in traffic signal control systems including reducing queue lengths and delay at signalized intersections, decreasing the travel time, increasing traffic flow and reducing traffic exhaust emissions. Table 3.2 illustrates objectives, which have been optimized in traffic signal optimization approaches. Among these targets, delay reduction, travel time decrease of flow improvement can be achieved concurrently. If the delay is reduced, travel time will be decreased and flow will be increased consequently. However, environmental target sometimes conflicts with other objectives and have not received an adequate attention. A significant portion of researches and applications of optimization in transportation management considers problems with a single objective. However, more than one objective are usually involved in the real-world problems. Further information about the optimization objective in traffic signal control systems is introduced in the following.

3.2.1 Optimization Objectives in Traffic Signal Control

A. Reducing traffic delay is one of the most critical objectives in traffic management as it directly relates to travel time, fuel consumption and discomfort of drivers. Consequently, reducing delay at intersections is the main and fundamental objective in traffic signal optimization. The first signal timing optimization program was introduced for an isolated intersection by Webster and Cobbe (1966) to minimize the traffic delay using the Webster formulation. Guangwei et al. (2007) introduced a formula of the average delay of all vehicles travelled through the intersection. This formulation is based on the information of individual vehicles obtained from a microscopic traffic simulator. The

TABLE 3.2: Optimization objectives in traffic signal optimization using MOEAs.

No.	References	Optimization Objectives					
		Stops	Flow	Delay	Emission	Queue	Time
1	Guangwei et al. (2007)			√			
2	Feng and Xiaoguang (2008)			√	√	√	
3	Fang and Elefteriadou (2008)					√	
4	Zhou et al. (2008)				√		√
5	Qun (2009)			√			
6	Zhang et al. (2009)					√	
7	Ben et al. (2010)				√		
8	Sanchez-Medina et al. (2010)		√		√		√
9	Kouvelas et al. (2011)		√				
10	Shen et al. (2011)		√				
11	Chin et al. (2011)			√			
12	Adacher (2012)			√			
13	Passow et al. (2012)			√	√		
14	Shen et al. (2013)	√		√			
15	Yan et al. (2013)		√			√	
16	Chen and Chang (2014)		√	√			
17	Kai et al. (2014)			√			
18	Abushehab et al. (2014)						√
19	Tung et al. (2014)						√
20	Zakariya and Rabia (2016)			√			
21	Gao et al. (2016)			√			
22	Sabar et al. (2017)			√			
23	Armas et al. (2017)				√		√
24	Mihaita et al. (2018)						√

weighting factor, which is assigned for transit vehicles, is taken into account when calculating the average delay. Feng and Xiaoguang (2008) obtained desired objective values, such as overall delay, queue length, and travel time, by using the Cell Transmission Model (CTM). Webster delay model is adopted in Qun (2009) to construct the function of the average delay per vehicle. After that, a formula of the total delay is derived and minimized. The total delay on each direction in Adacher (2012) is obtained from the TRANSYT and the proposed algorithm minimized the linear combination of the total delays in every direction. Shen et al. (2013) minimized both the average stop time and the average delay time. A macroscopic traffic simulation model is used in Chen

and Chang (2014) to formulate the total delay function and the total throughput of the traffic scenario. The total delay time in Kai et al. (2014) was derived from output files generated by SUMO and the optimization approach, proposed by the authors, minimized the average vehicle delay, which is determined by the total delay time divided by the total vehicles. Two different formulas of the cycle length were proposed in Zakariya and Rabia (2016) to minimize the delay of the intersection. The first cycle length formula is recalibrated from the Webster's minimum delay cycle length formula which is introduced by Webster to estimate the optimal cycle length of an isolated signalized intersection. The second model is the exponential function of the non-linear regression model. Another way of calculating traffic delay was described in Sabar et al. (2017), which is the difference between the free-flow travel time and the estimated travel time needed to finish the route. The fitness function is, therefore, a time-dependent form derived from the travel time Davidson's function which is used to predict travel time based on several parameters, for example, traffic volume, free flow travel time, and the capacity.

B. Reducing queue length is another major objective that has been used in traffic signal optimization approaches. How to calculate queue lengths in real-time at signalized intersections is a long-lasting problem and it becomes more difficult under over-saturated conditions, Wu and Yang (2013). The objective function in Fang and Elefteriadou (2008) was the sum of average queues per lane for all approaches. Each approach is assigned a weight and its average queue length per lane is the sum of the initial queue at the beginning of the interval and the number of vehicles that arrived during the interval minus the number of discharged vehicles. In Zhang et al. (2009), the sum of the vehicles waiting for leaving the queue which is derived from a real-time traffic flow model was minimized. The queue ratio maintenance optimization function in Yan et al. (2013) was defined based on the average queue length per phase, which was derived from the output of the traffic scenario simulated by VISSIM.

C. Increasing traffic flow is one of the main objectives of traffic signal optimization systems. Traffic flow rate is a parameter of a traffic network. It is defined as the number of vehicles which passed a reference point in a given period of time. In Kouvelas et al. (2011), traffic flow measurement was obtained from the traffic simulated scenario, which has available detectors installed on links. Both Sanchez-Medina et al. (2010) and Shen et al. (2011) simply defined the traffic flow as the total number of vehicles that left the

traffic network during the simulation and considered the traffic flow as the objective function. Yan et al. (2013) maximized the weighted vehicle number of the intersection. Each movement is allocated a weight coefficient and its actual traffic flow is a function of cycle length, green time, and time. All these parameters are derived from the traffic simulator.

D. Reducing traffic exhaust emissions is becoming a critical criterion when designing traffic signal optimization systems because air pollution and other detrimental impacts on the environment from the transport sector have greatly impacted to the society. Pollution emissions from vehicles can be measured by emission models which are generally classified by macroscopic models and microscopic models. There have been a number of studies introduced optimization models for traffic signal control to mitigate the damage of vehicle emissions. Zhou et al. (2008) considered the total exhaust emission as the sum of the idle emission and the running emission of each entrance link of the intersection. Traffic flow and length of links are included in the formula of the total emission. The idle emission is estimated based on the standard idle emission factor, traffic flow and the traffic delay of the link, which was calculated based on Webster delay model. Emission value was calculated using the International Vehicle Emissions (IVE) model in Ben et al. (2010). Air quality information in Passow et al. (2012) was obtained from monitors, models, and satellites. Although researchers often put the environmental targets after the traffic demand requirements but environment-related objectives are becoming a major concern of research of the transportation sector, especially in traffic signal control.

E. Reducing travel time is another objective in traffic signal optimization. Abushehab et al. (2014) carried out a number of experiments to determine algorithms and their corresponding parameters which are more suitable in traffic signal optimization to produce the minimum total travel time. Sanchez-Medina et al. (2010) defined mean travel time as the average time which a vehicle takes to finish its route. Travel time in Zhou et al. (2008) was designed as the sum of the running time and the delay at intersections. The running time function used by the author is the BRP function which is developed by Bureau of Public Road while the delay time is determined by the Webster model. Average travel time for all vehicles to finish their routes was defined as the fitness value in Tung et al. (2014).

Recently, studies in traffic signal optimization normally consider multiple objectives,

as a result, MOEAs are one of very promising algorithms to simultaneously optimize multiple objectives. In MOEAs, solutions are compared using their fitness value which is assigned based on the objective value. Consequently, objective calculation is very important. The next section introduces methods to calculate fitness value of candidate solutions.

3.2.2 Objective Calculation using Mathematical Programming Methods

In traffic signal optimization, to assess the performance of a solution, the traffic signal system attempts to determine a connection between traffic signal parameters, such as cycle length, phase sequence, and green split, with estimated traffic parameters, for example, queue length, flows, and emissions. According to Chen and Chang (2014), traffic signal timing optimization can be classified in two main categories: mathematical programming methods and simulation-based approaches. The former scheme utilizes mathematical formulations to capture the characteristics of traffic flow models which is then utilized to optimize objectives in traffic management. Zhang et al. (2009) assumed that the arrival rate of the traffic flow follows the Poisson distribution and a real-time traffic flow sequences generating program is developed to simulate the objective function based on the number of vehicles waiting for release, the arrival rate, and the vehicle discharge rate. A delay model based on the platoons representation is used to estimate the total delay in Adacher (2012). The objective function in this approach is then constructed as a non-linear function of delay on each link approaching and leaving the intersection. However, the calculations of these mathematical models are often very complicated and difficult to meet real-time requirements Zhao et al. (2012) as the assumptions of the mathematical models are strict and therefore it is hard to model the real-time transport characteristics. Furthermore, the interrelationship between the traffic flows of complex intersections, such as queue spillback or blockage between through and turning lanes, cannot be adequately captured by mathematical programming formulations, Chen and Chang (2014). On the other hand, the simulation-based approaches aim at capturing the complex interactions between traffic characteristics. For that reason, more recently, researchers tend to optimize traffic signal timing by using simulation-based approaches such as combining multi-objective optimization algorithms with traffic simulators. This combination has been successfully applied to optimizing traffic signal timing in the real

world and assist decision makers evaluating and finding optimal values of traffic parameters for traffic management, Armas et al. (2017), Mihaita et al. (2018), Papatzikou and Stathopoulos (2015).

3.2.3 Objective Calculation using Simulation-based Methods

A number of different traffic simulators have been utilised in traffic signal optimization. Traffic simulators can be divided into three main categories which are microscopic, macroscopic and mesoscopic, Barcelo (2010). Macroscopic simulators describe the traffic at a high level of aggregation without considering its parts. They are mainly used in traffic analysis. The dynamics of every single vehicle in microscopic traffic models are modelled based on the interaction between the vehicle and other vehicles in the traffic stream. Mesoscopic models have an intermediate level of detail, for instance, describing the individual vehicle without their interactions. All these types of traffic simulators have been utilized in traffic signal optimization.

Papatzikou and Stathopoulos (2015) used DTALite, which is an open-source queue-based mesoscopic traffic simulator, to simulate the traffic scenario and measure the objective value of solutions. Mihaita et al. (2018) constructed a 3D mesoscopic traffic simulator using data from the traffic management center to test the algorithm proposed for improving the traffic flow and reducing travel time. Armas et al. (2016) simulated a traffic scenario using Multi-agent Transport Simulator (MATSim) to evaluate solutions generated by an evolutionary algorithm.

Chen and Chang (2014) models the interrelation between the formation of queues and movement of vehicles through a macroscopic traffic-flow model. Traffic characteristics achieved from the traffic model are utilized to produce a near-optimal set of signal timing plans via the optimization algorithms. Furthermore, this strategy may be used on-line with real-time data obtained from sensors which are installed on the roads. The macroscopic simulation tool METANET is utilized in Poole and Kotsialos (2016)) to model a traffic network, which is used for evaluating the proposed optimization algorithm. Macroscopic traffic simulators are good at describing overall traffic properties but lack the flexibility to simulate complicated behaviours, such as queue spillback or blockage. Consequently, microscopic simulators become more and more popular and have been widely used to simulate traffic scenario and to evaluate new solutions.

PARAMICS, a microscopic simulator, was used in Srinivasan et al. (2006) to simulate a traffic section in a business centre and return measurements of traffic parameters. Sanchez-Medina et al. (2010) developed a new cellular-automata based micro-simulator for simulating traffic behaviour and evaluating every solution. Fang and Elefteriadou (2008) selected AIMSUN, which stands for Advanced Interactive Micro-Simulation for Urban and non-urban Networks, to evaluate newly generated solutions. The authors utilized an application programming interface (API) to customize changes in traffic simulation. In Ben et al. (2010), urban traffic microscopic simulation model (UTMSM) was selected to combine with a traffic signal optimization method to reduce the vehicle emissions and improve the intersection traffic capacity. Kouvelas et al. (2011) also used the microscopic simulator AIMSUN to obtain evaluation criteria. The emulated traffic measurements, obtained by AIMSUN are then used by a real-time traffic optimization strategy to produce the traffic signal timing parameters. Yan et al. (2013) utilized VISSIM, which is a microscopic traffic simulator, to simulate the traffic scenario. At each time step, the optimization approach decides a new optimized signal timing plan in the next interval based on the current state. This desired signal timing plan is then evaluated by the VISSIM network. Tettamanti et al. (2014) also used VISSIM to evaluate the traffic measurements, which are then forwarded to the optimization algorithm. After that, new control signal parameters are returned to the simulated traffic scenario. SUMO is used in Kai et al. (2014) to compute fitness value of solutions. When a new solution is generated, it is transferred to the traffic scenario, which is built by SUMO, to update the traffic signal timing parameters. After the simulation is done, desired traffic parameters, such as average delay time, is calculated using the output files of SUMO. Abushehab et al. (2014) also used SUMO to simulate the traffic network.

Comparative studies of traffic simulators can be found in Mustapha et al. (2016) and Pell et al. (2017). SUMO is able to model intermodal traffic systems such as vehicles, pedestrians, and public transport. SUMO has a flexible architecture, Passos et al. (2011). While most of other microscopic traffic simulators only allows homogeneous vehicle, SUMO supports customized vehicle types such as cars and two wheelers, Patel et al. (2016). Furthermore, SUMO is open-source and free and it provides very helpful support for new users, therefore, it is very suitable for researchers and students to evaluate their own hypotheses.

3.3 Reducing Computational Cost using Surrogate Models

3.3.1 Computational Cost of Traffic Signal Optimization using MOEAs and Traffic Simulators

The macroscopic traffic simulators model the flow propagation in a mathematical way and they are not suitable enough for real-time adaptive control approaches as they do not consider individual vehicle arrivals which are needed for adaptive signal control systems, Guangwei et al. (2007), Yin et al. (2015). Whereas the microscopic traffic simulators are flexible to simulate complicated behaviour of traffic. MOEAs and microscopic traffic simulators have been applied for traffic signal optimization problems in a many studies. Although evaluating solutions using microscopic simulators has several advantages and they have been used widely, it also has to face a big challenge, which the computational burden required to evaluate a solution. Furthermore, the computation time will rapidly rise as the scale of the traffic network increases, such as in road network size and number of vehicles. When applying simulation-based MOEAs to optimize a transportation problem, the traffic simulator needs to be called every time an individual solution is evaluated. Therefore, the offspring evaluation requires much processing time in MOEAs and computation time of estimating fitness value of solutions becomes a burden, Shen et al. (2011). Consequently, this limits the number of evaluations and generations as well as the population size that can be used in MOEAs.

There are a significant number of studies optimizing traffic signaling optimization, however, only a few researches contemplated processing time in traffic signaling optimization. To reduce computational time of MOEAs in transportation optimization problems, Guangwei et al. (2007) utilizes a parallel genetic algorithm to reduce time responses of the optimization process. The population is divided into a number of sub-populations running on multiple processors, therefore, the parallel genetic algorithm is more efficient and faster and more suitable for complex problems. Sanchez-Medina et al. (2010) combines a genetic algorithm and a cellular-automata-based microscopic traffic simulator running on a scalable multiple-instruction-multiple-data(MIMD) multicomputer. This approach employed Beowulf Cluster to achieve parallelism to reduce total computational time. The optimization process is divided into a number of tasks which will be performed in different processes. Graphics Processing Units (GPU) were utilized in Shen

et al. (2011) and Shen et al. (2013) to reduce computational burden. In these studies, an evolutionary algorithm was used to search for optima. Generating new generation and checking termination conditions are solved via the CPU. Computational expensive tasks, including solution evaluation, selection, crossover and mutation operators, are implemented by GPUs. Traffic data are also loaded into a GPU memory, as a result, the GPU version executes much faster compared with the CPU version. However, this approach is not suitable in traffic scenarios where there are limited processing capabilities. Such scenarios are typical for local and distributed signal controllers, which offer very limited processing power while requiring optimized signal timings within a few cycles or minutes.

Surrogates are computational models used to predict objective values of solutions at a cheaper cost compared to original objective function. Surrogates are used to reduce the total number of evaluations using original objective function while remaining a reasonable good quality of results obtained. Using surrogates to estimate objective value of solutions greatly reduces the computational cost as the price to build and manage the surrogates is much more cheaper than that of evaluating objective value using the original objective function. Consequently, surrogate-assisted MOEAs are very promising to reduce the computation cost of traffic signal optimization problems caused by estimating objective value using a traffic simulator. Approximation techniques which are used to construct surrogates are discussed in the following section.

3.3.2 Techniques for constructing surrogates

A surrogate model is an approximation of a complex model and it is used as a substitute in many applications, usually with a purpose to reduce computational complexity. There are several techniques utilized to build a surrogate model. Table 3.3 shows techniques which have been used to construct surrogates and their references. From the conducted review of these techniques, Artificial Neural Networks (ANNs) emerges as one of the most popular technique used in constructing surrogate.

A. Fitness inheritance

Fitness values of some solutions are calculated based on the fitness values of individuals created previously during the evolution process. The fitness value of a new solution can

TABLE 3.3: Techniques for constructing surrogate in the literature.

No.	Techniques	References
1	Artificial Neural Networks	Ong, Nair and Lum (2006), Zhou et al. (2007), Sun et al. (2013), Jin et al. (2015), Bhattacharjee et al. (2016)
2	Regression Model	Branke and Schmidt (2005)
3	Gaussian Process	Zhou et al. (2007), Liu et al. (2014)
4	Fitness inheritance	Reyes-Sierra and Coello (2005), Fonseca et al. (2012)

be obtained from the fitnesses of its parents, and this is called "fitness inheritance", which has firstly been introduced in 1995 by Robert E. Smith (0995). In Fonseca et al. (2012), utilized a surrogate model which is a fitness inheritance scheme to assist a Genetic Algorithm (GA). The impact on the evolutionary search of three kinds of inheritance, including weighted inheritance, averaged inheritance, and parental inheritance, was compared. A fixed probability p_{sim} of solutions in each generation was chosen using the random selection scheme to be calculated by a simulation model. This approach analysed the search behaviour of the surrogate-assisted GA when the parameter p_{sim} is changed. The proposed algorithm also is compared to the traditional GA (with standard fitness evaluation procedure) using only simulations. The results indicate the capabilities of using cheaper inheritance surrogate to increase the number of generations which leads to further exploration of the search space. Consequently, the quality of final solutions has been improved. However, for highly non-linear problems, the fitness inheritance is not a good choice to build a surrogate as its results were not good, Ducheyne et al. (2003). Moreover, the results obtained by traditional GA are better than those achieved by weighted inheritance.

B. Regression Model

Branke and Schmidt (2005) built local models instead of a global approximation model. Regression is utilized to build local approximation schemes, which are based on previously evaluated neighbouring individuals. Every individual evaluated with original objective function can be taken into account by the approximation model. However, the larger dataset, the longer it takes to construct the approximation model. Therefore, all evaluated solutions are preserved and they are available for use but only the closest

neighbours are then selected to build estimation models. As a result, the construction of the approximation is still fast as only information of related individuals is actually used to construct the model. An uncertainty measure of an individual is defined based on the Euclidean distance from that solution to all data points, which are utilized to build the local surrogate.

C. Gaussian Process

A multi-layer surrogate-assisted evolutionary optimization approach including global and local surrogates was proposed in Zhou et al. (2007). To construct a global surrogate, the search processes standard EA for a number of generations to collect data points. Global surrogate, constructed using Data-parallel Gaussian process (DPGP), pre-screens promising solutions in the population. The pre-defined top ranking $\eta\%$ solutions in the population are then re-evaluated using original fitness function. For each solution belonging to $\eta\%$, a surrogate is constructed based on local radial basis function (RBF) using k nearest neighbouring samples from the preserved database. Therefore, in the surrounding area of an individual, each surrogate model represents a local fitness landscape. Any superior solution found during the local search using Lamarckian learning process is re-evaluated by the original objective function. Any new solution evaluated with the real fitness function during the local search procedure is added into the database to update the global surrogate.

A surrogate-assisted optimization algorithm for medium scale computational expensive optimization problems is introduced in Liu et al. (2014). Sammon mapping technique is utilized in this approach to reduce the size of the dimension of decision variables. All the solutions already evaluated and their fitness values are recorded in a database. Surrogate model is constructed using the Gaussian process to pre-screen newly generated children. The most promising solutions will be re-evaluated using the original objective function.

D. Artificial Neural Network

Ong, Nair and Lum (2006) proposed a robust optimization method to deal with uncertainties type II and type III for problems which are sensitive with uncertainties. Uncertainty type II appears in design variables while uncertainty type III is the result of fluctuations in operating conditions. A random noise vector is inserted to the genotype

before evaluating fitness value. The Multiple Evaluation Model (MEM) is used to determine the effective individuals' fitness in the population. The search proceeds with the standard robust GA and worst MEM for first z generations and all evaluated solutions are recorded in a database. For each solution in the population, k nearest design points are chosen from the database to construct a local surrogate model using Radial Basic Function and this individual undergoes a local search strategy to find out the worst case performance. The fitness of the individual is set to be the worst-case value.

Another strategy is introduced in Sun et al. (2013) to handle uncertainties in human fitness assignment. The main purpose is to construct a surrogate model based on Co-training Semi-supervised Learning (CSSL). The main process of the proposed algorithm is as follows. A population with a large number of solutions are randomly initialized. It is then clustered into z sub-populations. The individual which is closest to the centre of the cluster is then calculated by the user and inserted into $L(t)$, which is a database including labelled samples. All the other unevaluated solutions of each cluster are stored in unlabelled data set $U(t)$. CSSL is then adopted to construct the surrogate. An evaluation reliability formulation is defined and integrated into the error function. Two radial basis function networks (RBFN) are adopted as two co-training learners. Outputs of the two approximation models are aggregated based on the estimation confidence. A pre-defined number of solutions, including the centre of clusters, are re-evaluated by the user in each iteration. The newly evaluated individuals are to update the surrogate when evaluation error is larger than a pre-defined threshold. Potentially good individuals, for example, solutions having a higher estimated fitness value and a better estimation confidence or individuals which could increase the diversity of the population, should be selected for reproduction.

Jin et al. (2015) builds a local surrogate model for every solution. After a pre-defined number of generations, a database is formed, which includes M individuals evaluated using original fitness function. To construct local models, each data point s_i in the database is assigned to k training sets TD_i, which consist of k nearest solutions in the population of s_i. Each solution x_i in the population has a dataset TD_i which is used to build the local surrogate. When a new offspring is generated, k local surrogates of its k nearest solutions in the population are used to create an ensemble surrogate and offspring fitness value is estimated based on this ensemble surrogate.

In Bhattacharjee et al. (2016), all exact individual evaluations are recorded into a database to serve for building local surrogates. For each offspring, k nearest neighbours in the database are used to construct approximation models. Multiple local surrogates are employed using different types of surrogate techniques including Radial Basic Function, Gaussian Process, and Polynomial Response Surface to predict the values of each objective and constraints and the best local surrogate among them is chosen. Mean square error (MES) is utilized to validate the fidelity of surrogates.

Lim et al. (2010) proposes a generalized framework for combining different surrogates in the evolutionary search to mitigate the negative effects introduced by the approximation error of the surrogates and to get benefits obtained by the use of surrogates. Each solution in the population simultaneously undergoes local searches using smoothing and aggregated surrogates. Surrogate M_1, which is an ensemble of n surrogate models used, helps to reduce the negative impacts of inaccurate estimation of the surrogate. The smoothing surrogate model M_2, which transforms the function into one with fewer minima, therefore, increasing the convergence rate of the evolutionary search. The higher quality solution among locally improved individuals obtained from M_1 and M_2 is used to replace the initial individual. However, constructing multiple surrogates leads to increasing computational cost as well as the complexity of the model.

Recent studies clearly illustrated that the selection of surrogate techniques affects the performance of evolutionary search approaches, Lim et al. (2010). However, there are many approximation techniques available in the literature. Therefore, it is almost impossible to know which approximation technique is suitable for a problem if the knowledge about the problem landscape is limited. One technique, working successfully in an instance, might not work well on others, Goel et al. (2007), Acar and Rais-Rohani (2009). A number of researches have been carried out to assess and compare the performance of approximation techniques, Carpenter and Barthelemy (1992), Jin et al. (2001), T. Simpson and Mistree (1998). The results indicate that there is no clear conclusion about which model is definitely better to the others. More than one criterion should be considered when selecting an approximation model. Jin (2005) suggested that if the number of available samples is limited and the input space is highly-dimensional, a neural network is then recommended.

3.3.3 Surrogate Assisted Optimization in Transportation

To the best of our knowledge, traffic signal optimization using surrogate models is not common in the literature of current signal control approaches and there are few studies consider using surrogate in optimizing the traffic signal timing. Osorio and Bierlaire (2009) introduces an approximate queuing network model to optimize congested urban road network networks. This study focuses on fixed-time signal control for coordinated intersections. The interaction between flows on upstream and downstream roads is taken into account. Roads are mapped into sets of queues and finite capacity queueing theory is utilized to capture the interaction between consecutive roads. A set of nonlinear equations is formulated to describe the correlation between decision variables, exogenous parameters, for example, the route choice decisions and total demand or the topology of traffic network, and endogenous variables such as the probability and the capacities of spillbacks. This approach assumes that exogenous variables are fixed, which are not practical since urban traffic is highly dynamic and uncertain.

A surrogate model was constructed in Gil et al. (2018) using a fuzzy model to define the optimal cycle length and the green duration ratios. The optimization objective, which is traffic delay, was obtained from the Intelligent Driver Model based microscopic traffic simulator, which was developed at the authors' department. A fuzzy model was constructed to the model the relationship between objective values (total delay time in this case) and related traffic parameters, for example, the green time ratio, the cycle length and traffic flows. A number of simulation runs were implemented to collect data describing the total delay time and its influential parameters. The fuzzy model was created based on this data. The position of the fuzzy sets of the new rule was optimized using PSO. To optimize the objective function, any optimization technique can help and PSO would be a straightforward option. The accuracy of the approximation model mainly depends on the data collected from the traffic simulator. If the data does not cover all different types of traffic flows, and traffic light cycle's type, the fuzzy surrogate model would not be able to adequately replicate the relationship between the total delay time and its related traffic parameters, as a result, the approximation error would be high. Furthermore, it is well-known in the literature of surrogate-assisted optimization that the surrogate model should be used together with the traffic simulator to avoid false

optima. Therefore, a management model should be introduced to decrease the negative impact of the approximation error.

In conclusion, surrogate-assisted MOEAs are promising to reduce computational cost in traffic signal optimization problems. However, the management model is needed to use surrogates properly and effectively.

3.4 Conclusion

Traffic signal control systems have a big impact on travel costs and the environment. Traffic signal optimization is an importation approach to increase the effectiveness of the control system. The optimization of parameters of signal timing is a computationally complex problem and the measurement of objective values, such as delay time, flow, and travel time is one of the fundamental issues in this research area. Different methods have been used to solve this task and traffic simulators is one of the most popular tools. Although estimating objective value by using traffic simulators has several advantages, such as higher accuracy and flexibility to capture the dynamic of transport, the simulation run is very time-consuming. Computational time rises rapidly as the scale of the traffic network increases.

Multi-objective evolutionary algorithms are superior to traditional searching approaches and they have been widely used to produce optimal signal timings. However, population-based MOEAs have slow convergence speed and the computational time would be a burden when population-based MOEAs are combined with traffic simulators to address the traffic signal optimization problems. Therefore, a simulation-based multi-objective evolutionary algorithm, which can produce good results with limited population size, a small number of generations or a few numbers of objective evaluations would be desirable in traffic signal optimization problems.

Combining a local search and a global may accelerate the search to optima. Furthermore, local search also might help reduce the population size of evolutionary algorithms. Therefore, a hybrid of an evolutionary algorithm and a local search can improve the performance of traffic signal optimizations system.

Surrogates are approximate models which can be used to estimate objective values of solutions at a cheaper cost compared to original objective function. Surrogates has received increasingly interest in recent years. Surrogate-assisted MOEAs have been used for reducing computational cost of optimizing expensive problems. Consequently, surrogate-assisted MOEAs are promising for solving multi-objective traffic signal timing. Surrogate can be constructed using several approximation techniques. Surrogate should be combined with the original objective function, traffic simulators in case of traffic signal optimization problems, to reduce the computational cost while avoiding mislead the search to false optima.

Chapter 4

Methodology

4.1 Introduction

Multi-objective Evolutionary Algorithms (MOEAs) have been widely utilized in the traffic signal optimization problems in order to provide effective control methods for urban traffic networks which are highly complex, uncertain and dynamic. However, running MOEAs on traffic optimization problems is time-consuming Shen et al. (2013). For example, with a small traffic simulation introduced in Bieker et al. (2015), it takes 25 seconds to run one simulation using a PC with Inter(R) Core(TM) i5-6500 CPU 3.2GHz. If the evolutionary process includes 20 generations and there are 60 solutions in the population, the number of solution evaluations needed is 1200 and therefore the traffic simulator has to run 1200 simulations. Consequently, the time to run simulations in the evolutionary process is about 8.3 hours. Furthermore, the computation time will rapidly rise as the scale of the traffic network increases, such as in road network size and number of vehicles. Consequently, in real-time traffic signal management where optimized solutions need to be provided in real time, optimization approaches which have the ability to provide good solutions at a reasonable processing time, especially at an early stage, is preferable. Anytime behavior of an algorithm is its ability to provide as good a solution as possible at any time during its execution and continuously improves the quality of the results as computation time increases. Therefore, optimization approaches for urban traffic signal control systems, which have good anytime behavior, are desirable. Moreover, in transportation optimisation, small population sizes are

inevitable for scenarios where processing capabilities are limited but require quick response times. Nevertheless, most existing algorithms are mainly focused on the quality of solutions at the end of the optimization process and population size has not been considered as an indicator when evaluating the effectiveness of an optimization algorithm. Therefore, NS-LS has been introduced which is a multi-objective optimization strategy to improve anytime behavior and which can work effectively with various population sizes. Furthermore, local search is integrated into the evolutionary search to accelerate the convergence rate.

The multi-objective optimization for traffic signal control system using traffic simulators is computationally expensive. Therefore, a surrogate model is constructed to estimate the fitness value of candidate solutions and this model is used together with SUMO to eliminate the false optimum. SA-LS, which is an enhancement of NS-LS, is introduced for traffic signal optimization problems. The number of traffic simulator-based evaluations is reduced, as a result, the number of generations will be increased. Therefore, anytime behaviour of SA-LS would be improved.

This chapter is organized as follows: Section 4.2 describes the motivation and the flow of the local search method while the framework and design of NS-LS are provided in Section 4.3. The construction process and updating rules of the surrogate model are shown in Section 4.4. Fitness evaluation scheme is introduced in Section 4.5. Details of SA-LS algorithm are provided in Section 4.6 and Section 4.7 concludes the chapter.

4.2 The local search strategy

A local search strategy is introduced to improve anytime behaviour of the evolutionary algorithm for traffic signal optimization problems. Firstly, this section explains how neighbours are created from a candidate solution and reference solutions. Afterwards, the local search method is provided in the second part of this section.

4.2.1 Creating neighbours of a solution

There are many neighbours surrounding a solution and choosing appropriate neighbours to evaluate is critical in local search strategies. Given a solution $R_i^{(t)}$ in $j^{(th)}$ subpopulation, $R_u^{(t)}$ and $R_v^{(t)}$ are two solutions selected from front F_1 and F_2 sorted from the population using a non-dominated sorting algorithm. The neighbour $nb_{R_i^{(t)}}$ of solution $R_i^{(t)}$ is defined using the following formula:

$$nb_{R_i^{(t)}} = R_i^{(t)} + \alpha * (R_u^{(t)} - R_v^{(t)}) \tag{4.1}$$

where α is a perturbation index which complies with Gaussian distribution $N(\mu, \sigma^2)$, where σ and μ are the standard deviation and mean values of the Gaussian probability distribution, respectively. If μ is set too large, it would magnify $(R_u^{(t)} - R_v^{(t)})$ value which would, in turn, affect the convergence of the algorithm. Similarly, if σ is too big, the value of α will be too big or too small, thus affecting the convergence speed of the search (Chen et al. (2015)). On the other hand, if σ is set too small, the effectiveness of perturbation would not be observable.

4.2.2 Motivation of the local search method

Suppose that we have two reference solutions $R_u^{(t)}$ and $R_v^{(t)}$, a neighbour $nb_{R_i^{(t)}}$ of solution $R_i^{(t)}$ is created using formula 4.1. An example of a neighbour creation is illustrated in Figure 4.1 with $\alpha = 0.5$. There are two worthwhile conclusions drawn from this neighbour creation as the following:

1. The direction from $R_i^{(t)}$ to its neighbour $nb_{R_i^{(t)}}$ is the same with the direction from $R_v^{(t)}$ to $R_u^{(t)}$.
2. The distance between $R_i^{(t)}$ and $nb_{R_i^{(t)}}$ is proportional to the length of $(\vec{R_u^{(t)}} - \vec{R_v^{(t)}})$ and it also depends on the parameter α.

From the above conclusions, another observation is made. If the following conditions:

1. $R_u^{(t)}$ dominates $R_v^{(t)}$;

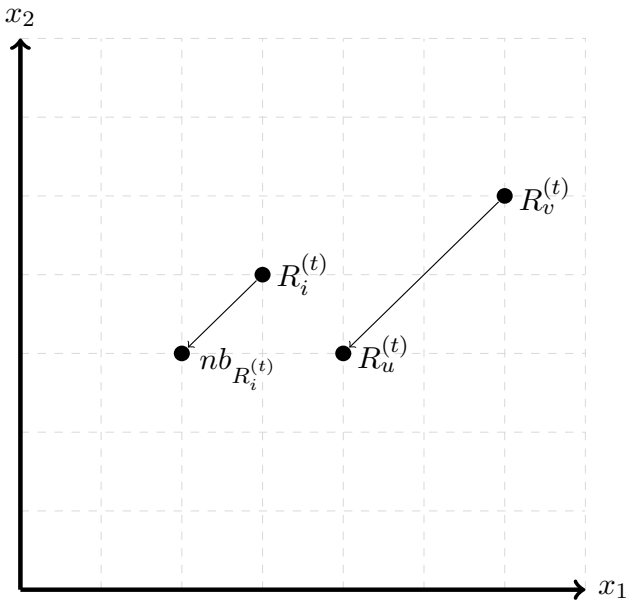


FIGURE 4.1: The neighbour creation: a neighbour $nb_{R_i^{(t)}}$ is created from solution $R_i^{(t)}$ based on two other reference solutions $R_u^{(t)}$ and $R_u^{(t)}$ using equation 4.1 with $\alpha = 0.5$.

2. $R_u^{(t)}$, $R_v^{(t)}$, and $R_i^{(t)}$ are close enough or they are in the same local area.

are satisfied, the direction from $R_v^{(t)}$ to $R_u^{(t)}$ would be the direction from one point to a better point, as a result, neighbour $nb_{R_i^{(t)}}$ would dominate solution $R_i^{(t)}$ and the neighbour $nb_{R_i^{(t)}}$ would be closer to a local optimum.

From the above reason, we introduce a new local search method, which can predict a potential search direction to quickly find a superior neighbour. The population is classified into several subpopulations and neighbour is created using candidate and reference solutions in the same subpopulation. Furthermore, the reference solutions are selected from the first and second non-dominated fronts. Therefore, the two conditions mentioned above are satisfied, and as a result, the neighbour is likely to be better than the original solution. Details of the proposed local search algorithm is introduced in the next section.

4.2.3 The flow of the proposed local search

The pseudo code of this new local search algorithm is illustrated in Algorithm 2 and its procedure is outlined in the following:

Algorithm 2 The pseudo code of the local search algorithm

Input: *population* $R^{(t)}$ *including* $2N$ *solutions.*
Output:
    1. Population $P^{(t)}$ *including* N *best solutions.*
    2. L_{temp} *containing solutions evaluated by SUMO during the local search process.*
Procedure:
$L_{temp} = \emptyset$
Sorting $R^{(t)}$ *into a number of different non-dominated fronts* F_i, $i = 1, 2, \ldots$
Cluster $R^{(t)}$ *into* J *sub-populations using a k-means algorithm.*
For $j^{(th)}$ *sub-population* ($j \in [1, J]$):
    $SUP_1 = \emptyset$, $SUP_2 = \emptyset$
    L^j_{sub} *is the length of* $j^{(th)}$ *sub-population.*
    For $k^{(th)}$ *solution in* $j^{(th)}$ *sub-population* ($j \in [1, L^j_{sub}]$):
        If $k^{(th)}$ *solution belongs to* F_1 **then**
            $SUP_1 \leftarrow k^{(th)}$ *solution.*
        If $k^{(th)}$ *solution belongs to* F_2 **then**
            $SUP_2 \leftarrow k^{(th)}$ *solution.*
    If $|SUP_1| \geq 2$ *and* $|SUP_1| \geq 1$ **then**
        Select one solution $R^{(t)}_i$ belonging to SUP_1.
        Select one solution $R^{(t)}_u$ belonging to SUP_1 and one solution $R^{(t)}_v$ in SUP_2.
        Create a neighbour $nb_{R^{(t)}_i}$ *of* $R^{(t)}_i$ *using formula 4.1.*
        Assign a fitness value for the neighbour $nb_{R^{(t)}_i}$.
        $L_{temp} \leftarrow$ *solutions evaluated by SUMO during the local search.*
        $R^{(t)} \leftarrow nb_{R^{(t)}_i}$.
End For
Sorting $R^{(t)}$ *into different non-dominated fronts* $F_i, i = 1, 2, \ldots$
$P^{(t+1)} \leftarrow \emptyset$, $i \leftarrow 0$
Until $|P^{(t+1)}| + |F_i| \leq N$*:*
    $P^{(t+1)} \leftarrow P^{(t+1)} \cup F_i$
    $i \leftarrow i + 1$
End Until
Crowding-sort($F_i, <_c$)
Descending-order-sort(F_i)
$P^{(t+1)} \leftarrow P^{(t+1)} \cup F_i[1 : (N - |P^{(t+1)}|)]$
Return: $P^{(t+1)}$ and L_{temp}

Step 1: Classify a population $R^{(t)}$ to a number of non-dominated fronts F_i, $i = 1, 2, ...$ using a non-dominated sorting algorithm.

Step 2: Cluster $R^{(t)}$ into J sub-populations on the decision variable space using a k-means algorithm. This helps determine neighbouring solutions of a particular solution.

Step 3: For each sub-population:

- *Step 3.1*: Select all solutions which belong to both the current sub-population and front F_1. These selected solutions are then reserved in SUP_1.
- *Step 3.2*: Create a set SUP_2, which includes solutions belong to both the current sub-population and front F_2.
- *Step 3.3*: If there are more than 2 solutions in SUP_1, randomly select one solution $R_i^{(t)}$ and another individual $R_u^{(t)}$ in SUP_1.
- *Step 3.4*: Randomly select one solution $R_v^{(t)}$ in SUP_2.
- *Step 3.5*: Create a neighbour $nb_{R_i^{(t)}}$ of $R_i^{(t)}$ using formula 4.1. $R_u^{(t)}$ dominates $R_v^{(t)}$ as $R_u^{(t)}$ and $R_v^{(t)}$ belong to F_1 and F_2 respectively. Furthermore, $R_i^{(t)}$, $R_u^{(t)}$, and $R_v^{(t)}$ are close together as they are all in the current sub-population. Thus $nb_{R_i^{(t)}}$ is likely to be superior to $R_i^{(t)}$.
- *Step 3.6*: Assign a fitness value for the new created neighbour $nb_{R_i^{(t)}}$ using SUMO.
- *Step 3.7*: Add $nb_{R_i^{(t)}}$ to $R^{(t)}$.

Step 4: Non-dominated sort $R^{(t)}$ into different fronts $F_i, i = 1, 2, \ldots$.

Step 5: Replace the population $P^{(t+1)}$ by N best solutions selected from $R^{(t)}$ using their rank and crowding distance. Return $P^{(t+1)}$ and L_{temp} as the output of the local search procedure.

As we can see from the above procedure that the search direction in the proposed local search method is not randomly chosen. A potential direction is selected before starting the searching process, by only selecting $R_u^{(t)}$ in front F_1 and $R_v^{(t)}$ in front F_2. As a result, the chance to immediately find out a superior neighbour from the first search would be increased. Moreover, the neighbour $nb_{R_i^{(t)}}$ is not only compared to $R_i^{(t)}$ but also all solutions in $R^{(t)}$ and other neighbours newly created in the local search process. Consequently, the elitism of the algorithm is conserved and no good solutions are missed once they are found.

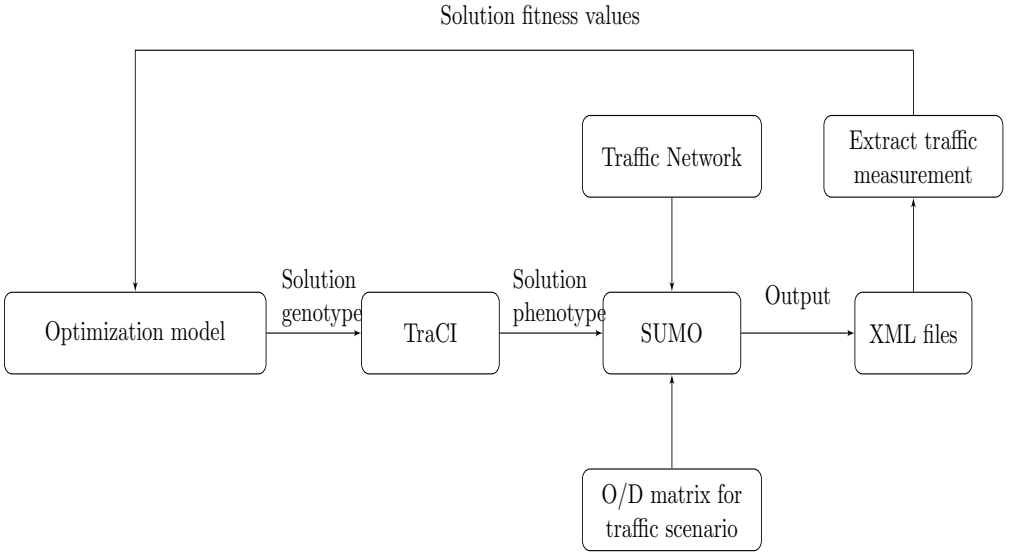


FIGURE 4.2: The overall optimisation framework of NS-LS.

4.3 NS-LS algorithm

In this section, NS-LS, which is an evolutionary algorithm for traffic signal optimization, is proposed to improve anytime behaviour. NS-LS uses the local search method introduced in the previous section to accelerate the convergence rate of the search, therefore, the anytime behaviour of the algorithm will be increased. The architecture, framework and the flow of NS-LS are described in the following.

4.3.1 Overview of NS-LS

NS-LS is a simulation-based optimization algorithm proposed for traffic signal control. It includes an optimizer and a traffic simulator which is used to evaluate the performance of solutions generated by the optimizer. The overall framework of NS-LS is illustrated in Figure 4.2. This study uses SUMO to simulate traffic scenarios and return the results of traffic parameters needed to calculate the objective values. The optimizer and SUMO are connected through TraCI, which stands for Traffic Control Interface. TraCI uses a TCP based client/server architecture to provide access to SUMO. Whenever the optimization model needs to estimate the fitness value of a solution using SUMO, it has to send a request to SUMO, which acts as a server, to establish a connection. The traffic network and origin/destination (O/D) matrix of the traffic scenario are loaded into SUMO. The solution needed to be evaluated is sent from the optimizer to SUMO.

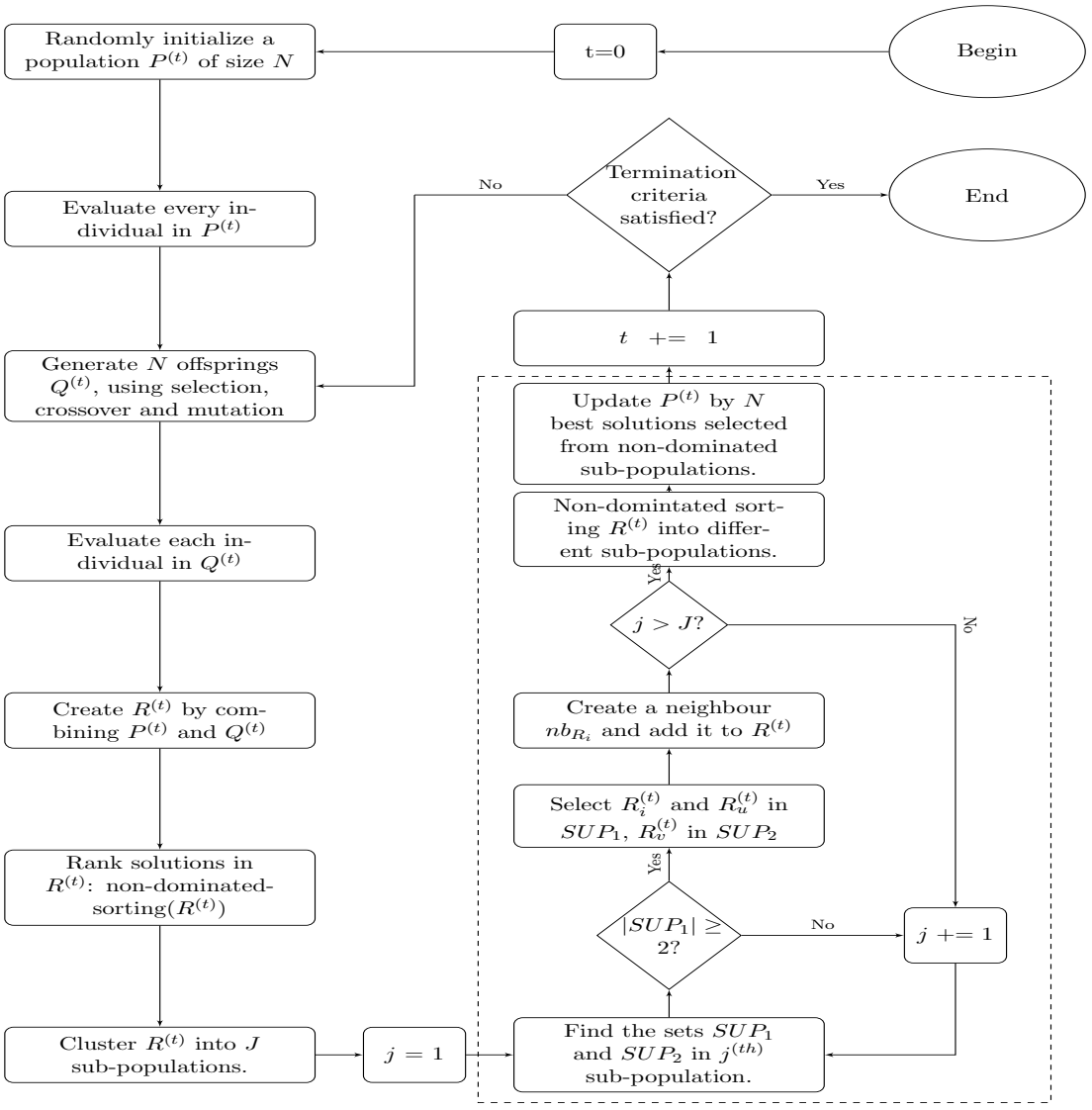


FIGURE 4.3: The framework of the optimization process in NS-LS. The left part of the figure is iterations of the optimization process while the local search strategy is on the right in the dashed box.

Non-dominated sorting evolutionary algorithm with Local Search (NS-LS) is an algorithm for traffic signal optimization problem based on Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and local search. NSGA-II is a well-known multi-objective optimization algorithm and is an instance of an Evolutionary Algorithm which was proposed by K. Deb and his students in 2002. NSGA-II is chosen because of the following reasons:

1. NSGA-II provides a superior performance compared to other multi-objective evolutionary strategies, Sun et al. (2003).

2. Deb, Pratap, Agarwal and Meyarivan (2002) pointed out that NSGA-II, in most problems, can find a set of solutions, which has a wider spread distribution and a better convergence rate compared to Pareto-archived evolution strategy and strength-Pareto evolutionary algorithm.

3. In NSGA-II, the original sorting algorithm used in NSGA is replaced by a fast non-dominated sorting approach, therefore, reducing the computational complexity of NSGAII.

The flowchart of the proposed NS-LS algorithm is shown in Figure 4.3. NS-LS contains two parts: the main iterative process and the local search. Introduction to these two components are provided in the following.

The main iterative process, which is on the left of the flowchart, utilises NSGA-II's evolutionary process to move a population of candidates towards the optimal front which is defined as the non-dominated set of the entire feasible search space. An initial population is randomly generated in the beginning of the optimization process, and this population undergoes a number of generations. In each generation, new offspring are created using selection and reproduction operators. $(\mu+\lambda)$ evolution strategy, which the fittest individuals are selected from λ parents and μ children, is utilized in NSGA-II to prevent the loss of good solutions found during the optimization process. Consequently, all the best solutions from parents and children would have chances to be chosen to be cloned to the next iteration. This iterative process is stopped when termination conditions are satisfied and the result of NS-LS are high performing individuals selected in the last generation.

The local search dashed box of the flowchart in Figure 4.3 has been integrated into NS-LS to speed up the convergence rate of the search. The population is classified into a number of small groups and the local search is employed to find superior neighbours in these local areas. Consequently, the population would move more quickly toward the optimal front. As a result, anytime behaviour of NS-LS is likely to be improved.

4.3.2 The flow of NS-LS

The pseudo code of the proposed algorithm NS-LS is given in Algorithm 3. From the input of the algorithm, which is the population size N, an initial population $P^{(0)}$

Algorithm 3 The pseudo code of NS-LS

1: **Input:** *Population size N*
2: **Procedure:**
3: $t \leftarrow 0$
4: $Q^{(t)} \leftarrow \emptyset$
5: $P^{(t)} \leftarrow$ *Random-population().*
6: *Evaluate*($P_i^{(t)}$), $i \in N$
7: **while** *not Termination-condition()* **do**
8: $a_1 = [1, 2, \ldots, N]$ // a_1 *is an array including N items from 1 to N.*
9: $a_2 = [1, 2, \ldots, N]$ // a_2 *is an array including N items from 1 to N.*
10: **for** $(k = 0; k < N; k++)$ **do**
11: $rand = rnd(k, N-1)$ //*a random number in range k and* $N-1$.
12: *swap*($a_1[k], a_1[rand]$)
13: $rand = rnd(k, N-1)$
14: *swap*($a_2[k], a_2[rand]$)
15: **end for**
16: **for** $(k = 0; k < N; k += 4)$ **do**
17: *parent1=tournament*($P^{(t)}_{a_1[k]}$,$P^{(t)}_{a_1[k+1]}$)
18: *parent2=tournament*($P^{(t)}_{a_1[k+2]}$,$P^{(t)}_{a_1[k+3]}$)
19: *crossover(parent1, parent2,* $Q_k^{(t)}$, $Q_{k+1}^{(t)}$)
20: *parent1=tournament*($P^{(t)}_{a_2[k]}$,$P^{(t)}_{a_2[k+1]}$)
21: *parent2=tournament*($P^{(t)}_{a_2[k+2]}$,$P^{(t)}_{a_2[k+3]}$)
22: *crossover(parent1, parent2,* $Q_{k+2}^{(t)}$, $Q_{k+3}^{(t)}$)
23: **end for**
24: **for** $(k = 1; k \leq N; k++)$ **do**
25: *mutation*($Q_k^{(t)}$)
26: **end for**
27: *Evaluate*($Q_i^{(t)}$), $i \in N$
28: $R^{(t)} \leftarrow P^{(t)} \cup Q^{(t)}$
29: *Sorting* $R^{(t)}$ *into different non-dominated fronts* F_i, $i = 1, 2, \ldots$
30: *local-Search*($R^{(t)}$)
31: *Re-sorting* $R^{(t)}$ *into different non-dominated fronts* F_i, $i = 1, 2, \ldots$
32: $P^{(t+1)} \leftarrow \emptyset, i = 0$
33: **Until** $|P^{(t+1)}| + |F_i| < N$: — — *is defined as the length of an array.*
34: $P^{(t+1)} \leftarrow P^{(t+1)} \cup F_i$
35: $i \leftarrow i + 1$
36: **End Until**
37: *Crowding-sort*($F_i, <_c$)
38: *Descending-Order-Sort*(F_i)
39: *Select first* $(N - |P^{(t+1)}|)$ *solutions in* F_i *and insert them into* $P^{(t+1)}$
40: $t \leftarrow t + 1$
41: **end while**
42: **Return** P.

including N individuals is randomly generated (line 5 in Algorithm 3). Each solution $P_i^{(0)}$ is represented as a vector including n variables. Each variable contains a green

duration of a phase in a traffic light cycle. Details of the solution representation are explained in the next section. Each solution is assigned a fitness value using a traffic scenario simulated by SUMO. SUMO simulates the traffic network using traffic light durations, contained in a solution, to estimate the objective values, such as traffic flow or delay time.

This initial population $P^{(0)}$ undergoes a transformation through a number of generations until the terminating conditions are satisfied. In $t^{(th)}$ iteration, the population $P^{(t)}$ is processed as follows:

Step 1 (lines 8 - 22): The binary tournament selection scheme and the crowded tournament selection operator are utilised to choose individuals for reproduction. Recombination and mutation operators are then used to produce children population $Q^{(t)}$ including N solutions from solutions.

Step 2 (line 27): Evaluate and assign the fitness value for offspring in $Q^{(t)}$ using SUMO.

Step 3 (lines 28 - 29): The elitism scheme is applied to keep the best solutions for the next generation. Both $P^{(t)}$ and $Q^{(t)}$ are taken into consideration to guarantee that no good solution is missed once it is found. A population $R^{(t)}$ including $2N$ solutions is created by combining parent and offspring populations. $R^{(t)}$ is then classified into a different non-dominated fronts $F_i, i = 1, 2, ...$ using a non-dominated sorting algorithm, for example Efficient Nondominated Sort (ENS), Zhang et al. (2015) and Efficient Non-domination Level Update Method (ENLU), Li et al. (2017). The algorithm utilized in this study is ENS. Minimization of objective functions is assumed. Each individual in $R^{(t)}$ is allocated a rank equal to the level of the non-dominated front where it belongs to (1 is the best level), therefore, every solution in a front has the same rank. Solutions in a front which has a smaller rank dominate individuals in fronts which have larger ranks.

Step 4 (line 30): To improve anytime behaviour of NS-LS, a local search method is applied to $R^{(t)}$. Firstly, $R^{(t)}$ is classified into J sub-populations using a k-mean algorithm. For each subpopulation, one neighbour is then created using three solutions belonging to fronts F_1 and F_2. Consequently, J neighbours have been created in this step. These new individuals found by the local search are then included into $R^{(t)}$. Thus the size of $R^{(t)}$ has been increased.

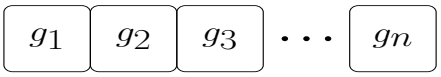


FIGURE 4.4: Chromosome representation where g_i is a variable representing the green duration of $i^{(th)}$ phase.

Step 5 (line 31): $R^{(t)}$ is then sorted using a non-dominated sorting algorithm. New population $P^{(t+1)}$ is filled up by N best solutions chosen from $R^{(t)}$ based on their ranks. Firstly, $P^{(t+1)}$ takes solutions from front F_1. If the size of F_1 is smaller than N, $P^{(t+1)}$ gets all individuals of F_1 and fills in its missing places by taking solutions in $F_i, i = 2, 3, ...$ This procedure is terminated if $|P^{(t+1)}| + |F_i| > N$.

Step 6 (lines 32 - 39): Perform the crowding distance assignment procedure described later in algorithm 4 and include the most widely spread $(N - |P^{(t+1)}|)$ solutions from sorted F_i to $P^{(t+1)}$. Consequently, diversity among non-dominated solutions is conserved.

Step 7: If the termination conditions are satisfied, the search process is terminated and returns $P^{(t+1)}$ as the output of NS-LS. If not, the search process repeats steps 1-6 using population $P^{(t+1)}$ in the next iteration.

It is worth to mention that the sorting procedure to divide $R^{(t)}$ into several non-dominated fronts in step 4 and the process to fill up $P^{(t+1)}$ can be implemented together. When a non-dominated front is found, the number of solutions in this front can be used to determine if it can be included in $P^{(t+1)}$. Consequently, the running time of the algorithm would be reduced.

4.3.3 Design of the evolutionary search

4.3.3.1 Chromosome Representation

Representation of a solution is illustrated in Figure 4.4. In this study, the proposed methodology aims to optimize the green duration of every stage in the signal plan of the signalized intersection to get the best objective values of the simulated traffic scenario, such as traffic flow and delay. A chromosome is represented as $X = (g_1, g_2, ...g_n)$ including n variables indicating green duration of n phases of the signal cycle. The optimization problem can be represented as follows:

$$\min / \max \quad f_m(X), \quad m \in [1, M] \tag{4.2}$$

where M is the number of objectives in the optimization problem. Each objective f_m can be maximized or minimized. Each element g_i of the vector X contains a positive integer, so real-number encoding is chosen in this study. The constraints and ranges of these variables are provided in Equations 4.3-4.6.

$$C = \sum_{i=1}^{n} g_i + \sum_{i=1}^{n} I_i \tag{4.3}$$

The cycle length C is calculated using Equation 4.3 as the sum of all the green durations g_i and inter-green times I_i. The range of C is defined in Equation 4.4, where C_{min} and C_{max} are minimum and maximum values of C, respectively.

$$C_{min} < C < C_{max} \tag{4.4}$$

The range of each green duration variable g_i is defined in 4.5:

$$g_i^{min} < g_i < g_i^{max}, i \in (1, n) \tag{4.5}$$

The minimum cycle length C_{min} is determined by the total of the minimum green durations and inter-green times of all phases in a cycle, as shown in Equation 4.6.

$$C_{min} = (\sum_{i=1}^{n} g_i^{min} + \sum_{i=1}^{n} I_i) \tag{4.6}$$

In this study, to set ranges for these variables, we follow the guidelines of the Highway Capacity Manual (HCM) 2010, Board (2010). The recommended values for minimum and maximum of green duration are 15 and 60 seconds, respectively. The inter-green time value is 3 seconds for each phase. The maximum cycle length C_{max} frequently set to 120 seconds and it can take a value of 140 seconds in some exceptional conditions, Teply et al. (2008). Board (2010)) recommends that the cycle length should take a minimum of 40 seconds or 60 seconds if pedestrians are included in the traffic control

Algorithm 4 The pseudo code of Crowding distance assignment procedure Crowding-sort($F_i, <_c$), code reproduced from Deb (2008)

1: **Input:** F_i*: front needed to assign crowding distance.*
2: **Procedure:**
3: *Step 1: set the number of solutions in* F_i *is* l, $l := |F_i|$. *For each solution in* F_i, *assign its initial crowding distance* $d_i := 0$.
4: *Step 2: for each subject function* $f_m, m \in [1, M]$, *sort the solutions in* F_i *in descending order of* f_m: $I^m = sort(f_m, <_c)$.
5: *Step 3: for* $m \in [1, M]$:
6: *1. Allocate a large crowding distance to the boundary solutions:* $d_{I_1^m} = d_{I_l^m} = \infty$.
7: *2. For other solutions* $j = 2$ *to* $(l-1)$: $d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}}$
8: **Return** *Crowding distance of solutions in front* F_i

system. However, pedestrians are not considered in this study, therefore, the minimum cycle length value is set to be 40 seconds while the maximum cycle length is 120 seconds.

4.3.3.2 Selection and Reproduction Operators

A. The crowded tournament selection operator $<_c$: Suppose that we need to compare two solution P_i^t and $P_j^{(t)}$, the crowded tournament selection operator $<_c$ is described as follows:

- **Step 1**: Check the dominance relation between P_i^t and $P_j^{(t)}$ and assign a crowding distance to each solution using the crowding distance assignment procedure in Algorithm 4.
- **Step 2**: Select the winner of the tournament using following rules:
  - If there exists one solution dominating the other: the one which dominates the other is the winner.
  - If these solutions are non-dominated: the one with better crowding distance becomes the winner of the tournament.

The first rule is used to guarantee that the chosen solution is the better solution which lies on a better non-dominated front. If two solutions are non-comparable as they are in the same front, the second rule is used to select the one residing in a less crowded area.

B. Selection strategy: binary tournament selection and crowded tournament selection operator are used to choose better solutions for reproduction as it is a useful and robust

selection method. Two solutions are randomly picked up and the better one is selected based on the tournaments. The chosen individual is then archived in the mating pool. Two other individuals are chosen and the better one is selected to fill in the mating pool. This process is iteratively carried out until the mating pool is full and each solution exactly participates in two tournaments.

Each solution $P_i^{(t)}$ has two attributes:

1. A rank r_i determined by front F_i where $P_i^{(t)}$ belongs.
2. A crowding distance d_i measures the search area around solution $P_i^{(t)}$ which does not accommodate any other solution.

The search space surrounding a solution which is not occupied by any other solutions is measured by the crowding distance. If this distance is large, it means that this solution is far from other solutions and vice versa. Evolutionary algorithms always try to find a set of solutions which are as diverse as possible to guarantee to have a good set of trade-off individuals among objectives. Consequently, solutions which have large crowding distance are preferable to reserve the diversity of the population.

B. Reproduction operators: Simulated Binary Crossover (SBX) is a commonly-used real-parameter recombination operator, introduced by Deb and his student in 1995, Deb and Agrawal (1995). Compared to other real-parameter crossover operators, SBX performs well in most continuous optimization problems. This crossover creates two offsprings from two parents. Children created by SBX tend to be closer to their parents and the spread of the children is proportional to that of the parent solutions, Deb (2008). As a result, SBX gives evolution strategy self-adaptive power. Consequently, SBX is adopted to create offsprings in this study. Another reproduction operator utilized in this study is Polynomial Mutation (PLM) which is one of the most widely used mutation operators. PLM can sample the entire search space of the decision variable even though the value to be mutated is close to one of the boundaries. PLM is able to sample the entire search space of the decision variable although the value of the variable to be mutated is near to the boundaries. Furthermore, big jumps in variable search space are permitted in PLM, as a result, the evolutionary search has better chances of getting out of a local optimum and can adjust an individual near the boundary, Hamdan (2010). Details of SBX and PM are explained as follows:

The procedure of computing two children $x_i^{(t+1)}$ and $x_j^{(t+1)}$ from their parents $x_i^{(t)}$ and $x_j^{(t)}$ using SBX is described as follows:

- **Step 1**: using uniform random algorithm to randomly create a number $u_i \in [0, 1]$.
- **Step 2**: the ordinate β_{qi} is calculated using a specified probability distribution function, so that the area under the probability curve from 0 to β_{qi} is equal to u_i and β_{qi} is computed using the following equation:

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{if} \quad u_i \leq 0.5; \\ (\frac{1}{2(1-u_i)})^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases} \tag{4.7}$$

- **Step 3**: after obtaining β_{qi}, the children are then calculated using the following equations:

$$x_i^{(t+1)} = 0.5[(1+\beta_{qi})x_i^{(t)} + (1-\beta_{qi})x_j^{(t)}] \tag{4.8}$$

$$x_j^{(t+1)} = 0.5[(1-\beta_{qi})x_i^{(t)} + (1+\beta_{qi})x_j^{(t)}] \tag{4.9}$$

where η_c is the distribution index and it is a user-specified parameter with any non-negative real value. The larger the distribution index, the higher the probability for generating "near-parent" solutions. If η_c gets a small value, it is likely that offspring will be created far from their parents. Furthermore, for a fixed η_c, the distance between offspring is proportional to that of their parents:

$$(x_i^{(t+1)} - x_j^{(t+1)}) = \beta_{qi}(x_i^{(t)} - x_j^{(t)}) \tag{4.10}$$

This property of SBX is very important as the solutions are randomly placed at the beginning of the evolutionary search, therefore, children can be created anywhere in the search area. When the population converges to local areas, far-distance offspring are not allowed because the search needs to focus on these narrow regions, Deb (2008).

Polynomial Mutation: PM also uses a probability distribution to alter values of decision variables in individuals, Deb and Goyal (1996). Each decision variable in a solution has a mutation probability P_m and it is recommended that $P_m = \frac{1}{n}$ where n is the number

of decision variables, as a result, one variable gets mutated per offspring on an average. For a given variable $x_i^{(t+1)}$, the mutated variable $y_i^{(t+1)}$ is created using the following procedure:

- **Step 1**: a random number *rand* within $[0, 1]$ is drawn.
- **Step 2**: if $rand > P_m$ then $x_i^{(t+1)}$ is not mutated and this procedure is terminated. If $rand \leq P_m$ then:
  - another random number r_i within $[0, 1]$ is created.
  - either of two parameters $\bar{\delta}_L$ and $\bar{\delta}_R$ are calculated from the polynomial probability distribution, as follows:

$$\begin{aligned} \bar{\delta}_L &= (2r_i)^{\frac{1}{(\eta_m+1)}} - 1, && \text{if} \quad r_i \leq 0.5, \\ \bar{\delta}_R &= 1 - [2(1-r_i)]^{\frac{1}{(\eta_m+1)}}, && \text{if} \quad r_i > 0.5. \end{aligned} \tag{4.11}$$

- **Step 3**: The mutated variable is then created, as follows:

$$\begin{aligned} y_i^{(t+1)} &= x_i^{(t+1)} + \bar{\delta_L}(x_i^{(t+1)} - x_i^{(L)}), && \text{for} \quad r_i \leq 0.5, \\ y_i^{(t+1)} &= x_i^{(t+1)} + \bar{\delta_R}(x_i^{(U)} - x_i^{(t+1)}), && \text{for} \quad r_i > 0.5. \end{aligned} \tag{4.12}$$

  where $x_i^{(U)}$ and $x_i^{(L)}$ are upper and lower bounds of $x_i^{(t+1)}$, respectively.

Distribution index η_m is a user-defined index parameter, which can take any non-negative value, and Deb and Agrawal (1995) has found that a value from 20 to 100 of η_m is adequate in most problems that they have tested. η_m determines the shape of the offspring distribution as it procedures distant offspring from the parent when it has small values and vice versa. There is a scope for crossover and mutation probabilities to be adapted and better ones could be found. However, this is not part of this study and have therefore been fixed to 0.9 and 0.1, respectively.

4.4 The surrogate model

As mentioned earlier that multi-objective optimization for traffic signal control systems using traffic simulators is computationally expensive. It means that it is time-consuming

to evaluate and assign the fitness value to an individual, which mainly depends on the number of fitness evaluations and the running time of traffic simulators. The latter one is determined by the scale of the traffic network and it is not considered in this work. Although NS-LS may work effectively in various conditions, the computational cost reduction is not considered in NS-LS. Consequently, we construct a surrogate model to estimate the fitness value of candidate solutions in the optimization process.

4.4.1 Constructing a surrogate model

4.4.1.1 Choosing the model

Feedforward Neural Networks (FNNs), also known as Deep Feedforward Networks or Multi-layer Perceptrons are suitable for modelling relationship between input variables and responses. FNNs have been widely used for a variety of approximation tasks. Therefore, an FNN is utilized to build the surrogate model in this work. The employed FNN includes one input layer, two hidden layers, and one output layer. The structure of the surrogate is illustrated in Figure 4.5.

The input layer includes n nodes corresponding to n phases of the signal cycle. $i^{(th)}$ input node receives a positive integer value indicating the green duration of $i^{(th)}$ phase. The output layer consists of m nodes, where m is the number of objective functions. The activation function deployed in this work is the sigmoid function:

$$\phi(z) = \frac{1}{1 + e^{-az}} \tag{4.13}$$

where a is a constant. An example of sigmoid function with $a = 4$ is plotted in Figure 4.6. The output of the sigmoid function is always between zero and one.

The other hyperparameters, such as the number of neurons in each hidden layer and the learning rate, are determined by grid-search and cross-validation techniques which are described in the next section.

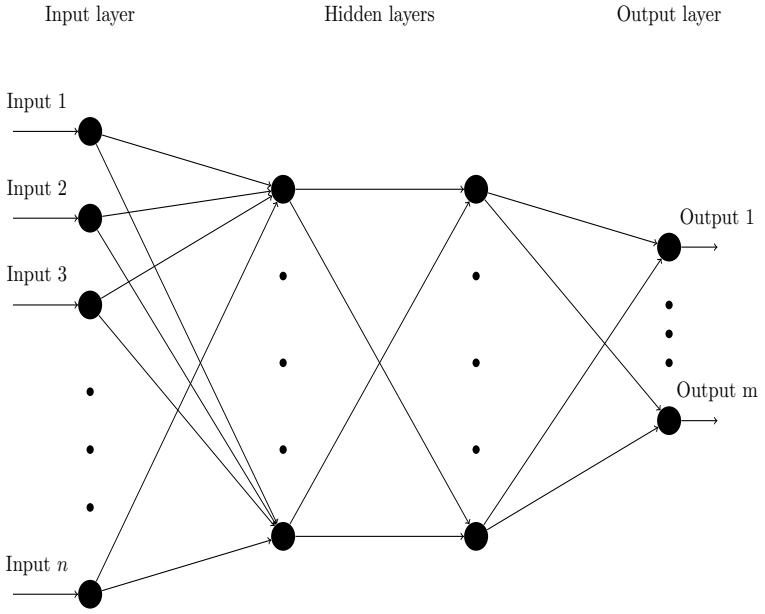


FIGURE 4.5: Overall structure of the surrogate model.

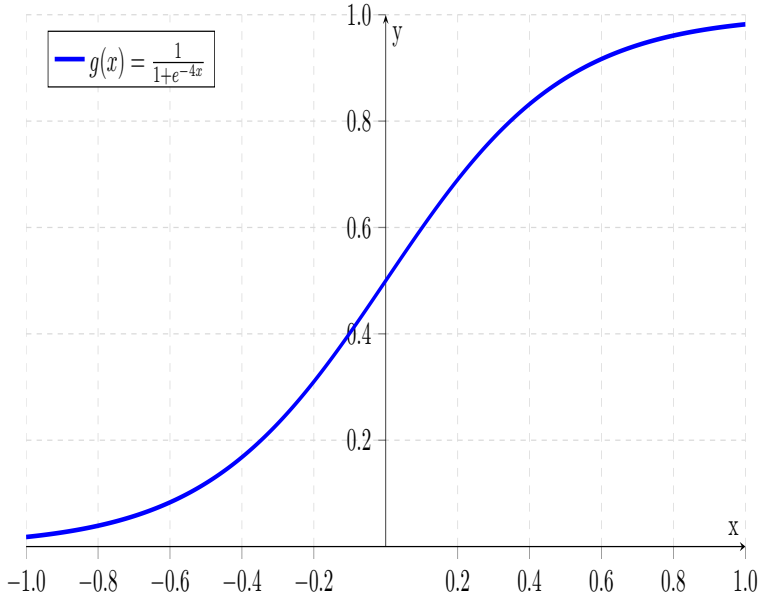


FIGURE 4.6: Sigmoid function with $a = 4$.

4.4.1.2 The training algorithm

The surrogate model is trained by Resilient Back Propagation Algorithm (RPROP). RPROP, introduced by Riedmiller and Braun (1993), is an efficient learning algorithm and considered as a well performing algorithm in terms of convergence speed, accuracy, and robustness. This learning algorithm directly adjusts the weight step based on the local gradient information. Each weight w_{ij} of the link from node i to node j has its individual update-value Δ_{ij} and is calculated as follows:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)}, & \text{if} \quad \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)}, & \text{if} \quad \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{else} \end{cases} \tag{4.14}$$

where $0 < \eta^- < 1 < \eta^+$ and E is the error produced by the training data. Size of the weight-update is determined by the update-value using the following rule:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{if} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)}, & \text{if} \quad \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0, & \text{else} \end{cases} \tag{4.15}$$

The number of learning steps in RPROP is significantly reduced in comparison to the original gradient-descent procedure. RPROP is fast, accurate, and easy to implement. Therefore, RPROP is a promising choice for training neural networks for applications having time related constraints.

4.4.1.3 The error function

The training data are used to train the surrogate. The input is passed through the network and the corresponding output is obtained from the network. The difference between the estimated output and the actual output is then calculated. This error is used to adjust the weights and bias of the neurons so that the error decrease gradually. A typical error functions for a learning algorithm is the mean square error (MSE) of the difference between the actual outputs and the outputs obtained from training samples. It is defined as follows:

$$E = \frac{1}{|L|} \sum_{p=1}^{|L|} e_p \tag{4.16}$$

where $|L|$ is the number of evaluated solutions using for training the surrogate and e_p is the error of $p^{(th)}$ sample, which is calculated by the following formula:

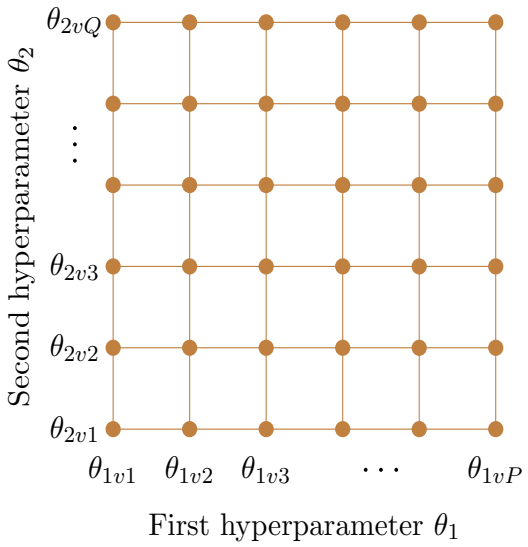


FIGURE 4.7: Grid search for hyperparameter fine-tuner.

$$e_p = \frac{1}{2}\sum_{m=1}^{M} ||\hat{y_m} - y_m||^2 \tag{4.17}$$

where M is the number of outputs, $\hat{y_m}$ is the $m^{(th)}$ output of estimated by the approximation model and y_m is $m^{(th)}$ expected output.

4.4.1.4 Hyperparameter tunning

Hyperparameter optimization or tuning is used to find an optimal set of hyperparameters for a machine learning algorithm. This optimal set can produce a model which minimize a predefined loss function on the given data. Grid search is a traditional technique for optimizing hyperparameters. It is an exhausted search which tests all possible combinations of hyperparameters. Although computational time of grid search may be longer than other techniques, it can be easily parallelized as each combination of these hyperparameters are independent. Furthermore, the number of hyperparameters as well as the number of solutions in the database L is are relatively small. Consequently, grid search is used to optimize hyperparameters in this research. An example of a grid search for two hyperparameters is illustrated in Figure 4.7, where the first hyperparameter includes P values while the second hyperparameter consists of Q values.

The n-fold cross-validation technique is utilized in the fine-tuning procedure to overcome under-fitting and over-fitting problems. The progress of the cross-validation progress is

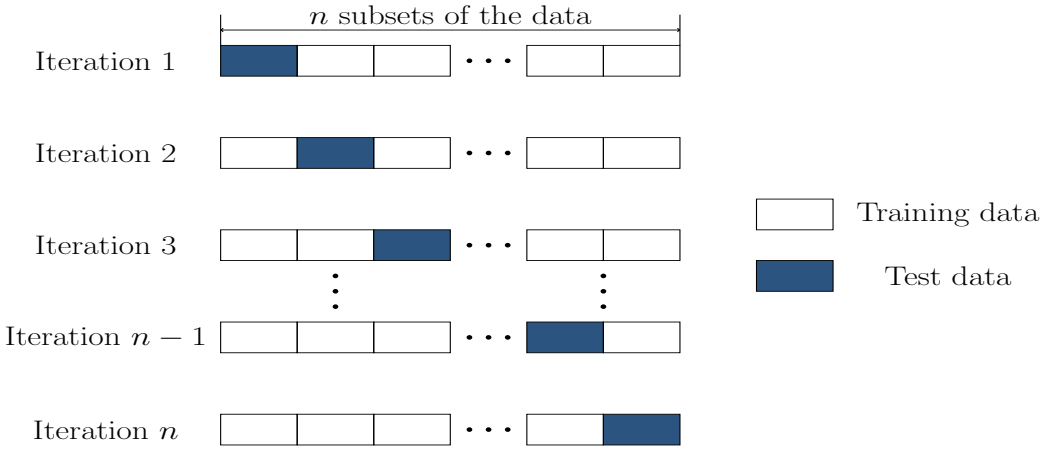


FIGURE 4.8: The n-fold cross validation technique.

Algorithm 5 The pseudo code the hyperparameter tuner using grid searching where L is the database consisting solutions evaluated by SUMO, H is a set of hyperparameters.

Procedure:
n=5
Divide L into n equal parts L_1, L_2,...,L_n
$A \leftarrow$ *Machine learning algorithm.*
for each θ in the H **do**
    for *i=1 to n* **do**
        $M_{i,\theta} = A(L \backslash L_i, \theta)$
        $E_i = L(M_{i,\theta}, L_i)$
    end for
    $E(\theta) = \frac{1}{n} \sum_{i=1}^{n} E_i$
end for
$\theta_{best} = min(E(\theta))$
$M_{\theta_{best}} = A(L, \theta_{best})$

illustrated in Figure 4.8. The training data is divided into n roughly equal parts. For each $i^{(th)}$ part, $i \in [1, n]$, keep this part as a test dataset and take the remaining parts as the cross-validation training set. Train the surrogate model using the cross-validation training set. Evaluate the model using the validation dataset to obtain an error E_i of the model. The cross-validation error E_c of the model is the average value of E_i, $i \in [1, n]$:

$$E_c = \frac{1}{n} \sum_{i=1}^{n} E_i \tag{4.18}$$

There is no formal rule for choosing the value of n. As n gets larger, the bias of the technique becomes smaller, Kuhn and Johnson (2013). A value of n equal to 5 or 10 is recommended as 5-fold and 10-fold cross-validations are proved to work effectively in most cases. In this research, $n = 5$ is selected for the cross-validation.

The pseudo code of the hyperparameter tuner is illustrated in Algorithm 5. H is a set of hyperparameters needed to be fine-tuned and L is the database consisting all solutions already evaluated by SUMO in the previous generations. Grid search and cross-validation techniques are then used to evaluate every possible combination of those values. The combination of hyperparameters which produces the lowest error is chosen for the overall structure of the surrogate model. The procedure of the hyperparameter fine-tuning is briefly described as follows:

- *Step 1*: Divide the data in the database L into n equal parts.
- *Step 2*: For each θ in H, leave one part of the data as the validation data and train the surrogate using the other parts. Evaluate the setting using the validation data and return an error value. Choose another part of data as the validation and iterate this procedure until all part have been selected to be the validation data. Average error of the surrogate associated with the θ is then calculated.
- *Step 3*: The best setting which is one with the smallest training error is chosen to construct the architecture for the surrogate. Re-train the surrogate using the best setting and return the best architecture of the surrogate.

During the optimization process, the size of L continuously increases as any solution newly evaluated by SUMO will be added into L. Different datasets may lead to different architectures of the surrogate. Therefore, this hypeparameter tuning process is performed during the optimization process.

4.4.2 Updating a surrogate model

It is important that the surrogate is updated properly so the evolutionary search can converge to a correct optimum. Firstly, the surrogate is checked whether or not it needs to be updated. If the error of the surrogate exceeds a given threshold, it will be updated using the database. Assume that all solutions newly generated in the current generation are achieved in a temporary database L_{temp}. The procedure to update the surrogate is described as follows:

- **Step 1**: estimate the average approximation error of all solutions in L_{temp} using the following equation:

$$err_{cur} = \frac{1}{2 * |L_{temp}|} \sum_{i=1}^{|L_{temp}|} \sum_{m=1}^{M} ||\hat{y_m} - y_m||^2 \tag{4.19}$$

  where $|L_{temp}|$ is the size of L_{temp}, M is the number of optimization objectives, $\hat{y_m}$ and y_m are values of $i^{(th)}$ objective, estimated by the surrogate and SUMO, respectively.

- **Step 2**: if $err_{cur} \leq \delta$, the surrogate is up to date, therefore, there is no need to update the surrogate and this procedure is terminated. If not, continue to the next step to update the surrogate.

- **Step 3**: all solutions in L_{temp} are added to the database L. The surrogate is re-trained using L based on k-fold cross-validation and grid search techniques.

By using the condition mentioned in Step 2, the computational cost can be reduced since the surrogate does not need to be reconstructed at every iteration. If the surrogate is not updated in the current generation, all solutions newly created and evaluated by SUMO in the current iteration are included into L for use in the succeeding generations. The more iterations the evolutionary search are done, the more solutions evaluated by SUMO are added into L. Therefore, the size of L increases during the optimization process. Consequently, the overall approximation accuracy of the surrogate would be improved.

4.5 Fitness evaluation scheme

After being trained, the surrogate model can partially replace SUMO to estimate the fitness value of solutions. This replacement may decrease the number of fitness evaluation significantly, however, it also can increase the approximation error, as a result, the search might converge to a false optimum. Therefore, it is very important that the surrogate should be combined with SUMO in an effective way. Furthermore, one of the critical questions when using the surrogate model is that which solutions should be evaluated by SUMO and which solutions might be estimated by the surrogate. In this

part, we introduce a fitness evaluation scheme to help decide which model a solution should use to evaluate its fitness value.

4.5.1 The motivation of the fitness evaluation scheme

Assume that there are 6 available samples $m_1, m_2, ..., m_6$, which are used to build an approximation model. Each available sample includes input and output data. Figure 4.9 illustrates the idea of using a fuzzy distance to determine which model is used to estimate the fitness value of a particular solution. The red solid line denotes the original fitness function where the blue dashed line is the approximation model and the dots represent the available samples. It is clearly shown in the figure that, the discrepancy between the original function and the approximation model is large when the number of available samples is not sufficient. For example, m_1, m_2, and m_3 are very near each other, so the approximation model can adequately learn the relationship between input and output of samples around m_1, m_2, and m_3. In other words, the approximation model can learn the part of the problem domain where is has sufficient training data. In contrast, m_4 and m_5 are further away from the other data and hence accurate interpolation cannot be guaranteed. Similarly, there is no available samples on the right-hand side of m_6, therefore, the learned model may not well extrapolate.

Suppose that there are 3 unseen inputs x_1, x_2, and x_3. The outputs calculated by the original fitness function are y_1, y_2, and y_3 while $\hat{y_1}$, $\hat{y_2}$, and $\hat{y_3}$ are the outputs estimated by the approximation model. In the figure, x_1 is close to m_1 and m_2, as a result, the estimated output $\hat{y_1}$ is very close to the output y_1 calculated by the original fitness function. The inputs x_2 and x_3 are not particular close to any available samples so their estimation errors $|y_2 - \hat{y_2}|$ and $|y_3 - \hat{y_3}|$ are very large. It is concluded that, if a new sample is found to be close to at least one of available samples, its estimation error might be small, therefore, the fitness value of this solution can be estimated using the surrogate model. In contrast, if it is far from all available samples, the error should be large, as a result, the real fitness function should be used and the model updated to include this new sample.

On the other hand, one critical criterion needs to be carefully considered when selecting the model to estimate the fitness value of one solution is how well the approximation

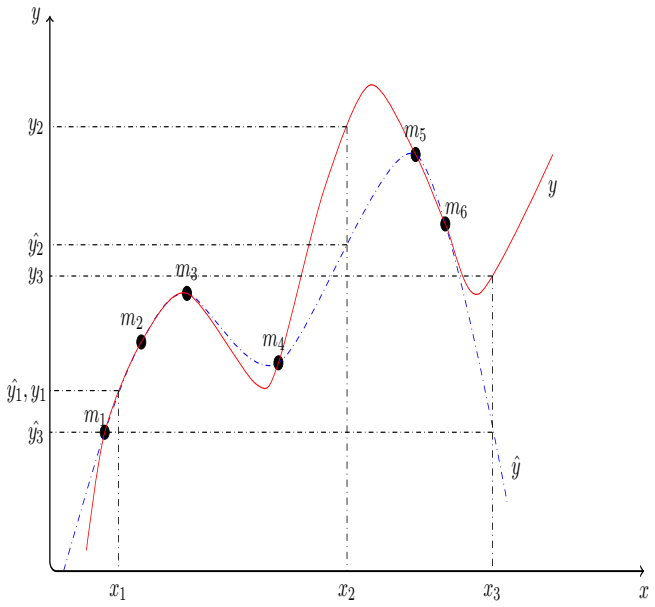


FIGURE 4.9: Relationship between distance and approximation error of new solutions and available solutions in the database. The red solid line represents the original fitness function while the blue dashed line is the surrogate and the dots $m_i, i \in [1, 6]$ denotes the available samples used to construct the approximation model.

model generalises. If the estimation error of the surrogate model is high, its approximation result would be not reliable, therefore, it should not be chosen to evaluate the solution. Consequently, the surrogate model is considered to be selected if and only if its approximation error is smaller than a pre-specified threshold.

For all the above reasons, we propose a fitness evaluation scheme based on Mean Square Error (MSE) of the approximation error obtained by the surrogate model and the closeness between the solution needed to be evaluated and samples used to construct the surrogate. The next section defines the closeness between a solution and a sample in the data set.

4.5.2 The closeness of two solutions

A solution in SA-LS is defined as a vector $(x_1, x_2, ..., x_n)$ where x_i, $i \in [1, n]$ is an positive integer indicating green duration of a phase in a cycle. For this work, a fuzzy set $\bar{A}$ is a set of long green times. According to the guidelines of the Highway Capacity Manual (HCM) 2010, the recommended value for maximum of green duration is 60 seconds Board (2010). Therefore, in this study, a green duration is defined as long if it exceeds 60 seconds and the membership function of $\bar{A}$ is described as follows:

$$\mu_{\bar{A}}(x_i) = \begin{cases} \frac{x_i}{60} & \text{if} \quad x_i \leq 60 \\ 1 & \text{if} \quad x_i > 60. \end{cases} \tag{4.20}$$

Therefore, a solution $(x_1, x_2, ..., x_n)$ can be represented as $(\mu_{\bar{A}}(x_1), \mu_{\bar{A}}(x_2), ..., \mu_{\bar{A}}(x_n))$ where $\mu_{\bar{A}}(x_i)$ is the degree of membership of x_i in the fuzzy set $\bar{A}$.

Assume that there are two solution $P_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, ..., x_{in}^{(t)})$ and $P_j^{(t)} = (x_{j1}^{(t)}, x_{j2}^{(t)}, ..., x_{jn}^{(t)})$ The fuzzy distance between two solutions is calculated using the following formula:

$$d(P_i^{(t)}, P_j^{(t)}) = \sqrt{\sum_{i,j,k=1}^{n} (\mu_{\bar{A}}(x_k i) - \mu_{\bar{A}}(x_k j))^2} \tag{4.21}$$

A threshold ϵ is used to determine the closeness of two solutions. If $d(X_i, X_j) \leq \epsilon$ then X_i and X_j are close. The minimum fuzzy distance between a solution and the data set is defined as the smallest fuzzy distance between that solution and all samples in the data set. This distance will be used to measure the closeness between a solution and a dataset.

4.5.3 The framework of the fitness evaluation scheme

The fitness evaluation scheme, employed in this study, is based on the closeness of the solution to the samples in the database, which is used to construct the surrogate model, and the MSE of the approximation error of the surrogate model. The basic idea of the fitness evaluation scheme is that a solution is evaluated by the surrogate if and only if this solution is close enough to the dataset and the surrogate is reliable. If the solution is very close to samples in the database but the surrogate itself is not reliable, its fitness value will not be estimated by the surrogate model. The closeness of the solution to the data set is defined by the minimum fuzzy distance definition. The framework of the scheme is described in Figure 4.10 and the pseudo code is illustrated in Algorithm 6.

Assume that there are $|L|$ solutions $X_m = (x_{m1}, x_{m2}, ..., x_{mn})$ in the database L where $m \in [1, |L|]$ and n is the number of variables, which are used to construct the surrogate and we need to calculate fitness value of solution $P_i^{(t)}$. Firstly, the reliability of the surrogate is assessed as follows: the data in the database L is separated into two different

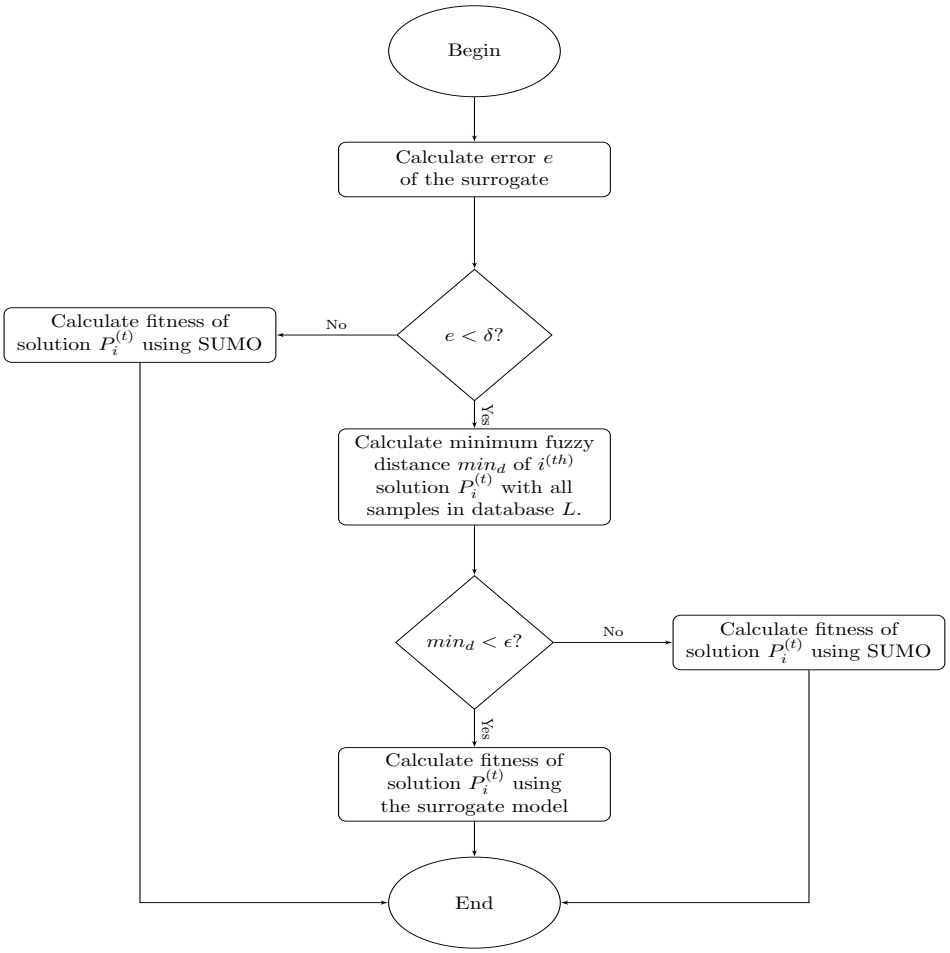


FIGURE 4.10: The framework of the fitness evaluation scheme.

sets which are training set and validation set. Firstly, the surrogate is trained by the training data set. After the training process is finished, the validation data is used to evaluate the performance of the surrogate model and the obtained error is e calculated by equation 4.16.

- If $e > \delta$, the surrogate is not reliable and fitness value of $P_i^{(t)}$ is calculated by SUMO.
- If $e < \delta$, fuzzy distance $d(P_i^{(t)}, X_m)$ between $P_i^{(t)}$ and $m^{(th)}$ sample X_m in database L, with $m \in [1, |L|]$, is calculated. The minimum distance min_d is then found out. If min_d is smaller than ϵ, $P_i^{(t)}$ is estimated by the surrogate. If not, its fitness value is calculated by SUMO.

Solutions estimated by SUMO are added into L for updating the surrogate model in the next generations.

Algorithm 6 The pseudo code of the fitness evaluation scheme

Input: *Solution* $P_i^{(t)}$*, the database* L *including* $|L|$ *solutions* $X_m(x_{m1}, x_{m2}, ..., x_{mn})$ *where* $m \in [1, |L|]$ *and* n *is the number of variables,* δ *and* ϵ*.*
Output: *Fitness value of solution* $P_i^{(t)}$
Procedure:
Split L *into two data sets: training and validation sets.*
Train the surrogate with the training set.
Calculate the error e *of the surrogate using the validation set.*
if $e > \delta$ **then**
    Calculate fitness value of $P_i^{(t)}$ *using SUMO.*
    $L_{temp} \leftarrow P_i^{(t)}$
else
    $min_d := 0.$
    for $m := 1$ to $|L|$ **do**
        Calculate the fuzzy distance $d(P_i^{(t)}, X_m)$ *between* $P_i^{(t)}$ *and* X_m*.*
        if $min_d > d(P_i^{(t)}, X_m)$ **then**
            $min_d := d(P_i^{(t)}, X_m)$
        end if
    end for
    if $min_d < \epsilon$ **then**
        Calculate fitness value of $P_i^{(t)}$ *using the surrogate.*
    else
        Calculate fitness value of $P_i^{(t)}$ *using SUMO.*
        $L_{temp} \leftarrow P_i^{(t)}$
    end if
end if
Return *Fitness value of solution* $P_i^{(t)}$*.*

4.6 SA-LS algorithm

This section introduces a surrogate-assisted evolutionary algorithm SA-LS which is an enhancement of NS-LS to overcome this limitation. The main contribution of SA-LS is a surrogate model constructed through solutions which have been already evaluated by SUMO in previous generations. This surrogate model is more computationally efficient than SUMO and it is partially used together with SUMO to evaluate the fitness value for solutions during the evolutionary search. Consequently, the total number of solutions evaluated by SUMO in one generation would be reduced. Consequently, if a pre-defined total number of solution evaluations using SUMO is used as the termination condition for different evolutionary algorithms, the number of generations in the entire search of SA-LS would be larger than that of the other algorithms. Therefore, using the same number of expensive solution evaluations, SA-LS would obtain better results. Furthermore, the

local search employed in NS-LS is also utilized in SA-LS to effectively accelerates the convergence of NS-LS.

4.6.1 Overview of SA-LS

Surrogate-assisted multi-objective evolutionary algorithms have been widely used to reduce the computational cost of fitness evaluation in optimizing expensive problems. Using surrogate models to estimate the fitness value of solutions greatly reduce the computational burden since the cost for constructing surrogate models and managing them is lower than that of evaluating fitness values using expensive objective functions. Consequently, surrogate-assisted evolutionary algorithms are very promising for solving traffic signal optimization problems. Therefore, we proposed SA-LS which is an enhancement of NS-LS algorithm introduced in the previous section using a surrogate model. This model is able to estimate the fitness value of solutions and it is partly used with SUMO to evaluate fitness values in every generation of the optimization process to avoid the search converging to a false optimum.

Effective use of SUMO and the surrogate model is extremely important to reduce the computational cost of SA-LS. An adaptive generation-based evolution control is utilized in this work to deals with the following questions:

1. How to balance the reduction of the number of expensive fitness evaluations and the approximation accuracy of the surrogate model?
2. Which solutions should be evaluated by the surrogate model instead of SUMO?
3. How to update the surrogate model?

This model management strategy is critical to the effective working of the optimization process. A suitable strategy would significantly reduce computational cost while leading the evolutionary search to converge to correct optima.

The main framework of the proposed algorithm SA-LS is illustrated in Figure 4.11. The main new contributions proposed in this chapter are highlighted by the shaded blocks, including a surrogate construction procedure, a fitness evaluation scheme, a database to accommodate all solutions evaluated by SUMO, and a surrogate update strategy. The algorithm contains three main parts:

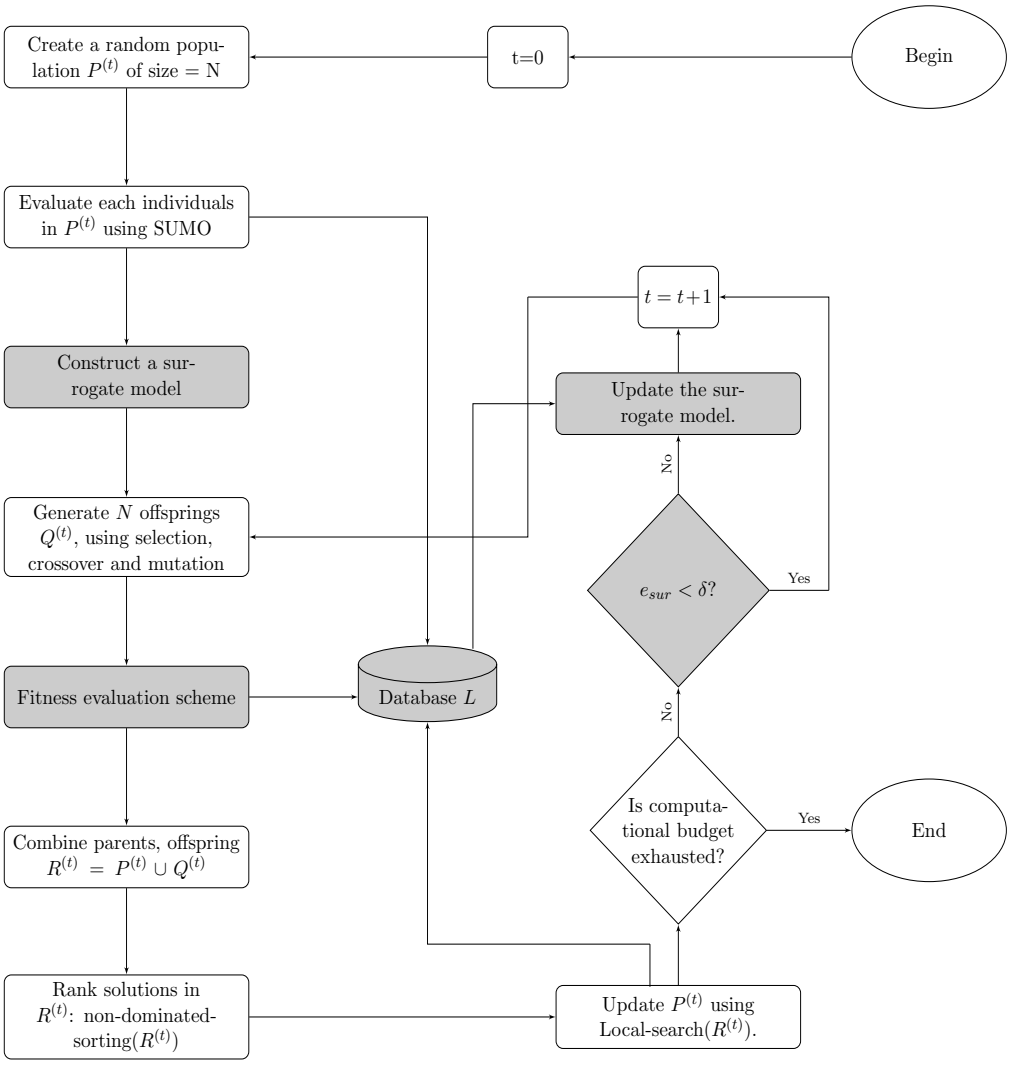


FIGURE 4.11: The framework of the proposed algorithm SA-LS with the new contributions are in shaded blocks. err_{cur} and δ are average approximation error and pre-defined threshold of the surrogate, respectively.

1. The main iterative process is inherited from NS-LS. The main aim of this part is to move the population of solutions toward the optimal front through a number of generations. This part includes binary tournament selection scheme and reproduction procedures to create offspring, as a result, SA-LS can explore new search areas. SA-LS is terminated when a pre-defined maximum number of solution evaluations using SUMO is reached.

2. Surrogate construction procedure, a fitness evaluation scheme, and a surrogate update strategy are introduced to enhance the performance of NS-LS. A Feed-forward Neural Network (FNN) is chosen to construct the surrogate model. Fitness value of a solution is estimated by either SUMO or the surrogate model. A fitness evaluation scheme is proposed based on a fuzzy distance and a Mean Squared Error

(MSE) of the surrogate model on the training data. This scheme decides which model, SUMO or the surrogate model, is used to estimate the fitness value of a particular solution. An update strategy for the surrogate model is also introduced to reduce the approximation error.

3. The local search method looks for neighbours which potentially have good fitness values in local areas. All solutions in the population and neighbours created in the local search procedure then participate in the selection process to choose the best solutions for the next generation, therefore, the population can move more quickly towards the optimal front. The local search utilized in SA-LS is also inherited from NS-LS.

During the first few generations of the optimization process, the number of samples archived in the database is small, therefore, the number of solutions estimated by SUMO would be small. In these generations, the local search would help SA-LS increase the convergence speed. When a large number of solutions are included in the database, more individuals in the population would be evaluated by the surrogate model. Consequently, the number of solutions evaluated by SUMO would be reduced and the computational time of a generation in SA-LS would be decreased.

4.6.2 The flow of SA-LS

The pseudo code of the proposed algorithm SA-LS is illustrated in Algorithm 7. Details of the flow of the optimization process are described as follows:

Step 1: Randomly initialize a population $P^{(0)}$ with N individuals.

Step 2: Evaluate the fitness value of N solutions in the population using SUMO.

Step 3: A Feedforward Neural Network is utilized to build a surrogate model which represents the global trend of the entire fitness landscape, using solutions evaluated by SUMO in the previous step. This surrogate is partly used, together with SUMO, to reduce the number of traffic simulator-based evaluations.

Step 4: A database L is created to store all solutions whose fitness values are estimated by SUMO. All individuals in $P^{(0)}$ are then archived in L.

Algorithm 7 The pseudo code of SA-LS

Input:
    N: population size of the optimization process,
    δ: a pre-defined threshold of the estimation error of the surrogate,
    maxEval: a pre-specified maximum number of fitness evaluations using SUMO,
    $P^{(t)}$ and $Q^{(t)}$ are parent and children populations of t^{th} iteration, respectively.
Procedure:
$t := 0$, $numEval = 0$, $Q^{(t)} \leftarrow \emptyset$;
Generate N random solutions and insert them into population $P^{(t)}$: $P^{(t)} \leftarrow$ Random-population().
Evaluate every solution in $P^{(t)}$ using SUMO. These solutions are then archived in a database L.
$numEval = numEval + N$
Use a FNN to construct a surrogate model using L.
while $numEval < maxEval$ **do**
    Generate an offspring population $Q^{(t)}$ of size N from $P^{(t)}$ by using selection, crossover, and mutation operators.
    Assign a fitness value for each solution in $Q^{(t)}$ using the fitness evaluation scheme and reserve all solutions evaluated by SUMO to the database L.
    $numEval = numEval + H$, where H is the number of solutions in $Q^{(t)}$ evaluated by SUMO.
    $R^{(t)} \leftarrow P^{(t)} \cup Q^{(t)}$
    Apply the local search to $R^{(t)}$: $P^{(t)}$ = local-search($R^{(t)}$).
    Compute the approximation error err_{cur} of the surrogate using L_{temp}, where L_{temp} is a temporary database containing solutions evaluated by SUMO during the local search process.
    $L \leftarrow L \cup L_{temp}$
    if $err_{cur} > \delta$ **then**
        Update the surrogate model using L.
    end if
    $t \leftarrow t + 1, P^{(t+1)} \leftarrow P^{(t)}$
end while
Return *P*.

Step 5: The initial population undergoes a transformation through a number of generations until the termination criteria are satisfied. In $t^{(th)}$ iteration, the population $P_i^{(t)}$ undergoes the following sub-steps:

- *Step 5.1*: Create a children population $Q^{(t)}$, including N individuals, from population $P_i^{(t)}$ using tournament selection, recombination and mutation operators.
- *Step 5.2*: The fitness evaluation scheme is deployed to evaluate and assign a fitness value for each solution in $Q^{(t)}$. This fitness evaluation scheme determines which model, SUMO or the surrogate, is selected to evaluate a solution. Any solution, which is estimated by SUMO, is then added into L.

- *Step 5.3*: A population $R^{(t)}$ including $2N$ solutions is formed by including all solutions in $P^{(t)}$ and $Q^{(t)}$. Apply the local search method, which is illustrated in section 4.2, to $R^{(t)}$.

- *Step 5.4*: Update the population for the next generation $P^{(t+1)}$ by the new population found in the local search.

- *Step 5.5*: Check whether or not the surrogate model needs to be updated. All solutions, which are newly evaluated by SUMO in steps 4.2 and 4.3, are used to evaluate the correctness of the surrogate model.

Step 6: If the termination condition is reached then the algorithm is terminated. Because the fitness evaluation using SUMO is computational expensive, therefore, $maxEval$, which is a pre-defined maximum number of fitness evaluations using SUMO, is used as the termination condition of SA-LS. If $maxEval$ is not reached, the search goes to the next iteration and continues searching by repeating step 4 until the termination condition is satisfied.

In other proposed surrogate-assisted evolutionary optimization algorithms, surrogate models are only constructed after several initial generations. The reason is that the surrogates need enough data to adequately learn the underlying relationship between input and output data. If not, the approximation error would be very large, which leads the search to inadequate solutions. Therefore, during the first few generations, only the expensive fitness function is used to evaluate solutions and these exact fitness values are archived in a database for constructing the surrogates later. This period is also called the collecting data phase. However, in this study, the surrogate is built from the beginning of the optimization process, and it is checked whether or not it needs to be updated in every iteration. To decrease the negative effects which would be introduced by the surrogate due to the lack of training data at the beginning of the optimization process, the fitness evaluation scheme is utilized to decide which model would be used to evaluate the fitness value of a particular solution. A solution will be evaluated by SUMO if the conditions of the fitness evaluation scheme are satisfied. In that case, the number of traffic simulator-based evaluations may be reduced. Although the number of surrogate-based evaluations may be small in the first few generations, there may be more solutions evaluated by the surrogate if it is constructed from the first generation.

On the other hand, any new solution which is evaluated using SUMO during the optimization process is added into the database L to update the surrogate. Therefore, the number of solutions in L has been continuously increased during the optimization process. Consequently, the approximation error of the surrogate would be decreased and there are probably more solutions evaluated by SUMO when the number of generations increases.

The surrogate model is used together with the traffic simulator to evaluate the goodness of candidate solutions. Individuals which satisfy two conditions of the fitness evaluation scheme are estimated by the surrogate model and the others are evaluated by the traffic simulator. Consequently, the number of traffic simulator-based evaluations in each generation of the optimization process will be reduced. Therefore, the proposed algorithm uses less computational expensive evaluations compared to evolutionary algorithms which do not use surrogate models. In other words, the computational burden of the proposed algorithm is reduced.

4.7 Conclusion

Anytime behavior of an algorithm is the ability to produce good results at any running time. In transportation optimization problems using MOEAs, good anytime behavior of optimization approaches is enviable. Furthermore, the ability to work effectively with small population sizes is also important in transportation optimization problems because of lacking processing capabilities while requiring quick responses. This section describes NS-LS, an optimization approach for urban traffic signal control systems. A local search method is presented to help the search move faster toward the optimal front. Consequently, NS-LS would able to produce good solutions at any running time, as a result, it would have good anytime behaviour. This is crucial in traffic management since urban traffic networks are highly dynamic and require near real-time solutions while traditional multi-objective optimization procedures are often extremely time-consuming, and therefore, infeasible for use in online traffic management.

Multi-objective transportation optimization problems, which utilizes a traffic simulator to estimate a fitness value, are computationally expensive. Therefore, to reduce the burden of computational cost of fitness evaluation, we proposed SA-LS which is an

enhancement of NS-LS. SA-LS is a surrogate-assisted evolutionary algorithm for transportation optimization problems. A feed-forward neural network is used to construct the surrogate model. Unlike surrogate models introduced in the previous studies, the surrogate model in this study is immediately built from the first generation of the iterative process. A management model based on fuzzy distance is introduced to use the surrogate correctly and effectively. Solutions, which are newly evaluated by SUMO during the optimization run, are archived in a database and the surrogate is frequently updated during the searching procedure using this database. Consequently, the surrogate can be effectively used from the beginning of the optimization process rather than waiting for several generations. A fitness evaluation scheme is introduced in this chapter to decide which model, surrogate or SUMO, is used to evaluate a solution. Furthermore, a local search method is utilized to improve anytime behaviour of the optimization algorithm. The combination of the surrogate model and the enhanced local search have the potential to reduce the number of solution evaluations using SUMO while producing good anytime behaviour.

Chapter 5

Experimental Setup

5.1 Introduction

To evaluate the performance of the proposed algorithms introduced in Chapter 4, a number of experiments are established using both benchmark test functions and traffic scenarios. To assess the performance of the proposed algorithms, NS-LS and SA-LS are compared to NSGA-II and MOEA/D using a number of performance indicators. This chapter describes the selected traffic scenarios and the experimental settings used to evaluate the algorithms. Furthermore, the procedure to achieve objective values of a solution from output files of SUMO is explained in this chapter. Indicators for evaluating and comparing the algorithms are also provided.

The chapter is organized as follows: two traffic scenario Andrea Costa and Pasubio used to evaluate the performance of the proposed algorithms are introduced in Section 5.2. The signal plan of the signalized intersections in the scenarios where the optimization algorithms are applied is also provided in this section. Method to extract traffic parameters from output files of SUMO is provided in Section 5.3. Section 5.4 introduces employed indicators for performance assessment. Experiments using benchmark functions are explained in Section 5.5.1 while Section 5.5.2 introduces experiments using real-time traffic scenarios simulated by SUMO. Section 5.6 concludes this chapter.

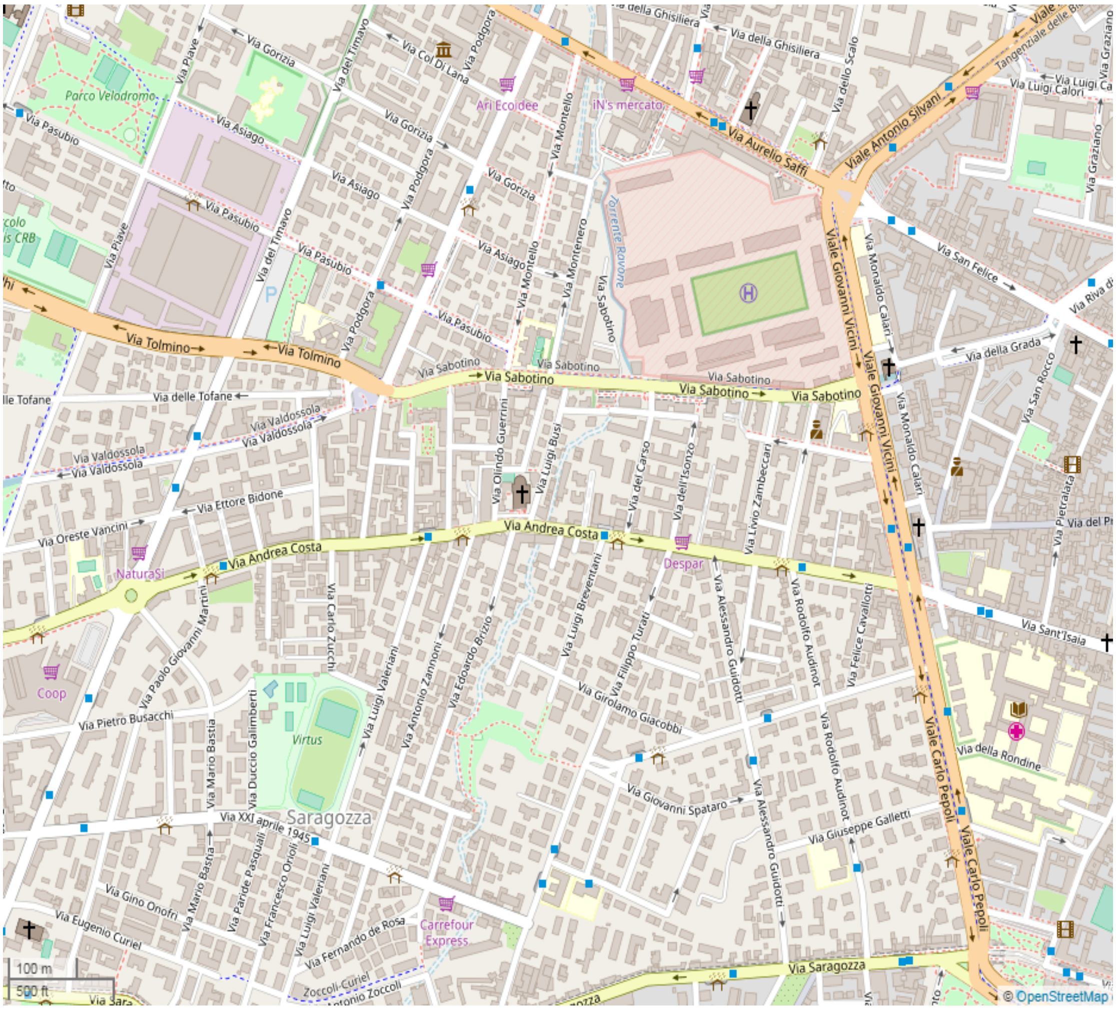


FIGURE 5.1: The traffic network of Andra Costa extracted from Open Street Map, Bieker et al. (2015).

5.2 Traffic scenarios

In transportation, it is difficult to evaluate a newly proposed traffic signal optimization approach under real-world conditions, as a result, traffic simulator is a preferable and popular tool. There is a lack of publicly available traffic data as well as traffic signal timing plans. Further, it is extremely time-consuming to construct traffic scenarios from scratch. Therefore, to address this problem, there are several available traffic scenarios, which have been introduced to help researches evaluate and investigate their research hypotheses with little effort at preparing experimental scenarios.

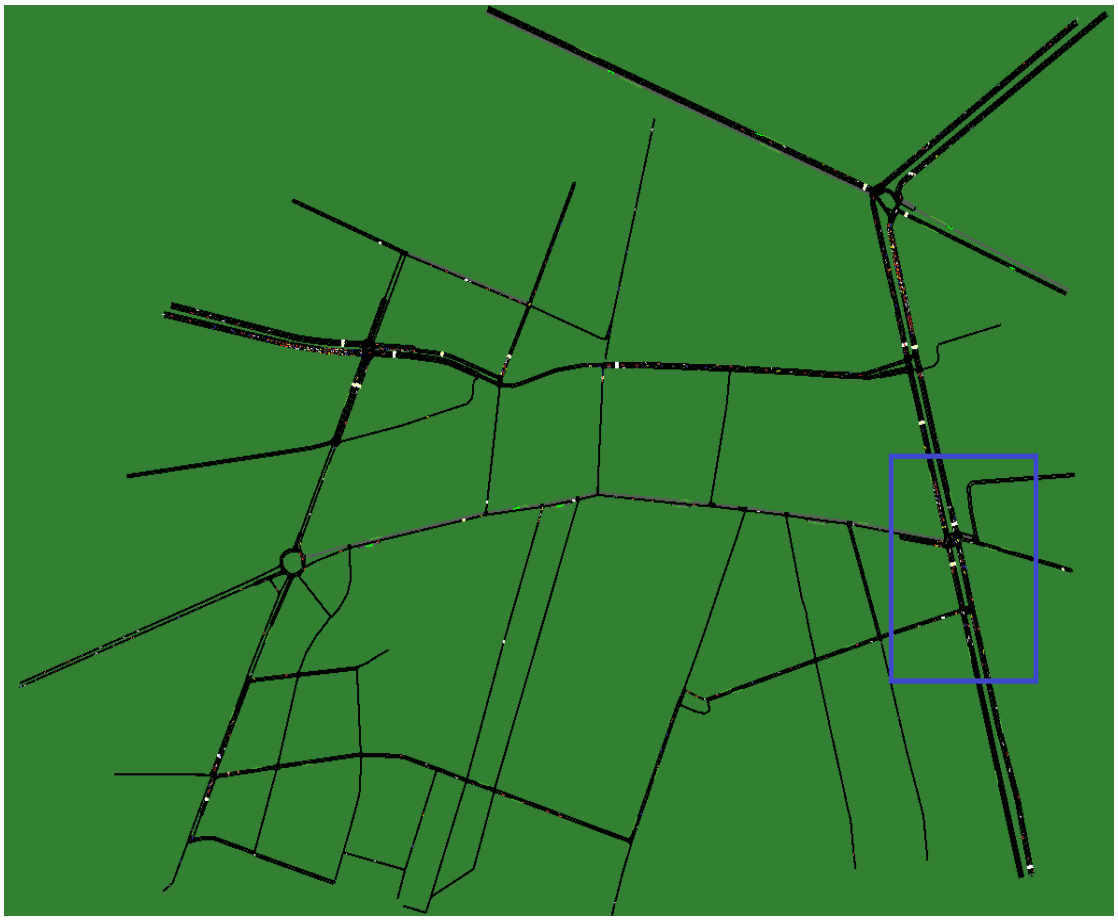


FIGURE 5.2: The Andrea Costa traffic map simulated by SUMO, the intersections where are optimized using the proposed algorithms are covered in the blue box.

5.2.1 Introduction to the traffic scenario of Andrea Costa

Andrea Costa is a scenario from Bologna city in Italy, which was provided to the research community by Bieker et al. (2015). This scenario describes the traffic at peak hour from 8:00 am to 9:00 am and is a good tool to evaluate traffic signal control algorithms as it is easy to use and is adequately provided with necessary components such as traffic network, traffic demand, and additional infrastructure. The road network of Andrea Costa extracted from Open Street Map is shown in Figure 5.1 and a typically pruned traffic network of the same area, which is simulated in SUMO, is shown in Figure 5.2. The case study where used to evaluate the performance of the algorithm is covered in the blue box on the right hand side of the figure. Some fundamental information about this scenario are as follows:

Traffic network: Andrea Costa scenario models the area located east of Bologna city, outside the inner city ring road. It consists of the area around the football stadium and

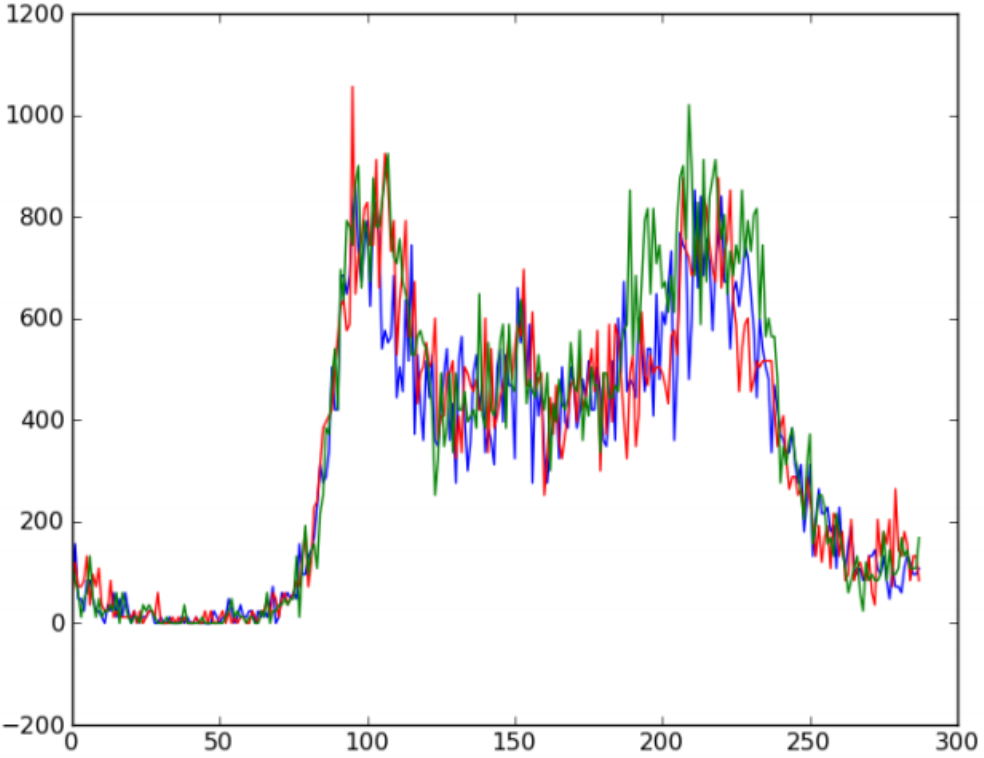


FIGURE 5.3: The traffic flow of three days in Bologna city provided by the municipality, taken from Bieker et al. (2015). The blue line indicates day 11, the red shows the traffic flow of day 12 and the green represents the day 13. The traffic flow is aggregated into intervals of 5 minutes.

this scenario is made to simulate the traffic movements of the area when big events, for example, concerts or football matches happen. Small streets are excluded from the scenario. Furthermore, public bus transport activity is also included in the scenario, consisting of bus stops, bus routes, and schedules. Furthermore, the positions of inductive loops and their measures are also provided by the municipality of Bologna.

Traffic light: Real traffic signal plans are available in this scenario including 7 traffic light control programs. Positions, types, and signal plans of traffic lights are provided in the acosta_tls.add.xml file[1].

Traffic demand of the scenario is constructed based on the dataset provided by the municipality of Bologna. The dataset consist of traffic measures collected from 636 detectors installed along the roads. They were measured in 3 usual working days, from Monday $11^{(th)}$ to Thursday $13^{(th)}$ November 2008. The traffic flow is aggregated into interval of 5 minutes indicating the number of vehicles passing through the detectors

[1]https://sumo.dlr.de/wiki/Data/Scenarios

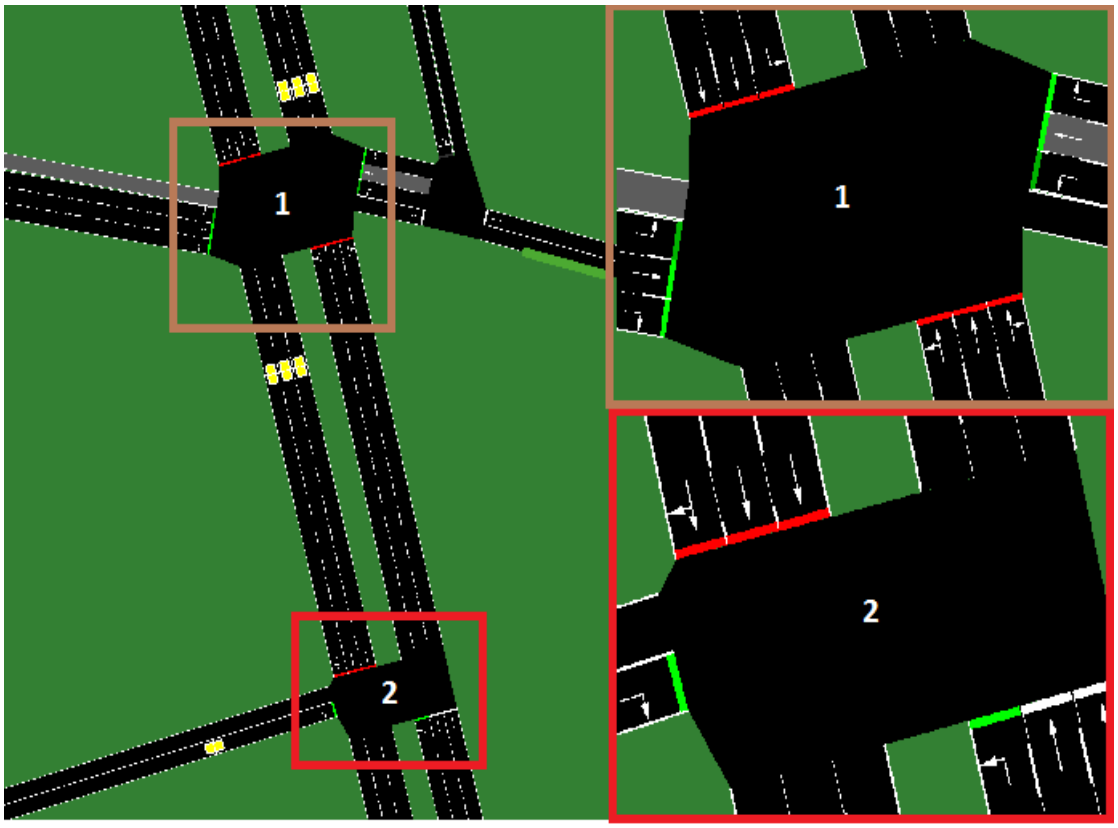


FIGURE 5.4: Case study area in Andrea Costa covered in the blue box on Figure 5.2 is on the left hand side of the figure. The signal control program includes two neighbouring intersections. The enlarged pictures of these two intersections are on the right hand side.

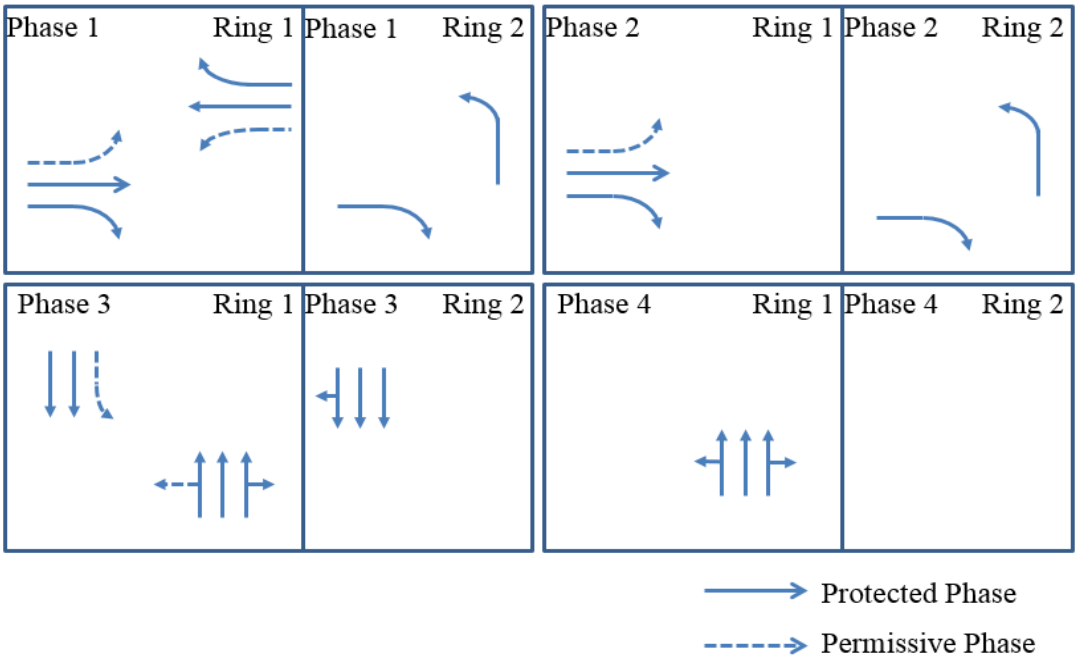


FIGURE 5.5: Phases of the signal control program of the case study in Andrea Costa.

within 5 minutes. The traffic flow of three days are illustrated in Figure 5.3. The dataset are then used to analyse and simulate the traffic demand of the scenario in peak hour from 8:00 am to 9:00 am and there are total 8622 vehicles simulated in this scenario. The traffic demand of the scenario is described in acosta.rou.xml file. The real measure data is also utilized to compare to the results of the simulation in order to validate the simulation.

Case study: the signal control program, which have been selected for the purpose of optimization within the proposed work, consists of two coordinated intersections and are shown in the left hand side of Figure 5.4. The enlarged pictures of the two intersections are on the right hand side. The signal control program of the case study illustrated in Figure 5.5 including four different phases. Ring 1 and Ring 2 describe the movements of the first and second intersections, respectively.

5.2.2 Introduction to the traffic scenario of Pasubio

Pasubio is another traffic scenario of Bologna city in Bieker et al. (2015). The scale of the road network in this scenario is relatively small, as a result, it does not take too much time to run the simulation. Furthermore, this scenario provides necessary elements, including traffic infrastructure and demand, for researchers in investigating their research questions. Therefore, this scenario is select to evaluate the transferability of the proposed algorithm. A traffic network of Pasubio, taken from Open Street Map, is illustrated in Figure 5.6.

Traffic network: Pasubio, which is next to Andrea Costa, includes the area around the hospital and main routes to the football stadium. The traffic network of Pasubio, which is simulated using SUMO, is provided in 5.7 and the case study where the optimization algorithms are applied to is covered in the blue box. The network also consists of public transport infrastructure, such as bus stops positions and bus routes. Traffic lights information, including positions, types, and signal plans, and positions of induction loops are also provided in this scenario.

Traffic demand: are collected from induction loops installed along the roads. All details about these detectors are provided in the pasubio_detectors.add.xml file. Similar to Andrea Costa, this scenario describes the demand for the peak hour from 8:00 am to

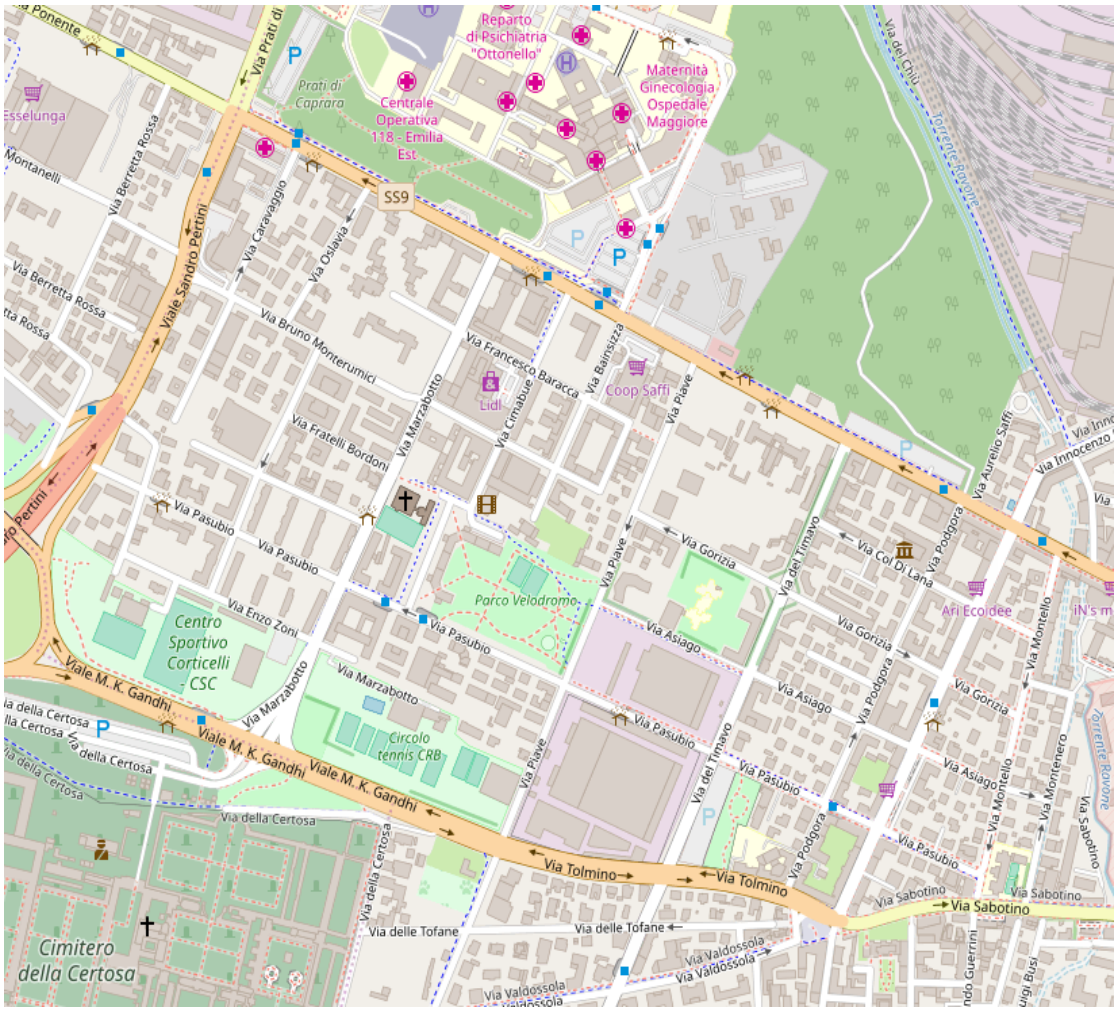


FIGURE 5.6: A traffic network of Pasubio taken from Open Street Map, Bieker et al. (2015).

9:00 am and there are 8681 vehicles simulated in this traffic scenario. The traffic demand of this scenario is determined from the dataset supplied by the local government of Bologna city. The procedure of collecting and processing the traffic flow of the Pasubio scenario is similar with the Andrea Costa scenario and the traffic demand is represented in pasubio.rou.xml file.

Case study: the intersections, which are selected to assess the performance of the proposed algorithms, are illustrated in the left hand side of Figure 5.8. The enlarged images of the intersections providing lanes and directions of movements at the crossed roads are shown in the right hand side of the figure. These two neighbouring intersections are coordinated by a common signal control program. Figure 5.9 illustrates the phases of the signal control program in the case study of Pasubio.

FIGURE 5.7: The Pasubio road network simulated by SUMO.

The authors validated the two scenarios by comparing the results of the simulation to the real world measurements. The comparison results show that these scenarios are able to simulate relatively well the number of vehicles (Bieker et al. (2015)). Consequently, they are good choices to evaluate the proposed algorithms with less effort to establish traffic scenarios.

To calculate the fitness value of a particular solution using SUMO, the optimizer sends values of signal timing parameters stored in variables of that solution, such as cycle length or green durations, to the traffic scenario via TraCI. SUMO simulates the traffic scenario using signal timing data provided via TraCI and returns XML output files. Required measurement results are then extracted from the output files and sent back to the optimization model in order to assign a fitness value to that solution. Further information about extracting traffic parameter results is provided in the next section.

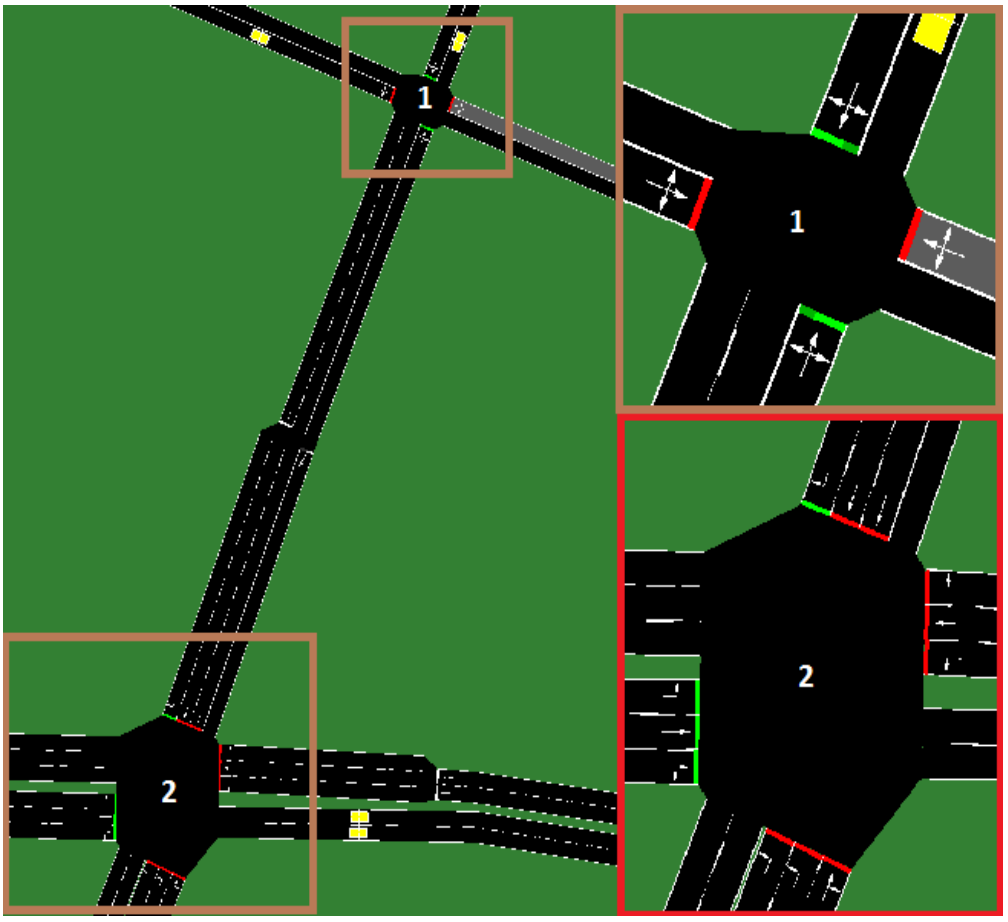


FIGURE 5.8: Case study area in Pasubio covered in the blue box on Figure 5.7 is on the left hand side of the figure. The signal control program consists of two neighbouring intersections. The enlarged pictures of the two intersections are on the right hand side.

5.3 Extracting optimization objective values from SUMO output

A large number of different traffic parameters can be extracted from output files of SUMO. By default, SUMO's output files are written in the XML-format. There are some measures which are automatically generated in the output files, such as trip information which is aggregated information about each vehicle's journey, emissions, and vehicle route information. The unavailable measures can be obtained by defining within additional files. Two objectives optimized in this study are minimizing the average time loss and maximizing the average traffic flow of traffic network. The procedures to obtain these objective values are described in the following.

A. Average time loss is the average time lost due to the intersection of all vehicle in the traffic network. Time loss is available in the tripinfos.xml file, which is an output of

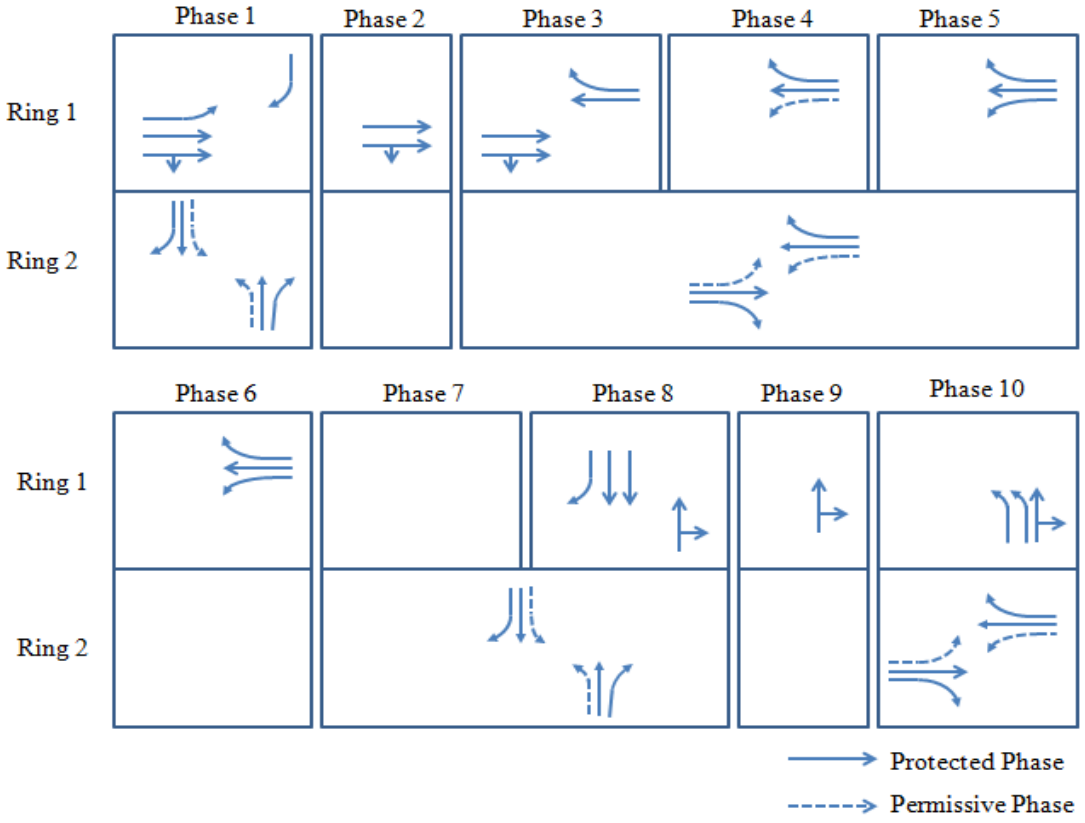


FIGURE 5.9: Phases of the signal control program of the case study in Pasubio.

```
<tripinfo id="Silvani_11_0" depart="0.00" departLane="203[0]_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" a
duration="88.00" routeLength="1114.20" waitSteps="0" timeLoss="7.00" rerouteNo="0" devices="tripinfo_Silvani_11_0" vTyp
<tripinfo id="Silvani_11_1" depart="2.00" departLane="203[0]_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" a
duration="87.00" routeLength="1114.20" waitSteps="0" timeLoss="6.79" rerouteNo="0" devices="tripinfo_Silvani_11_1" vTyp
<tripinfo id="Togliatti_2_8" depart="11.00" departLane="85_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="92.00" routeLength="1149.15" waitSteps="0" timeLoss="8.53" rerouteNo="0" devices="tripinfo_Togliatti_2_8" vTyp
<tripinfo id="Costa_1_4" depart="15.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="5.00" arr
duration="96.00" routeLength="1160.89" waitSteps="0" timeLoss="9.68" rerouteNo="0" devices="tripinfo_Costa_1_4" vType=
<tripinfo id="Costa_1_2" depart="9.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="4.00" arri
duration="104.00" routeLength="1160.89" waitSteps="3" timeLoss="17.69" rerouteNo="0" devices="tripinfo_Costa_1_2" vTyp
<tripinfo id="Costa_12_0" depart="0.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="115.00" routeLength="1413.22" waitSteps="0" timeLoss="12.65" rerouteNo="0" devices="tripinfo_Costa_12_0" vTy
<tripinfo id="Pepoli_8_16" depart="27.00" departLane="210_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="96.00" routeLength="693.93" waitSteps="32" timeLoss="45.31" rerouteNo="0" devices="tripinfo_Pepoli_8_16" vTyp
<tripinfo id="Costa_12_1" depart="6.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="4.00" arr
duration="119.00" routeLength="1413.22" waitSteps="0" timeLoss="15.98" rerouteNo="0" devices="tripinfo_Costa_12_1" vTy
<tripinfo id="Costa_3_21" depart="55.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="1.00" ar
duration="71.00" routeLength="853.79" waitSteps="0" timeLoss="8.75" rerouteNo="0" devices="tripinfo_Costa_3_21" vType=
<tripinfo id="Pepoli_8_21" depart="36.00" departLane="210_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="91.00" routeLength="693.93" waitSteps="26" timeLoss="40.64" rerouteNo="0" devices="tripinfo_Pepoli_8_21" vTyp
<tripinfo id="Costa_12_3" depart="12.00" departLane="78[0]_0" departPos="0.00" departSpeed="0.00" departDelay="5.00" ar
duration="116.00" routeLength="1413.22" waitSteps="0" timeLoss="13.64" rerouteNo="0" devices="tripinfo_Costa_12_3" vTy
<tripinfo id="Pepoli_8_23" depart="39.00" departLane="210_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="90.00" routeLength="693.93" waitSteps="24" timeLoss="39.24" rerouteNo="0" devices="tripinfo_Pepoli_8_23" vTyp
<tripinfo id="Pepoli_8_25" depart="42.00" departLane="210_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="88.00" routeLength="693.93" waitSteps="21" timeLoss="38.07" rerouteNo="0" devices="tripinfo_Pepoli_8_25" vTyp
<tripinfo id="Pepoli_8_28" depart="48.00" departLane="210_0" departPos="0.00" departSpeed="0.00" departDelay="0.00" arr
duration="84.00" routeLength="693.93" waitSteps="17" timeLoss="33.66" rerouteNo="0" devices="tripinfo_Pepoli_8_28" vTyp
```

FIGURE 5.10: A part of a trip information output file from the Andrea Costa scenario. This file is produced after the simulation finished containing departure and arrival times, time loss, and route length and other information.

```
▼<additional>
  <e1Detector id="0.127_2.73_6_1__l0" lane="104_0" pos="23" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.73_6_1__l1" lane="104_1" pos="23" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.73_4_1__l0" lane="15_0" pos="77.542050042" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.73_2_1__l0" lane="103_0" pos="118" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.73_2_1__l1" lane="103_1" pos="118" freq="1800" file="e1_output.xml"/>
 ▼<!--
    e1Detector id="2.19_2.19_6_1__l0" lane="83_0" pos="40" freq="1800" file="e1_output.xml"/
  -->
 ▼<!--
    e1Detector id="2.19_2.19_2_1__l0" lane="83_0" pos="37" freq="1800" file="e1_output.xml"/
  -->
  <e1Detector id="2.19_2.18_4_1__l0" lane="86_0" pos="31.3626881232" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_2.18_4_1__l1" lane="86_1" pos="31.3626881232" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_2.18_4_1__l2" lane="86_2" pos="31.3626881232" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_2.20_8_1__l0" lane="72[0]_0" pos="32.8995678051" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_2.20_8_1__l1" lane="72[0]_1" pos="32.8995678051" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_2.20_8_1__l2" lane="72[0]_2" pos="32.8995678051" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.19_6_1__l0" lane="11_0" pos="0" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.19_6_1__l1" lane="11_1" pos="0" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.19_6_1__l2" lane="11_2" pos="0" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_0.127_2_1__l0" lane="67_0" pos="42" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_0.127_2_1__l1" lane="67_1" pos="42" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.82_2.19_8_1__l0" lane="137_0" pos="158.698022913" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.19_0.127_8_1__l0" lane="180_0" pos="105" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.20_2.19_4_1__l0" lane="17_0" pos="100" freq="1800" file="e1_output.xml"/>
  <e1Detector id="2.20_2.19_4_1__l1" lane="17_1" pos="100" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.20_2_1__l0" lane="18_0" pos="1" freq="1800" file="e1_output.xml"/>
  <e1Detector id="0.127_2.20_2_1__l1" lane="18_1" pos="1" freq="1800" file="e1_output.xml"/>
```

FIGURE 5.11: A part of the acosta_detectors.add.xml file. This file is created to add detectors to the road network of the scenario. Each detector declared in this file includes identification of that detector, lane and position where it is installed and other information.

SUMO and generated as soon as a vehicle finish its route and is removed from the traffic network. An example of a tripinfos.xml file, containing the information about expected starting time, departure and arrival times, time loss, and many others of each vehicle in the scenario, is shown in Figure 5.10. The average time loss is calculated using the following formula:

$$\bar{TL} = \frac{1}{N_{veh}} \sum_{i=1}^{N_{veh}} tl_i \tag{5.1}$$

where $\bar{TL}$ is the average time loss of vehicles traveled through the intersection, N_{veh} is the total number of vehicles passed the intersection during the simulation running time, and tl_i is the time loss of i^{th} vehicle.

B. Average traffic flow is the average number of vehicles that passed through a point in a time unit. To measure traffic flow, a number of induction loop detectors (E1) were installed on incoming and out-coming lanes of the intersection. All these installed detectors E1 are defined in an additional file. For example, in the traffic scenario of Andrea Costa, these detectors are stated in the acosta_detectors.add.xml file within the scenario dataset. A part of this file from Andrea Costa scenario is illustrated in Figure 5.11. This file consists of declarations of detectors such as id, lane where the detector laid

```
<detector xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/det_e1_file.xsd">
  <interval begin="0.00" end="1800.00" id="0.127_2.73_6_1__l0" nVehContrib="30" flow="60.00" occupancy="0.62" speed="13.00" length="4.85" nVehEntered="30"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.73_6_1__l1" nVehContrib="209" flow="418.00" occupancy="20.30" speed="9.43" length="4.81" nVehEntered="209"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.73_4_1__l0" nVehContrib="185" flow="370.00" occupancy="17.14" speed="8.65" length="4.86" nVehEntered="185"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.73_2_1__l0" nVehContrib="120" flow="240.00" occupancy="5.63" speed="10.69" length="4.85" nVehEntered="120"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.73_2_1__l1" nVehContrib="169" flow="338.00" occupancy="15.26" speed="7.79" length="4.84" nVehEntered="169"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.18_4_1__l0" nVehContrib="42" flow="84.00" occupancy="0.85" speed="13.20" length="4.82" nVehEntered="42"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.18_4_1__l1" nVehContrib="119" flow="238.00" occupancy="2.46" speed="13.15" length="4.89" nVehEntered="119"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.18_4_1__l2" nVehContrib="99" flow="198.00" occupancy="2.03" speed="13.10" length="4.83" nVehEntered="99"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.20_8_1__l0" nVehContrib="22" flow="44.00" occupancy="0.46" speed="12.61" length="4.75" nVehEntered="28"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.20_8_1__l1" nVehContrib="347" flow="694.00" occupancy="7.55" speed="12.72" length="4.82" nVehEntered="355"/>
  <interval begin="0.00" end="1800.00" id="2.19_2.20_8_1__l2" nVehContrib="131" flow="262.00" occupancy="3.20" speed="11.94" length="4.81" nVehEntered="131"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.19_6_1__l0" nVehContrib="254" flow="508.00" occupancy="39.52" speed="9.62" length="4.81" nVehEntered="254"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.19_6_1__l1" nVehContrib="94" flow="188.00" occupancy="6.83" speed="10.52" length="4.89" nVehEntered="94"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.19_6_1__l2" nVehContrib="100" flow="200.00" occupancy="13.42" speed="7.87" length="4.86" nVehEntered="107"/>
  <interval begin="0.00" end="1800.00" id="2.19_0.127_2_1__l0" nVehContrib="1" flow="2.00" occupancy="0.05" speed="12.80" length="12.00" nVehEntered="1"/>
  <interval begin="0.00" end="1800.00" id="2.19_0.127_2_1__l1" nVehContrib="292" flow="584.00" occupancy="5.97" speed="13.14" length="4.83" nVehEntered="292"/>
  <interval begin="0.00" end="1800.00" id="2.82_2.19_8_1__l0" nVehContrib="0" flow="0.00" occupancy="0.00" speed="-1.00" length="-1.00" nVehEntered="0"/>
  <interval begin="0.00" end="1800.00" id="2.19_0.127_8_1__l0" nVehContrib="357" flow="714.00" occupancy="7.23" speed="13.20" length="4.81" nVehEntered="357"/>
  <interval begin="0.00" end="1800.00" id="2.20_2.19_4_1__l0" nVehContrib="73" flow="146.00" occupancy="1.48" speed="13.21" length="4.83" nVehEntered="73"/>
  <interval begin="0.00" end="1800.00" id="2.20_2.19_4_1__l1" nVehContrib="121" flow="242.00" occupancy="2.52" speed="12.96" length="4.81" nVehEntered="121"/>
  <interval begin="0.00" end="1800.00" id="0.127_2.20_2_1__l0" nVehContrib="194" flow="388.00" occupancy="21.83" speed="9.33" length="4.81" nVehEntered="194"/>
```


FIGURE 5.12: A part of the e1_output.xml file from Andrea Costa scenario.

on and position of the detector on the lane, the output file. This file is then added into the run.sumo.cfg file, which is the configuration file of the simulated scenario. The file where contains the output information of the detectors is also indicated in the detector additional file. A part of the detector output file is shown in Figure 5.12. It shows the number of vehicles that have completely passed the detector within the interval and the flow which is the number of vehicles extrapolated to an hour.

The process for calculating the average traffic flow of the traffic network is described as follows:

- **Step 1**: Install detectors on incoming and out-coming lanes of the intersections.
- **Step 2**: Define all installed detectors in an additional file and add this file to the configuration file of the simulation scenario.
- **Step 3**: Read the detector output file and extract the value of the traffic flow.
- **Step 4**: Calculate the average traffic flow using following equation:

$$\bar{F} = \frac{1}{N_e} \sum_{i=1}^{N_e} f_{e_i} \tag{5.2}$$

Where $\bar{F}$ is the average traffic flow at the intersection, N_e is the total number of detectors which have been installed on incoming and out-coming lanes of the intersections. f_{e_i} is the number of vehicles passed the detector e_i in a period of time.

5.4 Indicators for Performance Assessment

In multi-objective optimization problems, good quality indicators are very important to the performance evaluation of algorithms. There are a number of performance indexes that can be utilized for measuring the performances of the different algorithms, Helbig and Engelbrecht (2013), Zitzler et al. (2003). In general, there are three aspects that need to be considered when evaluating the performance of algorithms: accuracy aspect, which relates to the convergence of the approximation set or how close the approximation set is to the Pareto optimal front; diversity aspect, which defines the relative distance between solutions in the approximation set; and cardinality aspect, relating to the number of solutions achieved in the approximation set. In this work, the following indexes are used to assess the performance of the algorithms, that consider all the above stated aspects.

5.4.1 Hypervolume

The result of a survey in Riquelme et al. (2015) shows that Hypervolume (HV) is the most used metric. HV considers all three aspects: accuracy, diversity, and cardinality. HV calculates the largeness of the objective space enclosed by an approximation set and a specific reference point. HV can evaluate an approximation set obtained by an algorithm without knowledge of Pareto optimal front, as a result, it is suitable for optimization problems whose Pareto front is unknown. The HV calculation includes a reference point $R(R_1, R_2, ..., R_M)$, that is dominated by all solutions in an approximation set. Therefore, the accuracy of HV depends only on the choice of the reference point. HV is calculated using the following formula:

$$HV(A) = Leb(\bigcup_{x \in A} [f_1(x), R_1] \times ... \times [f_M(x), R_M]) \tag{5.3}$$

where M and A are the number of objectives and the approximation set obtained by an algorithm, respectively. $Leb(S)$ is the Lebesgue measure of a set S.

The algorithms are compared using statistical measurements of the obtained HV. Each algorithm is run 20 independent times and mean of HV on these 20 runs are calculated and this value is used as a main metric to compare the performance of the algorithms. The number of solution evaluations using SUMO in each generation of NS-LS and SA-LS

are different. The execution of the algorithms is terminated when the maximum number of evaluation using SUMO is reached. Therefore, the number of generations in different runs are different. The output of $k^{(th)}$ run includes I pairs (n_k^i, HV_k^i) where I is the number of generations, n_k^i is the number of evaluations using SUMO and HV_k^i is the corresponding HV in $i^{(th)}$ generation. The average number of evaluations is determined by:

$$\bar{n} = \frac{1}{K}\sum_{k=1}^{K} n_k^i \tag{5.4}$$

where K is the total runs. Average HV on 20 runs corresponding to the average number of evaluations n_{ave} is calculated using the the following formula:

$$\bar{HV} = \frac{1}{K} \times (\sum_{k=1}^{K} n_k^i) \times \frac{1}{K} \times (\sum_{k=1}^{K} \frac{HV_k^i}{n_k^i}) \tag{5.5}$$

Furthermore, standard deviation is also used to measure how the the obtained HV spread out from the mean using the following equation:

$$S_{HV} = \sqrt{\frac{\sum_{k=1}^{K}(HV_k - \bar{HV})^2}{K-1}}$$

where S_{HV} is the standard deviation, HV_k and $\bar{HV}$ are hypervolume of $k^{(th)}$ run and mean of HV on K runs, respectively. Best, worst, and median are also computed to provide further comparison of the algorithms. Best and worst are the maximum and minimum hypervolume values obtained in 20 runs, respectively. Arrange all HV obtained in 20 runs in ascending order and median is the average of the two middlemost numbers in this ordered set.

5.4.2 C-metric

The set coverage (C-metric): Suppose that A and B are two approximation sets of the optimization problem. The C-metric is defined as follows:

$$C(A,B) = \frac{|u \in B| \ni v \in A : v \preceq u|}{|B|} \tag{5.6}$$

The C-metric is used to compare the convergence of two approximation sets. $C(A, B)$ refers to the percentage of the solutions in B which are dominated by at least one solution in A. $C(A, B) > C(B, A)$ suggests that set A has better convergence than set B.

5.4.3 Diversity Indicators

Furthermore, to evaluate the diversity performance of the algorithms, two diversity indicators are used. The first diversity performance measure is the spacing metric of Schott (S), which measures how evenly the points of the approximated Pareto front are distributed in the objective space. Spacing is calculated as:

$$S = \sqrt{\frac{1}{N-1}\sum_{m=1}^{N}(d_{avg} - d_m)^2} \tag{5.7}$$

with $d_m = min_{j=1,\ldots,N} \sum_{k=1}^{M} |f_{km}(x) - f_{kj}(x)|$ where N is the number of the solutions in the found Pareto front and M is the number of objective functions. d_{avg} is the average value of all d_m values. The smaller S is, the more evenly distributed the solutions.

However, S does not provide any information with regards to the spread of the solutions. Therefore, maximum spread measurement is utilized as the second diversity indicator.

$$MS = \sqrt{\sum_{k=1}^{N}(max_k - min_k)^2} \tag{5.8}$$

where max_k and min_k are maximum and minimum values of the k^{th} objective, respectively. MS measures the length of the diagonal of the hyperbox that is created by the extreme values of the non-dominated set. The bigger MS is, the more widely spread the solutions.

TABLE 5.1: Experimental parameters settings for NS-LS, SA-LS, and NSGA-II in Experiment 1.

No.	Symbol	Description	Value
1	N	Population size	120
2	$maxEval$	Number of evaluations using the real fitness function	24000
3	p_c	Crossover probability	0.9
4	p_m	Mutation probability	0.1
5	P_{mv}	Mutation probability of a variable in an individual	$1.0/n$
6	n	Number of variables in an individual	30

5.5 Experimental design for evaluating the performance of the algorithms

5.5.1 Experiment 1 - Benchmark functions

Before assessing the algorithms using real traffic scenarios, we generally evaluate their performance using Benchmark test functions. The main purpose of this experiment is to prove that the proposed algorithms work correctly. This experiment is not used to conclude which algorithms are better as the proposed algorithms are designed for traffic signal optimization problems. These optimization problems are extremely time-consuming to assign the fitness value for solutions.

There are several advantages of testing the performance of the algorithms using Benchmark test functions, such as ease of implementation, faster execution and the fact that their optimal fronts are already known. The ZDT test suite is one of the most widely employed Benchmark test functions for multi-objective in the evolutionary algorithm literature, Deb, Thiele, Laumanns and Zitzler (2002). Two test functions to be implemented in the current work are Zitzler–Deb–Thiele's function N.1: (ZDT1) and Zitzler–Deb–Thiele's function N.2 (ZDT2). The parameter settings for the evolutionary process of the algorithms are illustrated in Table 5.1.

Objective functions: this experiment uses ZDT1 and ZDT2 as the optimization functions. The ZDT1 test function is described as follows:

$$minimize \begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x) \times [1 - \sqrt{\frac{f_1(x)}{g(x)}}] \end{cases} \tag{5.9}$$

where

$$g(x) = 1 + 9\frac{(\sum_{i=2}^{n} x_i)}{(n-1)} \tag{5.10}$$

ZDT2 test function is given as:

$$minimize \begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x) \times [1 - (\frac{f_1(x)}{g(x)})^2] \end{cases} \tag{5.11}$$

where

$$g(x) = 1 + 9\frac{(\sum_{i=2}^{n} x_i)}{(n-1)} \tag{5.12}$$

Chromosome representation: individual x is represented as a vector $(x_1, x_2, \ldots, x_n)$ where n is the number of variables in one solution and $n = 30$ is selected in this work as suggested in Huband et al. (2006). Each variable x_i takes a real number in range $[0, 1]$.

Number of runs: each algorithm runs 20 independent times. Solutions found in every generation are recorded to calculate results of the performance indicators.

Performance indicators: HV is utilized to evaluate the performance of the algorithms. Mean of HV in 20 runs over generations achieved by the algorithms in every generation are compared.

TABLE 5.2: Experimental parameters settings for NS-LS, SA-LS, and NSGA-II in Experiments 2 and 3.

No.	Symbol	Description	Value
1	N	Population size	20, 40, 60, and 80
2	$maxEval$	Number of evaluations using SUMO	1200
3	p_c	Crossover probability	0.9
4	p_m	Mutation probability	0.1
5	P_{mv}	Mutation probability of a variable	$1.0/n$
6	n	Number of variables	13 (in Experiment 2) 9 (in Experiment 3)

5.5.2 Experiments using real-time traffic scenarios simulated by SUMO

This section describes two experiments which have been established to evaluate the performance of NS-LS, SA-LS, NSGA-II, and MOEA/D using Andrea Costa and Pasubio scenarios.

5.5.2.1 Experiment 2 - Andrea Costa scenario

Objective functions: the traffic scenario of Andrea Costa is used to evaluate the performance of the algorithms. Objectives optimized in this experiments are minimizing time loss and maximizing the flow of the traffic network. Experimental parameter settings are shown in Table 5.2. The number of variables n indicates the number of phases in a cycle of the traffic light control program. There are 13 and 9 phases in Experiments 2 and 3, respectively, as a result, n takes values of 13 and 9 in these two experiments.

Performance indicators: HV, C-metric, S, and MS are used to compare the algorithms in this experiment. Moreover, other related statistical measurements such as standard deviation, median, min and max values of the HV are also calculated.

Number of runs: it is necessary to perform multiple runs of each algorithm to more accurately evaluate the performance. Therefore, in this experiment, each algorithm independently run 20 times. Solutions found in every generation and their objective values are recorded to calculate the results of the performance indicators.

Number of solution evaluations using SUMO: for each run, 1200 evaluations using the traffic simulator are performed.

Population size: for each algorithm, population sizes of 20, 40, 60, and 80 are used to evaluate the performance of the algorithms in different population sizes.

5.5.2.2 Experiment 3 - Pasubio scenario

It is important that the proposed algorithms are transparent to different traffic networks. This means the proposed algorithms should work effectively in various traffic conditions. Consequently, we establish this experiment to evaluate the transferability of our proposed algorithm by using another traffic scenario to evaluate the algorithms.

The traffic scenario used in this experiment is Pasubio. Other parameters of this experiment are the same with Experiment 2 in the previous section.

5.6 Conclusion

Constructing a traffic scenario is time-consuming as real traffic data and traffic signal plan are not always available. Several traffic scenarios have been introduced and they are available to the public. This chapter introduces two traffic scenarios, Andrea Costa and Pasubio, used to evaluate the performance of SA-LS, NS-LS, NSGA-II, and MOEA/D. A number of assessment indicators are also presented, including HV, C-metric, S, and MS. They measure both convergence and diversity results of the algorithms. Before assessing the performance of the algorithm using traffic scenarios, we evaluate them using Benchmark test functions to have an overview of the general performance of the algorithms. A number of experiments using different test functions and traffic scenarios have been established to evaluate the performance of the algorithms.

Chapter 6

Experimental Results

6.1 Introduction

Anytime behaviour or ability to provide good solutions at any point during execution is preferable in traffic signal optimization. Moreover, in transportation optimization, small populations sizes are necessary for traffic scenarios where processing capacities are limited. NS-LS is a multi-objective optimization algorithm for traffic signal control based on NSGA-II and local search.

Multi-objective evolutionary algorithms (MOEAs) are superior to traditional optimization approaches and they have been widely used in traffic signal optimization problems. To assign fitness value to a solution, traffic simulators are one of the popular tools as they have several advantages such as high accuracy and ability to capture dynamics of time-dependent traffic phenomena. However, the simulation is very time-consuming, especially in large-scale traffic networks. Surrogate-assisted evolutionary algorithms help reducing the running time while give optimised solution. To the best of our knowledge, traffic signal optimization using surrogate models is not common in the literature of current signal control approaches. Only a few studies consider using surrogates with some assumptions which are not suitable and practical in real-time urban traffic management. SA-LS which is an enhancement of NS-LS utilizes a surrogate model to reduce the burden of the computational cost of fitness evaluation. The computational cost of the surrogate model is much lower than that of traffic simulators, therefore, it is partially used to estimate the objective values of solutions.

Before assessing the performance of the proposed algorithms using traffic scenarios, we evaluate their correctness using ZDT1 and ZDT2 test functions. These Benchmark test functions have been widely used in the evolutionary algorithm literature.

Two traffic scenarios, Andrea Costa and Pasubio, have been utilized to demonstrate the performance of NS-LS and SA-LS. These scenarios are good tools to evaluate traffic signal control algorithms as they adequately provide necessary components such as traffic network, traffic demands, and additional infrastructure. Furthermore, these scenarios are available to the public to help researchers evaluate their hypotheses in a reproducible and comparable experimental setup.

To evaluate the performance of the proposed algorithms, several experiments established in Chapter 5 have been implemented and their results are illustrated in this chapter. The performance of NS-LS, SA-LS, NSGA-II, and MOEA/D are compared using both convergence and diversity indicators. Furthermore, for the experiments using traffic scenarios, the distribution of optimal solutions obtained by the algorithms with various population sizes are studied and analysed. Time loss and flow over generations from Experiments 2 and 3 are also studied.

This chapter is organized as follows: Section 6.2 shows the result of Experiment 1 which uses Benchmark test function ZDT1 and ZDT2 as objective functions while Section 6.3 presents the results of experiments using two traffic scenarios Andrea Costa and Pasubio to evaluate the performance of the algorithms. Section 6.4 concludes this chapter.

6.2 Experiment 1: ZDT1 and ZDT2 test functions

The overall performance of NS-LS, SA-LS, NSGA-II, and MOEA/D in experiment 1 are shown and discussed in this section. In this experiment, each algorithm runs 20 times. In each run, HV obtained by each algorithm in every generation is calculated. The number of solution evaluations using ZDT functions in each iteration is also recorded. Average HV on 20 runs of the algorithms are then computed using Equation 5.5 and this metric is used as the main performance indicator to assess the algorithms in this experiment. The objective functions utilized in this experiment are ZDT1 and ZDT2 and the results are presented in the following paragraphs.

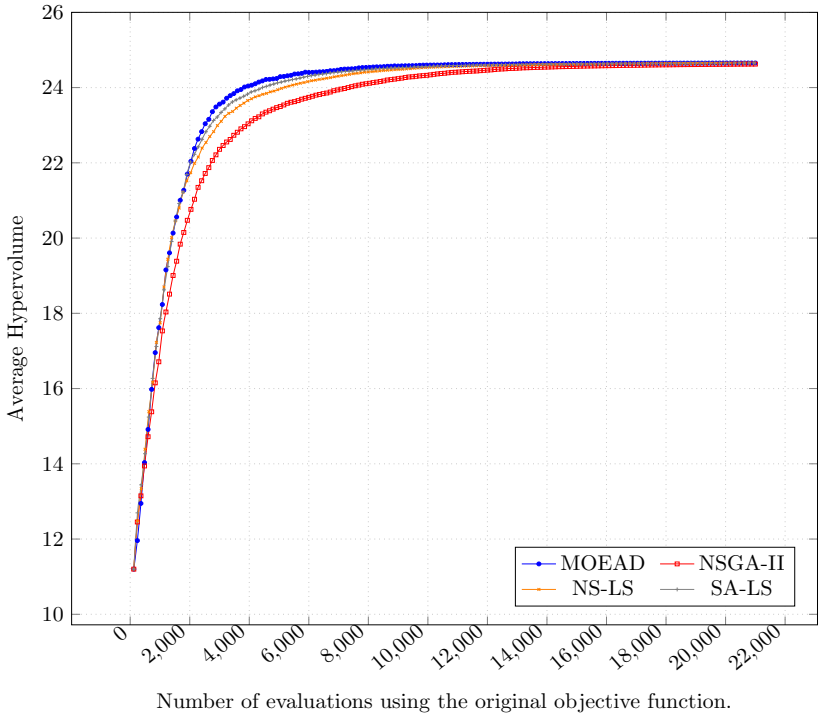


FIGURE 6.1: The mean of HV on 20 runs obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D over the number of evaluations using the original objective function. The objective function is ZDT1.

The means of HV obtained by the four algorithms over the number of evaluations using ZDT1 function are illustrated in Figure 6.1. It is clear from the figure that, during the optimization process, the proposed algorithms NS-LS and SA-LS gain higher HV values compared to NSGA-II. Consequently, using the same number of evaluations evaluated by ZDT1 function, NS-LS and SA-LS achieve better performance than NSGA-II. On the other hand, NS-LS, SA-LS, and MOEA/D gain almost identical performance at the first 2000 evaluations. MOEA/D outperforms NS-LS and SA-LS in between 2000 and 8000 evaluations, afterwards, they obtain similar HV values. SA-LS achieves slightly better performance compared to NS-LS in the middle of the optimization process. Furthermore, NS-LS, SA-LS, and MOEA/D reach the Pareto-optimal front earlier than NSGA-II.

The mean HV value obtained by these algorithms using ZDT2 as the objective function are shown in Figure 6.2. The conclusions we can draw from this figure are similar to the results of the experiment using ZDT1 as the objective function:

1. NS-LS and SA-LA outperform NSGA-II as during the entire optimization process, mean values of HV achieved by NS-LS, SA-LS, and MOEA/D are always higher than that of NSGA-II;

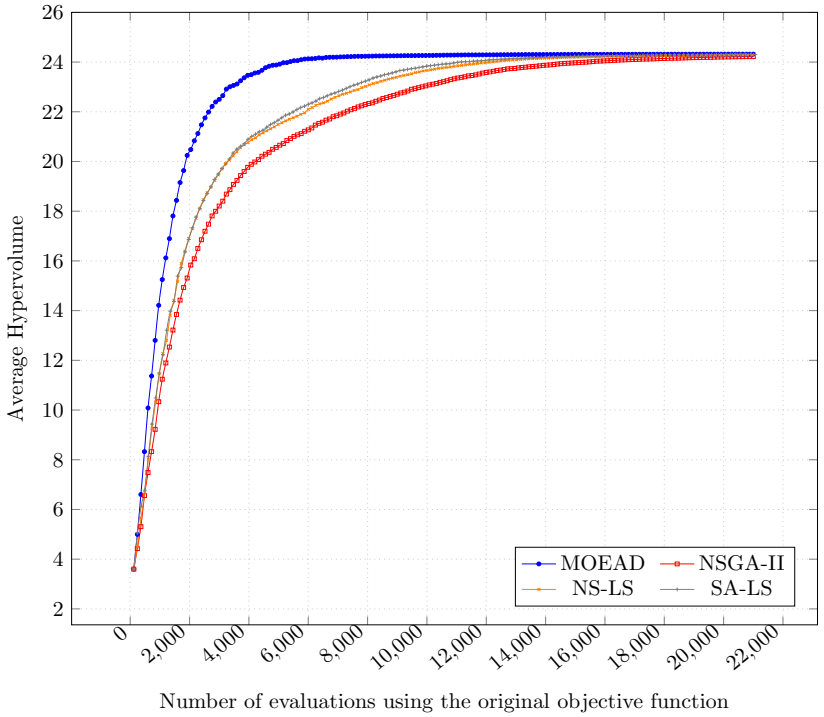


FIGURE 6.2: Mean of HV on 20 runs obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D over the number of evaluations using the original objective function. The objective function is ZDT2.

2. SA-LS finds better solutions than NS-LS in between 4000 and 12000 evaluations, despite mean of HV values acquired by NS-LS and SA-LS using ZDT2 as the objective function are almost identical in the first 4000 evaluations of the optimization process.

3. The convergence of MOEA/D is slightly better than the others in ZDT1 while the optimization result of MOEA/D has considerable stronger convergence than other algorithms in ZDT2. Furthermore, MOEA/D reaches the Pareto-optimal front earlier than the other algorithms;

In conclusion, the results of this experiment show that our proposed algorithms perform in a similar way to NSGA-II and MOEA/D and hence they work correctly. Furthermore, NS-LS and SA-LS always outperform NSGA-II in ZDT1 and ZDT2 test functions. Although MOEA/D achieves higher average HV in the middle of the optimization process with ZDT1 test function and in the entire optimization process with ZDT2 test function, in overall, the results of NS-LS and SA-LS are promising. Experiments 2 and 3 investigate the performance of the proposed models using two traffic scenarios and the results are shown in the following section.

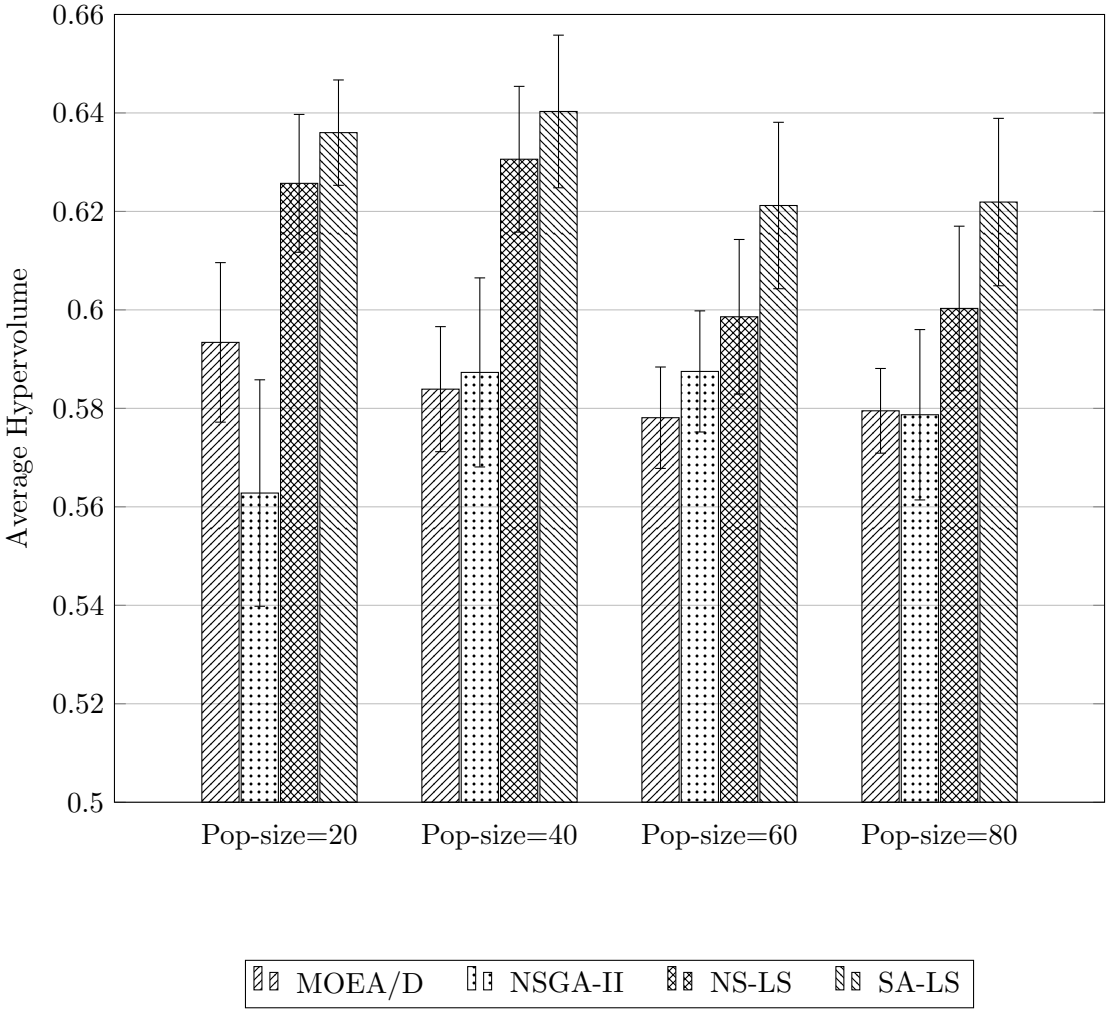


FIGURE 6.3: Average HV with standard deviation on 20 independent runs obtained by MOEA/D, NSGA-II, NS-LS, and SA-LS at the end of the optimization process in Experiment 2.

6.3 Results of experiments using traffic scenarios

6.3.1 Results of Experiment 2 - Andrea Costa

This experiment utilizes Andrea Costa traffic scenario to evaluate the performance of the algorithms. Four population sizes which are 20, 40, 60, and 80 are used in the evolutionary search. Each algorithm runs 20 independent times. Hypervolume is the main performance indicators used to measure the anytime behavior of the algorithms. The optimization results of the algorithms are further compared using diversity indicators such as S and MS.

TABLE 6.1: A solution obtained by SA-LS algorithm in the final generation with the population size 20 in Experiment 2.

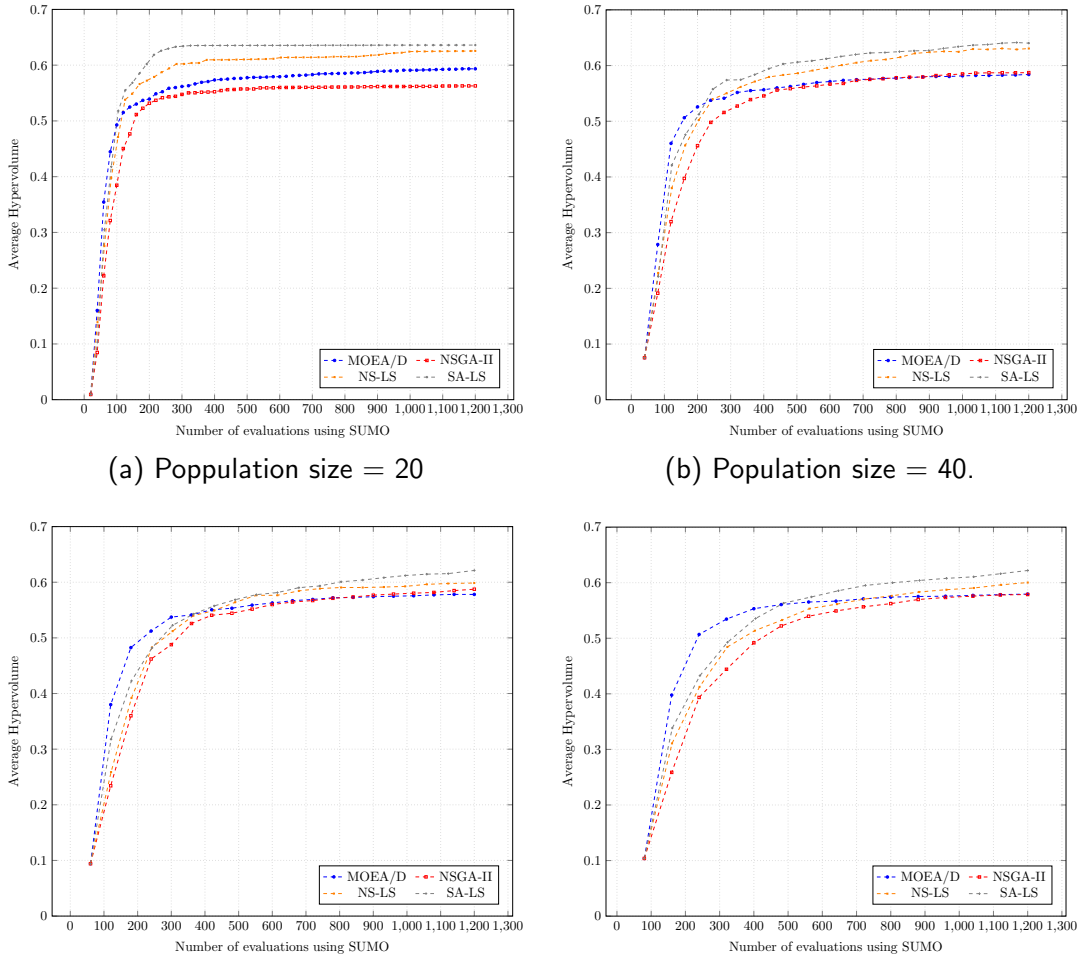
Phase	Signal	Duration value
1	GGgrrrrrGGgrrrGGrrrr	45
2	yyyrrrrrGGgrrrGGrrrr	5
3	rrrGGGGgrrrGGgrrGGGG	26
4	rrrGGGGGrrrrrrrrrrr	14

There are N solutions produced by each optimization algorithm in the final generation with N is the population size. An example of a solution obtained by SA-LS algorithm with the population size 20 in Experiment 2 is introduced in Table 6.1. The first column represents the phase order while traffic signal is shown in the second column. Symbol G indicates an exclusive green light which means that vehicles can pass the junction with priority while character g describes the green light without priority. Symbol r represents red light for a signal and y means amber (yellow) light. The last column provides the duration value of phases in second.

6.3.1.1 Hypervolume Metric

Average HV with standard deviation on 20 independent runs returned by the algorithms at the end of the optimization process runs are illustrated in Figure 6.3. It can be seen from the figure that NS-LS and SA-LS always obtain higher mean values of HV than NSGA-II and MOEA/D in all four different population sizes. Consequently, we can conclude that our proposed algorithms NS-LS and SA-LS work more effectively than NSGA-II and MOEA/D in various population sizes, especially in small population sizes, which is important in traffic management. In the comparison between NS-LS and SA-LS, average HV acquired by SA-LS is always higher than that of NS-LS in different population sizes. Therefore, SA-LS is superior to NS-LS.

Figure 6.4 presents average HV achieved by the four algorithms over the number of evaluations using SUMO in different population sizes. As illustrated in the four sub-figures, using the same number of solution evaluations implemented by SUMO, NS-LS and SA-LS always achieve higher HV values compared to NSGA-II during the entire



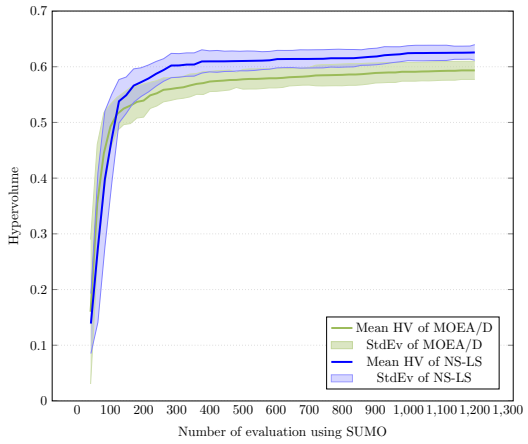
(a) Poppulation size = 20 (b) Population size = 40.

(c) Population size 60. (d) Population size 80.

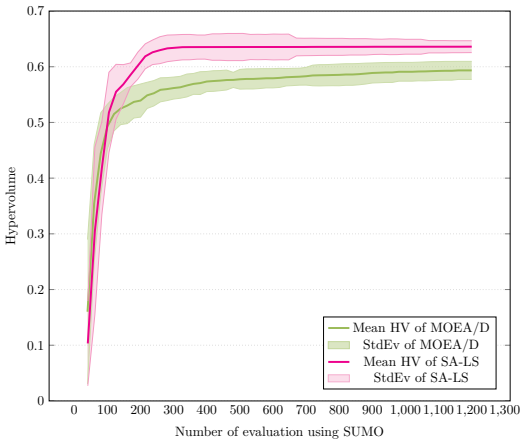
FIGURE 6.4: Mean of HV on 20 runs obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D over the number of evaluations using SUMO in Experiment 2.

optimization process in all four different population sizes. Moreover, average HV obtained by SA-LS is always higher or equal to that of NS-LS over the number of traffic simulator-based evaluations on different population sizes.

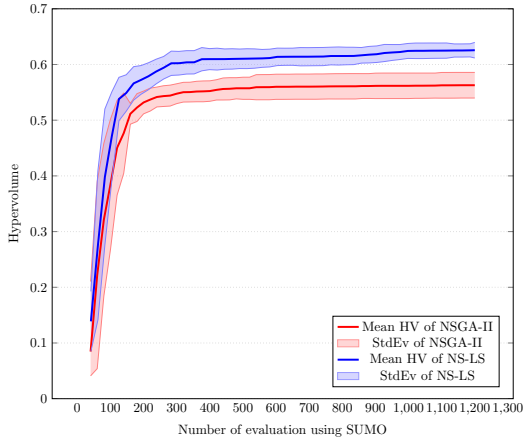
When the population size is 20, average HV obtained by MOEA/D is slightly higher than that achieved by NS-LS and SA-LS in the first 100 evaluations conducted with SUMO. When the population size increases, the number of evaluations that NS-LS and SA-LS used to catch up MOEA/D also increases. When the population size is 40, the optimization result of MOEA/D is better than the proposed algorithms in the first 200 evaluations using SUMO. In case of the population size is 60, NS-LS and SA-LS outperforms MOEA/D in the period from 350 evaluations until the end of the optimization process. When the population size is 80, average HV of MOEA/D is larger



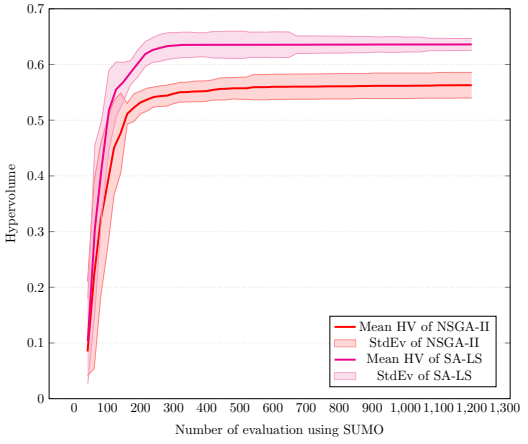
(a) MOEA/D and NS-LS



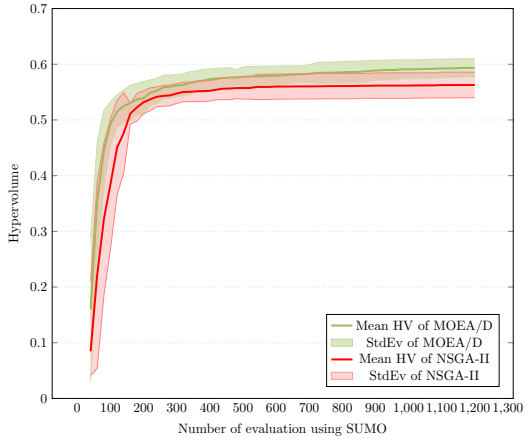
(b) MOEA/D and SA-LS.



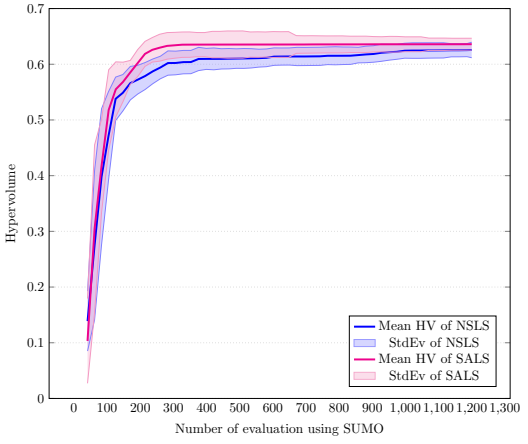
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.



(e) MOEA/D and NSGA-II.



(f) NS-LS and SA-LS.

FIGURE 6.5: Mean HV with standard deviation of MOEA/D, NSGA-II, NS-LS, and SA-LS on 20 different runs in population size 20 in Experiment 2.

than HV obtained by NS-LS and SA-LS in the first 500 and 700 traffic simulator-based evaluations, respectively. Thereafter, average HV obtained by MOEA/D is still lower than that of NS-LS and SA-LS. It can be concluded that although MOEA/D performs better than NS-LS and SA-LS at the beginning of the optimization process, thereafter the proposed algorithms catch up and performs much better.

Figure 6.5 illustrates the comparison between pairs of algorithms in terms of mean with standard deviation over the number of traffic simulator-based evaluations in population size 20. These results of population sizes 40, 60, and 80 are presented in Appendix B. This figure shows how consistent the algorithms are over various runs and generations. The bold line in the middle of the color-filled area is the mean HV obtained by each algorithm. The width of the color-filled area is the width of the standard deviation around the mean HV achieved by different runs. We can see from the sub-figures that standard deviations are relatively uniform between algorithms and different population sizes. Therefore, the variability of the optimization result obtained by the algorithms are relatively small over various runs and generation.

Table 6.2 presents the best, worst, median, mean, and standard deviation HV values obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D by the end of the optimization process through 20 independent runs in experiment 2. These results shows that the convergence rate of NS-LS and SA-LS is better than that of MOEA/D and NSGA-II as the mean values of HV acquired by NS-LS and SA-LS is always higher than MOEA/D and NSGA-II in various population sizes. Best HVs found by NS-LS and SA-LS are also larger than those obtained by MOEA/D and NSGA-II in different population sizes. The worst HVs in population sizes 20, 40, and 80 are produced by NSGA-II while MOEA/D got the worst HV in the case of population size 60. There are is much difference between the standard deviation of the algorithms over various population sizes.

Figure 6.6 illustrates the distribution of solutions in the non-dominated set obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D at the end of the optimization process in various population sizes. They are the best solutions found by an algorithm and can be used to identified the Pareto-optimal front of the optimization problem. For each algorithm, to obtain the non-dominated set, solutions in final generation of all 20 different runs are combined and then sorted using a non-dominated sorting algorithm. Only solutions in the first non-dominated front are then plotted in the figure.

TABLE 6.2: Best, worst, median, mean, and standard deviation of HV obtained by MOEA/D, NSGA-II, NS-LS, and SA-LS in Experiment 2, each over 20 independent runs and for different population sizes.

Algorithms	HV value	Pop size=20	Pop size=40	Pop size=60	Pop size=80
MOEA/D	Best	0.625	0.610	0.597	0.596
	Worst	0.576	0.574	0.560	0.562
	Median	0.588	0.578	0.580	0.580
	Mean	0.593	0.584	0.578	0.580
	Stdev	0.016	0.013	0.010	0.007
NSGA-II	Best	0.599	0.616	0.600	0.609
	Worst	0.534	0.555	0.569	0.552
	Median	0.563	0.586	0.588	0.581
	Mean	0.563	0.587	0.588	0.579
	Stdev	0.023	0.019	0.012	0.017
NS-LS	Best	0.681	0.632	0.644	0.623
	Worst	0.572	0.585	0.565	0.566
	Median	0.618	0.629	0.585	0.597
	Mean	0.626	0.631	0.599	0.600
	Stdev	0.014	0.015	0.016	0.017
SA-LS	Best	0.951	0.641	0.619	0.634
	Worst	0.571	0.593	0.582	0.576
	Median	0.628	0.638	0.593	0.608
	Mean	0.636	0.640	0.621	0.622
	Stdev	0.011	0.016	0.017	0.017

In general, most non-dominated solutions found by the four algorithms are relatively close to each other over four different population sizes. Solutions in the non-dominated set achieved by NS-LS and SA-LS are distributed closer to the optimal front than those of MOEA/D and NSGA-II. As illustrated in the sub-figure 6.6a, solutions in the non-dominated set obtained by NSGA-II are distributed further away from the those in the non-dominated sets obtained by NS-LS, SA-LS, and MOEA/D. Therefore, the optimization results of NSGA-II does not have as good a convergence rate as the other algorithms in the population size 20. Most solutions found by NSGA-II are distributed closer to the optimal front than those of MOEA/D when the population sizes are 40 and 60. Consequently, optimization results of our proposed algorithms are better than NSGA-II and MOEA/D. In sub-figures from 6.6b to 6.6d, non-dominated fronts of the algorithms are placed further way from each other than those in the first sub-figure.

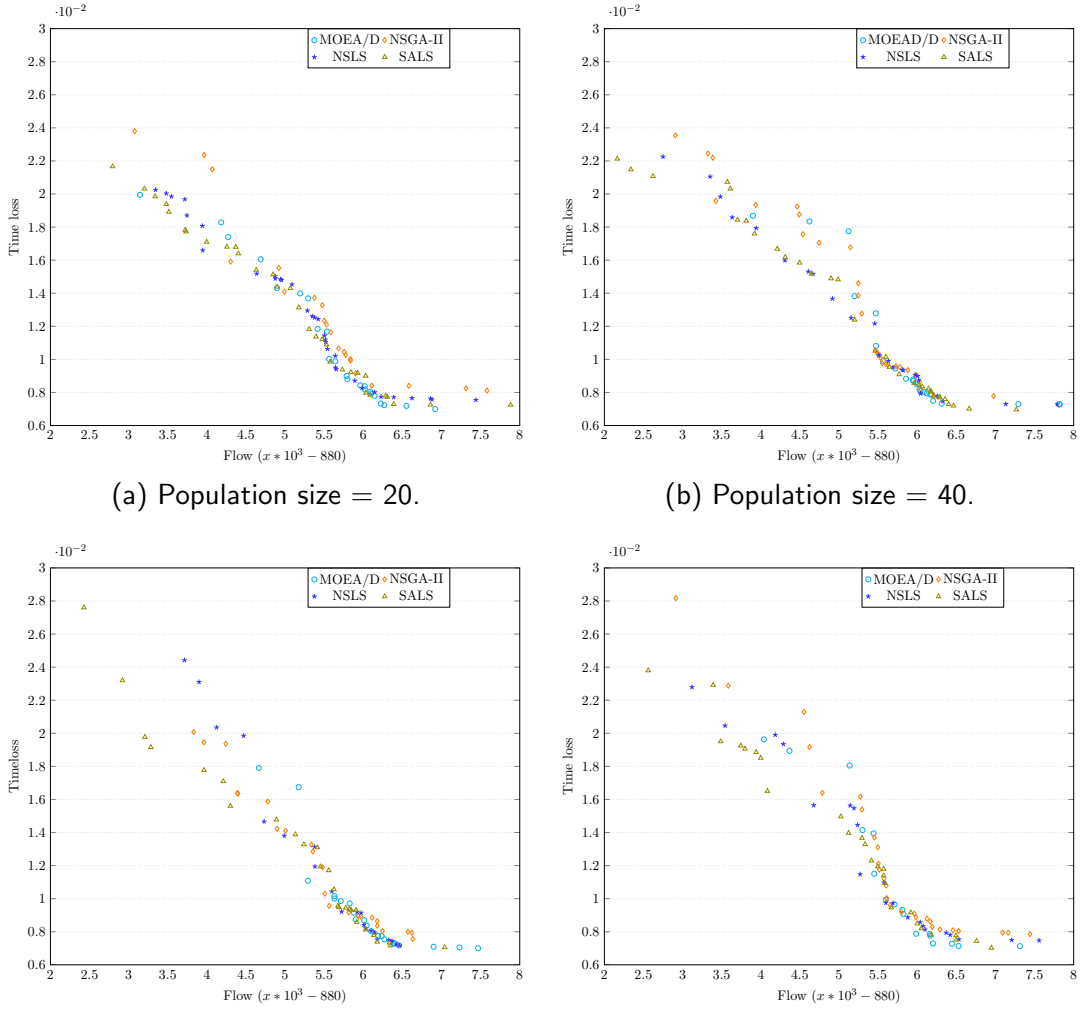


FIGURE 6.6: Distribution of solutions in the non-dominated set achieved by NS-LS, SA-LS, NSGA-II, and MOEA/D at the end of the optimization process in Experiment 2. These solutions are selected from the final solutions of 20 runs.

6.3.1.2 C-metric results

MOEA/D, NSGA-II, NS-LS, and SA-LS are compared in Table 6.3 in terms of C-metrics. $C(A, B)$ refers to the proportion of the solutions in set B which are dominated by at least one solution in A. This metric is used to compare the convergence rate of two algorithms. $C(A, B) < C(B, A)$ indicates that A has better convergence rate than B. C-metric of each algorithm in this study is calculated from its approximated optimal set which consists of the non-dominated solutions obtained from the final generation of all 20 independent runs.

As we can see from this table, $C(MOEA/D, NS - LS) < C(NS - LS, MOEA/D)$ in

TABLE 6.3: C-metric obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D at the end of the optimization process in Experiment 2. For each algorithm, this metric is calculated using the non-dominated set obtained from the final generations of 20 runs.

C-metric	Pop size 20	Pop size 40	Pop size 60	Pop size 80
C(MOEA/D, NS-LS)	0.2609	0.2500	0.1925	0.3462
C(NS-LS, MOEA/D)	0.4242	0.3478	0.3333	0.4118
C(MOEA/D, SA-LS)	0.3125	0.2727	0.1481	0.2727
C(SA-LS, MOEA/D)	0.4348	0.3478	0.5714	0.5294
C(NSGA-II, NS-LS)	0.0303	0.0833	0.3333	0.0385
C(NS-LS, NSGA-II)	0.8095	0.6207	0.3913	0.6429
C(NSGA-II, SA-LS)	0.1250	0.1515	0.2593	0.1364
C(SA-LS, NSGA-II)	0.8571	0.6552	0.6522	0.7857
C(NS-LS, SA-LS)	0.1875	0.2727	0.2963	0.4545
C(SA-LS, NS-LS)	0.6061	0.4167	0.5238	0.4615

all four population sizes. Hence, the percentage of solutions in MOEA/D dominated by at least one solution in NS-LS is larger than the percentage of solutions NS-LS which is dominated by at least one solutions in MOEA/D. Consequently, NS-LS has better convergence rate than MOEA/D. It is also indicated in the table that NS-LS and SA-LS converge faster than NSGA-II as $C(NSGA-II, NS-LS) < C(NS-LS, NSGA-II)$ and $C(NSGA-II, SA-LS) < C(SA-LS, NSGA-II)$. Similarly, solutions in MOEA/D are dominated by solutions in SA-LS in different population sizes. $C(NS-LS, SA-LS) < C(SA-LS, NS-LS)$ in various population size and this implies that SA-LS is better than NS-LS in terms of the convergence rate.

6.3.1.3 Diversity results

MOEA/D, NSGA-II, NS-LS, and SA-LS are further compared in Table 6.4 in terms of Schott (S) and Maximum Spread (MS) metrics. S measures how evenly the solutions in the approximated optimal front are distributed in the objective space. The smaller S is, the more evenly solutions are distributed. MS measures the spread of solutions and the larger MS is, the more widely they are spread. The optimization results will be better if the distribution of approximated optimal solutions can cover the whole period. S and MS results of each algorithm are calculated from the non-dominated set which is achieved from the final solutions of 20 independent runs.

TABLE 6.4: S and MS metrics achieved by NS-LS, SA-LS, NSGA-II, and MOEA/D in Experiment 2. For each algorithm, these diversity metrics are calculated using the non-dominated set obtained from the final generations of 20 runs.

Population size		20	40	60	80
MOEA/D	S	0.0006197	0.0004628	0.0004954	0.0005136
	MS	0.013498	0.012061	0.011263	0.012922
NSGA-II	S	0.0004979	0.0003535	0.0002802	0.0012178
	MS	0.016317	0.016277	0.012811	0.020818
NS-LS	S	0.0002914	0.0004875	0.0005419	0.0005878
	MS	0.013353	0.015803	0.017468	0.015943
SA-LS	S	0.0003636	0.0003970	0.0010946	0.0004980
	MS	0.015291	0.016001	0.021057	0.017339

It is clear from Table 6.4 that, in population size 20, solutions obtained by NS-LS and SA-LS are more evenly distributed than those of MOEA/D and NSGA-II as S values acquired by NS-LS and SA-LS are smaller than S values derived by MOEA/D and NSGA-II. NSGA-II spreads more evenly than the other algorithms in population sizes 40 and 60 while solutions in the approximated front acquired by SA-LS are distributed most uniformly in population size 80.

In term of maximum spread measurement, NSGA-II and SA-LS are the two most widely spread in population sizes 20 and 40, 80. SA-LS is the algorithm which has largest MS value in population size 60, therefore, its non-dominated solutions in the approximated optimal front spread most widely. The worst MS are produced by MOEA/D when the population sizes are 40, 60 and 80 while NS-LS has smallest MS in the case of population size 20.

Summary of results

It is clear from the results of this experiment that the proposed algorithms have the capacity to efficiently optimize traffic signal timings using different population sizes 20, 40, 60, and 80. At the end of the optimization process, NS-LS and SA-LS always have larger mean HV values than MOEA/D and NSGA-II in all the four population sizes. The mean HVs achieved by the proposed algorithms over the traffic simulator-based evaluations are better than those of NSGA-II. Although MOEA/D outperforms NS-LS and SA-LS in terms of mean HV at the beginning of the optimization process, mean HVs of our algorithms are larger than those of MOEA/D in the most period of the

optimization. As a results, we can conclude that the proposed algorithms have better anytime behavior compared to MOEA/D and NSGA-II in various population sizes.

The optimization results of SA-LS is always better than that of NS-LS in four population sizes as mean HV achieved by SA-LS is always higher than that obtained by NS-LS over the number of traffic simulator-base evaluations. Therefore, the surrogate works effectively to reduce the number of simulation-based evaluations in each generation. For that reason, the number of generations in the evolutionary process can be increased. Consequently, using the same number of evaluations performed by SUMO, SA-LS has better anytime behaviour than NS-LS as well as MOEA/D and NSGA-II.

In terms of diversity measurement, solutions in the approximated optimal fronts achieved by NS-LS and SA-LS are distributed more evenly than those of MOEA/D and NSGA-II in population size 20. Solutions of SA-LS spread most widely among the algorithms in population size 80. In other cases, SA-LS and NS-LS are slightly worse than MOEA/D and NSGA-II in term of S metric. NSGA-II and SA-LS are the most widely spread in population sizes 20, 40, and 80. It can be concluded that our algorithms have a good diversity.

To evaluate the transferability of the proposed algorithms, Experiment 3 is conducted to examine the ability of NS-LS and SA-LS to work effectively in various traffic conditions. Another traffic scenario is used in the next experiment and the algorithms are assessed using the same measurement metrics as in this experiment. The results of Experiment 3 are described in the following section.

6.3.2 Results of Experiment 3

In this experiment, NS-LS, SA-LS, NSGA-II, and MOEA/D are evaluated using Pasubio traffic scenario. Similar to experiment 2, each algorithm runs 20 independent times in each population size. Four different population sizes are used, which are 20, 40, 60, and 80. HV is the main indicator as it measures the convergence rate of algorithms as well as it is used to compare anytime behavior between algorithms. Furthermore, the algorithms are also assessed using diversity metrics. The results of the algorithms are presented in the following paragraphs.

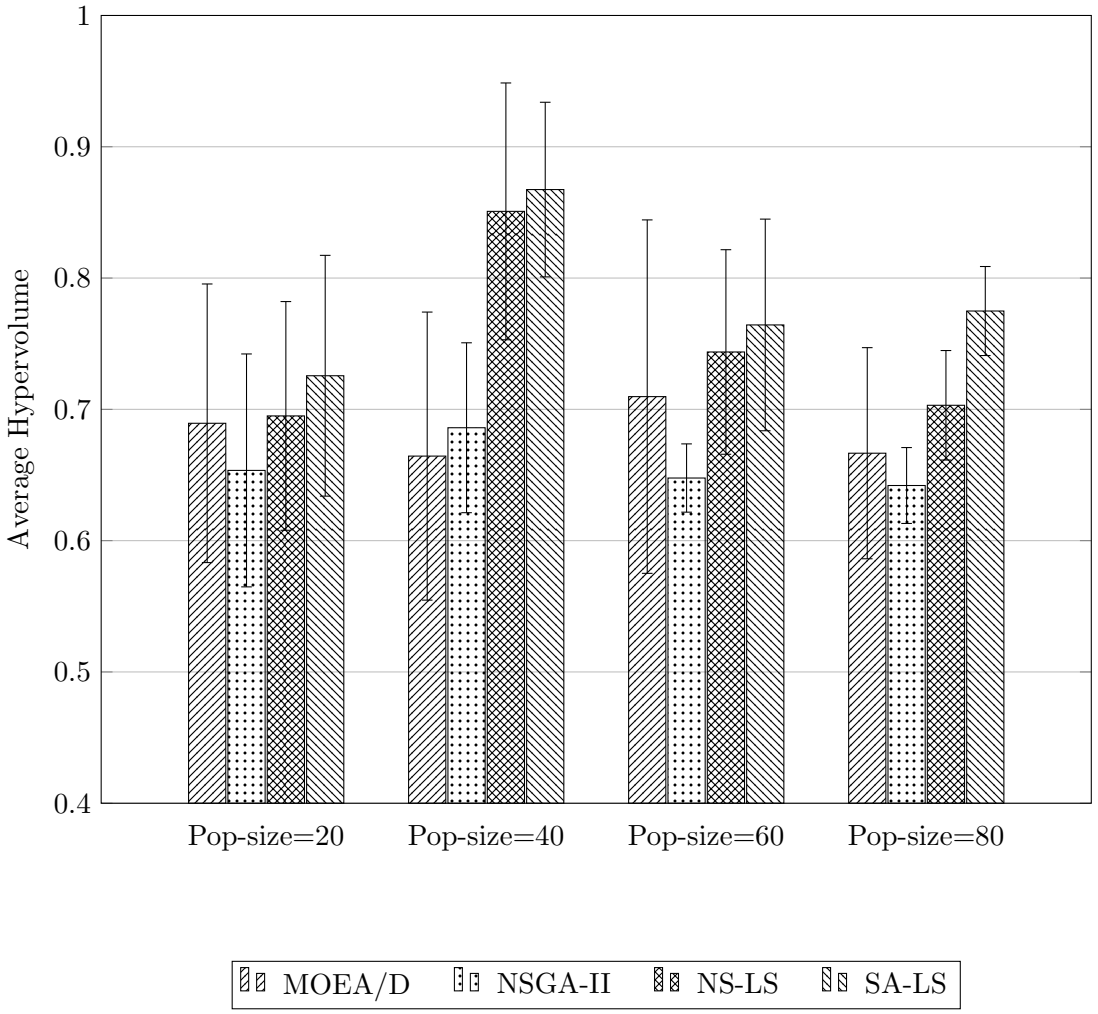
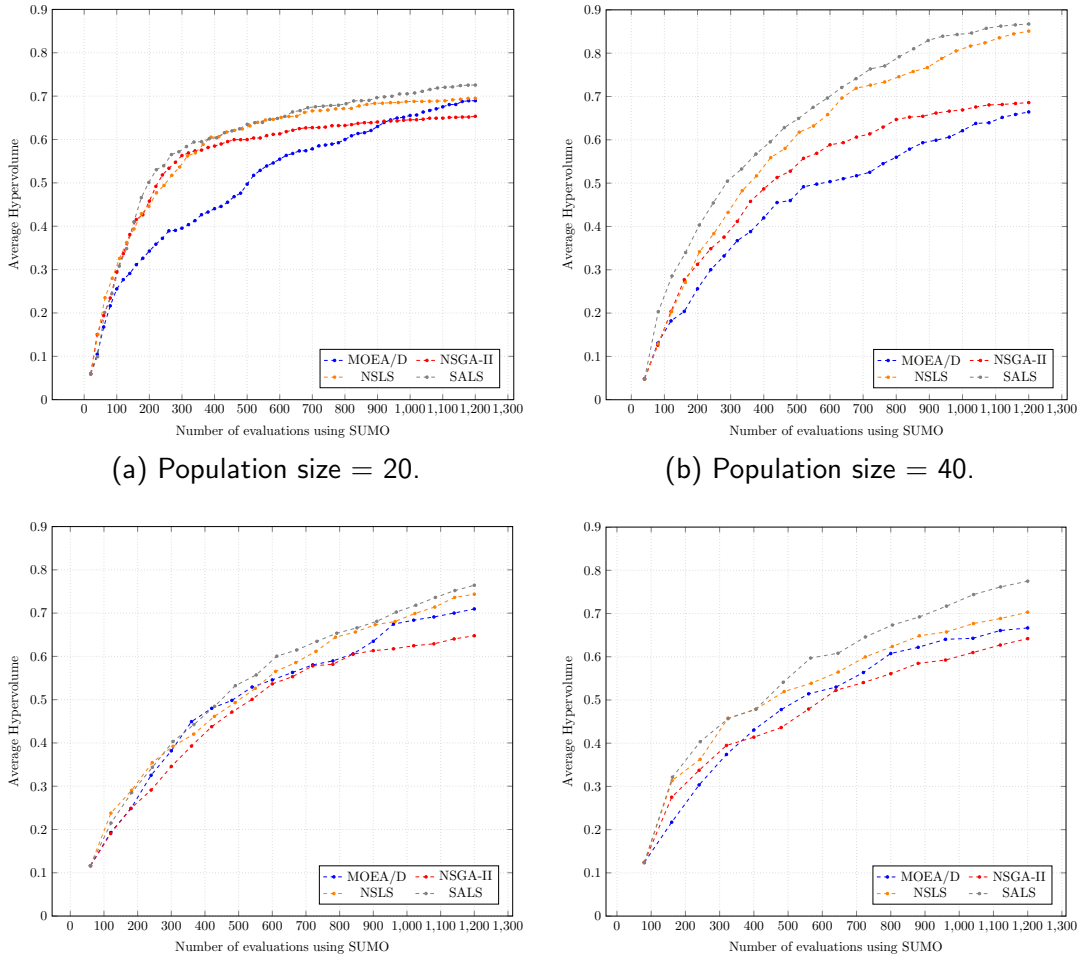


FIGURE 6.7: Average HV with standard deviation on 20 independent runs obtained by MOEA/D, NSGA-II, NS-LS, and SA-LS at the end of the optimization process in Experiment 3.

6.3.2.1 Hypervolume results

The mean with standard deviation obtained at the end of the optimization process on 20 runs of the algorithms are illustrated in Figure 6.7. Overall, NS-LS and SA-LS always have higher mean HV than NSGA-II and MOEA/D in the four population sizes. These results validate that NS-LS and SA-LS have better anytime behaviour compared to NSGA-II and MOEA/D in various population sizes. In comparison between SA-LS and NS-LS, mean HV obtained by SA-LS is always larger than that of NS-LS in all different population sizes. Consequently, SA-LS has better anytime behaviour than NS-LS even in small population sizes. This result confirm the conclusion of the experiment 2.

Performance of the algorithms over the number of traffic simulator-based evaluations



(a) Population size = 20.

(b) Population size = 40.

(c) Population size = 60.

(d) Population size = 80.

FIGURE 6.8: Mean of HV on 20 runs obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D over the number of evaluations using SUMO in Experiment 3.

in different population sizes is illustrated in Figure 6.8. Generally, NS-LS and SA-LS achieved larger mean HVs than NSGA-II and MOEA/D over the entire optimization process in all four different population sizes. Therefore, it can be concluded that NS-LS and SA-LS have better anytime behaviour than NSGA-II and MOEA/D over the number of evaluations evaluated by SUMO.

In Figure 6.8a, although MOEA/D has a significant lower convergence rate than NS-LS and SA-LS in the first 800 evaluations, its mean HV obtained at the end of the optimization process is slightly smaller than that of NS-LS. The optimization results of NSGA-II is slightly better than NS-LS in between 200 and 300 evaluations. When the population sizes are 40 and 80, NS-LS and SA-LS outperform NSGA-II and MOEA/D during the entire optimization process as illustrated in Figures 6.8b and 6.8d. In Figure

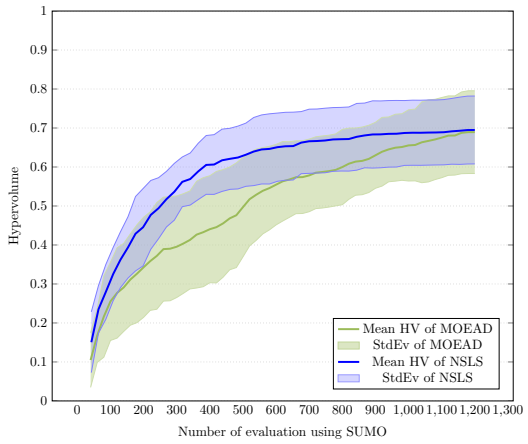
6.8c, MOEA/D gains larger average HV than NS-LS in between 300 and 560 evaluations while mean of HV achieved by SA-LS is slightly smaller than that of MOEA/D in between 350 and 400 evaluations.

When the population size is 20, NS-LS outperforms SA-LS in the first 150 evaluations. Thereafter, SA-LS achieves better average HV than NS-LS. Similarly, NS-LS is better than SA-LS in the first 250 evaluations, then this is converted and SA-LS outperforms NS-LS until the end of the optimization process. In other population sizes, mean HV obtained by SA-LS is always equal or larger than that acquired by NS-LS over number of traffic simulator-based evaluations. Consequently, the combination of the local search with a surrogate works effectively as it helps to enhance anytime behaviour of the optimization algorithm in traffic signal timing optimization.

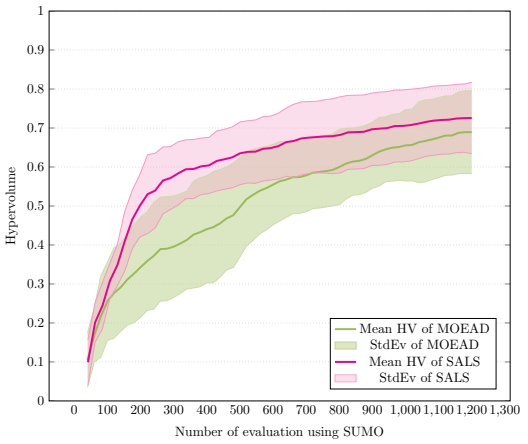
Figure 6.9 represents the mean HV with standard deviation achieved by MOEA/D, NSGA-II, NS-LS, and SA-LS over 20 runs for population size 20. The results of population sizes 40, 60, and 80 are provided in Appendix C. Overall, standard deviations obtained by the algorithms in this experiment are larger than those in the previous experiment. Moreover, standard deviation obtained by the algorithms in this experiment is not uniform. Among the algorithms, SA-LS has the smallest standard deviation, therefore, SA-LS works the most consistently over various runs. Standard deviation gained by NS-LS is slightly smaller than that of NSGA-II. MOEA/D achieved the largest standard deviation compared to the other algorithms. Consequently, the variability of HV over the runs produced by NS-LS is smaller than that of NSGA-II and MOEA/D.

In Figure 6.9b, although mean HV obtained by MOEA/D is always smaller than that achieved by SA-LS, HVs obtained at the end of the optimization process of MOEA/D in some runs are larger than those of SA-LS. Furthermore, standard deviation obtained by MOEA/D is large in the entire process while standard deviations of the other algorithms are small at the beginning and after that, they increase.

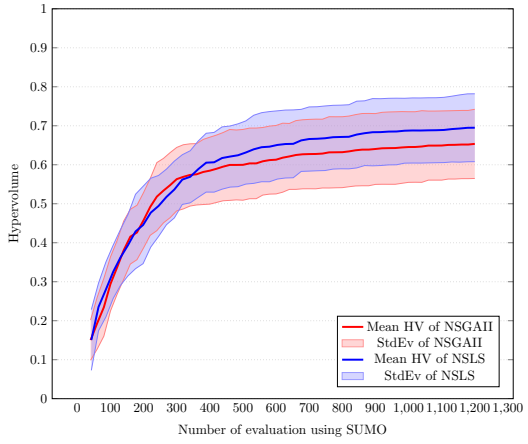
In Experiment 2, HV increase relatively slowly from around 400 traffic simulator-based evaluations until the end of the optimization process. However, in this experiment, the algorithms still converge at the end of the optimization process. Therefore, the algorithms probably need more evaluations using traffic simulator before their optimization results can reach the optimal front. However, this study does not conduct the experiment with a larger number of evaluations using SUMO as this remarkably increases



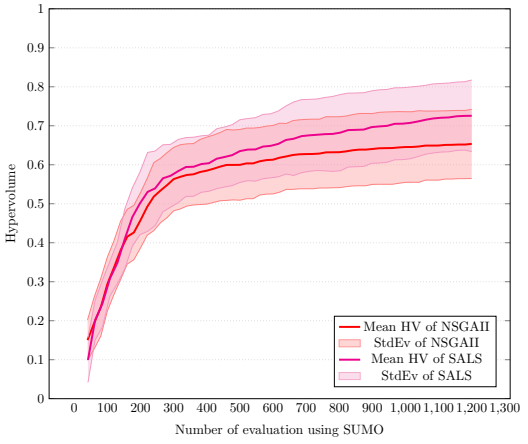
(a) MOEA/D and NS-LS



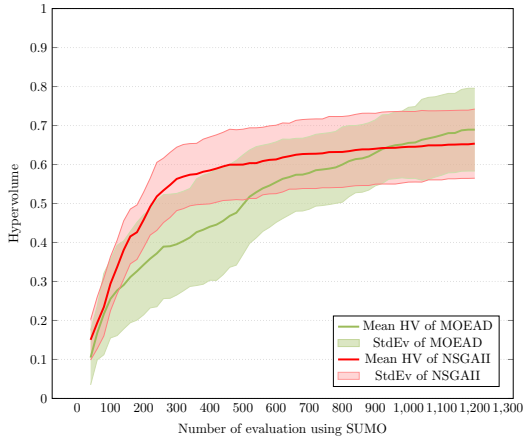
(b) MOEA/D and SA-LS.



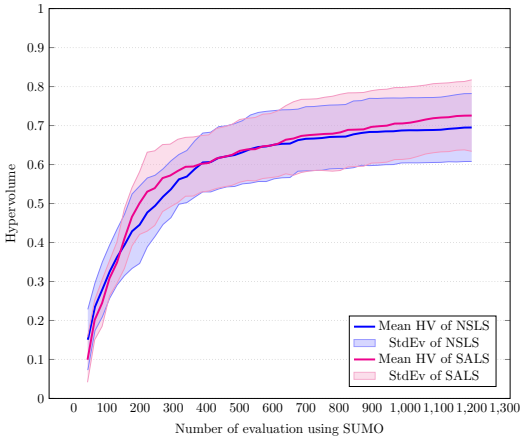
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.



(e) MOEA/D and NSGA-II.



(f) NS-LS and SA-LS.

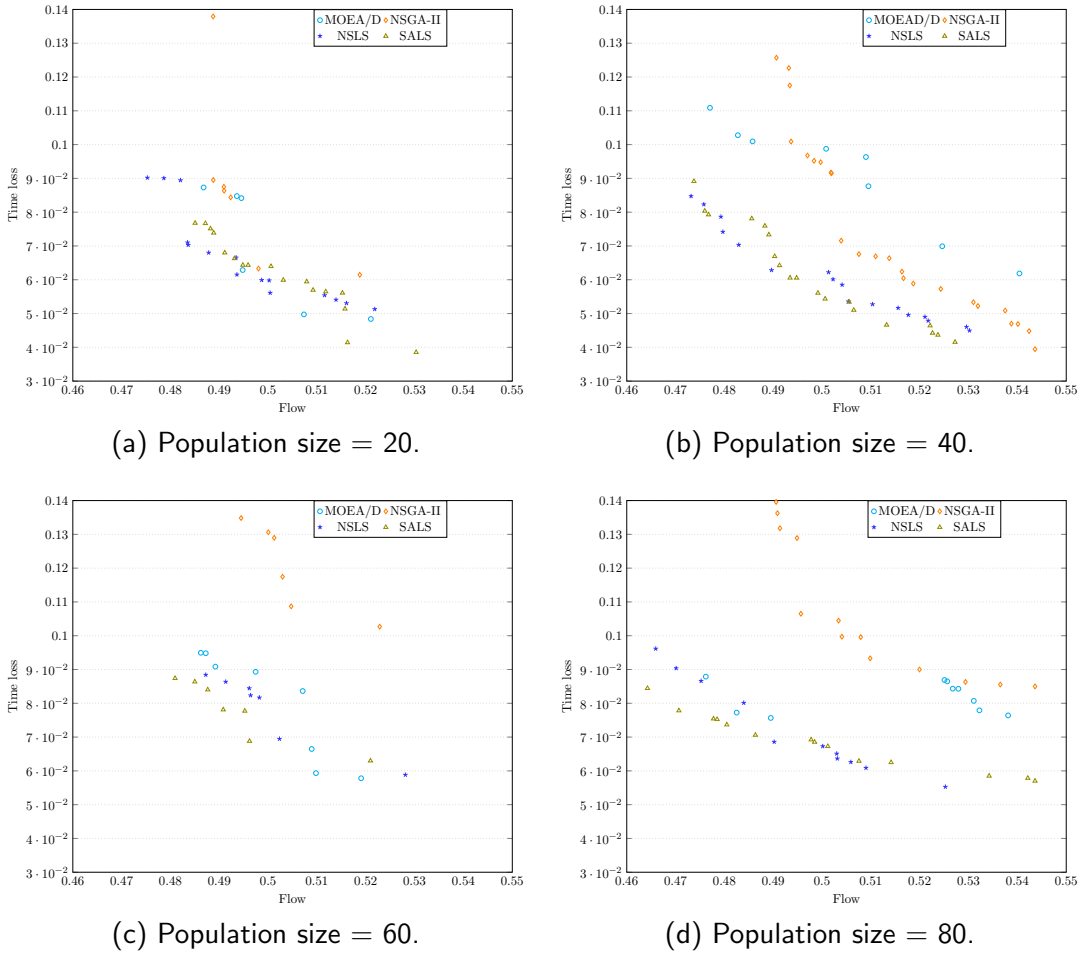
FIGURE 6.9: Mean HV with standard deviation of MOEA/D, NSGA-II, NS-LS, and SA-LS on 20 different runs in population size 20 in Experiment 3.

TABLE 6.5: Best, worst, median, mean, and stdev of HV obtained by NS-LS, SA-LS, and NSGA-II over 20 independent runs in Experiment 3.

Algorithms	HV value	Pop size=20	Pop size=40	Pop size=60	Pop size=80
MOEA/D	Best	0.882	0.899	0.900	0.806
	Worst	0.549	0.464	0.423	0.572
	Median	0.671	0.661	0.730	0.640
	Mean	0.689	0.664	0.710	0.667
	Stdev	0.106	0.110	0.135	0.080
NSGA-II	Best	0.784	0.830	0.695	0.685
	Worst	0.538	0.609	0.620	0.587
	Median	0.632	0.689	0.643	0.642
	Mean	0.654	0.686	0.648	0.642
	Stdev	0.089	0.065	0.026	0.029
NS-LS	Best	0.837	0.961	0.872	0.755
	Worst	0.562	0.697	0.637	0.618
	Median	0.691	0.894	0.710	0.696
	Mean	0.695	0.851	0.744	0.703
	Stdev	0.087	0.098	0.078	0.042
SA-LS	Best	0.751	0.968	0.881	0.795
	Worst	0.628	0.781	0.632	0.714
	Median	0.720	0.863	0.758	0.786
	Mean	0.726	0.867	0.764	0.775
	Stdev	0.092	0.067	0.081	0.034

the running time of the experiment as one simulation takes around 25 seconds using a PC with Intel(R) Core(TM) i5-6500 CPU 3.2GHz. Moreover, the main purpose of our proposed algorithms is to have a good anytime behaviour. The optimization results of the algorithms in this experiment with 1200 traffic simulator-based evaluations already confirm the good anytime behaviour of NS-LS and SA-LS as well as their ability to work in small population sizes which is important in traffic signal timing optimization problems.

Best, worst, median, mean, and standard deviation of the algorithms in all four different population sizes on 20 runs are shown in Table 6.5. In terms of the best HV, MOEA/D gets the highest HV values among the algorithms in population sizes 20, 60, and 80 although mean HVs of MOEA/D are smaller than those of NS-LS and SA-LS. The best HV in population size 40 has been achieved by SA-LS. The worst HV in population size 20 belongs to NSGA-II while in case of population size 40, 60, and 80, MOEA/D is the algorithm which has the worst HV.



(a) Population size = 20.

(b) Population size = 40.

(c) Population size = 60.

(d) Population size = 80.

FIGURE 6.10: Distribution of solutions in the non-dominated set achieved by NS-LS, SA-LS, NSGA-II, and MOEA/D at the end of the optimization process in Experiment 3. These solutions are selected from the final solutions of 20 runs.

MOEA/D has the largest standard deviation among the four algorithms in all four different population sizes. SA-LS has the smallest standard deviation compared to the other algorithms in the populations size 20. It indicates that SA-LS work more consistently than the other algorithms as the smaller standard deviation implies the smaller variability of HV over various runs. The optimization result of NSGA/II has smallest standard deviation among them in population sizes 40, 60, and 80. In the case of population size 20, standard deviation of NSGA-II and NS-LS are similar and they are smaller than that of MOEA/D. Standard devitions of SA-LS in population sizes 20, 40, and 80 are relatively smaller than those of NS-LS while these values are similar in population size 60.

The distribution of non-dominated solutions in the approximated optimal fronts obtained by the algorithms are represented in Figure 6.10. Overall, solutions in the approximated

TABLE 6.6: C-metric obtained by NS-LS, SA-LS, NSGA-II, and MOEA/D at the end of the optimization process in Experiment 3. For each algorithm, this metric is calculated using the non-dominated set obtained from the final generations of 20 runs.

C-metric	Pop size 20	Pop size 40	Pop size 60	Pop size 80
C(MOEA/D, NS-LS)	0.4706	0.1765	0.1429	0.2667
C(NS-LS, MOEA/D)	0.6667	0.5000	0.4000	0.8750
C(MOEA/D, SA-LS)	0.2667	0.0526	0.1429	0.0714
C(SA-LS, MOEA/D)	0.6667	0.7143	0.7000	0.8750
C(NSGA-II, NS-LS)	0.2000	0.0000	0.0000	0.0000
C(NS-LS, NSGA-II)	0.8571	1.0000	0.7778	1.0000
C(NSGA-II, SA-LS)	0.0000	0.0000	0.0000	0.0000
C(SA-LS, NSGA-II)	1.0000	1.0000	1.0000	1.0000
C(NS-LS, SA-LS)	0.2000	0.3158	0.0000	0.3636
C(SA-LS, NS-LS)	0.7500	0.5882	0.8571	0.4286

optimal front of NS-LS and SA-LS in this experiment are closer to the optimal front than those of MOEA/D and NSGA-II. This results confirm that NS-LS and SA-LS have better convergence rates compared to MOEA/D and NSGA-II. In population sizes 60 and 80, non-dominated solutions obtained by NSGA-II are distributed far from those of the other three algorithms. Number of solutions in the approximated front achieved by NSGA-II in population size 20 is smaller than that of the other algorithms. In population size 40, solutions of NSGA-II are placed closer to the approximated front of NS-LS and SA-LS than those of MOEA/D.

6.3.2.2 C-metric results

The algorithms are further compared in Table 6.6 in terms of C metric which is another convergent indicator. Similar to the previous experiment, C-metric is calculated using non-dominated solutions obtained from 20 runs. In terms of comparison between MOEA/D and NS-LS, we can see that NS-LS always has better convergence rate than MOEA/D as $C(MOEA/D, NS-LS) < C(NS-LS, MOEA/D)$ in all the four population sizes. Similarly, the percentage of non-dominated solutions in SA-LS which are dominated by at least one solution in MOEA/D is smaller than

the percentage of solutions in MOEA/D which are dominated by at least one solution in SA-LS. It is also indicated from this table that NS-LS and SA-LS are superior than NSGA-II in terms of convergence rate as $C(NS-LS, NSGA-II) < C(NSGA-II, NS-LS)$ and $C(SA-LS, NSGA-II) < C(NSGA-II, SA-LS)$. Furthermore, $C(NS-LS, SA-LS) < C(SA-LS, NS-LS)$ in all four different population sizes. Consequently, we can say that the optimization result of SA-LS has better convergence rate than that of NS-LS. These conclusions are similar with C-metric results in Experiment 2.

6.3.2.3 Diversity results

Diversity measurement of the algorithms are presented in Table 6.7 with results of S and MS metrics. S measures how evenly the solutions are distributed in the obtained non-dominated set and an algorithm which finds a set of non-dominated solutions having smaller S is better. Solutions in the non-dominated sets obtained by NS-LS, and SA-LS are more evenly distributed in the objective space than those of MOEA/D and NSGA-II in population size 20 as S values of NS-LS and SA-LS are significantly smaller than those of MOEA/D and NSGA-II. S values achieved by NS-LS, SA-LS, and NSGA-II are equivalent in population size 40 and they are a half of S acquired by MOEA/D. Hence, solutions of NS-LS, SA-LS, and NSGA-II are more uniformly spaced than those of MOEA/D. NS-LS achieves better S value than NSGA-II while SA-LS gains the worst S value with population size 60. NS-LS and SA-LS obtain smaller S than MOEA/D in the population size 80. Therefore, solutions in the non-dominated set obtained by NS-LS and SA-LS are more evenly distributed than those of MOEA/D.

In population size 80, NS-LS and SA-LS have the largest MS values, as a result, NS-LS and SA-LS have the largest distance between boundary solutions among the algorithms. Therefore, NS-LS and SA-LS have more diverse sets of solutions than NSGA-II and MOEA/D. NS-LS and SA-LS also have a better set of trade-off solutions among objectives than MOEA/D with the population size 20. Non-dominated solutions obtained by SA-LS spread wider than those in MOEA/D with population size 40 as its MS is higher than that achieved by MOEA/D. Hence, the non-dominated solutions of SA-LS are more diverse than those obtained by MOEA/D.

Summary of results

TABLE 6.7: S and MS metrics achieved by NS-LS, SA-LS, NSGA-II, and MOEA/D in Experiment 3. For each algorithm, these diversity metrics are calculated using the non-dominated set obtained from the final generations of 20 runs.

Population size		20	40	60	80
MOEA/D	S	0.009270	0.004912	0.003212	0.016792
	MS	0.059719	0.069708	0.061932	0.062962
NSGA-II	S	0.017646	0.002325	0.006537	0.002805
	MS	0.082148	0.109862	0.070413	0.076106
NS-LS	S	0.001312	0.002595	0.004035	0.004649
	MS	0.067378	0.069498	0.055814	0.095896
SA-LS	S	0.001903	0.002531	0.009569	0.003899
	MS	0.060661	0.071571	0.046873	0.083977

The results of this experiment confirm that our proposed algorithms can work effectively in various population sizes as their mean HV achieved at the end of the optimization process are always higher than those of NSGA-II and MOEA/D in all four different population sizes 20, 40, 40, and 80. This mean that the local search utilized in NS-LS and SA-LS effectively accelerate the convergence rate of the evolutionary search, and as a result, NS-LS and SA-LS have better anytime behaviour than MOEA/D and NSGA-II.

Mean HV obtained by SA-LS over the traffic simulator-based evaluations also indicates that SA-LS has better anytime behaviour than NS-LS. This means that, in Pasubio traffic scenario, the surrogates also work effectively in the fitness evaluation process to reduce the number of traffic simulator-based evaluations in each generation of the optimization process. As a result, the number of generations may be increased. Therefore, SA-LS outperforms SA-LS in terms of HV and has better anytime behaviour.

The results in this experiment confirms the conclusions in the previous experiment with Andrea Costa scenario and it can be concluded that the proposed algorithms are able to work well in different traffic scenarios.

6.4 Conclusion

NS-LS, SA-LS, MOEA/D, and NSGA-II are evaluated using Benchmark ZDT1 and ZDT2 and two traffic scenarios Andrea Costa and Pasubio. The results of the experiment using ZDT1 and ZDT1 test functions prove that our algorithm work correctly although they do not have obvious advantages in Benchmark functions. However, this

was conducted only to valued the correctness of our algorithms and not to evaluate their efficiency. These algorithms are then evaluated using Andrea Costa and Pasubio.

In both two experiments using Andrea Costa and Pasubio, NS-LS always outperform MOEA/D and NSGA-II in all population sizes. This means that the local search method introduced in chapter 4 works effectively as it can quickly find superior neighbours and accelerate the convergence rate of the evolutionary algorithm. Consequently, the anytime behaviour of NS-LS has been improved.

A hybrid of the local search method and a surrogate based technique proved to enhance the anytime behaviour of NS-LS. The optimization result of SA-LS has better convergence rate than NS-LS, MOEA/D, and NSGA-II. The mean HV obtained by SA-LS over traffic simulation-based evaluations is always higher or equal to that of NS-LS. Therefore, the surrogate is effective in reducing the number of evaluations using the computationally expensive traffic simulator.

Both NS-LS and SA-LS work effectively in various population sizes, especially in small population sizes. The optimization results of NS-LS and SA-LS with the population size 20 in experiments 2 and 3 indicate their efficiency as their mean HV are always larger than NSGA-II and MOEA/D. Non-dominated solutions obtained by NS-LS and SA-LS are also more diverse and sparsely spaced in the Pareto-optimal region than those of MOEA/D in both traffic scenarios with population size 20. Consequently, the proposed algorithms outperform MOEA/D and NSGA-II in terms of convergence and diversity measurements in small population sizes. HV obtained by NS-LS and SA-LS are always larger than those of MOEA/D and NSGA-II when the population size increases to 40, 60, and 80. It means that they have better convergence compared to MOEA/D and NSGA-II over various population sizes. Results of Experiments 2 and 3 indicates the ability of the proposed algorithms to work in different traffic scenarios.

Chapter 7

Conclusions, Recommendations, and Future Work

MOEAs and microscopic traffic simulators have been widely applied in traffic signal optimization problems. However, running simulator-based MOEAs on traffic optimization problems is time-consuming. Therefore, optimization methods with good anytime behaviour indicating the ability to return good solutions at any running time are preferable. In this study, NS-LS and SA-LS are proposed for traffic signal optimization problems which have good anytime behaviour and can work effectively in various population sizes. In NS-LS, a local search is performed in the iteration process of the evolutionary search to accelerate the convergence speed, therefore, improve anytime behaviour of the algorithm. A surrogate model is utilized in SA-LS to reduce the computational cost of evaluating the fitness values of candidate solutions. The combination of the surrogate model and local search improves anytime behaviour of SA-LS. The proposed algorithms can work effectively in various population sizes.

This chapter provides a summary of the research, revisits the propositions and draws conclusions of this work. The chapter is split into five parts: in Section 7.1, the research propositions are revisited. Key findings and contributions of the research are highlighted in Section 7.2 and Section 7.3. Limitations of this study is discussed in Section 7.4. Tasks for a future study are mentioned in Section 7.5.

7.1 Propositions

This research has investigated the benefits of using surrogate-assisted evolutionary algorithms and local search to improve anytime behaviour in traffic signal control optimization problems. As such, the following propositions have been carefully set and are now going to be discussed in detail.

Proposition 1: *A local search method can be used to improve anytime behaviour of multi-objective optimization algorithms in traffic signal optimization problems.*

A local search method was introduced in this research to improve anytime behaviour of the well-known NSGA-II in traffic signal control systems. The population $R^{(t)}$ of the evolutionary search is sorted into a number of non-dominated fronts F_1, F_2, F_3, ..., F_m. Solutions in front F_1 dominates solutions in front F_2. Similarly, solutions in front F_2 are better than solutions in front F_3. A clustering algorithm is utilized to classify the population $R^{(t)}$ into a number of sub-populations. For each sub-population, select solutions belonging to front F_1 and archive them in set SUP_1. Solutions which belongs to both the current sub-population and front F_2 are reserved in set SUP_2. If there are at least two solutions in SUP_1 and one solution in SUP_2, then randomly select solution $R_i^{(t)}$ and $R_u^{(t)}$ from SUP_1. $R_v^{(t)}$ is randomly selected from SUP_2. A neighbour $nb_{R_i^{(t)}}$ of $R_i^{(t)}$ is then created using $R_u^{(t)}$ and $R_v^{(t)}$. This neighbour is assigned a fitness value and added into the population $R^{(t)}$. This procedure is applied for all sub-populations. Afterward, best solutions are selected from the population $R^{(t)}$. In this study, a local search method is applied in the iterative process of NSGA-II, to improve anytime behaviour of NSGA-II in traffic signal timing optimization problems. This novel algorithm has been named NS-LS.

To evaluate the efficiency of the proposed local search method, NS-LS is compared to two well-known optimization methods, NSGA-II and MOEA/D. Two benchmark test functions and two traffic scenarios are utilized to assess the performance of the algorithms. ZDT1 and ZDT2 test functions are fast and have been well studied and their Pareto optimal fronts are already known. Therefore, they are used to evaluate the correctness of the proposed algorithms. Andrea Costa and Pasubio are two traffic scenarios which are available to the public to help students and researchers evaluated their own hypotheses in a reproducible and comparable experimental setup.

Three experiments have been conducted in this study. ZDT1 and ZDT2 are used to evaluate the correctness of the proposed algorithms while Experiment 2 uses Andrea Costa traffic scenario to evaluate the performance of the algorithms. Pasubio scenario is used in Experiment 3 to assess the optimization results of the algorithms. Four population sizes, which are 20, 40, 60, and 80 are used to evaluate the performance of the algorithms over various population sizes, especially in small population sizes. The Hypervolume has been selected as the main indicator for optimization performance. Furthermore, the algorithms are also evaluated using C-metric which is a convergence metric.

The results of Experiment 1 show that NS-LS works correctly and it outperforms NSGA-II in terms of average HV over the number of evaluations using a microscopic traffic simulation. Mean HV obtained by NS-LS is always larger than that of NSGA-II and MOEA/D in all population sizes. The result of C-metric also indicates that NS-LS has better convergence rate than NSGA-II and MOEA/D. Therefore, NS-LS has better anytime behavior than NSGA-II and MOEA/D. Consequently, the local search introduced in NS-LS is effective in improving anytime behavior of multi-objective optimization algorithms in traffic signal optimization problems.

Proposition 2: *A method based on an approximation model can be designed to evaluate candidate solutions in traffic signal optimization problems.*

An approximation model, a surrogate, is constructed using a Feedforward Neural Network(FNN) consisting of one input layer, two hidden layers, and one output layer. The number of nodes in the input layer is equivalent to the number of parameters in the traffic signal timing setting needed to be optimized. The number of nodes in the output layer is equal to the number of objectives in the optimization problems.

Hyperparameters of the surrogate are fine-tuned using the grid search technique. Grid search trains the surrogate for all combinations of hyperparameters and measures the performance using k-fold cross-validation. The combination with the smallest error is chosen to construct the architecture for the surrogate.

The surrogate is trained by the Resilient Back Propagation Algorithm (RPROP). Any solution evaluated by the traffic simulator during the optimization process is added into a database for constructing and updating the surrogate. By using solutions evaluated

by the traffic simulator in previous generations for the purpose of training the surrogate model, it can learn the relationship between signal timings of a traffic signal system and traffic parameters such as flow and delay.

The surrogate is updated during the optimization process to increase the accuracy of the approximation result. At each generation, solutions newly evaluated by the traffic simulator in that iteration are used to estimate the approximation error of the surrogate. If the error is larger than a pre-defined threshold, the surrogate needs to be updated.

After being trained, the surrogate can replace the traffic simulator to estimate the fitness value of candidate solutions. However, this replacement may lead the search to a false optimum. Therefore, an appropriate management model is also proposed to use the surrogate effectively and properly.

We can, therefore, conclude that an approximation model, a surrogate, can indeed be constructed to partially replace the computationally expensive microscopic traffic simulator to evaluate candidate solutions in traffic signal optimization problems.

Proposition 3: *A local search method can be combined with an approximation model to enhance anytime behavior of evolutionary search in traffic signal optimization problems, especially in small population sizes.*

In transportation optimization problems, small population sizes can be important for scenarios where limited processing capabilities meet demand for quick response time. Such scenarios are typical for local and distributed signal controllers, which offer very limited processing power while requiring optimised signal timings within a few cycles or minutes.

A novel surrogate-assisted evolutionary algorithm is introduced for traffic signal optimization problems based on the local search and the surrogate (SA-LS). The surrogate is utilized to estimate the fitness value of some candidate solutions and it is used together with the traffic simulator to evaluate other solutions. Therefore, the number of traffic simulator-based evaluations in a generation of the evolutionary process is reduced. Consequently, using the same number of evaluations conducted by a traffic simulator, the number of generations in the optimization process of the proposed algorithm will be increased. Furthermore, the local search is integrated into the iterations to accelerate the convergence rate of the evolutionary search. As a result, the combination of the local

search and the surrogate enhances the performance of SA-LS in terms of the convergence rate.

The experimental setup to evaluate SA-LS is similar to that of NS-LS. The performance of SA-LS is evaluated using the three experiments which were explained in Chapter 6. In Experiment 1, the correctness of SA-LS is examined using ZDT1 and ZDT2 test functions. The result of this experiment reveals that SA-LS work correctly and it always obtains a larger mean HV than NS-LS and NSGA-II.

SA-LS is evaluated and compared to NS-LS, NSGA-II, and MOEA/D using traffic scenarios. Andrea Costa traffic scenario is used in Experiment 2 to evaluate the performance of the algorithms. It is clear from the results of this experiment that SA-LS always has larger mean HV than NS-LS, NSGA-II, and MOEA/D for all different population sizes. Mean HV obtained by SA-LS over the number of traffic simulator-based evaluations is also better than those of NS-LS, NSGA-II, and MOEA/D. Therefore, SA-LS has better anytime behaviour than NS-LS, NSGA-II, and MOEA/D.

SA-LS is also evaluated in Experiment 3 using Pasubio traffic scenario to examine the ability of SA-LS to work effectively in various traffic conditions. The results of this experiment confirm the conclusion of Experiment 2 that SA-LS has better anytime behaviour than the other algorithms as its mean HV is always higher than those achieved by NS-LS, NSGA-II, and MOEA/D. Furthermore, SA-LS can work effectively in different population sizes, especially in small population sizes. In transportation, optimising with small population sizes is critical in cases, which have limited processing capacity but require a quick response. Therefore, this means that the proposed algorithms are capable of solving traffic signal optimization problems.

The evaluation results in Experiments 2 and 3 indicate that a combination of a local search with an approximation model works effectively in different population sizes. Moreover, combining the local search and the surrogate enhances anytime behaviour of evolutionary search in traffic signal optimization problems.

7.2 Key findings of the research

Here are some key findings of this study:

A surrogate-assisted evolutionary algorithm can be combined with a local search strategy to enhance its anytime behaviour in traffic signal optimization problems.

Approximation models can be utilized to construct surrogate models which can be used to estimate fitness values of candidate solutions in traffic signal optimization using multi-objective evolutionary algorithms. Solutions which are evaluated by a traffic simulator during the optimization process are archived in a database to construct and update the surrogate model. This surrogate is able to learn the relationship between the inputs which are phase durations and the outputs that are traffic parameters needed such as traffic flow and delay. In each generation of the optimization process, new traffic simulator-based solutions are added into the database in order to update the surrogate model in order to improve the approximation accuracy in the oncoming iterations.

A local search can be utilized inside every generation of a surrogate-assisted evolutionary algorithm to improve anytime behaviour of the optimization algorithm in traffic light signal control systems. Some candidate solutions will be estimated by the surrogate model while the fitness value of the other solutions is evaluated by the traffic simulator. A management model of the surrogate needs to be introduced to determine which will be used to estimate the fitness value of a candidate solution, the surrogate model or the traffic simulator. By using the surrogate model in assessing the goodness of solutions, the number of traffic simulation-based evaluations in each generation of the optimization process is reduced. As a result, using the same number of traffic simulation-based evaluations, the surrogate-assisted optimization algorithm has a larger number of generations than the non-surrogate algorithm. Therefore, anytime behaviour of the surrogate-based optimization algorithm will be enhanced. The local search is utilized in every iteration of the optimization process to accelerate the convergence rate. As a result, this combination improves anytime behaviour of the evolutionary algorithm in traffic signal optimization problems.

In traffic signal optimization problems, a local search method can be integrated inside the iteration process of evolutionary algorithms to improve anytime behaviour.

This study proposed a local search method to improve anytime behaviour of a multi-objective optimization algorithm in traffic light control systems. The proposed local

search is integrated into the iteration process of the evolutionary algorithm to speed up the convergence rate. In this local search strategy, a potential direction is selected before starting the searching process by classifying the population into different sub-populations and hierarchical fronts. The reference solutions which are used to create neighbor solutions are selected in the same sub-populations and the first two fronts. The created neighbor is likely to dominate the original solution. Thus, the chance to immediately find out a superior solution from the first search in this proposed local algorithm is increased. Hence, anytime behaviour of the evolutionary algorithm would be enhanced.

Fuzzy distance can be used as an indicator to decide which model should be used to evaluate the goodness of a candidate solution in the optimization process.

It is difficult to obtain a surrogate model with very high accuracy due to the lack of available data. If only the surrogate model is used to estimate the fitness value of candidate solutions, the evolutionary search will likely converge to a false optimum. Therefore, the surrogate model is used together with a traffic simulator in an effective way to predict the goodness of candidate solutions. A critical question is what which solutions should be evaluated by the traffic simulator and which solutions might be estimated by the surrogate model. A fitness evaluation scheme was introduced to solve this question. In this scheme, a candidate solution should be estimated by the surrogate model if this solution is close to the dataset and the surrogate model is reliable. If a solution is close to the dataset, the searching area surrounding that solution is well studied, as a result, the approximation error should be small. Furthermore, if the estimation error of the surrogate model is high, it should not be chosen to evaluate the solution.

The closeness of a candidate solution and the dataset is measured by the minimum fuzzy distance between that solution and the dataset which is defined as the smallest fuzzy distance between the solution and all the samples in the dataset. If the minimum fuzzy distance between a solution and the database is smaller than a threshold and the surrogate model is reliable, that solution will be estimated by the surrogate model. Otherwise, its fitness value will be evaluated by the traffic simulator.

7.3 Key contributions of the research

Major contributions of the thesis are summarized as follows:

1. A local search to look for superior neighbours is introduced. Firstly, the population of the evolutionary search is sorted into several non-dominated fronts. By classifying the population into a number of subpopulations, solutions which are close to each other are allocated into a subpopulation. For each subpopulation, only solutions in first two non-dominated fronts are used to created neighbours. The search direction is determined by the solutions in the equation used to create the neighbour. By selecting the two solutions in two different fronts to participate in the neighbour creation, this local search has the ability to predict potential search directions. Consequently, this local search can accelerate the convergence rate of the evolutionary search.

2. A multi-objective evolutionary algorithm (NS-LS) based on Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and as local search is proposed for improving anytime behaviour in traffic signal timing. The main iterative process of the algorithm utilizes NSGA-II's evolutionary process to move a population of candidates towards the optimal front. The local search is integrated into the iteration process of the evolutionary algorithm to quickly find superior solutions. Therefore this algorithm can produce good solutions at any running time, as a result, it has good anytime behaviour.

3. A evaluation model is constructed to estimate the fitness value of candidate solutions in the optimization process. A feedforward neural network is used to build the surrogate and this model is trained using solutions which are already evaluated by the traffic simulator in the previous generations. This surrogate is able to learn the relationship between the signal timing settings and the traffic parameters needed such as flow and time lost. The surrogate is partially used with the traffic simulator to estimate the fitness value of candidate solutions. Any solutions which are newly evaluated by the traffic simulator during the optimization process are added into the database. The model is updated during the optimization process using previously evaluated solutions to improve the approximation accuracy.

4. A surrogate-assisted multi-objective evolutionary optimization algorithm (SA-LS) for traffic light signal control in urban intersections is introduced. This algorithm is an enhancement of NS-LS and it is based on the surrogate to reduce the number of traffic simulator-based evaluations in a generation of the evolutionary search. This algorithm utilizes the surrogate model to estimate the fitness value of candidate solutions. Both traffic simulator and the surrogate are used together in the fitness evaluation process to prevent the evolutionary search from obtaining false optimum. Furthermore, the local search is also used in the iterations of the evolutionary search to quickly find superior neighbours, as a result, the convergence rate of the algorithm can be increased. The combination of the local search and the surrogate can improve the anytime behaviour of the evolutionary algorithm in traffic signal optimization problems.

5. A fitness evaluation scheme is proposed to maintain a reasonable good quality of the optimization result. This scheme effectively chooses a model between the surrogate and the traffic simulator SUMO to estimate fitness values of solutions. The fitness evaluation scheme is used to guarantee that the surrogate is used effectively. This scheme selects the model to estimate the fitness value of a candidate solution based on the fuzzy distance between that solution and the solutions in the database which is used to construct the surrogate. The scheme also considers the approximation error of the surrogate when choosing the model. If the error is not smaller than a pre-defined threshold, the surrogate is assumed to be unstable and it will not be used in the estimation process until more solutions newly evaluated by the traffic simulator are added into the database and the surrogate is then re-trained.

7.4 Limitations of the Research

The proposed optimization algorithms have not been tested in actual real-world traffic networks. It is very difficult to test and evaluate traffic strategies in a real-world traffic network. Therefore, traffic simulators are a popular tool to evaluate the performance of proposed algorithms. However, there are several drawbacks of using simulation scenarios to evaluate the performance of algorithms such as the simulation environment is not as

dynamic as the real-world traffic network and it is difficult to simulate uncertainties of real traffic networks in traffic simulators.

In this study, the coordination of several neighbouring intersections has not considered and included in the research. For a group of neighbouring intersections, coordination is mainly used to maintain speed and reduce stops and delays along an arterial route. Thus, coordinating adjacent intersections should be involved when optimizing the and we leave this task for a future study.

7.5 Recommendations and Future Work

The coordination of neighbouring intersections would be considered in optimizing traffic signal: main traffic flows of urban traffic networks are often on arterial roads. Performance of traffic signal controls on these arterial roads significantly affect the efficiency of the whole traffic network. Coordination signal plans are developed to decrease stops and delays on arterial roads. Therefore, coordinated traffic signal should be considered when optimizing traffic signals of multiple neighboring intersections to improve the performance of the traffic network.

Multi-modal traffic signal controls would be considered properly: modern urban traffic networks commonly consist of multiple travel modes such as pedestrians, buses, emergency vehicles, and bicycles. All these travel modes should be considered when optimizing traffic signal. Different types of traffic modes have their own characteristics, and therefore, they should be treated differently.

More objectives could be optimized in a future work such as delay of buses, fuel consumption, and number of stops. Especially, air pollution caused by traffic vehicles is a major concern. Air pollution is often more serious at signalized intersections. Therefore, reducing air pollutants should be involved in optimizing traffic signal.

The optimisations could be run for longer to see differences in convergence at near optimas. Especially in Pasubio traffic scenario, the optimization algorithms need more evaluations conducted with SUMO to reach the optimal front.

Appendix A

Published Papers

Here are the papers published during the research period:

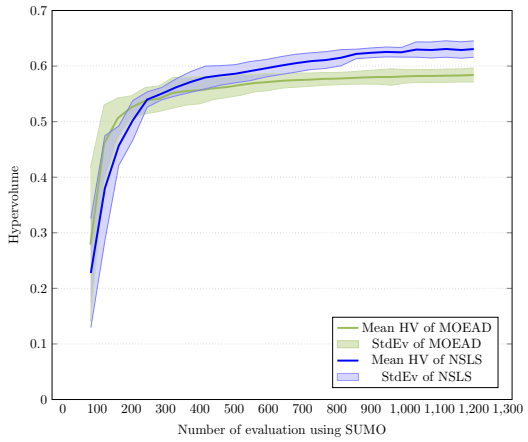
1. Phuong Thi Mai Nguyen ; Benjamin N. Passow ; Yingjie Yang. Improving anytime behavior for traffic signal control optimization based on NSGA-II and local search. 2016 International Joint Conference on Neural Networks (IJCNN).

2. Phuong Thi Mai Nguyen ; Benjamin N. Passow ; Yingjie Yang. An enhancement of non-dominated soritng genetic algorithm iI for multi-objective urban traffic signal control. International Student Workshop 2016, Miedzygorze, Poland.

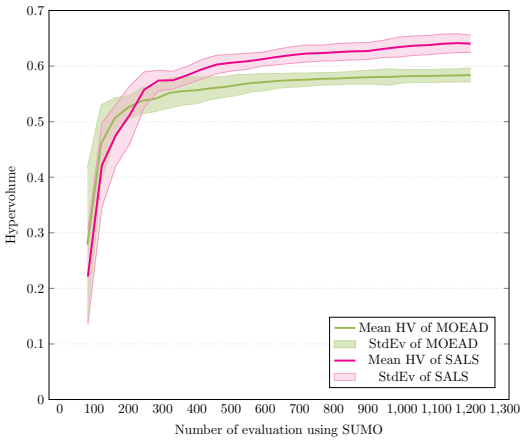
Appendix B

Mean hypervolume with standard deviation of the algorithms in Experiment 2

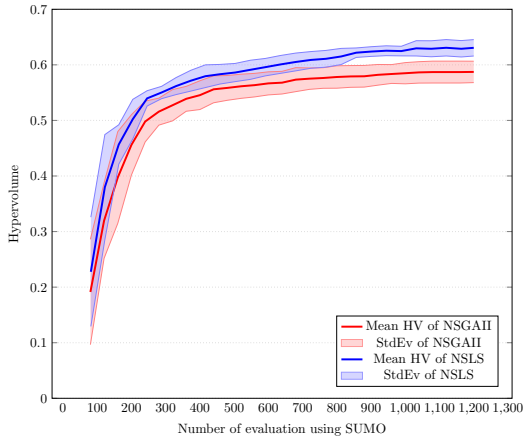
This section provides mean of hypervolume with standard deviation obtained by NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 runs in Experiment 2 using traffic scenario of Andrea Costa with different population sizes. Figures B.1, B.2, and B.3 illustrates the mean and standard deviation of HV achieved by the four algorithms on 20 runs with the population size 40, 60, and 80 respectively.



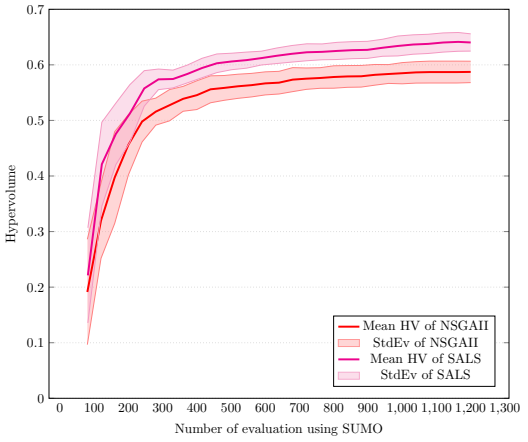
(a) MOEA/D and NS-LS



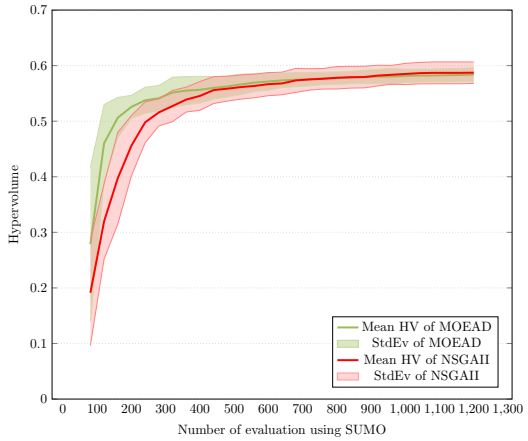
(b) MOEA/D and SA-LS.



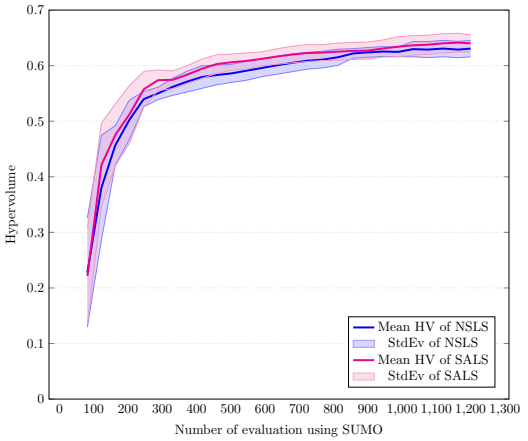
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.

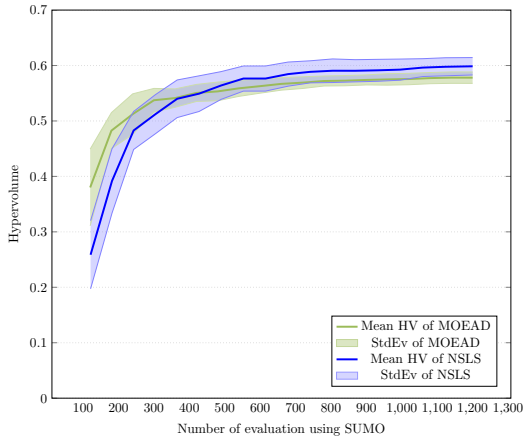


(e) MOEA/D and NSGA-II.

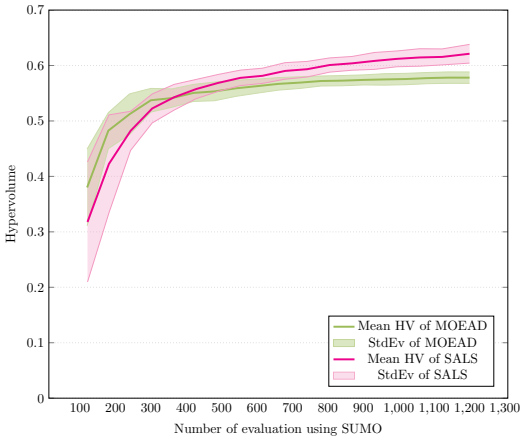


(f) NS-LS and SA-LS.

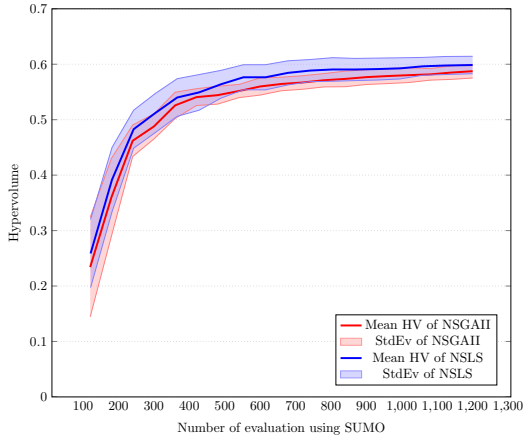
FIGURE B.1: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 40 in Experiment 2.



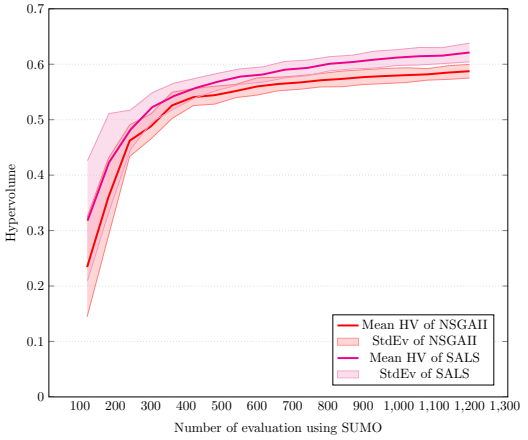
(a) MOEA/D and NS-LS



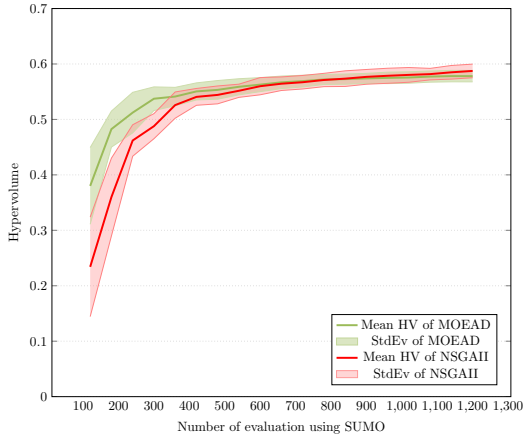
(b) MOEA/D and SA-LS.



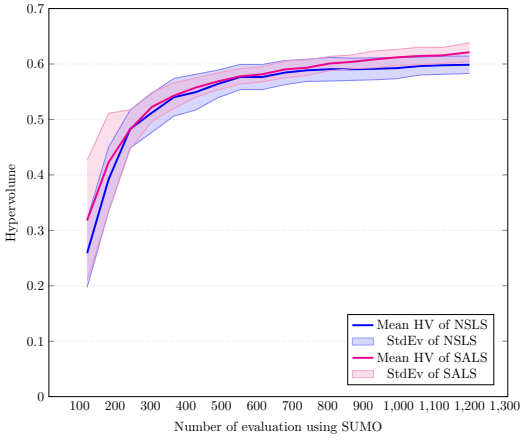
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.

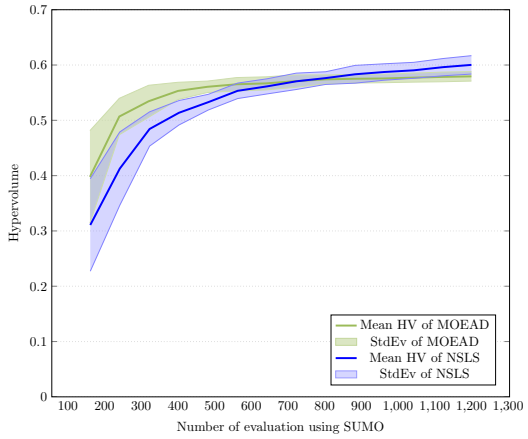


(e) MOEA/D and NSGA-II.

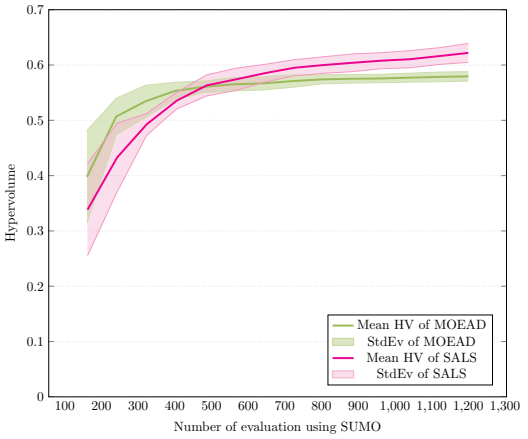


(f) NS-LS and SA-LS.

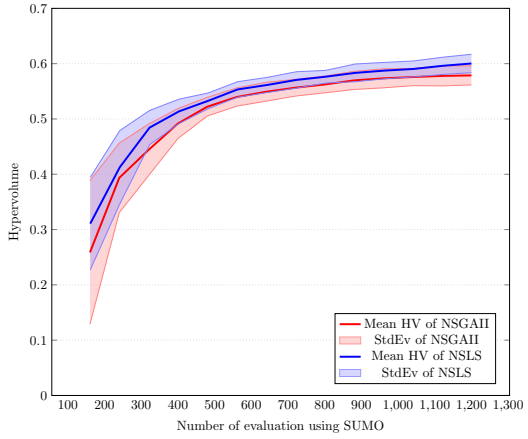
FIGURE B.2: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 60 in Experiment 2.



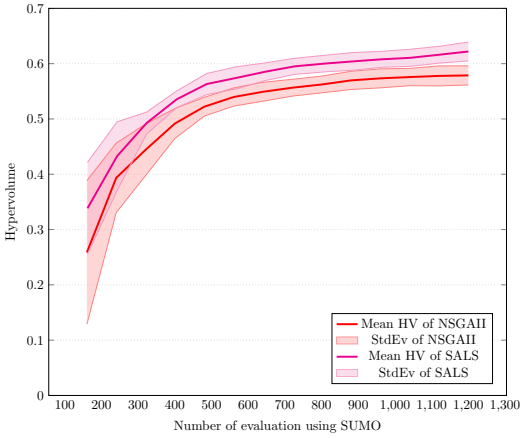
(a) MOEA/D and NS-LS



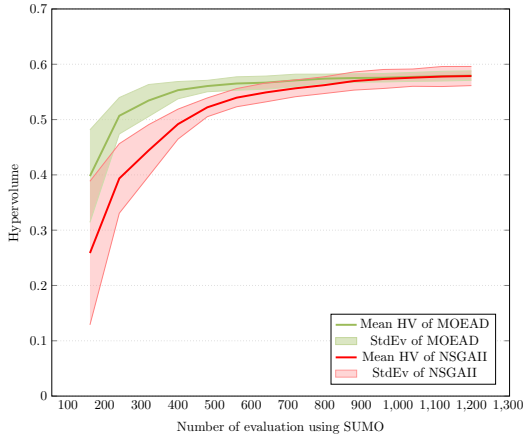
(b) MOEA/D and SA-LS.



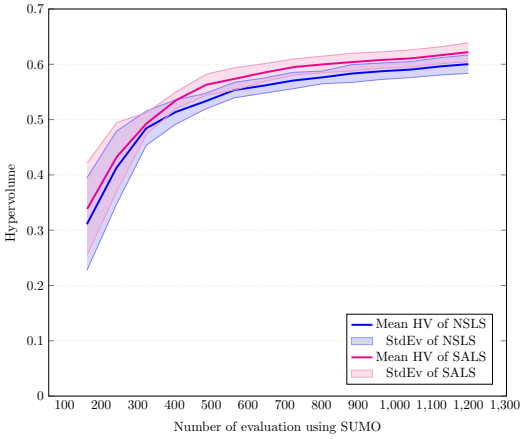
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.



(e) MOEA/D and NSGA-II.



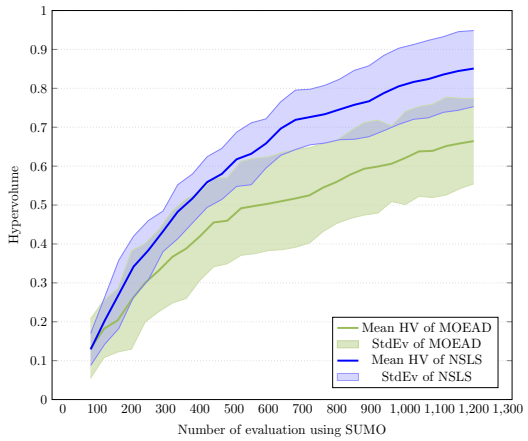
(f) NS-LS and SA-LS.

FIGURE B.3: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 80 in Experiment 2.

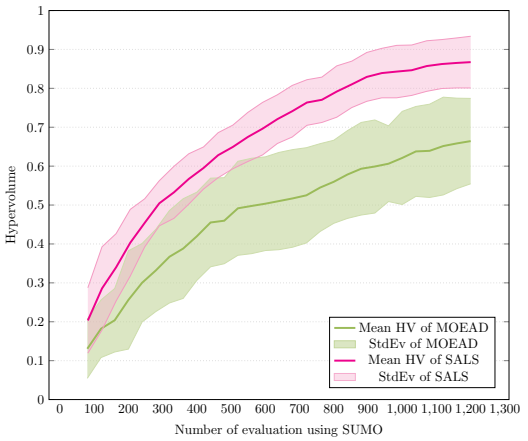
Appendix C

Mean hypervolume with standard deviation of the algorithms in Experiment 3

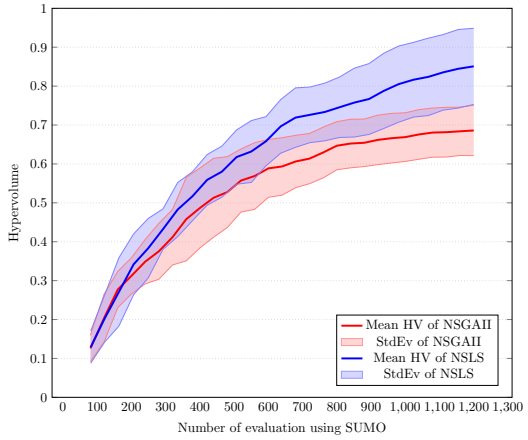
Mean with standard deviation of hypervolume achieved by NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 runs in Experiment 3 using traffic scenario of Pasubio with different population sizes are illustrated in this section. Figure C.1 shows the mean and standard deviation of HV achieved by the four algorithms on 20 runs with the population size 40 while those data of population sizes 60 and 80 are presented in Figures C.2, and C.3.



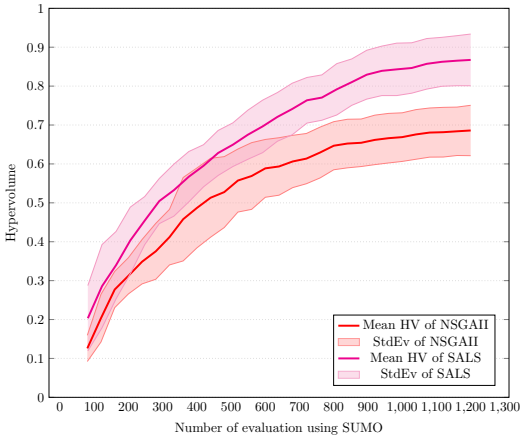
(a) MOEA/D and NS-LS



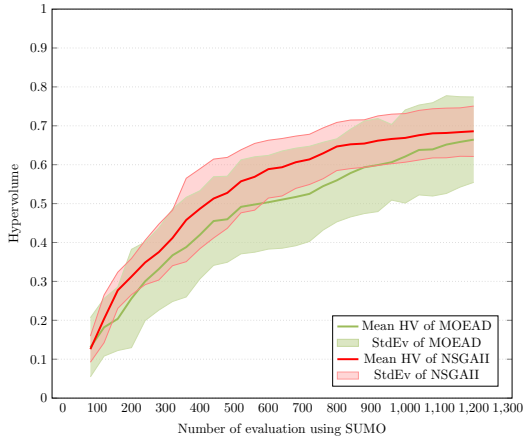
(b) MOEA/D and SA-LS.



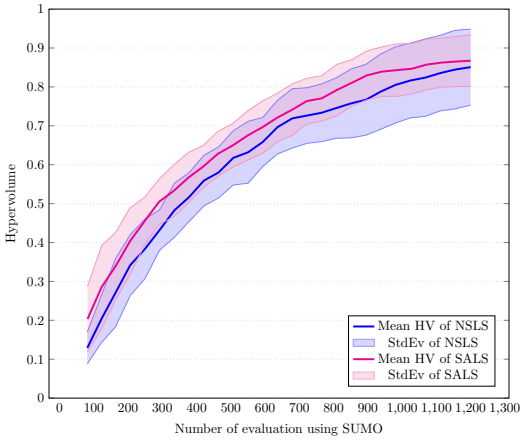
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.

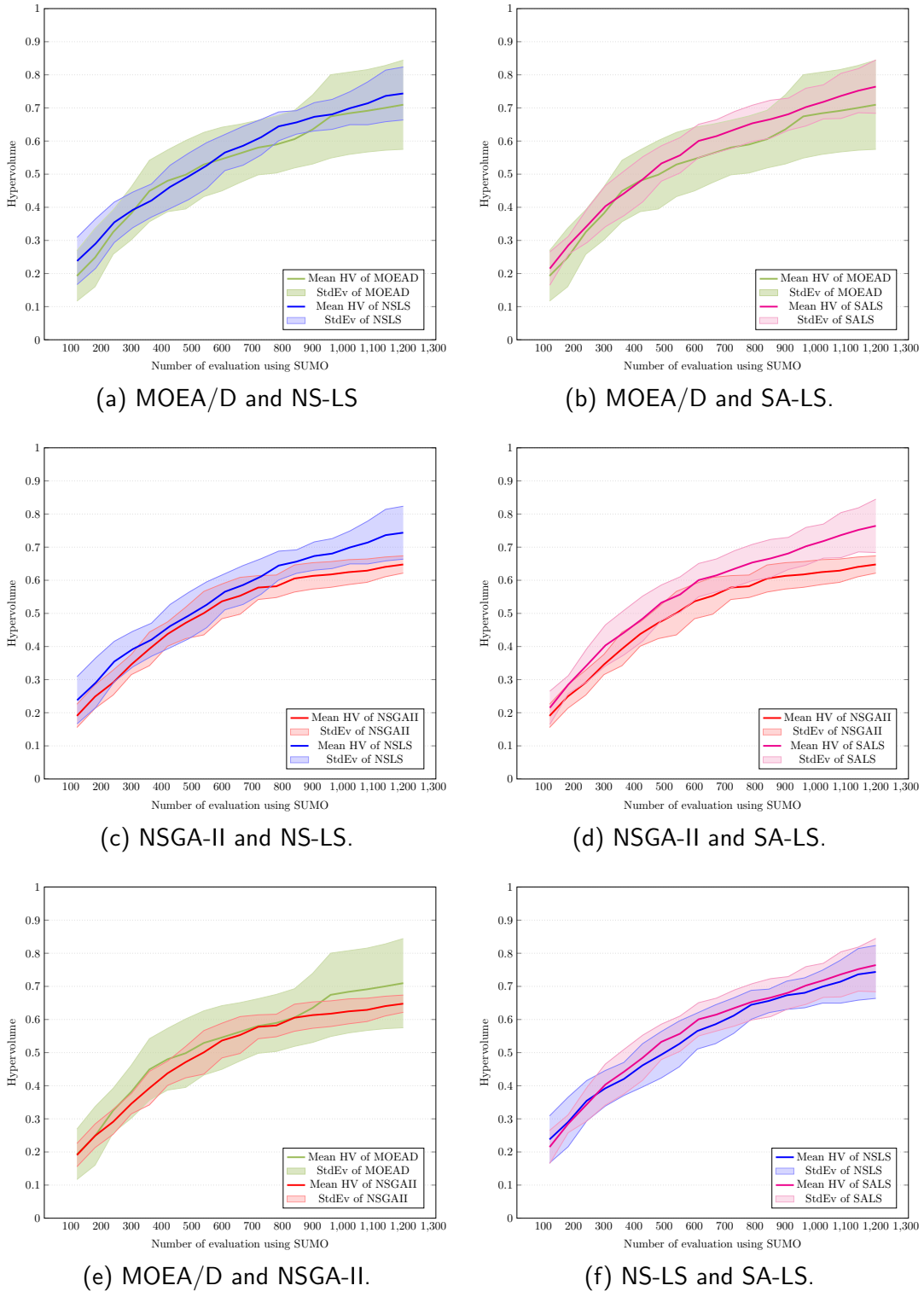


(e) MOEA/D and NSGA-II.



(f) NS-LS and SA-LS.

FIGURE C.1: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 40 in Experiment 3.



(a) MOEA/D and NS-LS

(b) MOEA/D and SA-LS.

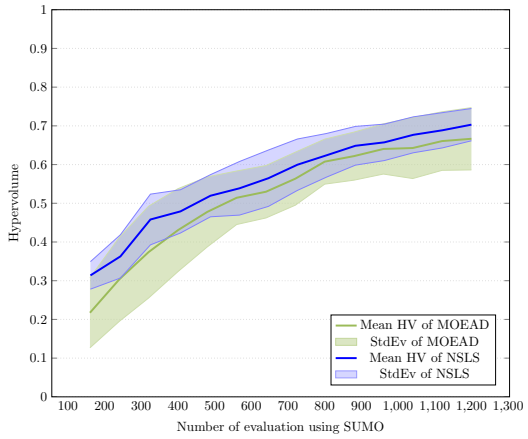
(c) NSGA-II and NS-LS.

(d) NSGA-II and SA-LS.

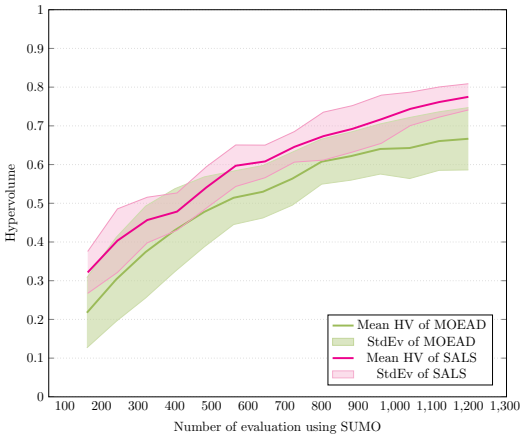
(e) MOEA/D and NSGA-II.

(f) NS-LS and SA-LS.

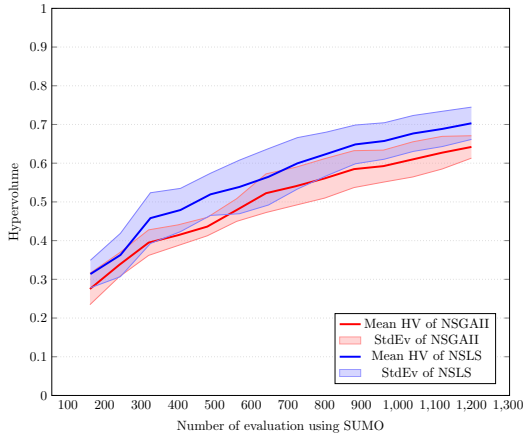
FIGURE C.2: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 60 in Experiment 3.



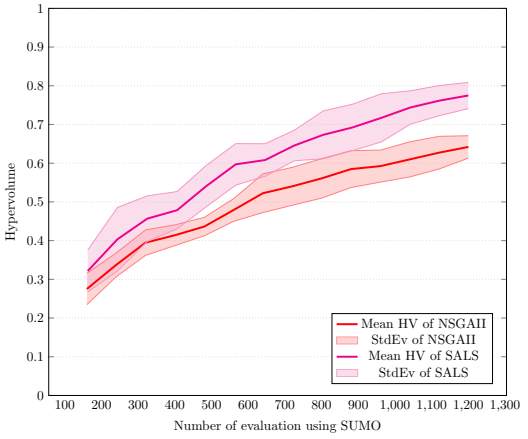
(a) MOEA/D and NS-LS



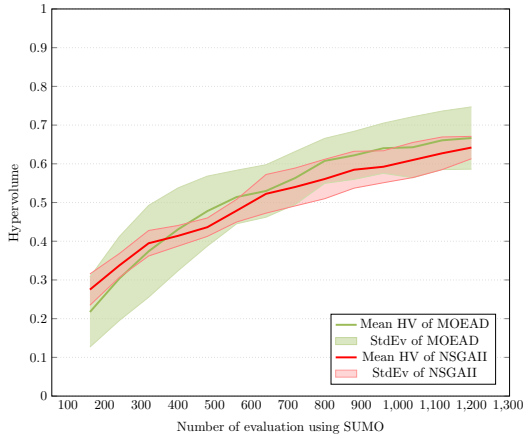
(b) MOEA/D and SA-LS.



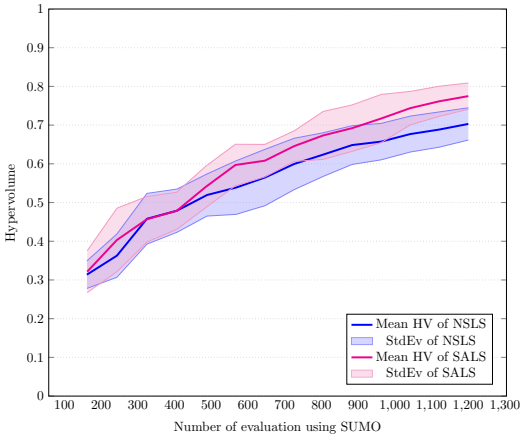
(c) NSGA-II and NS-LS.



(d) NSGA-II and SA-LS.



(e) MOEA/D and NSGA-II.



(f) NS-LS and SA-LS.

FIGURE C.3: Mean HV with standard deviation of NS-LS, SA-LS, MOEA/D, and NSGA-II on 20 different runs with population size 80 in Experiment 3.

Bibliography

Abushehab, R., Abdalhaq, B. and Sartawi, B. (2014), Genetic vs. particle swarm optimization techniques for traffic light signals timing, *in* 'Computer Science and Information Technology (CSIT), 2014 6th International Conference on', pp. 27–35.

Acar, E. and Rais-Rohani, M. (2009), 'Ensemble of metamodels with optimized weight factors', *Structural and Multidisciplinary Optimization* **37**(3), 279–294.
URL: *https://doi.org/10.1007/s00158-008-0230-y*

Adacher, L. (2012), 'A global optimization approach to solve the traffic signal synchronization problem', *Procedia - Social and Behavioral Sciences* **54**, 1270 – 1277. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris.
URL: *http://www.sciencedirect.com/science/article/pii/S1877042812043030*

Araghi, S., Khosravi, A. and Creighton, D. (2015), 'A review on computational intelligence methods for controlling traffic signal timing', *Expert Systems with Applications* **42**(3), 1538 – 1550.
URL: *http://www.sciencedirect.com/science/article/pii/S0957417414005429*

Armas, R., Aguirre, H., Daolio, F. and Tanaka, K. (2016), Traffic signal optimization and coordination using neighborhood mutation, *in* '2016 IEEE Congress on Evolutionary Computation (CEC)', pp. 395–402.

Armas, R., Aguirre, H., Daolio, F. and Tanaka, K. (2017), 'Evolutionary design optimization of traffic signals applied to quito city', *PLOS ONE* **12**(12), 1–37.
URL: *https://doi.org/10.1371/journal.pone.0188757*

B D Venter, M. J. V. and Barcelo (2001), The advantages of micro simulation in traffic modelling with reference to n4 platinum toll road, *in* '20th South African Transport Conference - Meeting the Transport Challenges in Southern Africa'.

Ban, X. J., Hao, P. and Sun, Z. (2011), 'Real time queue length estimation for signalized intersections using travel times from mobile sensors', *Transportation Research Part C: Emerging Technologies* **19**(6), 1133 – 1156.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X11000143*

Barcelo, J. (2010), *Models, Traffic Models, Simulation, and Traffic Simulation*, Springer New York, pp. 1–62.

Basudhar, A., Dribusch, C., Lacaze, S. and Missoum, S. (2012), 'Constrained efficient global optimization with support vector machines', *Structural and Multidisciplinary Optimization* **46**, 201–221.

Behrisch, M., Bieker, L., Erdmann, J. and Krajzewicz, D. (2011), Sumo-simulation of urban mobility, an overview, *in* 'The Third International Conference on Advances in System Simulation.', Vol. 2011, pp. 63–68.

Ben, Z., Lei, S. and Dan, C. (2010), Traffic intersection signal-planning multi-object optimization based on genetic algorithm, *in* 'Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on', pp. 1–4.

Bhattacharjee, K. S., Singh, H. K., Ray, T. and Branke, J. (2016), Multiple surrogate assisted multiobjective optimization using improved pre-selection, *in* '2016 IEEE Congress on Evolutionary Computation (CEC)', pp. 4328–4335.

Bieker, L., Krajzewicz, D., Morra, A., Michelacci, C. and Cartolano, F. (2015), *Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014*, Springer International Publishing, Cham, chapter Traffic Simulation for All: A Real World Traffic Scenario from the City of Bologna, pp. 47–60.

Board, N. R. C. U. T. R. (2000), *Highway Capacity Manual*, Transportation Research Board.

Board, N. R. C. U. T. R. (2010), *HCM 2010: Highway Capacity Manual*, Transportation Research Board.
URL: *https://books.google.co.uk/books?id=oe7hnAAACAAJ*

Board, T. R., National Academies of Sciences, E. and Medicine (2010), *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*, The National Academies Press, Washington, DC.
URL: *https://www.nap.edu/catalog/14364/adaptive-traffic-control-systems-domestic-and-foreign-state-of-practice*

Bourinet, J.-M. (2016), 'Rare-event probability estimation with adaptive support vector regression surrogates', *Reliability Engineering & System Safety* **150**, 210 – 221.
URL: *http://www.sciencedirect.com/science/article/pii/S0951832016000387*

Branke, J. and Schmidt, C. (2005), 'Faster convergence by means of fitness estimation', *Soft Computing - A Fusion of Foundations, Methodologies and Applications* **9**(1), 13–20.
URL: *http://dx.doi.org/10.1007/s00500-003-0329-4*

Caraffini, F., Neri, F., Passow, B. N. and Iacca, G. (2013), 'Re-sampled inheritance search: high performance despite the simplicity', *Soft Computing* **17**(12), 2235–2256.
URL: *http://dx.doi.org/10.1007/s00500-013-1106-7*

Carpenter, W. and Barthelemy, J. F. (1992), A comparison of polynomial approximations and artificial neural nets as response surfaces, Technical report, Technical Report 92-2247, AIAA.

Chen, B., Zeng, W., Lin, Y. and Zhang, D. (2015), 'A new local search-based multiobjective optimization algorithm', *Evolutionary Computation, IEEE Transactions on* **19**(1), 50–73.

Chen, J. and Xu, L. (2006), Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm, *in* 'Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on', pp. 1–7.

Chen, Y., Kim, J. and Mahmassani, H. (2014), Pattern recognition using clustering algorithm for scenario definition in traffic simulation-based decision support systems, *in* 'Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on', pp. 798–803.

Chen, Y.-Y. and Chang, G.-L. (2014), 'A macroscopic signal optimization model for arterials under heavy mixed traffic flows', *Intelligent Transportation Systems, IEEE Transactions on* **15**(2), 805–817.

Cheshmehgaz, H. R., Haron, H. and Sharifi, A. (2015), 'The review of multiple evolutionary searches and multi-objective evolutionary algorithms', *Artificial Intelligence Review* **43**(3), 311–343.
URL: *https://doi.org/10.1007/s10462-012-9378-3*

Chin, Y., Yong, K., Bolong, N., Yang, S. and Teo, K. (2011), Multiple intersections traffic signal timing optimization with genetic algorithm, *in* 'Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on', pp. 454–459.

Chow, H. K. L. H. F. (2010), 'Adaptive traffic control system: Control strategy, prediction, resolution, and accuracy', *Journal of Advanced Transportation* .

Council, L. C. (2019).
URL: *https://www.leicester.gov.uk/transport-and-streets/roads-and-pavements/area-traffic-control/*

Deb, K. (2008), *Multi-objective optimization using evolutionary algorithms*, Wiley.

Deb, K. and Agrawal, R. (1995), 'Simulated binary crossover for continuous search space', *Complex System 9(2), 115-148* .

Deb, K. and Goyal, M. (1996), 'A combined genetic adaptive search (geneas) for engineering design', *Computer Science and Informatics* **26**, 30–45.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), 'A fast and elitist multiobjective genetic algorithm: Nsga-ii', *Evolutionary Computation, IEEE Transactions on* **6**(2), 182–197.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2002), Scalable multi-objective optimization test problems, *in* 'Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)', Vol. 1, pp. 825–830 vol.1.

Diaz-Manriquez, A., Toscano, G., Barron-Zambrano, J. H. and Tello-Leal, E. (2016), 'A review of surrogate assisted multiobjective evolutionary algorithms', *Computational Intelligence and Neuroscience* **2016**.

Djalalov, M. (2013), The role of intelligent transportation systems in developing countries and importance of standardization, *in* 'ITU Kaleidoscope: Building Sustainable Communities (K-2013), 2013 Proceedings of', pp. 1–7.

Dong, C., Huang, S. and Liu, X. (2010), Urban area traffic signal timing optimization based on sa-pso, *in* 'Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on', Vol. 3, pp. 80–84.

DOrey, P. and Ferreira, M. (2014), 'Its for sustainable mobility: A survey on applications and impact assessment tools', *Intelligent Transportation Systems, IEEE Transactions on* **15**(2), 477–493.

Dubois-Lacoste, J., López-Ibáñez, M. and Stützle, T. (2015), 'Anytime pareto local search', *European Journal of Operational Research* **243**(2), 369 – 385.

Ducheyne, E., De Baets, B. and De Wulf, R. (2003), Is fitness inheritance useful for real-world applications?, *in* C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele and K. Deb, eds, 'Evolutionary Multi-Criterion Optimization', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 31–42.

Emmerich, M., Giotis, A., Uezdenir, M., Baeck, T. and Giannakoglou, K. (2002), Metamodel-assisted evolution strategies, *in* 'Parallel Problem solving from Nature, LNCS, Springer'.

Espinoza, F. P., Minsker, B. S. and Goldberg, D. E. (2003), Performance evaluation and population reduction for a self adaptive hybrid genetic algorithm (sahga), *in* 'GECCO'.

Fang, F. and Elefteriadou, L. (2008), 'Capability-enhanced microscopic simulation with real-time traffic signal control', *Intelligent Transportation Systems, IEEE Transactions on* **9**(4), 625–632.

Feng, S. and Xiaoguang, Y. (2008), Optimization algorithm of urban road traffic signal plan based on nsgaii, *in* 'Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on', Vol. 2, pp. 398–401.

Fonseca, L. G., Lemonge, A. C. C. and Barbosa, H. J. C. (2012), A study on fitness inheritance for enhanced efficiency in real-coded genetic algorithms, *in* '2012 IEEE Congress on Evolutionary Computation', pp. 1–8.

Fushiki, T. (2011), 'Estimation of prediction error by using k-fold cross-validation', *Statistics and Computing* **21**(2), 137–146.
URL: *https://doi.org/10.1007/s11222-009-9153-8*

Gao, K., Zhang, Y., Sadollah, A. and Su, R. (2016), 'Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search', *Applied Soft Computing* **48**, 359 – 372.
URL: *http://www.sciencedirect.com/science/article/pii/S1568494616303556*

Gao, K., Zhang, Y., Zhang, Y. and Su, R. (2017), A meta-heuristic with ensemble of local search operators for urban traffic light optimization, *in* '2017 IEEE Symposium Series on Computational Intelligence (SSCI)', pp. 1–8.

Geman, S., Bienenstock, E. and Doursat, R. (1992), 'Neural networks and the bias/variance dilemma', *Neural Computation* **4**(1), 1–58.
URL: *https://doi.org/10.1162/neco.1992.4.1.1*

Gil, R. P. A., Johanyak, Z. C. and Kovacs, T. (2018), 'Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation', *International Journal of Artificial Intelligence* **16**(1), 20–40.

Goel, T., Vaidyanathan, R., Haftka, R. T., Shyy, W., Queipo, N. V. and Tucker, K. (2007), 'Response surface approximation of pareto optimal front in multi-objective optimization', *Computer Methods in Applied Mechanics and Engineering* **196**(4), 879 – 893.
URL: *http://www.sciencedirect.com/science/article/pii/S0045782506002179*

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goodyer, E., Ahmadi, S., Chiclana, F., Elizondo, D., Gongora, M., Passow, B. N. and Yang, Y. (2013), 'Computational intelligence and its role in enhancing sustainable transport systems', *International Journal for Traffic and Transport Engineering (IJTTE)* **1**, 180–186.

Guangwei, Z., Albert, G. and Sherr, L. D. (2007), 'Optimization of adaptive transit signal priority using parallel genetic algorithm', *Tsinghua Science and Technology* **12**(2), 131–140.

Hamdan, M. (2010), 'On the disruption-level of polynomial mutation for evolutionary multi-objective optimisation algorithms.', *Computing and Informatics* **29**, 783–800.

Hamza-Lup, G., Hua, K., Le, M. and Peng, R. (2008), 'Dynamic plan generation and real-time management techniques for traffic evacuation', *Intelligent Transportation Systems, IEEE Transactions on* **9**(4), 615–624.

Heaton, J. (2008), *Introduction to Neural Networks for Java, 2nd Edition*, 2nd edn, Heaton Research, Inc.

Helbig, M. and Engelbrecht, A. P. (2013), 'Performance measures for dynamic multi-objective optimisation algorithms', *Information Sciences* **250**, 61 – 81.

Hess, S., Quddus, M., Rieser-Schüssler, N. and Daly, A. (2015), 'Developing advanced route choice models for heavy goods vehicles using gps data', *Transportation Research Part E: Logistics and Transportation Review* **77**, 29 – 44.
URL: *http://www.sciencedirect.com/science/article/pii/S1366554515000113*

Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA.

HSM (2010), *Highway Safety Manual*, Washington, D.C. : American Association of State Highway and Transportation Officials.

Huband, S., Hingston, P., Barone, L. and While, L. (2006), 'A review of multiobjective test problems and a scalable test problem toolkit', *IEEE Transactions on Evolutionary Computation* **10**(5), 477–506.

Husain, A. and Kim, K. (2010), 'Enhanced multi-objective optimization of a microchannel heat sink through evolutionary algorithm coupled with multiple surrogate models', *Applied Thermal Engineering* **30**(13), 1683 – 1691.
URL: *http://www.sciencedirect.com/science/article/pii/S1359431110001377*

Jin, C., Qin, A. K. and Tang, K. (2015), Local ensemble surrogate assisted crowding differential evolution, *in* '2015 IEEE Congress on Evolutionary Computation (CEC)', pp. 433–440.

Jin, R., Chen, W. and Simpson, T. (2001), 'Comparative studies of metamodelling techniques under multiple modelling criteria', *Structural and Multidisciplinary Optimization* **23**(1), 1–13.
URL: *https://doi.org/10.1007/s00158-001-0160-4*

Jin, Y. (2005), 'A comprehensive survey of fitness approximation in evolutionary computation', *Soft Comput.* **9**(1), 3–12.
URL: *http://dx.doi.org/10.1007/s00500-003-0328-5*

Jin, Y. (2011), 'Surrogate-assisted evolutionary computation: Recent advances and future challenges', *Swarm and Evolutionary Computation* **1**(2), 61 – 70.
URL: *http://www.sciencedirect.com/science/article/pii/S2210650211000198*

Jones, D., Mirrazavi, S. and Tamiz, M. (2002), 'Multi-objective meta-heuristics: An overview of the current state-of-the-art', *European Journal of Operational Research* **137**(1), 1 – 9.
URL: *http://www.sciencedirect.com/science/article/pii/S0377221701001230*

Kadali, B. R. and Vedagiri, P. (2016), 'Review of pedestrian level of service', *Transportation Research Record Journal of the Transportation Research Board* pp. 37–47.

Kai, Z., Gong, Y. J. and Zhang, J. (2014), Real-time traffic signal control with dynamic evolutionary computation, *in* 'Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on', pp. 493–498.

Kittelson & Associates, I. (2008), Traffic singal timing manual, Technical report, Federal Highway Administration.

Konak, A., Coit, D. W. and Smith, A. E. (2006), 'Multi-objective optimization using genetic algorithms: A tutorial', *Reliability Engineering & System Safety* **91**(9), 992 – 1007. Special Issue - Genetic Algorithms and Reliability.
URL: *http://www.sciencedirect.com/science/article/pii/S0951832005002012*

Kotusevski, G. and Hawick, K. (2009), 'A review of traffic simulation software', *Res. Lett. Inf. Math. Sci.* **13**, 35–54.

Kouvelas, A., Aboudolas, K., Papageorgiou, M. and Kosmatopoulos, E. (2011), 'A hybrid strategy for real-time traffic signal control of urban road networks', *Intelligent Transportation Systems, IEEE Transactions on* **12**(3), 884–894.

Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L. (2012), 'Recent development and applications of sumo-simulation of urban mobility', *International Journal on Advances in Systems and Measurements* **5**.

Krajzewicz, D., Hertkorn, G., Rossel, C. and Wagner, P. (2019), 'Sumo homepage: http://sumo.sourceforge.net'.

Kuhn, M. and Johnson, K. (2013), *Applied Predictive Modeling*, Springer.

L. Graening, Y. Jin, B. S. (2005), Efficient evolutionary optimization using individual-based evolution control and neural networks: A comprarative study, *in* 'European Symposium on Artificial Neural Networks'.

Li, K., Deb, K., Zhang, Q. and Zhang, Q. (2017), 'Efficient nondomination level update method for steady-state evolutionary multiobjective optimization', *IEEE Transactions on Cybernetics* **47**(9), 2838–2849.

Lim, D., Jin, Y., Ong, Y. and Sendhoff, B. (2010), 'Generalizing surrogate-assisted evolutionary computation', *IEEE Transactions on Evolutionary Computation* **14**(3), 329–355.

Liu, B., Zhang, Q. and Gielen, G. G. E. (2014), 'A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems', *IEEE Transactions on Evolutionary Computation* **18**(2), 180–192.

Liu, G., Han, X. and Jiang, C. (2008), 'A novel multi-objective optimization method based on an approximation model management technique', *Computer Methods in Applied Mechanics and Engineering* **197**(33), 2719 – 2731.
URL: *http://www.sciencedirect.com/science/article/pii/S0045782508000029*

Liu, H. X., Wu, X., Ma, W. and Hu, H. (2009), 'Real-time queue length estimation for congested signalized intersections', *Transportation Research Part C: Emerging Technologies* **17**(4), 412 – 427.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X09000230*

Lopez-Ibanez, M. and Stutzle, T. (2014), 'Automatically improving the anytime behaviour of optimisation algorithms', *European Journal of Operational Research* **235**(3), 569 – 582.

Mathew, T. V. (2014), Signalized intersection delay models, *in* 'Lectures Notes in Transportation System Engineering', Civil IITB-IIT Bombay.

Mihaita, A. S., Dupont, L. and Camargo, M. (2018), 'Multi-objective traffic signal optimization using 3d mesoscopic simulation and evolutionary algorithms', *Simulation*

Modelling Practice and Theory **86**, 120 – 138.
URL: *http://www.sciencedirect.com/science/article/pii/S1569190X18300686*

Mladenovic, N. and Hansen, P. (1997), 'Variable neighborhood search', *Computers & Operations Research* **24**(11), 1097 – 1100.
URL: *http://www.sciencedirect.com/science/article/pii/S0305054897000312*

Mustapha, S., Abdeslam, E. and Elbelrhiti, E. (2016), 'A comparative study of urban road traffic simulators', *MATEC Web Conf.* **81**, 05002.
URL: *https://doi.org/10.1051/matecconf/20168105002*

Neri, F. and Cotta, C. (2012), 'Memetic algorithms and memetic computing optimization: A literature review', *Swarm and Evolutionary Computation* **2**, 1 – 14.
URL: *http://www.sciencedirect.com/science/article/pii/S2210650211000691*

Nguyen, P. T. M., Passow, B. N. and Yang, Y. (2016), Improving anytime behavior for traffic signal control optimization based on nsga-ii and local search, *in* '2016 International Joint Conference on Neural Networks (IJCNN)', pp. 4611–4618.

Ong, Y.-S., Lim, M.-H., Zhu, N. and Wong, K.-W. (2006), 'Classification of adaptive memetic algorithms: a comparative study', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **36**(1), 141–152.

Ong, Y.-S., Nair, P. B. and Lum, K. Y. (2006), 'Max-min surrogate-assisted evolutionary algorithm for robust design', *IEEE Transactions on Evolutionary Computation* **10**(4), 392–404.

Osorio, C. and Bierlaire, M. (2009), 'A surrogate model for traffic optimization of congested networks: an analytic queueing network approach'.
URL: *http://infoscience.epfl.ch/record/152480*

Pan, I. and Das, S. (2015), 'Kriging based surrogate modeling for fractional order control of microgrids', *IEEE Transactions on Smart Grid* **6**(1), 36–44.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. and Wang, Y. (2003), 'Review of road traffic control strategies', *Proceedings of the IEEE* **91**(12), 2043–2067.

Papatzikou, E. and Stathopoulos, A. (2015), 'An optimization method for sustainable traffic control in urban areas', *Transportation Research Part C: Emerging Technologies*

55, 179 – 190. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X15000479*

Passos, L. S., Rossetti, R. J. F. and Kokkinogenis, Z. (2011), Towards the next-generation traffic simulation tools: a first appraisal, *in* '6th Iberian Conference on Information Systems and Technologies (CISTI 2011)', pp. 1–6.

Passow, B., Elizondo, D., Goodyer, E., Leigh, R., Lawrence, J., Shah, S., Obszynska, J., Brown, S., Gustafsson, S. and Huebner, N. (2012), itraq - an integrated traffic management and air quality control system using space services, *in* '4th International Conference on Space Applications, Toulouse Space Show, Toulouse, France, 25-28 June 2012.'.

Patel, V., Chaturvedi, M. and Srivastava, S. (2016), 'Comparison of sumo and simtram for indian traffic scenario representation', *Transportation Research Procedia* **17**, 400 – 407. International Conference on Transportation Planning and Implementation Methodologies for Developing Countries (12th TPMDC) Selected Proceedings, IIT Bombay, Mumbai, India, 10-12 December 2014.
URL: *http://www.sciencedirect.com/science/article/pii/S2352146516306962*

Pell, A., Meingast, A. and Schauer, O. (2017), 'Trends in real-time traffic simulation', *Transportation Research Procedia* **25**, 1477 – 1484. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.
URL: *http://www.sciencedirect.com/science/article/pii/S2352146517304684*

Pirdavani, A., Brijs, T., Bellemans, T. and Wets, G. (2010), A simulation-based traffic safety evaluation of signalized intersections., *in* 'TRB, Road safety on four continents: 15th international conference, Abu Dhabi, United Arab Emirates.'.

Pontes, F., Amorim, G., Balestrassi, P., Paiva, A. and Ferreira, J. (2016), 'Design of experiments and focused grid search for neural network parameter optimization', *Neurocomputing* **186**(Supplement C), 22 – 34.
URL: *http://www.sciencedirect.com/science/article/pii/S0925231215020184*

Poole, A. and Kotsialos, A. (2016), 'Swarm intelligence algorithms for macroscopic traffic flow model validation with automatic assignment of fundamental diagrams', *Applied*

Soft Computing **38**, 134 – 150.
URL: *http://www.sciencedirect.com/science/article/pii/S1568494615005827*

Quddus, M. A., Rahman, F., Monsuur, F., de Ona, J. and Enoch, M. P. (2019), 'Analysing bus passengers' satisfaction in dhaka using discrete choice models', *SAGE Publications - National Academy of Sciences: Transportation Research Board* .
URL: *https://dspace.lboro.ac.uk/2134/36511*

Qun, C. (2009), Research on signal control of urban intersection based on genetic algorithms, *in* 'Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on', Vol. 1, pp. 193–196.

Reyes-Sierra, M. and Coello, C. A. C. (2005), A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization, *in* '2005 IEEE Congress on Evolutionary Computation', Vol. 1, pp. 65–72 Vol.1.

Riedmiller, M. and Braun, H. (1993), A direct adaptive method for faster backpropagation learning: the rprop algorithm, *in* 'IEEE International Conference on Neural Networks', pp. 586–591 vol.1.

Riquelme, N., Lucken, C. V. and Baran, B. (2015), Performance metrics in multi-objective optimization, *in* '2015 Latin American Computing Conference (CLEI)', pp. 1–11.

Robert E. Smith, B. A. Dike, S. A. S. (0995), Fitness inheritance in genetic algorithms, *in* 'SAC '95 Proceedings of the 1995 ACM symposium on Applied computing', pp. 345–350.

Robertson, D. I. (1986), 'Research on the transyt and scoot methods of signal coordination', *ITE JOURNAL* **56**, 36–40.

Robertson, D. I. and Bretherton, R. D. (1991), 'Optimizing networks of traffic signals in real time-the scoot method', *IEEE Transactions on Vehicular Technology* **40**(1), 11–15.

Rodriguez, J. D., Perez, A. and Lozano, J. A. (2010), 'Sensitivity analysis of k-fold cross validation in prediction error estimation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(3), 569–575.

Rosales-Perez, A., Gonzalez, J. A., Coello, C. A., Escalante, H. J. and Reyes-Garcia, C. A. (2015), 'Surrogate-assisted multi-objective model selection for support vector machines', *Neurocomputing* **150**, 163 – 172. Bioinspired and knowledge based techniques and applications The Vitality of Pattern Recognition and Image Analysis Data Stream Classification and Big Data Analytics.
URL: *http://www.sciencedirect.com/science/article/pii/S0925231214012612*

Sabar, N. R., Kieu, L. M., Chung, E., Tsubota, T. and de Almeida, P. E. M. (2017), 'A memetic algorithm for real world multi-intersection traffic signal optimisation problems', *Engineering Applications of Artificial Intelligence* **63**, 45 – 53.
URL: *http://www.sciencedirect.com/science/article/pii/S0952197617300854*

Sanchez-Medina, J., Galan-Moreno, M. and Rubio-Royo, E. (2010), 'Traffic signal optimization in "la almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing', *Intelligent Transportation Systems, IEEE Transactions on* **11**(1), 132–141.

Sanghamitra Bandyopadhyay, S. S. (2013), *Unsupervised Classification - Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, Springer.

Santana-Quintero, L. V., Montaño, A. A. and Coello, C. A. C. (2010), *A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 29–59.
URL: *https://doi.org/10.1007/978-3-642-10701-6_2*

Sharma, A., Bullock, D. M. and Bonneson, J. A. (2007), 'Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections', *Transportation Research Record* **2035**(1), 69–80.
URL: *https://doi.org/10.3141/2035-08*

Sheela, K. G. and Deepa, S. N. (2013), 'Review on methods to fix number of hidden neurons in neural networks', *Mathematical Problems in Engineering* **2013**.
URL: *https://doi.org/10.1155/2013/425740*

Shen, Z., Wang, K. and Wang, F.-Y. (2013), Gpu based non-dominated sorting genetic algorithm-ii for multi-objective traffic light signaling optimization with agent based modeling, *in* 'Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on', pp. 1840–1845.

Shen, Z., Wang, K. and Zhu, F. (2011), Agent-based traffic simulation and traffic signal timing optimization with gpu, *in* 'Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on', pp. 145–150.

Sheng-hai, A., Byung-Hyug, L. and Dong-Ryeol, S. (2011), A survey of intelligent transportation systems, *in* 'Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on', pp. 332–337.

Srinivasan, D., Choy, M. C. and Cheu, R. L. (2006), 'Neural networks for real-time traffic signal control', *IEEE Transactions on Intelligent Transportation Systems* **7**(3), 261–272.

Stephanopoulos, G., Michalopoulos, P. G. and Stephanopoulos, G. (1979), 'Modelling and analysis of traffic queue dynamics at signalized intersections', *Transportation Research Part A: General* **13**(5), 295 – 307.
URL: *http://www.sciencedirect.com/science/article/pii/0191260779900281*

Sun, D., Benekohal, R. and Waller, S. (2003), Multiobjective traffic signal timing optimization using non-dominated sorting genetic algorithm, *in* 'Intelligent Vehicles Symposium, 2003. Proceedings. IEEE', pp. 198–203.

Sun, X., Gong, D., Jin, Y. and Chen, S. (2013), 'A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning', *IEEE Transactions on Cybernetics* **43**(2), 685–698.

T. Simpson, T. Mauery, J. K. and Mistree, F. (1998), Comparison of response surface and kriging models for multidisciplinary design optimization, Technical report, Technical Report 98-4755, AIAA.

Tamura, S. and Tateishi, M. (1997), 'Capabilities of a four-layered feedforward neural network: four layers versus three', *IEEE Transactions on Neural Networks* **8**(2), 251–255.

Teply, S., D.I.Allingham, Richardson, D. and Stephenson, B. (2008), Canadian capacity guide for signalized intersections, Technical report, Canadian Institute of Transportation Engineers.

Tettamanti, T., Luspay, T., Kulcsar, B., Peni, T. and Varga, I. (2014), 'Robust control for urban road traffic networks', *Intelligent Transportation Systems, IEEE Transactions on* **15**(1), 385–398.

Tong, H., Hung, W. and Cheung, C. (2000), 'On-road motor vehicle emissions and fuel consumption in urban driving conditions', *Journal of the Air & Waste Management Association* **50**(4), 543–554.
URL: *https://doi.org/10.1080/10473289.2000.10464041*

Tung, H.-Y., Ma, W.-C. and Yu, T.-L. (2014), Novel traffic signal timing adjustment strategy based on genetic algorithm, *in* 'Evolutionary Computation (CEC), 2014 IEEE Congress on', pp. 2353–2360.

van Essen, H., Schroten, A., Otten, M., Sutter, D., Schreyer, C., Zandonella, R., Maibach, M. and Doll, C. (2011), External costs of transport in europe, Technical report.
URL: *http://www.cedelft.eu/*

Wang, Y., Yang, X., Liang, H. and Liu, Y. (2018), 'A review of the self-adaptive traffic signal control system based on future traffic environment', *Journal of Advanced Transportation* .

Webster, F. and Cobbe, B. (1966), 'Traffic signals', *Road Research Laboratory Technical Paper No.56, London, U.K* .

Webster, F. V. (1958), 'Traffic signal settings', *Road Research Technique Paper No. 39, Road Research Laboratory, London, 1958* .

WHO (2018), Global status report on road safety, Technical report, World Health Organization.

Witheridge, S., Passow, B. and Shell, J. (2014), Logan's run: Lane optimisation using genetic algorithms based on nsga-ii, *in* 'Neural Networks (IJCNN), 2014 International Joint Conference on', pp. 63–68.

Wu, A. and Yang, X. (2013), 'Real-time queue length estimation of signalized intersections based on rfid data', *Procedia - Social and Behavioral Sciences* **96**, 1477 – 1484. Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th {COTA} International Conference of Transportation Professionals

(CICTP2013).
URL: *http://www.sciencedirect.com/science/article/pii/S1877042813022945*

Yan, L., Lijie, Y., Siran, T. and Kuanmin, C. (2013), 'Multi-objective optimization of traffic signal timing for oversaturated intersection', *Mathematical Problems in Engineering* .

Yin, B., Dridi, M. and El Moudni, A. (2015), Adaptive traffic signal control for multi-intersection based on microscopic model, *in* '2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)', pp. 49–55.

Zakariya, A. Y. and Rabia, S. I. (2016), 'Estimating the minimum delay optimal cycle length based on a time-dependent delay formula', *Alexandria Engineering Journal* **55**(3), 2509 – 2514.
URL: *http://www.sciencedirect.com/science/article/pii/S1110016816302034*

Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X. and Chen, C. (2011), 'Data-driven intelligent transportation systems: A survey', *Intelligent Transportation Systems, IEEE Transactions on* **12**(4), 1624–1639.

Zhang, M., Zhao, S., Lv, J. and Qian, Y. (2009), Multi-phase urban traffic signal real-time control with multi-objective discrete differential evolution, *in* 'Electronic Computer Technology, 2009 International Conference on', pp. 296–300.

Zhang, X., Tian, Y., Cheng, R. and Jin, Y. (2015), 'An efficient approach to nondominated sorting for evolutionary multiobjective optimization', *IEEE Transactions on Evolutionary Computation* **19**(2), 201–213.

Zhao, D., Dai, Y. and Zhang, Z. (2012), 'Computational intelligence in urban traffic signal control: A survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(4), 485–494.

Zheng, Y.-J., Zhang, M.-X., Ling, H.-F. and Chen, S.-Y. (2015), 'Emergency railway transportation planning using a hyper-heuristic approach', *Intelligent Transportation Systems, IEEE Transactions on* **16**(1), 321–329.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N. and Zhang, Q. (2011), 'Multiobjective evolutionary algorithms: A survey of the state of the art', *Swarm and*

Evolutionary Computation **1**(1), 32 – 49.
URL: *http://www.sciencedirect.com/science/article/pii/S2210650211000058*

Zhou, S., Yan, X. and Wu, C. (2008), Optimization model for traffic signal control with environmental objectives, *in* 'Natural Computation, 2008. ICNC '08. Fourth International Conference on', Vol. 6, pp. 530–534.

Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J. and Lum, K. Y. (2007), 'Combining global and local surrogate models to accelerate evolutionary optimization', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(1), 66–76.

Zitzler, E., Laumanns, M. and Bleuler, S. (2004), A tutorial on evolutionary multiobjective optimization, *in* X. Gandibleux, M. Sevaux, K. Sörensen and V. T'kindt, eds, 'Metaheuristics for Multiobjective Optimisation', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–37.

Zitzler, E. and Thiele, L. (1998), Multiobjective optimization using evolutionary algorithms - a comparative case study, *in* 'Proceedings of the 5th International Conference on Parallel Problem Solving from Nature', PPSN V, Springer-Verlag, London, UK, UK, pp. 292–304.
URL: *http://dl.acm.org/citation.cfm?id=645824.668610*

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and da Fonseca, V. G. (2003), 'Performance assessment of multiobjective optimizers: an analysis and review', *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132.