An-Najah National University

Faculty of Graduate Studies

Optimization of Traffic Signals Timing Using Parameter-less Metaheuristic Optimization Algorithms

By

Thaer Thaher

Supervisor

Dr. Baker Abdulhaq

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Advanced Computing, Faculty of Graduate Studies, An-Najah National University, Nablus- Palestine.

2018

Optimization of Traffic Signals Timing using Parameter-less Metaheuristic Optimization Algorithms

By

Thaer Thaher

This thesis was defended successfully on 22 /7/2018, and approved by:

Defense Committee Members	**Signature**
- Dr. Baker Abdulhaq/ Supervisor	……………………………...
- Dr. Majdi Mafarja/ External Examiner	……………………………...
- Dr. Ahmed Awad/ Internal Examiner	……………………………...

Acknowledgement

First and foremost, I would like to thank our Almighty God for giving me the strength and knowledge to undertake and complete this research study. I'm also very grateful for the following people for their support and encouragement.

Firstly, no thanking words will be enough to express my appreciation to Dr. Baker Abdulhaq for his invaluable guidance and advice, suggestions, patience, encouragement and continuous support as my research supervisor. In addition, the door to Dr. Abdulhaq office was always open whenever I had a question about my research and writing. I'm very thankful to him for giving me an opportunity to work in the field of traffic and optimization algorithms.

Special thanks to my instructors of the advanced computing program at An-Najah National University, Dr. Adnan Salman, Dr. Fadi Draidi, Dr. Sameer Matar, Dr. Mohammad Najeeb, Dr. Ali Barakat, Dr. Anwar Saleh, and Dr. Abdel-Razzak Natsheh for enhancing my knowledge and ability to complete my research.

Furthermore, I would like to express my sincere thanks to the members of the discussion committee for their time and effort in reviewing this study.

I also thank my cousin Marah Hamdi for her help with proofreading this work.

Finally, I must express my greatest gratitude to my parents, my wife and children, my brothers and sisters for their endless love and continuous encouragement throughout my years of study.

My deep appreciation is also extended to all individuals, even those who played a little role, for their efforts in making this work a success. Without your tireless efforts, this work would not have been possible.

الإقرار

انا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

"Optimization of Traffic Signals Timing Using Parameter-less Metaheuristic Optimization Algorithms"

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل أية درجة علمية أو لقب علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى

Declaration

The work provided in this thesis, unless otherwise referenced, is the researcher's own work and has not been submitted elsewhere for any other degree or qualification.

Student's name: **اسم الطالب:** ثائر أحمد درويش ظاهر

Signature: **التوقيع:**

Date: **التاريخ:**

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACO	**A**nt **C**olony **O**ptimization
ATT	**Average Travel Time**
CORSIM	**Cor**ridor **Sim**ulation
GA	**G**enetic **A**lgorithm
GSA	**G**ravitational **S**earch **A**lgorithm
HC	**H**ill **C**limbing
HCM	**H**ighway **C**apacity **M**anual
HS	**H**armony **S**earch
MA	**M**emetic **A**lgorithm
MOTION	**M**ethod for the **O**ptimization of **T**raffic **S**ignals in **O**nline **C**ontrolled **N**etworks
OPAC	**O**ptimized **P**olicies for **A**daptive **C**ontrol
PASSER	**P**rogression **A**nalysis and **S**ignal **S**ystem **E**valuation **R**outine
Psize	**P**opulation **Size**
PSO	**P**article **S**warm **O**ptimization
RHODES	**R**eal-time **H**ierarchical **O**ptimized **D**istributed and **E**ffective **S**ystem
SA	**S**imulated **A**nnealing
SCATS	**S**ydney **C**oordinated **A**daptive **T**raffic **S**ystem
SCOOT	**S**plit **C**ycle and **O**ffset **O**ptimization **T**echnique
SUMO	**S**imulation of **U**rban **Mo**bility
TLBO	**T**eaching **L**earning **B**ased **O**ptimization
TRANSYT	**TRA**ffic **N**etwork **S**tudy **T**ool
TS	**T**abu **S**earch
TSIS	**T**raffic **S**oftware **I**ntegrated **S**ystem
TSOP	**T**raffic **S**ignals **O**ptimization **P**roblem
WTLBO	**W**eighted **T**eaching **L**earning **B**ased **O**ptimization

Optimization of Traffic Signals Timing Using Parameter-less Metaheuristic Optimization Algorithms

By
Thaer A. Thaher
Supervisor
Dr. Baker Abdulhaq

Abstract

Traffic congestion is a common challenge in urban areas, so several methods are used to reduce it. A powerful solution that can reduce the congestion problem is by developing a real-time traffic light control system with an optimization technique to minimize the overall traffic delay through optimizing the traffic signals timing. Researchers have proposed several simulation models and used various techniques to optimize the traffic signals timing.

The purpose of this research is to evaluate and compare the performance of several meta-heuristic techniques in tackling the Traffic Signals Optimization Problem (TSOP). In this work, recently published algorithms that do not have specific parameters (the parameter-less) such as Teaching-Learning-Based Optimization (TLBO) and Jaya are applied to solve the traffic signals optimization problem. These algorithms have not been applied to the considered problem yet.

A stochastic micro-simulator called 'Simulation of Urban Mobility' ***(SUMO)*** is used as a tool to implement and evaluate the performance and convergence speed of each algorithm. Three road networks of different

sizes: small, medium and large containing 13, 34 and 141 phases respectively are simulated to study the scalability of algorithms.

The performance of TLBO and Jaya algorithms are compared to three algorithms that have some parameters that need to be set such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Weighted Teaching-Learning-Based Optimization (WTLBO). The study also considers the effect of common controlling parameters (i.e. the population size) on the performance of the evaluated algorithms.

After conducting many experiments, the comparisons and discussions have shown that TLBO and Jaya outperformed WTLBO, GA, and PSO for small and medium-sized networks. Moreover, TLBO achieved the best performance and scalability for the complex network.

1. Introduction

1.1 Research Background and Motivation

Traffic jams are becoming a major problem that faces most countries in the world, especially developing ones. There is a steady increase in the population rate and thus an increase in the number of roads, and vehicles that cause traffic congestion (Gao et al., 2016). As a result, drivers and travelers are facing many problems such as air pollution, time wasting, fuel consuming, frustration, economic loss and other serious problems (Abushehab et al., 2014).

There is a number of suggested solutions to alleviate the problem. Urban planners tried to tackle this phenomenon through building new lanes, bridges and expanding them (Kumar & Sing, 2017). However, it did not meet the anticipated success. The first problem with this solution is that it is expensive, and it is impossible to do that in urban cities due to the residential areas and nearby buildings (Bazzan & Ana, 2007). Researchers are therefore resorting to the optimal utilization of the available infrastructure (Hu et al., 2015).

In traffic systems, there is a relationship between the timing of the traffic lights and the total traveling time for all vehicles in the network, so the adjustment of signal timing can give more green time to an intersection with heavy traffic or shorten or even skip a phase that has little or no traffic waiting. Thus, it may lead to increase or decrease the travel time for vehicles (Xie et al., 2014). when choosing the average travel time as a measure of efficiency for the traffic network, the best values for the time of traffic lights are those that give the minimum average travel time for all vehicles.

Due to the limitation of the supplied resources from the current infrastructure, smart traffic light control, and coordination system are becoming highly required to guarantee that traffic moves as smoothly as possible (Gao et al., 2016). These smart systems can be developed by replacing the traditional traffic light systems with smart ones that self-adjust timing based on the historical data collected by detectors (sensors, cameras) (Aljaafreh & Al-Oudat, (2014). According to *Warberg et al*. (2008), the correct utilization of smart traffic signals might increase the road's capacity [The maximum number of vehicles obtainable on a given roadway over a period of time] in the Greater Copenhagen area by 5 to 10%.

The desired objective of the problem is to obtain a global optimal scheduling of traffic lights which enhances the traffic conditions comprehensively (Hu et al., 2015). In urban networks, there are hundreds of intersections which are controlled by traffic lights. These traffic lights require a proper control and coordination to achieve the desired objective (Gao et al., 2016). However, how to optimize the timings of hundreds of traffic signals, has become a complex and challenging problem (Hu et al., 2015).

The traffic lights scheduling can be considered as an NP-hard problem (Sklenar et al., 2009). It is a real-world problem where the optimal solution is unknown (Adacher, 2012). It is difficult to develop a closed-form mathematical model to describe the stochastic behavior of traffic system (Yun & Park, 2006). In addition, the greater the number of traffic lights, the greater the problem search space, then the complexity of the search will be much higher (Talbi, 2009).

The vast majority of the real-world optimization problems in several areas such as transportation, engineering, manufacturing, and so on are NP-hard problems (Talbi, 2009). For complex optimization problems (e.g. NP-hard or global optimization), exact algorithms are not appropriate to be used because the amount of required time to find the optimal solution may increase exponentially relative to the dimensions of the problem (Beheshti & Shamsuddin, 2013). Hence, heuristic methods are more suitable to solve complex problems with a high-dimensional search space where it tends to find a good solution in a reasonable amount of time (Talbi, 2009). Heuristic methods can be classified into two types: specific heuristic designed for specific purpose problems (problem-dependent) and metaheuristic developed to solve a wide range of problems (problem-independent) (Talbi, 2009; Beheshti & Shamsuddin, 2013)

Metaheuristics algorithms have shown superior performance in solving a very large variety of optimization problems such as scheduling problems, parameter optimization, feature selection, automatic clustering, Neural Network training and so~~n~~ on (Mafarja & Mirjalili, 2018; Torres-Jimenez & Pavon, 2014). Recently, those algorithms have become popular for solving the traffic signals scheduling problem (Garcia-Nieto et al, 2013; Abushehab et al., 2014).

Metaheuristic techniques are classified into two categories according to the number of solution being processed in each iteration: single solution-based algorithms and population-based algorithms (Luke, 2013). Most of the population-based metaheuristic algorithms are inspired by naturally occurring phenomena (Talbi, 2009). They can be classified into four major groups: evolution-based (e.g. GA), swarm-based (e.g. PSO), physics-based (e.g. Simulated Annealing 'SA'), and human-based (e.g. Harmony Search 'HS') (Panimalar, 2017). Two contradictory approaches need to be balanced in all these techniques to achieve suitable performance: diversification (exploration of the search space) and intensification (exploitation of the best solution found) (Yang, 2010; Talbi, 2009).

Metaheuristic algorithms have their own specific parameter(s) in addition to the common control parameters like population size, the number of generations and elite size (Rao, 2016). The effectiveness of algorithms is sensitive to parameters' values. The wrong choice for the values of parameters will either increase the computational effort or lead to a wrong optimal solution. (Rao et al., 2012)

Parameter values selection is either assumed according to past experience or tuned to suit each new problem (Neumuller & Wagner, 2011). However, finding good values for parameters is difficult and time-consuming. The search for the optimal parameter values can be seen as an optimization problem itself (Neumuller et al., 2012). For these reasons, the search is still ongoing to modify algorithms with adaptive parameters methods or find new algorithms that are free of parameters.

Population size is a common parameter to all population-based techniques. It has a significant influence on the performance and convergence of metaheuristic algorithms, and therefore must be taken into consideration (Diaz-Gomez & Hougen, 2007 ; Roeva et. al, 2014; Mora-Melia et.al, 2017;). Several studies have examined the effect of population size on the effectiveness of algorithms, some studies have shown that small population size leads to the lack of sufficient diversity and will not provide good solutions (Koumousis & Katsaras, 2006), and other studies also have argued that large population size may leads to undesirable results (Lobo & Goldberg, 2004; Chen et. al, 2012 ; Roeva et al, 2014; Mora-Melia et al, 2017). Therefore, more investigation should be done to find an appropriate approximation for the population size parameter that yields better solutions.

Traffic system is a complex, dynamic, and adaptive system. It consists of interacting sub-systems which depends heavily on stochastic behaviors, and thus lead to unpredictable outcomes (López-Neri et al., 2010). Therefore, there is no closed mathematical form that can be used as a model which is capable of describing all the stochastic behavior of the traffic system components (Krajzewicz et al., 2002). Hence, simulation is an effective way for the experimental studies of the traffic system (Olstam, & Tapani, 2004).

The process of Traffic Signals Optimization Problem (TSOP) consists of two sub-problems: the optimization algorithm and the simulation model which is used to evaluate the objective function (Adacher, 2012). In this study, a microscopic traffic simulator called SUMO 'Simulation of Urban Mobility' integrated with parameterless metaheuristic algorithms called TLBO and Jaya have been used to determine the best time for each traffic signal and thus minimize the delay time for vehicles.

Recently, various optimization techniques have been used to solve the problem of traffic light optimization (Abushehab et al., 2014). However, due to the stochastic behavior of these techniques, there is no guarantee to find the optimal solution (Luke, 2013). Also, they may suffer from poor performance in solving some problems. Besides, the No-Free-Lunch (NFL) theorem confirms that there is no algorithm that can be considered the best to solve all optimization problems (Wolpert & Macready, 1997). Therefore, the answer to "which algorithm is most appropriate to solve the problem" remains open (Abdalhaq & Abu Baker, 2014). These reasons motivated us to investigate the efficiency of recently published algorithms such as TLBO and Jaya in the field of traffic signals timing optimization for the first time in literature.

1.2 Research Objectives

The main aim of this study is to develop a computational framework that is based on the integration of SUMO and an efficient metaheuristic optimizer which offers a better solution to TSOP and thus lead to minimize the average travel time of all vehicles. To achieve the main aim of this thesis, the following objectives were formulated:

- To apply different metaheuristic algorithms to optimize the traffic signals timing.
- To identify the effect of common controlling parameters such as population size on the performance of each algorithm for the optimization of traffic signals timing. And then estimate the most suitable population size for the considered algorithms.
- To identify the scalability of the algorithms through evaluating them on simple and complex networks.

1.3 Research Hypotheses

There are three research hypotheses that need to be tested at this phase of the research:

- The choice of common controlling parameter(s) values such as population size has a great impact on the performance of the algorithms to optimize traffic signals timing.
- The parameter-less algorithms such as TLBO and Jaya outperform the other traditional algorithms such as GA and PSO in solving the optimization of traffic signals timing problem.
- The performance of the algorithms varies depending on the size and characteristics of the network to be resolved.

1.4 Significance of the Research

The findings of this research will redound to the benefit of society, as well as specialists and researchers in the field of traffic system development. The growing of traffic congestion in urban traffic networks justifies the need for more effective approaches that alleviate this problem. Thus, the Ministry of Transport and Municipalities that apply the recommendations derived from the results of this study may alleviate traffic congestion and subsequent problems such as air pollution, fuel consumption, time wasting, and frustration.

In this study, recently published parameter-less algorithms (i.e. TLBO and Jaya) have been used to optimize the duration of traffic light phases in order to minimize the average of travel time for the vehicles. An improved version of TLBO called weighted TLBO (WTLBO), which is introduced by *Satapathy et al* (2013), is also tested. The performance and convergence rates of these algorithms have been compared with tuned GA and PSO algorithms selected from *Abushehab et al.* (2014) research. To study the scalability of each algorithm, the three different road networks, that have different characteristics and different number of traffic lights, have been simulated.

The findings of this study will raise the awareness of researchers about a better solution for TSOP. It will also give them a perception of the effectiveness of the metaheuristic techniques that have been tested in this study, especially the parameter-less algorithms, and thus determine the most appropriate algorithm for the traffic signals timing optimization.

1.5 Thesis Structure

This thesis consists of six chapters. The rest of the thesis is organized as follows:

Chapter two introduces a theoretical background that covers an introduction to optimization problem and solution techniques. Then, the metaheuristic optimization techniques such as TLBO, Jaya, WTLBO, GA, and PSO are reviewed. Furthermore, it introduces the modeling and simulation approaches to traffic systems.

Chapter three introduces the literature review in modeling and simulation of traffic systems, and then it reviews the approaches that have been used to optimize traffic light timing, including mathematical optimization models, simulation-based approaches, and metaheuristic techniques.

Chapter four explains the methodology which is used to answer the study questions. The methodology focuses on the use of a suitable microscopic traffic simulator integrated with an efficient metaheuristic optimization technique. In addition, chapter four presents the cases of the study, the model design of traffic signal optimization problem, the experimental setup, procedures, and statistical analysis.

Chapter five presents the simulation results and data analysis in the form of descriptive and inferential statistics. Furthermore, the performance and convergence speed of each tested algorithm is also discussed.

The last chapter summarizes the conclusions and recommendations. It also outlooks promising directions for future work.

2. Theoretical Background

2.1 Introduction to Optimization

Optimization is the process of finding the best solutions that give the maximum or minimum of a function (Chong & Zak, 2013). The optimum search methods are known as mathematical programming methods. In every optimization problem, there are the following elements: 1) search space which is the set of possible solutions. 2) cost function (objective function) which is the model that is used to evaluate solutions. 3) constraints (possibly empty) which is a set of conditions for the input variables that are required to be satisfied. (Neumüller& Wagner, 2011)

An optimization problem has the following form:

$$Maximize\ /\ Minimize\ f(x)$$
$$Subject\ to\ x\ \in\ \Omega \qquad (2.1)$$

Where:

- f: $R^n \rightarrow R$ is the objective function to be minimized or maximized.
- $x = [x_1, x_2, \ldots\ldots., x_n]^T \in R^n$ is an n-vector of parameters (*decision variables*)
- Ω: is a subset of R^n which is called *constraint set* or *feasible set.*

  The constraints are called *functional constraints* when Ω can be defined by some functions. It takes the form: $\Omega = \{x : h(x) = 0 , g(x) \leq =0 \}$

The above optimization problem can be defined as finding the best values of decision variables for vector x from all candidate vectors in Ω which minimize/maximize the objective function *f*. The optimization problem is either constraint or unconstraint. A previous standard is a general form for a constraint problem. If $\Omega = R^n$ then the problem is unconstraint. (Chong & Zak, 2013)

A variety of real-world problems can be formulated as an optimization problem. Indeed, optimization techniques are widely used to solve many real-world problems in several areas, such as automatic control systems, electronic design, chemical, mechanical, and civil design problems (Boyd & Vandenberghe, 2015, p.3). Furthermore, they are also used to solve traffic problems such as network designs and TSOP (Garcia-Nieto et al). The technique selection depends on the nature and the characteristics of the problem to be solved (Talbi, 2009, p. 3-9).

Optimization methods can be classified in several ways (see Figure 2.1), one of these classifications divides them into exact methods and heuristic methods depending on the complexity of the problem (Beheshti & Shamsuddin, 2013). Exact methods, such as dynamic programming, constraint programming, backtracking methods, branch-and-X methods (branch-and-bound, branch-and-cut, branch-and-price) guarantee finding the optimal solution for the problem being solved, they are suitable to solve small instances of difficult problems where the required time increases polynomially relative to the dimensions of the problem (Rothlauf, 2013, P.45). Whereas heuristic methods do not guarantee that globally optimal solution can be found in some class of problems, they can find "near optimal" solution in a reasonable amount of time (Talbi, 2009, P.21). In combinatorial optimization problems with a high-dimensional search space, finding all possible solutions are consuming time and resources. By searching over a large set of feasible solutions, heuristic methods can often find good solutions with less computational effort and therefore they are appropriate to solve this class of problems (Beheshti & Shamsuddin, 2013).

In general, heuristic methods can be classified into two types: specific heuristic and metaheuristic. Specific heuristic methods are problem-dependent and they are developed to solve very specific purpose problems. On the other hand, metaheuristic methods are a high-level problem-independent, so they are suitable to solve a wide range of problems (Talbi, 2009, P.21).

2.2 Metaheuristic Optimization Techniques

Metaheuristic techniques are a kind of stochastic optimization methods where some degree of randomness and probability is employed to find the (near) optimal solutions (Neumüller & Wagner, 2011). These methods explore the search space to find good solutions without guaranteeing the optimal solution. They are suitable for (*I knew it when I see it*) problems (Luke, 2013). In such problems, we do not have previous information about how the best solution seems. When we are given a candidate solution, its goodness or suitability can be evaluated using the objective function. (Luke, 2013)

Metaheuristic algorithms can be classified in many ways; one of the most popular categorizations is depending on the number of solutions being processed in each iteration. Single solution based (S-based) algorithms are algorithms that manipulate one solution in each iteration in the optimization process, while the population-based (P-based) algorithms manipulate a set of solutions (called population) in each iteration of the optimization process (*Luke, 2013*). Simulated Annealing (SA), Tabu Search (TS), and Great Deluge (GD) are examples of the S-based Metaheuristic algorithms. Genetic Algorithm (GA), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) are examples of P-based Metaheuristic algorithms.

Moreover, depending on the nature of inspiration, where most of the population-based metaheuristic algorithms are nature-inspired (Talbi, 2009), they can be classified into four major groups: evolution-based (e.g. GA, ES), swarm-based (e.g. PSO, TLBO, Jaya, and ACO, and), physics-based (e.g. SA, GSA), and human-based (e.g. HS). (Arockia, 2017).

In p-based metaheuristic algorithms, the optimization process is accomplished in two main phases: exploration (or diversification), and exploitation (or intensification). In exploration, a large scale of regions of the search space is examined to generate diverse solutions, so that reducing the chance of getting trap into a local minimum (Beheshti & Shamsuddin, 2013). On the other hand, exploitation means to examine the promising regions more carefully to find better solutions (Talbi, 2009). However, a proper trade-off between these two components is required to achieve the global optimality (Yang, 2010, P.5).

Metaheuristic algorithms are probabilistic algorithms and thus require their own specific parameters in addition to the common controlling parameters (Rao & Patel, 2012). These algorithms are highly sensitive to the parameter settings. Missing to fine tune the values for those parameters will negatively affect the performance of the employed algorithm (Neumuller et al. 2012). Considering this fact, recently published parameter-less algorithms called TLBO and Jaya have been introduced and shown a good performance in solving a variety of problems (Rao et al., 2011; Rao, 2016).

In this study, to solve the TSOP, the performance of parameter-less algorithms (e.g., TLBO and Jaya) was compared to the performance of algorithms that have their own parameters (e.g., WTLBO, GA, and PSO).

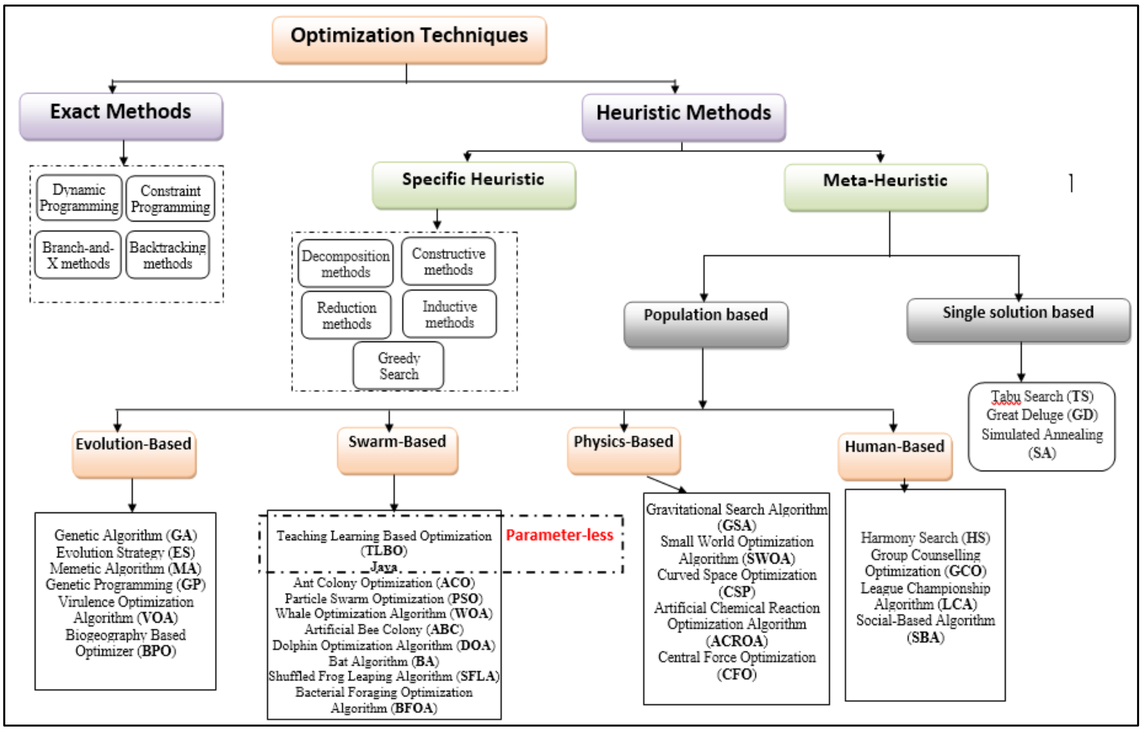


Figure 2.1: Optimization techniques classification

2.2.1 Parameter-less Algorithms

Different from other evolutionary and swarm intelligence based algorithms, these algorithms are free of any specific parameters and require only common controlling parameters like population size, number of iterations, and elite size. This category contains two recently published algorithms: TLBO and Jaya. (Rao, 2016b)

2.2.1.1 Teaching-Learning-Based Optimization (*TLBO*) Algorithm

TLBO is a population-based heuristic optimization method introduced by *Rao et al*. (2011). It simulates the teaching-learning process of the classroom, where learners represent the population, while the subjects which are given to learners represent the decision variables (Rao et al., 2011). The learners' results are equivalent to the fitness value of the optimization problem. The best learner (The learner who has the highest knowledge in the entire population according to the fitness value) is chosen as the teacher.

In TLBO, the optimization process is divided. The first one is called 'Teacher Phase' and the second one is called 'Learner Phase'. In the teacher phase, the learning process depends on the teacher himself/herself, but in the learner phase, the learning process is done through the interaction between learners. The two phases are explained in the next section (Rao, 2015).

Teacher Phase

In this phase, the teacher relies on his/her ability to transfer knowledge to the learners to raise their grades and thus to improve the mean results of the class (Rao et al., 2011). As shown in Fig 2.2, the teacher T_A makes an effort to shift the current mean of the learner M_A towards his/her level and gets a new mean M_B (Rao et al., 2012).

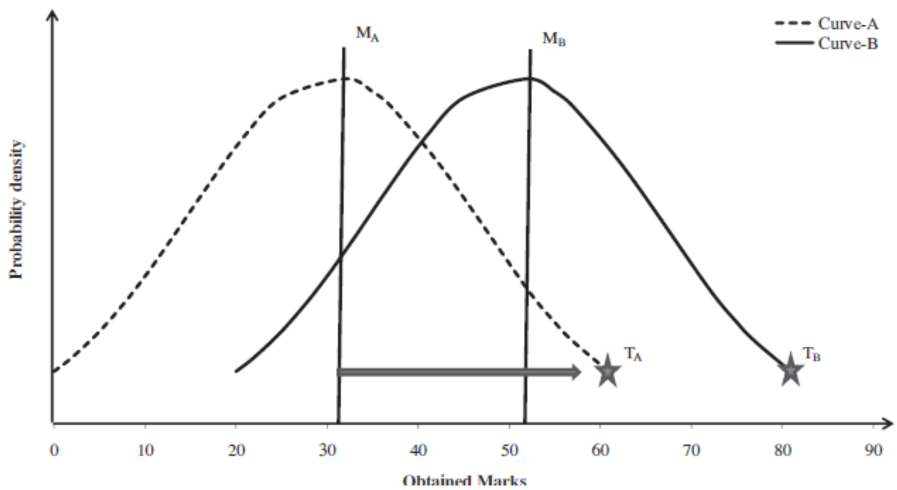


Figure 2.2: Distribution of marks for a group of learners (Rao et al., 2011)

The existing solution is modified according to Eqs. (2.2) and (2.3). The new solution is accepted if it gives better function value; otherwise, we keep the old one (Rao, 2016a).

$$Difference\_Mean_{j,i} = r * (X_{j,kbest,i} - TF * M_{j,i}) \quad (2.2)$$

$$X_{j,k,i}^{new} = X_{j,k,i}^{old} + Difference\_Mean_{j,i} \quad (2.3)$$

where:

i: represents the current iteration.
j: represents the subject (*j*=1 ….m)
k: represents the learner (*k*=1 …. n)
r: is a uniformly distributed random number within (0,1).
$X_{j,kbest,i}$: represents the result of the teacher (i.e. best learner) in subject *j*
TF: is the Teaching Factor which randomly calculated as in Eq. (2.4)
$M_{j,i}$: represents the mean result of all learners in subject *j*.
$Difference\_Mean_{j,i}$ represents The difference between the teacher result and the current mean result of the learners in each subject
$X_{j,k,i}$: represents the result of learner *k* in subject *j*.
$X_{j,k,i}^{new}$:is the updated value of the existing $X_{j,k,i}$.

$$TF = round[1 + r(0,1) * (2 - 1)] \quad (2.4)$$

The teaching factor (*TF*) determines the value of mean to be change (Satapathy et al., 2013). After performing several experiments on several benchmark functions, it is concluded that the efficiency of the algorithm is better when the value of TF is either 1 or 2 (Rao et al., 2011). Its value is calculated randomly by the algorithm using Eq. (2.4), so it is not an input parameter (Rao et al., 2011).

It can be observed that r and TF are both random parameters which are used for a stochastic purpose. The values of these parameters affect the performance of the algorithm (Rao et al., 2012). However, their values are calculated during the manipulation of the algorithm, and therefore do not need to be tuned. Thus, TLBO is called an algorithm-specific parameter-less algorithm (Rao et al. 2012; Rao, 2016). However, *Rao and Patel* (2012) have introduced an improved version of TLBO with the concept of an adaptive *TF* where its value is not always 1 or 2 but varies in automatically between [0,1].

Learner phase

This phase simulates learning through interactions among learners. A learner can gain knowledge through discussion and communication with another learner who has a better knowledge. For a given learner X_p, another learner X_q ,which is different from it (i.e. p $\neq$ q), is randomly chosen. The new values for learner X_p are updated as in Eq. (2.5).[1]

$$X_{j,p,i}^{new} = \begin{cases} X_{j,p,i}^{old} + r\,(X_{j,p,i}^{old} - X_{j,Q,i}^{old}) \quad \text{if } f(x_p) < f(x_q) & \text{(2.5a)} \\ X_{j,p,i}^{old} + r\,(X_{j,Q,i}^{old} - X_{j,p,i}^{old}) \quad \text{if } f(x_q) < f(x_p) & \text{(2.5b)} \end{cases}$$

where $f(x_p)$, $f(x_q)$ are the function values for learners X_p and X_q respectively. , $X_{j,p,i}^{new}$ is the updated value of the existing $X_{j,p,i}^{old}$. –The new solution is accepted if it gives a better function value, otherwise we keep the old one.

[1] The equation (4) is for minimization problems, the reverse is true for maximization.

The pseudo code for TLBO operation is illustrated in Algorithm 2.1, and the flow chart shown in figure 2.3.

Algorithm 2.1: TLBO (Zou et al., 2015)

```
Initialize N (number of learners), D (number of dimensions), and termination criteria
Generate initial population (the learners)
Calculate the fitness value for each learner
X* = the best solution
While (termination criteria is not met);
        {Teacher Phase}
        Choose the best learner as X_Teacher
        calculate the mean for each design variable
                for each learner
                        Calculate T_F using Eq. (2.4)
                        Update the existing solution according to Eqs. (2.2) and (2.3)
                end for
                Evaluated the new learners
                Accept the new solutions if it is better than the old one
        {Learner Phase}
                for each learner
                        Randomly select another learner that is different from it
                        Use Eq (2.5) to update the existing solution
                end for
                Evaluate the new learners
               Accept the new solution if it is better than the old one
        Update X* if there is a better solution
end while
Return X*
```

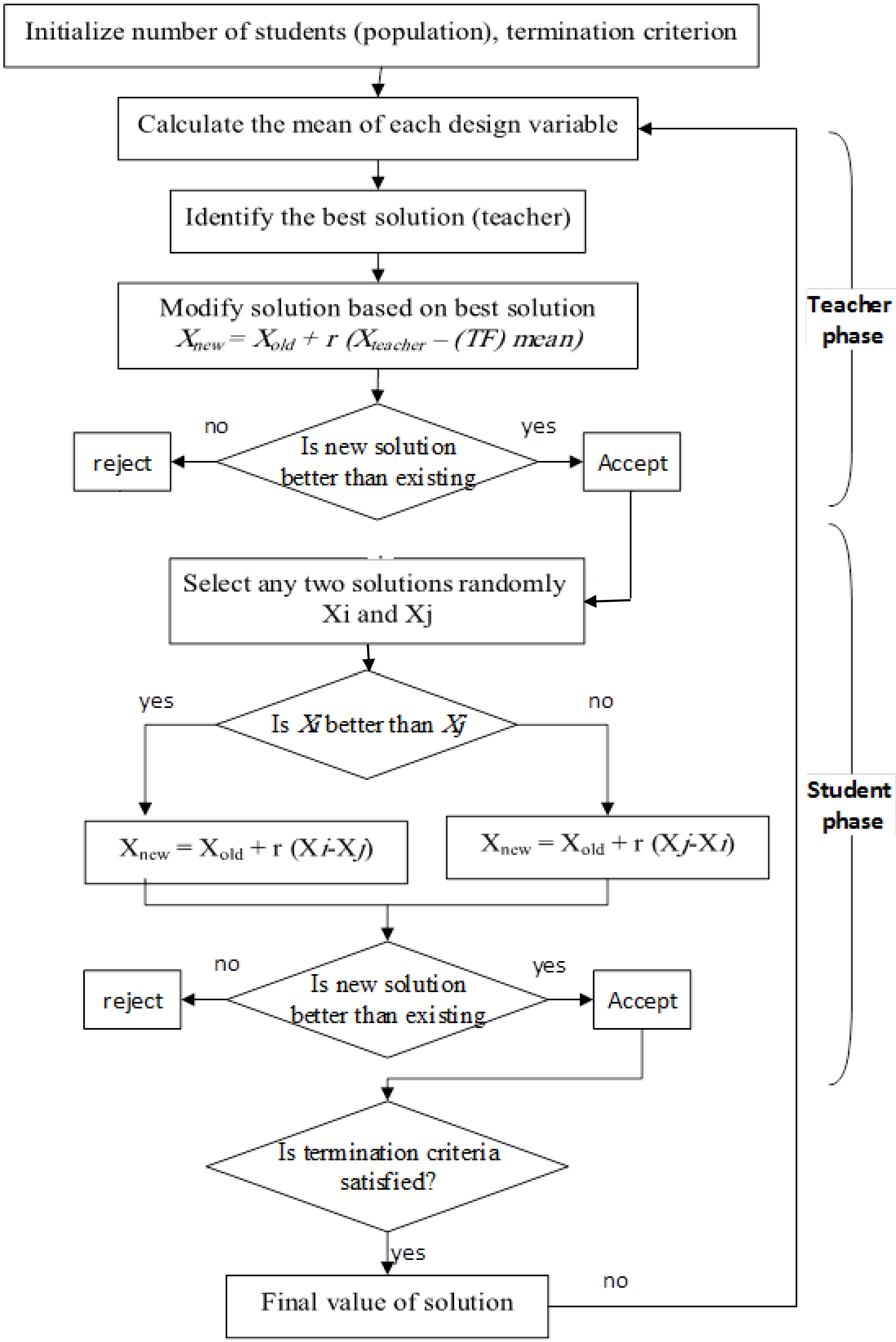


Figure 2.3: Flowchart of TLBO algorithm (Rao et. al, 2011)

2.2.1.2 Jaya Algorithm

Ventaka Rao (2016b) proposed a new optimization algorithm and called it Jaya. This algorithm is very similar to TLBO; both are classified as algorithm-specific parameter-less algorithms, but unlike TLBO, Jaya has only one phase and it is relatively simple to apply (Rao, 2016b; Pandey, 2016)

Jaya algorithm has a victorious nature (Pandey, 2016). It always tries to get closer to the best solution and tries to move away from the worst solution (Rao, 2016b). For this reason, the algorithm was named **Jaya** (which is a Sanskrit word meaning **victory**).

To illustrate the algorithm's work, suppose that we have '*m*' number of design variables (i.e. *j*=1,2,……, m), the population size '*n*' (i.e. k=1, 2, …., n). Suppose that the *best* and the *worst* respectively indicate the best solution and the worst solution obtained so far. Each variable of every candidate solution is updated using Eq. (2.6).

$$X_{j,k,i}^{new} = X_{j,k,i} + r1_{j,i}\left(X_{j,best-i} - \left|X_{j,k,i}\right|\right) - r2_{j,i}\left(X_{j,worst-i} - \left|X_{j,k,i}\right|\right) \qquad (2.6)$$

where i represents the current iteration, $X_{j,k,i}$ represents the value of the j^{th} variable for the k^{th} solution in the i^{th} iteration, $r1_{j,i}$ and $r2_{j,i}$ are two uniformly distributed random numbers in the range of [0,1] for the j^{th} variable in the i^{th} iteration, $X_{j,best-i}$ and $X_{j,worst-i}$ respectively represent the value of the j^{th} variable for the best and worst solutions. $X_{j,k,i}^{new}$ is the updated value of the existing $X_{j,k,i}$.

The new solution is accepted if it gives better function value; otherwise, we keep the old one. It is clear from Eq. (2.6) that the obtained solution always moves towards the best solution by the expression ($r1_{j,i}\left(X_{j,best-i} - \left|X_{j,k,i}\right|\right)$) and moving away from the worst solution by the expression ($-r2_{j,i}\,(X_{j,worst-i} - |X_{j,k,i}|)$) (Rao, 2016b). The absolute value of the variable is used instead of a signed variable for the exploration purpose (Rao et al., 2016). The new solution is accepted if it gives a better function value; otherwise we keep the old one. The pseudo code of Jaya is shown in Algorithm 2.2, and the flow chart is shown in figure 2.4.

Algorithm 2.2: Jaya algorithm (Pandey, 2016)

S1 Initialize
$PS \leftarrow Population\_size$
$NDV \leftarrow Number\_of\_Design\_Variables$
$TER\_COD \leftarrow Termination\_Condition$
S2 Until the termination condition not satisfied, Repeat S3 to S5
S3 Evaluate the best and worst solution
Set $best \leftarrow Best\_Solution\_Population$
Set $worst \leftarrow Worst\_Solution\_Population$
S4 Modify the solution
$X_{j,k,i}^{new} = X_{j,k,i} + r1_{j,i}\left(X_{j,best-i} - \left|X_{j,k,i}\right|\right) - r2_{j,i}\,(X_{j,worst-i} - |X_{j,k,i}|)$
S5 if (solution corresponding to $X_{j,k,i}^{new}$ better than that correspnding to $X_{j,k,i}$)
Update the previous solution
Else
No update in the previous solution
S6 Display the optimum result

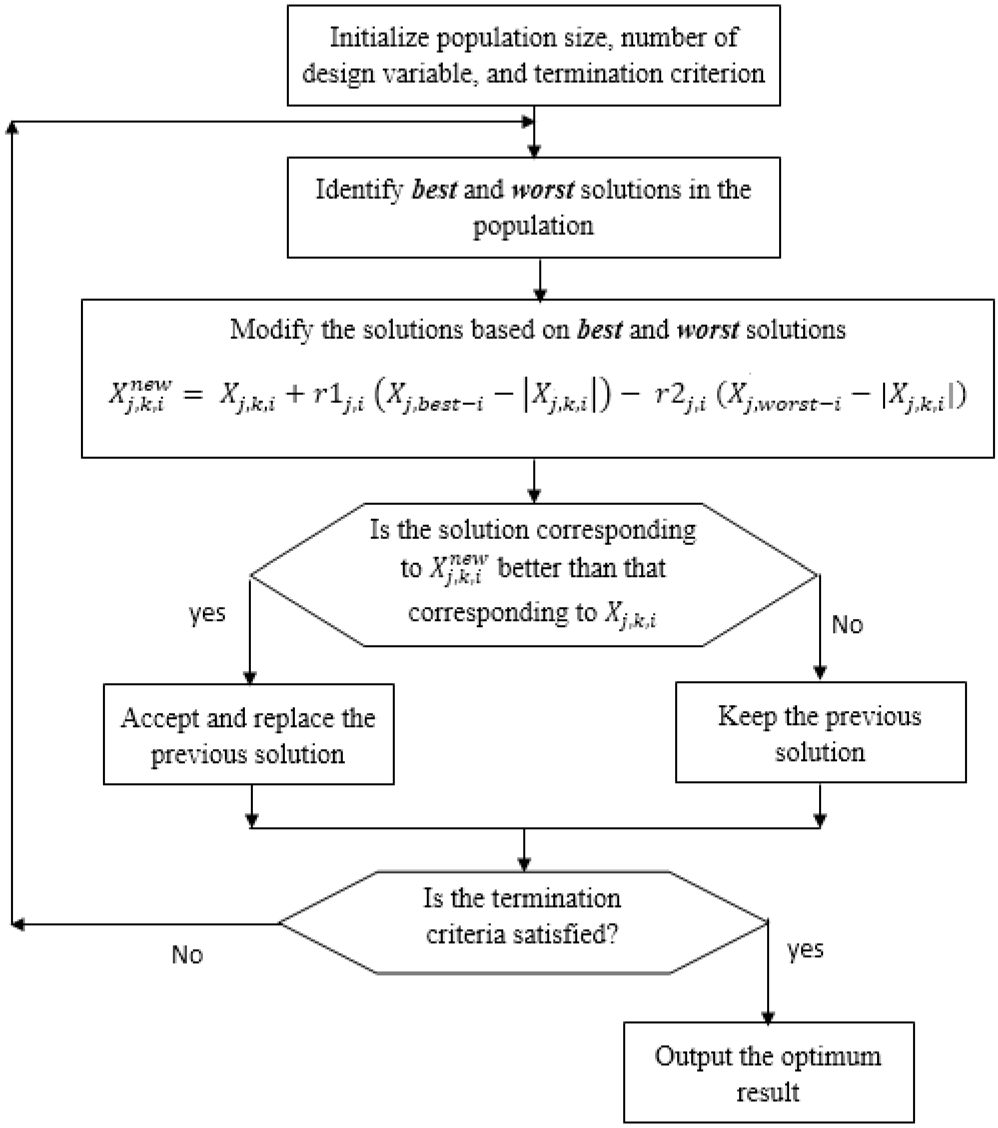


Figure 2.4: Flowchart of Jaya algorithm (Rao, 2016b)

2.2.2 Algorithms that Require Parameters

Unlike parameter-less algorithms, these algorithms require their own specific parameters in addition to the common controlling parameters like population size and the number of generations which are common in all population-based heuristic algorithms. For example, GA requires three main parameters (selection operator, mutation probability, and crossover probability); PSO requires inertia weight, cognitive, and social parameters;

ABC uses limit, and a number of onlooker bees, employed bees, scout bees; and other algorithms such as ACO, HS, DE, etc. use specific parameters (Rao, 2016).

We will briefly introduce the algorithms which were used in this research such as GA, PSO, and WTLBO in the next section.

2.2.2.1 Genetic Algorithm

Genetic algorithm is a probabilistic technique that was originally developed by John Holland in the late 1960s and early 1970s (Holland,1975). It simulates the phenomenon of natural evolution and hence it is classified within the evolutionary optimization methods (Chong & Zak, 2013).

GA is a population-based method which uses multiple solutions at the same time. It starts with an initial set of individuals that represents the candidate solutions, and it then involves a set of operations to generate a new set of individuals. These operations are called selection, crossover, and mutation (Chong & Zak, 2013).

The algorithm starts by selecting two pairs of individuals (called *parents*) according to their fitness scores. Individuals with high fitness have more chance to be selected for reproduction. The selected parents will be improved by the evolutionary operators (crossover and mutation) in the next iteration of the optimization process to form new solutions (offspring).

In the second stage, the crossover operation takes a pair of parents and recombine them to give a pair of offspring. Pairs of parents for crossover are chosen randomly from the selected group. After a crossover is performed, mutation take place by randomly changing the new offspring with a given probability. Mutation occurs to maintain diversity within the population and thus prevent premature convergence. The steps of traditional GA are shown in Algorithm 2.3 (Neumüller & Wagner, 2011).

The performance is influenced mainly by these two operators

Algorithm 2.3: GA algorithm

1: $P \leftarrow generate\ initial\ population$
2: evaluate P
3: **while** termination criteria not met **do**
4: $P_{selected} \leftarrow select\ two\ solutions\ from\ P$
5: $P_{offspring} \leftarrow recombine\ individuals\ from\ P_{selected}$
6: Mutate $P_{offspring}$
7: Evaluate $P_{offspring}$
8: $P \leftarrow P_{offspring}$ (*update population*)
9: **end while**
10: return P_0 (best solution)

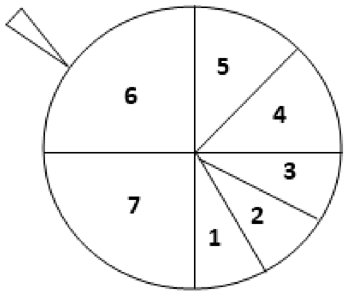
Selection Operator

There are different strategies for the selection operator which affects the convergence speed of GA (Goldberg & Deb, 1991). The common selection strategies are: roulette wheel selection, tournament selection, and rank-based selection (Talbi, 2009).

Roulette wheel selection is the most common selection method (Talbi, 2009). Each individual is assigned a probability of selection that is proportional to its relative fitness. For each individual *i*, the probability is calculated as follows:

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{2.7}$$

Where, n is the population size and f_i is the fitness of individual i. Therefore, the individual with better fitness has more opportunity to be selected as shown in Figure 2.5 (Beheshti & Shamsuddin, 2013). However, due to the possible presence of individual with high fitness that is always selected, this cause a premature convergence to a local optimum (Jebari, 2013).



individual	fitness
1	1
2	1
3	1
4	1.5
5	1.5
6	3
7	3

Figure 2.5: Roulette Wheel Selections (Talbi, 2009)

In *Tournament selection* method, a set of *k* individuals are randomly selected from the population; where *k* is the tournament group. The fittest individual is then selected after the tournament is applied to the *k* individuals (Figure 2.6). This process is repeated μ times until μ individuals are selected.

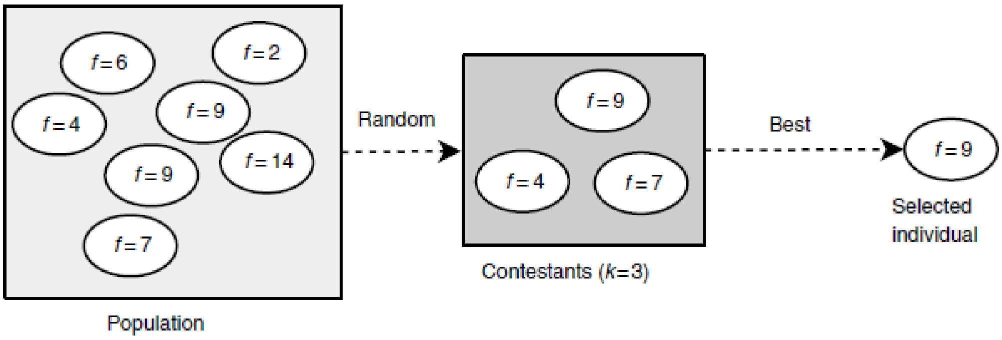


Figure 2.6: Tournament selection strategy (Talbi, 2009)

The main idea of *Rank-based selection* depends on using the rank of individuals instead of using their fitness. The best individual has rank *n* (population size) while the worst one has rank *1*. Each individual is assigned a probability of selection using the following liner formula (Jebari, 2013):

$$P_i = \frac{rank_i}{n\,(n-1)} \qquad (2.8)$$

where, n is the population size and $rank_i$ is the rank of individual i. Therefore, all the individuals have an opportunity to be selected (Beheshti & Shamsuddin, 2013) and hence reducing the problem of premature convergence (Figure 2.7).

individual	**A**	**B**	**C**
fitness	**4**	**1**	**5**

↓

rank	**2**	**1**	**3**

↓

probability	**0.34**	**0.16**	**0.5**

Figure 2.7: Linear Rank-based selection

In addition to the above selection methods, there are other methods that can be used such as exponential rank selection (Jebari, 2013), stochastic universal sampling (Talbi, 2009), competitive selection, and variable life span.

Crossover Operator

This is the first stage of evolutionary operators where a pair of parents are recombined to generate a pair of offspring. There are several methods to perform the crossover process such as one point, two points, and uniform crossover as shown in Figure 2.8 (Chong & Zak, 2013).

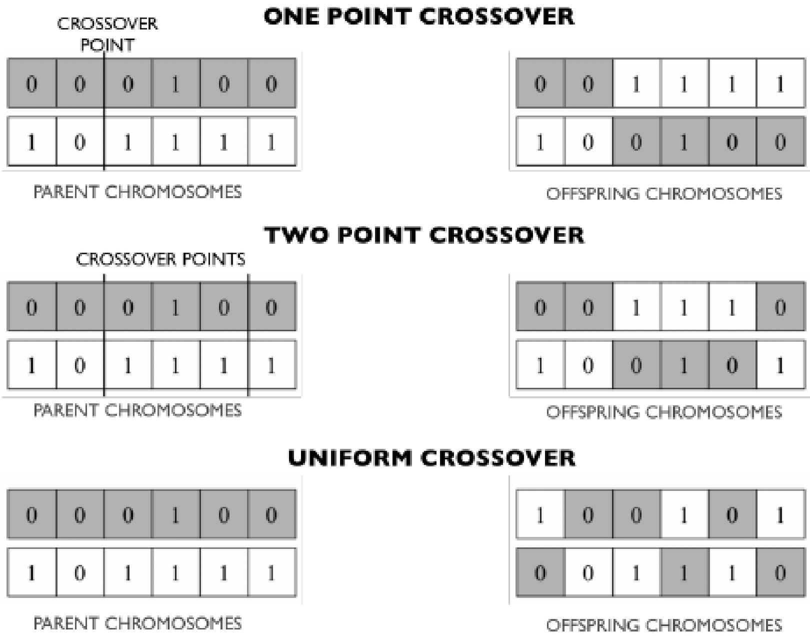


Figure 2.8: E~~e~~xample of one point, two points, and uniform crossover methods (Sastry et al., 2005)

Mutation Operator

It is the process of randomly changing some parts of individuals with a given probability. This operator helps to have better exploration process and thus escape from local optima (Mehboob et al., 2016).

2.2.2.2 Particle Swarm Optimization (PSO)

Swarm optimization is a stochastic optimization method which mimics the social behaviors of creatures that usually live in groups like bird flocking and fish schooling (Talbi, 2009). It was developed by *Kennedy and Eberhart (1995)*. PSO is a population-based optimization method, in which the population of particles is called a *swarm*. Each particle in the population is associated with two victors; position victor that represents its location according to the swarm, and the velocity that controls the direction of the next move of this particle (Luke, 2013).

During the optimization process each particle is evaluated using a fitness function, the fittest particle is denoted as global best (*gBest*), and the position that gives the best fitness value for a specific particle is denoted as a local best (*pBest*). Then, pBest (self-experiences) and gBest (social experiences) are used to update the position of the current particle hoping to get a better position than the current one (Garcia-Nieto et al, 2013). Each dimension of the velocity component is updated according to Eq. (2.9), while each dimension of the particle position is updated according to the Eq. (2.10) (Kennedy and Eberhart, *1995*)

$$xv_i = \underbrace{w}_{\text{Inertia wight}} * \ xv_i + \underbrace{cp * r * (pbest_i - \ x_i)}_{\text{self-experience}} + \underbrace{cg * r * (gbest_i - \ x_i)}_{\text{social-experience}} \qquad (2.9)$$

$$x_i = \ x_i + \ xv_i \qquad (2.10)$$

where:

xi: the i^{th} dimension of particle position

xv_i: the i^{th} dimension of the velocity component

r: a uniformly distributed random real number within [0, 1].

$pbest_i$: particle best value found so far of dimension i

$gbest_i$: global best value found so far of dimension i

w, cp, cg: tunable parameters. w (inertia weight), cp (weight of local information), cg (weight of global information)

In Eq. (2.9) The inertia weight parameter (*w*) controls the balance between exploration and exploitation. A smaller value of *w* assists the local exploitation, while a larger value of *w* encourages the global exploration (Kennedy, 1997; Beheshti & Shamsuddin, 2013). Therefore, this parameter has received increased attention in the research by introducing a dynamically adjusted inertia weight using different updating mechanisms such as linear and nonlinear decreasing methods (Arasomwan & Adewumi, 2013; Alkhraisat & Rashaideh, 2016).

The work of PSO can be summarized in Algorithm 2.4 (Kennedy & Eberhart, *1995*).

Algorithm 2.4: PSO algorithm

---

1. $initia(\theta)$ *// initial swarm usually random*

2. *for each particle* $x \in \theta$*:*
   for each dimension i
      // calculate velocity according to equation (2.9)
      // update particle position according to equation (2.10)
3. *While stop criteria not reached, Go to step 02*

2.2.2.3 Weighted Teaching-Learning Based Optimization

Satapathy et al (2013) proposed an improved version of traditional TLBO algorithm to improve the convergence speed. The authors added A new parameter called (weight) to the learning equations of TLBO, and hence the new algorithm was called Weighted TLBO (WTLBO). The principle of adding a new parameter was based on the natural phenomena of the learner's brain in forgetting the lessons learned in the last session. The value of the weight parameter (*w*) is linearly reduced from w_{max} to w_{min} according to Eq. (2.11).

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{max - iteration}\right) * i \tag{2.11}$$

Where w_{-max} and w_{-min} are a predetermined maximum and minimum values respectively, *max-iteration* is the maximum number of iterations, ***i*** is the current iteration. Hence, the learning equations (2.4) and (2.5) in TLBO become as following:

$$X_{j,k,i}^{new} = \mathrm{w} * X_{j,k,i}^{old} + Difference\_Mean_{j,i} \tag{2.12}$$

$$* X_{j,p,i}^{new} \begin{cases} w * X_{j,p,i}^{old} + r\,(X_{j,p,i}^{old} - X_{j,Q,i}^{old} \text{ if } f(x_p) < f(x_q) & (2.13a) \\ \\ w * X_{j,p,i}^{old} + r\,(X_{j,Q,i}^{old} - X_{j,p,i}^{old}) \text{ if } f(x_q) < f(x_p) & (2.13b) \end{cases}$$

WTLBO algorithm was compared to TLBO, PSO, DE algorithms using several benchmark functions. The results showed that WTLBO is faster than other algorithms (Satapathy et al, 2013).

2.3 Conclusion

Metaheuristic optimization techniques are suitable to solve complex and hard problems which cannot be solved by traditional optimization methods. They do not guarantee the optimal solution but they can find good solutions in a reasonable time even in large spaces of solutions. Many algorithms have been developed, some of which are suitable for solving a specific type of problems while the others are not. However, According to No-Free-Lunch (NFL) theorem, there is no optimization algorithm that is good enough to be suited for all optimization problems (Wolpert & Macready, 1997).

2.4 Modeling and Simulation of Traffic Systems

2.4.1 Introduction

A traffic system is a complex, dynamic and adaptive system. It consists of a number of interacting agents such as vehicles, pedestrians, traffic lights and some other sub-systems which lead to emergent outcomes that are often difficult (or impossible) to be predicted. (López-Neri et al., 2010).

Traffic conditions depend on the integrated and complex relationships between various variables such as passengers' behaviors, road laws, weather conditions, infrastructure, and other unpredictable conditions. Traffic cannot be described just by departure times and paths used during a period of time. It depends heavily on the travelers' behavior. (Krajzewicz et al., 2002).

This complexity makes it difficult to describe traffic using mathematical formulas. Therefore, there is no closed mathematical form that can be used as a model which is capable of describing all the stochastic behavior of the traffic system components (Krajzewicz et al., 2002). So, simulation is characterized as a powerful and cost-efficient tool to design, analyze, evaluate roads and to develop plans and proposals for their improvement. (Olstam, & Tapani, 2004)

Nowadays, the availability of data and the high processing power of computers makes it easier for researchers to simulate road networks much faster than real environment and thus an experiment that is conducted using simulations yields results in much less time than the same experiment when conducted in reality. (Bazzan & Ana, 2007; Kotushevski & Hawick, 2009).

Many model-based simulation packages such as VISSIM (PTV AG, 2015), CORSIM (FHWA, 2006), AIMSUN (Barceló, & Casas, 2006), PARAMICS (Ozbay et al., 2005) and SUMO (Krajzewicz et al., 2012) have been developed for traffic.

Traffic models can be classified based on several properties: Scale of independent variables (discrete, continuous and semi-discrete), level of details (microscopic, sub-microscopic, macroscopic, mesoscopic), the scale of applications (networks, stretches, links, intersections), representation of the processes (deterministic, stochastic) (Hoogendoorn & Bovy, 2001). The detail-level classification is commonly used because it specifies important criteria to be considered when choosing a traffic model such as accuracy, computation time, ability to achieve the objective, and suitability for large networks. In the following section, we discuss the modeling approaches based on the level of details.

2.4.2 Traffic Modeling Approaches Based on the Level of Details

In traffic flow models, there are different approaches to simulation models which are classified based on the level of details through which the system components are described. These models are macroscopic, microscopic, mesoscopic and sub-microscopic models (Hoogendoorn & Bovy, 2001; Abdalhaq & Abu Baker, 2014). The four approaches are represented in Figure 2.9.

2.4.2.1 Microscopic Models

The *microscopic* traffic flow model simulates the behavior of each individual vehicle-driver unit and its interactions with other vehicles in the street. This model is concerned with describing the network accurately and in details (Ehlert et al., 2017). The dynamic variables of the models represent microscopic properties like the position, velocity, and acceleration of single vehicles. Hence, a high computation time is needed to evaluate these parameters (Abushehab et al., 2014). This model assumes that there are two factors which determine the behavior of the vehicle: the vehicle's physical abilities to move and the driver's controlling behavior (Chowdhury et al., 2000).

2.4.2.2 Macroscopic Models

The macro-simulation has founded under the assumption that traffic streams are comparable to the fluid stream. Therefore, it ignores the behavior of the individual vehicle and concerns only with the traffic flow in a road network using aggregated quantities such as flow, density, and average speed (Mccrea & Moutari, 2010; Mitsakis et al., 2014). The lack of details used to describe the traffic system makes this model less complex than microscopic model, and therefore less computational time. It is also relatively easy to implement and allows users to execute several scenarios in a short time Therefore, in general, it is the most suitable for modeling large networks in real time or even faster (Olstam, & Tapani,2004; Burghout, 2004). However, the main drawback of this model is the lack of accuracy which limited its application in the cases where the interaction of vehicles is not crucial to the results of simulation (Olstam, & Tapani,2004)

2.4.2.3 Mesoscopic Models

The *mesoscopic* model combines the characteristics of the two previous models. It describes the traffic using both levels: the aggregate level of macroscopic models and the individual interactions behavior of microscopic models (Burghout, 2004). This model approximates the positions and behavior of vehicles but less accuracy than microscopic model (Olstam, & Tapani,2004). These models can be represented in several forms. One of these forms is a queue-server form (Mahut, 2001).

2.4.2.4 Sub-microscopic Models

The last class model of traffic simulation models is *sub-microscopic*. This model is similar to the microscopic one, but it describes more details about the vehicle-driver unit like the engine's rotation speed in connection with the vehicle speed or the driver's favored gear. However, this model needs longer computation time compared to simple microscopic model and therefore it is suitable for small networks (Krajzewicz et al., 2002; Hoogendoorn & Bovy, 2001).

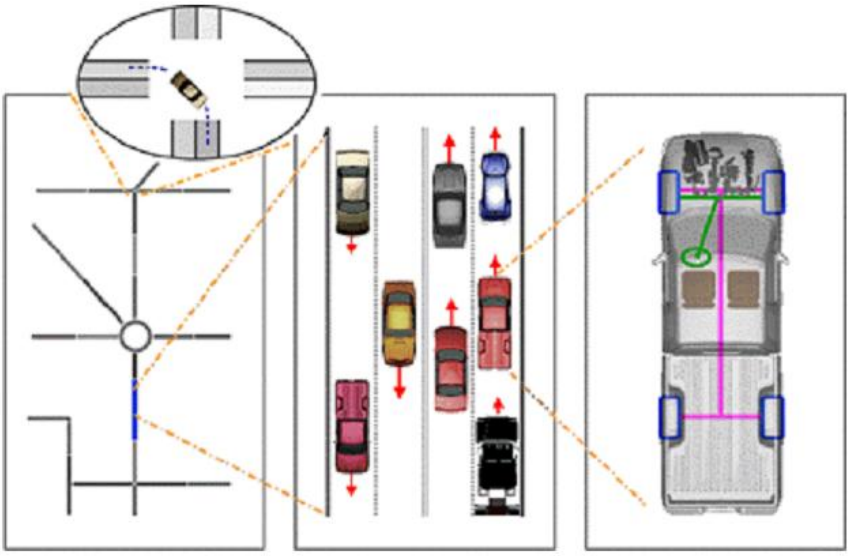


Figure 2.9: The different simulation granularities; from left to right: macroscopic, microscopic, sub-microscopic, within the circle: mesoscopic. (SUMO user documentation)

Although macroscopic and mesoscopic models are simpler and faster than microscopic models, their use is limited to certain cases where the interaction of individual vehicles is not decisive to the desired results. For example, they are inappropriate to analyze the merging areas. Besides, the accurate modeling of the adaptive signal control can be difficult in both macroscopic and mesoscopic models because when the positions of the vehicle are not known (i.e. macroscopic) or inaccurate (i.e. mesoscopic) it is difficult to simulate the activations of detectors used in the adaptive control system (Olstam, & Tapani,2004).

Moreover, the availability of data and high-performance computing environment makes the use of microscopic simulators less challenging to model large-scale networks accurately. For these reasons, we have used a microscopic traffic simulator (called SUMO) in this work.

2.4.3 SUMO Simulator:

"Simulation of Urban Mobility" (SUMO) is a microscopic road traffic simulation package which is available as an open source under the GPL License since 2001 (Krajzewicz, 2010). It was developed by the Institute of transportation systems at the German Aerospace Center (DLR). The main objective of developing SUMO was to provide researchers and engineers in the field of traffic with a tool to propose plans, implement and evaluate their own algorithms. SUMO is a multimodal, space-continuous and time-discrete simulation platform (DLR and contributors, n.d). See Table 2.1 for the main features of SUMO. (Abdalhaq & Abu Baker, 2014)

Table 2.1: SUMO features (Abdalhaq & Abu Baker, 2014)

Category	Features
Simulation	Complete workflow (network and routes import, DUA, simulation) Simulation
	Collision-free vehicle movement
	Different vehicle types
	Multi-lane streets with lane changing
	Junction-based right-of-way rules
	Hierarchy of junction types
	A fast OpenGL graphical user interface
	Manages networks with several 10.000 edges (streets)
	Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
	Interoperability with other application at runtime using Traci
	Network-wide, edge-based, vehicle-based, and detector-based outputs
Network	Many network formats (VISUM, Vissim, Shapefiles, OSM, Tiger, RoboCup, XML-Descriptions) may be imported
	Missing values are determined via heuristics
Routing	Microscopic routes - each vehicle has an own one
	Dynamic User Assignment
High portability	Only standard c++ and portable libraries are used
	Packages for Windows main Linux distributions exist
	High interoperability through the usage of XML-data only

SUMO as an open source software is widely used and popular because its source code is available for research, study, and modifications. This feature provides an additional help and a continuous support from other contributors (Kotushevski & Hawick, 2009).

Various sub-models were implemented in SUMO; each has a specific task in the simulation. These models are the car following Krauss model (Krauss,1998), lane change Krajzewicz model (Gawron,1998), route choice model, user assignment model and the traffic light model. SUMO is not only for traffic simulation, but it is a software package which includes several applications based on their purpose (i.e. network generation, demand generation, and simulation). This helps to prepare and perform the simulation of a traffic scenario. The main applications that are included in SUMO are listed in Table 2.2. (Krajzewicz et al., 2012)

Table 2.2: Main applications included in SUMO (Pattberg,n.d.)

Purpose	Application Name	Short Description
Simulation	SUMO	The microscopic simulation with no visualization; command line application
	SUMO-GUI	The microscopic simulation with a graphical user interface
Network generation	NETCONVERT	Network importer and generator; reads road networks from different formats and converts them into the SUMO-format
	NETEDIT	A graphical network editor.
	NETGENERATE	Generates abstract networks for the SUMO-simulation
Vehicles and Routes	DUAROUTER	Computes fastest routes through the network, importing different types of demand description. Performs the DUA
	JTRROUTER	Computes routes using junction turning percentages
	DFROUTER	Computes routes from induction loop measurements
	MAROUTER	Performs macroscopic assignment
	OD2TRIPS	Decomposes O/D-matrices into single vehicle trips
	POLYCONVERT	Imports points of interest and polygons from different formats and translates them into a description that may be visualized by **SUMO-GUI**
	ACTIVITYGEN	Generates a demand based on mobility wishes of a modeled population

SUMO is a microscopic simulation of vehicular traffic. Each vehicle behavior is simulated individually, and defined at least by a unique name, departure time, and the vehicle's route through the network. Moreover, the vehicle can be described in more details such as speed, position, type, and the amount of pollution or noise emission. See (Krajzewicz et al, 2012). These details are required in this research to achieve the desired simulation output (i.e. calculate the average travel time for vehicles). So, for the achievement of our study’s objective, a microscopic simulator was selected instead of a macroscopic one.

3. Literature Review

3.1 Traffic Lights Timing Optimization

The timing of traffic signals in roads and intersections has a significant impact on congestion. The correct scheduling for the duration of green and red lights is one of the most cost-effective techniques for facilitating the mobility within the urban traffic system. (Schneeberger & Park,2003)

Finding the proper duration of traffic lights phases is a complex optimization problem due to the unstable and random behavior of the urban traffic process (Sklenar et. al, 2009; Hu et. al., 2015). In addition, the complexity of the problem depends on the size of the network and the number of traffic lights. Hence, it could be difficult to solve such an optimization problem by -traditional mathematical optimization techniques (Damy, 2015).

The research on traffic signal optimization has been conducted since the early 1960s (Lu, 2015). In 1967, traffic was monitored using digital-computers installed in several cities (Denos & Gazis, 1967). Research is ongoing in this area to find innovative ways and to implement new algorithms to solve traffic signal timings optimization.

Many researches have been conducted to tackle the TSOP where different approaches have been used, including mathematical optimization models, and simulation-based approaches integrated with metaheuristic optimization techniques. (Warberg et al., 2008)

3.1.1 Mathematical Optimization Models

In the late 1950s, Webster has developed the principle of traffic signal optimization methodology for isolated intersections (Webster, 1958). He has developed a single intersection mathematical model for estimating the delays for vehicles at fixed-time traffic signals and for computing the optimum settings of such signals to minimize the overall vehicular delay.

Many researchers then have proposed mathematical optimization models for traffic signal timing, such as *Miller* (1963), *Gazis* (1964), *DAns* and *Gazis* (1975), *Michalopoulos* and *Stephanopoulos* (1978), *Akcelik* (1981), *Lieberman et al* (2000), *Ceder and Reshetnik* (2001), *Li* (2010), *Jiao and Sun* (2014). (Jiao, Z. Li, Liu, D. Li, & Y. Li, 2015). The main weakness in the use of mathematical models in this area is that it was used to optimize junctions as isolated units. (Abushehab et al, 2014)

3.1.2 Simulation-based Approaches

The traffic system is complex and random, so simulation is the most effective way of analyzing the different problems and gathering quantitative information about traffic system that changes dynamically (Olstam, & Tapani, 2004). Research studies about traffic simulation focused on two types of simulation models: macroscopic and microscopic models (Garcia-Nieto et al, 2013).

3.1.2.1 Off-line Optimization Tools

Off-line optimization tools are software packages which are based on historical data about traffic flow and therefore the scheduled time remains constant and does not change depending on the variety and stochastic aspects of traffic flow (Lu, 2015).

A variety of software packages have been developed to optimize traffic signal timing plans, such as SYNCHRO (Husch & Albeck, 2006) which is the most common software package used locally by municipalities, TRAffic Network Study Tool (TRANSYT) (Hale, 2005), Progression Analysis and Signal System Evaluation Routine (PASSER) (PASSER V, 2002), and the Traffic Software Integrated System - Corridor Simulation (TSIS/CORSIM) **(**Kaman Science Corporation, 1996**).**

These programs consist of two main parts: an optimizer that uses an optimization technique to search for the optimal settings which improve the system performance. In addition to a traffic simulation model, which is used to evaluate and assess the objective functions during the optimization process. (Álvarez & Hadi, 2014*)*.

TRANSYT, SYNCHRO, and PASSER are based on embedded macroscopic simulation models (Álvarez & Hadi, 2014), while

TSIS/CORSIM is based on a microscopic simulation model (Lu, 2015). The use of deterministic and macroscopic simulation-based signal optimization methods could lead to trap at the local optimum or even not good solution (Schneeberger & Park,2003). In addition, macroscopic models are limited in describing the behavior of each individual vehicle-driver unit and its interactions with other vehicles in the street.

Rouphail et al (2000) study indicated that the performance of the microscopic simulation-based approach is much better than the macroscopic simulation-based approach to solve the traffic light timing optimization problem (Schneeberger & Park,2003).

3.1.2.2 On-line Optimization Tools

Because urban traffic contains a variety of stochastic behaviors and time to time demand variations, some adaptive and real-time traffic control systems have been developed to adjust the traffic signal settings automatically to adapt to traffic conditions. Examples of these systems are Split Cycle and Offset Optimization Technique (SCOOT) (Robertson, & Bretherton, 1991), Sydney Coordinated Adaptive Traffic System (SCATS) (Lowrie,1982), Optimized Policies for Adaptive Control (OPAC) (Gartner, 1990), Real-time Hierarchical Optimized Distributed and Effective System (RHODES) (Mirchandani, & Head, 2001.), Method for the Optimization of Traffic Signals in Online Controlled Networks (MOTION) (Busch, & Kruse, 2001), and Balancing Adaptive Network Control Method

(BALANCE). However, there are other control systems in addition to the mentioned examples, yet SCOOT and SCATS are the most widely used internationally. (Jiao et al, 2015; Lu, 2015)

For more details about the components and the mission of each optimization tool, and the difference between them, look at (Lu, 2015; Ratrout, & Reza, 2014)

3.2 Review of TLBO and Jaya algorithms

TLBO and Jaya algorithms have been widely used in different real-world applications of engineering and science and have showed effectiveness in problem-solving (Rao, 2016a, 2016b). Table 3.1 presents examples of recently published papers related to TLBO and Jaya algorithms (Rao, 2016a).

Table 3.1: Recently published papers related to TLBO and Jaya

#	Algorithm	Authors	Year	Description
1	TLBO	Zou et al.	2015	An improved TLBO algorithm (LETLBO) with learning experience of other learners has been introduced.
2	TLBO	Yu et al.	2015	A self-adaptive multi-objective TLBO (SA-MTLBO) has been proposed.
3	TLBO	Xu et al.	2015	Proposed an effective TLBO algorithm to solve the flexible job shop scheduling problem.
4	Jaya	Rao et. al	2016	Dimensional optimization of a micro-channel heat sink using Jaya algorithm
5	TLBO	Qu et al	2017	An improved TLBO based memetic algorithm for aerodynamic shape optimization
6	Jaya	Rao & More	2017	Optimal design and analysis of mechanical draft cooling tower using improved Jaya algorithm
7	Jaya	Rao & Saroj	2017	A self-adaptive multi-population based Jaya algorithm for engineering optimization
8	TLBO	Kumar et. al	2018	A hybrid TLBO-TS algorithm for integrated selection and scheduling of projects
9	Jaya	Zhang & luo	2018	Parameter estimation of the soil water retention curve model with Jaya algorithm
10	Jaya	Sudhakar & Inbarani	2018	Intelligent Path Selection in Wireless Networks using Jaya Optimization
11	TLBO	Kiziloz et. al	2018	Novel multiobjective TLBO algorithms for the feature subset selection problem
12	Jaya	Ravipudi & Neebha	2018	Synthesis of linear antenna arrays using Jaya, self-adaptive Jaya and chaotic Jaya algorithms

3.3 Heuristic Optimization Techniques for TSOP

Metaheuristic optimization techniques have become popular in the field TSOP (Garcia-Nieto et al, 2013). Many well-known heuristic algorithms such as GA, PSO, TS, ACO, SA, HS have been used. However, the most common algorithm in this field is GA (Lu, 2015; Abushehab et al, 2014).

The following researchers have contributed to optimize the timing of the traffic signals. We classified them according to the algorithm(s) used.

3.3.1 Genetic Algorithm

Rouphail et al. (2000), discussed a strategy based on the integration between CORSIM microscopic simulator and the GA optimizer for the timing optimization of nine signalized intersections in the city of Chicago (USA). The outcomes gained from the proposed approach were compared to the outcomes of traditional signal optimization (TRANSYT-7F) after applying them to the study network. The authors concluded that the GA outperform TRANSYT-7F.

Schneeberger and Park (2003) evaluated SYNCHRO, TRANSYT-7F programs and the GA for traffic signal optimization. The case study was a network with 12 signalized intersections in Northern Virginia. A microscopic simulation model (VISSIM) was used to represent the case study. They tuned VISSIM parameters to ensure that the collected data were accurately represented. Five timing plans were investigated on the tuned VISSIM model. These plans were optimized timing plan from SYNCHRO, TRANSYT-7F, GA, in addition to the VDOT's former and current timing plans. As a result, the performance of the current VDOT's timing plan outperformed the other timing plans.

Farooqi et al. (2009) proposed their own traffic light simulator which is called THE to test the optimization algorithms that require chromosome encoding. They used GA to optimize the signals’ timing for a road network of 16 traffic lights, and after evaluating 10 chromosomes for 10 generations, the total waiting time for the cars was reduced efficiently from a random assignment of time.

Singh et al. (2009) proposed a real-time control methodology for traffic signals. They developed a traffic emulator using JAVA to represent the adaptive traffic conditions. It consisted of a four-legged isolated intersection with four traffic lights. The system was the real-time decision maker whether to extend the green time or not. They used GA with both 100 and 6 generations to find the optimal green time extensions that maximize the throughput. The new system was compared with the traditional fixed time traffic system. Based on the results obtained, they showed that the number of exit vehicles in the real-time system was larger than the fixed-time system, and thus a significant performance increases to 21.9 % in case of a real-time based system.

Qian et al. (2013) presented a traffic signal timing model with GA (AARGA) for optimizing the pollutant emission for isolated intersections. Shenzhen Lianhua- Xinzhou signal control intersection was selected to validate the proposed model and optimization algorithm. The obtained results indicated the effectiveness of using the presented algorithm.

Damay (2015) proposed a computational framework based on the SUMO microscopic simulator integrated with a tuned multi-objective GA (MOGA). The main aim of the study was to optimize the duration of green light phases and thus minimizing the total waiting time and the total pollution emissions. The proposed method was tested on a real network in the city of Rouen, France which contained 11 intersections, 168 traffic lights, and 40 possible turning movements. Furthermore, the author tuned the demand-related model of SUMO simulator to make the behavior of the simulation environment as closer to the real one as possible by using several algorithms: the Gradient Search Method (GSM), the Stochastic Search Method (SSA) where GA was used, and a hybrid algorithm called the Memetic Search Algorithm (MSA) which combined both the GSM and the SSA. The gained results demonstrated that MOGA algorithm was appropriate to optimize traffic light timing for a medium-sized network. Also, the hybrid algorithm MSA achieved satisfactory results for a medium-sized network.

3.3.2 Simulated Annealing

Sklenar et al. (2009) tried to optimize the traffic light time of three junctions at Konečného square in Brno, Czech Republic. The objective was to minimize the average waiting time in the queues of the system. To evaluate the objective function, they built a simulation model of Konečného square in Java using SSJ (Stochastic Simulation in Java - a Java library for stochastic simulation) and implemented Simulated Annealing algorithm (SA) for optimization. The obtained results were compared to VISSIM model provided by BKOM and they showed a good improvement.

3.3.3 Particle Swarm Optimization

In *Kachroudi and Bhouri* (2009), a predictive control strategy based on private and public vehicles models was used. The major objectives of the study were to improve the overall traffic conditions and to enable public transportation vehicles to move according to their schedules. Two versions of multi-objective PSO algorithm were applied for optimizing cycle programs. These versions were the original PSO and the modified algorithm GCPSO. To evaluate the strategy, a virtual urban road network made up of 16 signalized intersections and 51 links was used. The results exhibited that the proposed strategy is effective in achieving the wanted objectives.

Garcia-Nieto et al. (2013) proposed an optimization method based on PSO with the objective of optimizing the cycle programs of all traffic lights which lead to maximize the number of vehicles that reach their destinations and minimize the global trip time of all vehicles. The authors simulated large road networks with hundreds of traffic lights located in the cities of Sevilla and Malaga (in Spain) using a microscopic traffic simulator called SUMO. To validate the proposed method, they compared the obtained results against two methods: a random search algorithm and the SUMO cycle programs generator (SCPG). As a result, they concluded that PSO performance is better in terms of the throughput (the number of vehicles that actually leaves the network) and the global trip time than the two other compared algorithms.

Hu et al. (2015) presented a real-time optimization approach to schedule the traffic light in the large network using Inner and Outer cellular automaton integrated with Particle Swarm optimization (IOCA-PSO). The proposed method was compared to three methods: PSO, GA, and RANDOM method tested on a real urban network of Wuhan (China). The final results manifested that IOCA-PSO performance is better than other tested methods under different traffic conditions.

Zhao et al. (2016) employed the PSO algorithm for traffic signal optimization. The intersection of Huangshan road and Kexue Ave. in Hefei (China) was tested to find the optimal phase combination that minimizes the number of stops. The experiment results showed that that PSO method improved the traffic by decreasing the number of stops about 19.04% and thus reducing the total delay and CO emission.

Liang et al. (2017) proposed a method to optimize the overlapping phase combination for an isolated intersection. The objective was to minimize the total delay. First, the best group of possible phase combination was selected, then PSO method was used to optimize the green time for each phase in the selected group. The intersection of Xiuning Road and Hezuohua Road in Hefei (China)was chosen to examine the proposed method. At the end of the study, the reported results displayed a good improvement.

3.3.4 Ant Colony Optimization Algorithm

Renfrew and Yu (2009) research investigated the application of Ant Colony Optimization (ACO) to find the optimal signal timing plan that minimizes the delay average of the vehicles at an isolated intersection. ACO is used with a rolling horizon algorithm to achieve a real-time adaptive control. The intersection that was chosen to examine the algorithm was simple; only 2 phases and without turning lanes. Two variants of ACO algorithm were used, the Ant System (AS), and the Elitist Ant System (EAS). The simulation results indicated that the proposed approach was more efficient than traditional fully actuated control.

Jiajia and Zai'en (2012) used Ant Colony Algorithm (ACA) to optimize an objective function related to the cycle time and the saturation of an intersection. They used time delay, number of pauses and traffic capacity as a performance index. The performance of ACA algorithm was compared with Webster algorithm and GA. The ACA was founded to be effective and feasible in solving the signal timing optimization problem.

3.3.5 Harmony Search Algorithm

Ceylan Huseyin and *Ceylan Halim* (2012) solved the traffic signal settings in the Stochastic EQuilibrium Network Design (SEQND) by using Hybrid Harmony Search and Hill Climbing with TRANSYT (HSHCTRANS) model. In the proposed model, the local search method (HC) was used for fine-tuning the solution of global search method (HS). The proposed model was compared to HS and GA-based models. The gained results showed that HSHCTRANS performance is better than HS and GA-based models.

Dellorco et al. (2013) presented a bi-level methodology that combines traffic assignment and the traffic signal control to solve the Equilibrium Network Design Problem (ENDP). At the upper level, Harmony Search Algorithm (HSA) was used to optimize the traffic light timing, so to examine the effectiveness of HAS so that to solve the upper level of ENDP. The authors tested the performance of HSA, GA, and HC by calling TRANSYT-7F on a two junction network. It was found that

HAS was better than HC and GA, and thus the applicability of HAS to solve the traffic signal timing of ENDP problem.

Gao et al. (2016) proposed a scheduling framework for the urban traffic light control. Their methodology was based on Discrete Harmony Search (DHS) combined with three local search operators for optimization. Many computational experiments were conducted on a partial network in Singapore which was represented by a dynamic traffic flow model based on Daganzo’s cell transmission models. To evaluate the proposed algorithm, comparisons were made between the Fixed Cycle traffic control System (FCS) and the DHS before and after local search operators. It was found that the improved DHS is better than the standard DHS and FCS.

3.3.6 Multiple algorithms

The methodology of *Yun and Park* (2006) was based on the use of CORSIM microscopic traffic simulation model and heuristic optimizer. They investigated the performance of three optimization methods (i.e., GA, SA, and OptQuest Engine) on a real network of urban corridor in Fairfax, Virginia, the USA which contains 12 intersections with 82 traffic signals. The performance of the previous methods was compared with SYNCHRO optimizer under a microscopic simulation environment. The gained results exhibited that GA is better than SYNCHRO and the other two optimization methods presented in the study.

Abushehab et al. (2014) used a random optimization technique and nine metaheuristic algorithms (3 types of GA, PSO, and 5 types of TS algorithm) to optimize the traffic light signals timing for Nablus city center road network which contains 13 traffic lights. The objective function was to minimize the (ATT) for vehicles. A microscopic simulator called SUMO (Simulation of Urban Mobility) was used to simulate the case study and evaluate the objective function. They tuned the values of each algorithm parameters using *Rastrigin* benchmark function and hence determined the best parameters' values to solve the problem. They validated the obtained results by comparing the average results of optimization algorithms before and after tuning the parameters and also compared with the results of Webster, HCM methods, and SYNCHRO simulator. Furthermore, they conducted many experiments and found that benchmark iterative approach is suitable to determine the best parameters' values for algorithms and that the metaheuristic algorithms are better than traditional and mathematical models to optimize traffic light timing. The most efficient algorithms to solve the problem were GA Type 3, PSO (w=0.25, cg=3.5, and cp=1.25) and TS Type 5 (tau=10).

3.4 Other Approaches

Lu (2015) proposed a novel real-time methodology to optimize traffic signal timing for large network. The proposed approach was a hierarchical control system consisting of two levels: the upper level is for macro control strategies and the lower level is for micro parameters computations. So, two strategies were applied in the upper level. First, a network partition strategy in which the network was partitioned into smaller sub-networks based on the intersections' priority order computed by the sort model of priority order (SMoPO), TRANSYT tool was used to find the optimal order. Second, the network signal coordination strategy which makes the optimization problem much simpler. In the lower level, both cyclic flow and cyclic delay were used to propose a method for optimal relative offsets' estimation. A virtual network with 64 intersections and two real networks located in Braunschweig city (Germany) were simulated using SUMO simulator to test the proposed approach. The obtained simulation results showed that the proposed approach outperformed Webster method in terms of mean delay time, mean fuel consumption and mean PMx emission.

Jiao et al. (2015) proposed a multi-objective signal optimization model to improve the travel of people by minimizing the average of delay time per person and queue length. The proposed method is different from other methods because it aims to minimize the average of delay time per person instead of the delay of vehicles. VISSIM simulator was used as a tool to evaluate the model, which was coded using M language based on MATLAB. The proposed model tested on a real intersection in Beijing, China. The simulation experiments results showed the effectiveness of the proposed method.

Other techniques were applied to improve the traffic optimization problem such as fuzzy logic. Iscaro et al. (2013) speeded up the optimization process by using a set of fuzzy rules to detect the problem on the intersection before running the optimizer which was based on GA and SUMO simulator.

3.5 Summary of Literature Review

Different approaches have been proposed to solve the TSOP. Some mathematical optimization models have been developed based on Webster and HCM models. The road networks have become complex and dynamic, so most researchers turned to develop simulation-based approaches. Several off-line computer optimization tools have been developed like TRANSYT, SYNCHRO, and PASSER. To suit the stochastic behavior and time to time demand variations of traffic, an adaptive and real-time control systems like: SCOOT, SCATS, and OPAC were presented.

In the optimizer of traffic control system, the employed optimization technique applied plays an important role in determining the efficiency of the proposed approach. Metaheuristic optimization algorithms have become popular in the field of traffic signal timing. Most well-known heuristic algorithms have been applied applied, including GA (the most popular), PSO, TS, ACO, SA, and HS algorithms. However, none of the researchers have tested the modern TLBO and Jaya algorithms to optimize the traffic signal timing. Table 3.2 summarizes the previous studies that have tested metaheuristic methods to optimize the traffic signals in terms of evaluated algorithms, study area, simulation tool, optimized parameters and the objective function, where "positive" means that the conducted optimization method has been successful to improve the traffic conditions compared to traditional fixed time methods. While "negative" means that the optimization method did not give better timing plans than plans current.

Table 3.2: Summary of heuristic algorithms for traffic signal optimization

Optimization Methods	Authors	year	Simulation Tool	Optimization Parameter	objective Function	Network Type	Conclusion
Genetic Algorithm							
GA	Rouphail et al	2000	CORSIM	signal timing parameters cycle length, phase times and offsets	Minimize network delay and queue time	9 signalized intersections in Chicago city (USA)	positive
GA SYNCHRO, TRANSYT-7F	Schneeberger and Park	2003	VISSIM	offsets	Minimize the average travel time	Northern Virginia	negative
GA, SA, and OptQuest Engine	Yun and Park	2006	CORSIM	Signal timing	Minimize delay time (the total Queue time)	Fairfax, Virginia, USA road network	GA is the best
GA	Farooqi et al	2009	THE simulator	Signal timing	Minimize total wait time	Virtual network	positive
GA	Singh et al.	2009	Their own emulator	Signal timing (green time)	Maximize Throughput	a four-legged isolated intersection	positive
GA (AARGA and RGA)	Qian et al	2013	a traffic emission-saving and signal timing model	Emissions factors and Delay, green time, capacity	comprehensive performance index *CPI* that Minimize pollutant emission	Isolated intersection (Shenzhen Lianhua-Xinzhou)	AARGA is better
GA, PS, TS	Abushehab et al	2014	SUMO	Phases duration	Minimize ATT	Nablus city Center	GA type 3, PS, TS type 5 are the best
multiobjective GA (MOGA)	Damay	2015	SUMO	green light phases	minimize the total waiting time and the total pollution emission	Rouen, France	positive
Adaptive MA , GA	Sabar et. al	2017	AIMSUN	Signals timing	N/A	Brisbane, Australia, and Plock, Poland	MA is better than GA and traditional fixed-time
Evolutionary Algorithm							
Multi objective EA	Mihaiţa et. al	2018	3D mesoscopic simulation model, FlexSim	Signals plan	Maximize the capacity	Nancy Grand Cœur, France	positive
Simulated Annealing							

SA	Sklenar et al.	2009	SSJ Stochastic simulation in Java	Phases duration	minimize the average waiting time	Konečného square in Brno, Czech Republic	positive
SA , GP	Moghimi et. al	2018	a bi-level optimization model	Signal timing and link capacity expansion	Minimize total travel time	Virtual network	positive
Particle Swarm Optimization							
multiobjective PSO	Kachroudi and Bhouri	2009	Multimodal mathematical model	Green splits and offsets	minimize the total number of PV in the network and minimize the quadratic difference between the real position of the buses and a pre-specified position	virtual urban (16 signalized intersections)	positive
PSO	Garcia-Nieto et al	2013	SUMO	cycle programs	Maximize Throughput	Sevilla and Malaga (Spain)	positive
IOCA-PSO PSO, GA, random	Hu et al.	2015	VISSIM	phase scheduling (timing control, the phase sequence control and the special phase controls)	minimize the proportion of the waiting time to the running time and the the proportion of the red light time to the green light time	Wuhan case	OCA-PSO is better
PSO TRANSYT	Zhao et al	2016	N/A	Phase combination	Minimize the number of stops	the intersection of Huangshan road and Kexue Ave. in Hefei	positive
PSO	Liang et al	2017	VISSIM	Phase combination and green time of each phase	Minimize delay time	The intersection of Xiuning Road and Hezuohua Road in Hefei (China)	positive
Ant Colony Optimization Algorithm							
ACO (AS and EAS)	Renfrew and Yu	**2009**	Dynamic Mathematical model	Signal timing Cycle length and green time	Minimize total delay time	Simple Four-legged isolated traffic intersection	positive
ACA, GA Webster	Jiajia and Zai'en	2012	N/A	Cycle time, saturation	Minimize a function of cycle time and saturation leads to Minimize time delay, number of stops, and maximize capacity	N/A	ACA is better
Harmony Search Algorithm							
Hybrid HS and HC with TRANSYT(HSHC TRANS), HS, GA	Ceylan Huseyin and Ceylan Halim	2012	TRANSYT	Signals timing	minimize network performance index (PI) combination of delay and number of stops	a virtual signalized road network (Allsop and Charlesworth's example network)	HSHCTRA is better than HS and GA
HSA, GA, HC	Dellorco et al	2013	TRANSYT	Signal phases time	Minimize the PI (delay and number of stops)	Simple 2 junction test network, Allsop and Charlesworth's network	HAS is better than GA and HC
Discrete Harmony Search (DSH)	Gao et al	**2016**	dynamic traffic flow model based on Daganzo's cell transmission models	Cycle time	Minimize network-wise total delay	Partial network in Singapore.	DHS with local search operator is better

The answer to "*which algorithm is the most appropriate to solve the problem"* remains open. In this study, recently published parameter-less algorithms called TLBO and Jaya were used to optimize the duration of traffic light phases in order to minimize the average travel time for vehicles.

3.6 Weaknesses of the Previous Research

Despite the important achievements in the reviewed approaches, there are some weak points which can be summarized as follows:

- Mathematical models are suitable to optimize a single intersection. It is difficult to develop a closed-form mathematical formulation to describe the stochastic behavior of traffic the system components for many intersections.
- Most methods are investigated on a special traffic network with limited elements (traffic lights, intersections, vehicles, roads etc.), and thus they are not interested in studying the behavior and scalability of the algorithms on other large networks.
- Most studies used only one technique of metaheuristic optimization. Optimization algorithms vary in speed to get the optimal solution. The speed factor is very important especially when it deals with a real-time traffic light system. Some latest variants of optimization algorithms such as TLBO and Jaya are not considered.
- All traditional optimization algorithms require their own specific parameters in addition to the common controlling parameters. The choice of the best parameters' values is considered as an optimization problem. Although the presence of parameters allows users to adapt the behavior of the algorithm, there are some points to be considered:

1) Finding good parameters' values is time-consuming and the wrong choice may lead to wrong optimal solutions.
2) The performance of the algorithm depends on the values of parameters, so we may need to calibrate the parameters' values for each new targeted problem.

- Some researchers assumed the values of algorithm parameters. Abushehab R. used a benchmark function (*Rastrigin*) to find the best parameters values. But, there is no relation between the benchmark function optimization problem and the traffic light signals timing problem. Therefore, if an optimization algorithm is the best in solving the benchmark function, it may not be the best in solving traffic light signals timing problem, and the opposite is true.

4. The Methodology of the Study

4.1 Introduction

The proper scheduling of the traffic lights reduces congestion in urban areas (Kaur & Agrawal, 2014). Many methodologies have been conducted to solve this problem. Simulation-based approaches integrated with metaheuristic optimizer have been extensively used to optimize the traffic signals timing problem. (Hewage & Ruwanpura, 2004; Karakuzu & Demirci, 2010; Lim et al., 2001; Garcia-Nieto et al, 2013; Abushehab et al., 2014).

To answer the questions raised in chapter one, this thesis relied on a simulation-based approach by using an efficient metaheuristic optimization algorithm integrated with a suitable traffic simulator to find the near optimal schedule for traffic signals timing. The framework used to optimize the traffic signals timing can be summarized in Figure (4.1).

This study combined both quantitative and experimental research type. Several experiments have been carried out to investigate and compare the performance of five optimization algorithms on three different networks. Furthermore, the statistical analysis of the experimental results was performed using ANOVA and Tukey HSD post-hoc tests. We performed Welch's ANOVA and Games-Howell post hoc tests when the assumption of homogeneity of variances was not met.

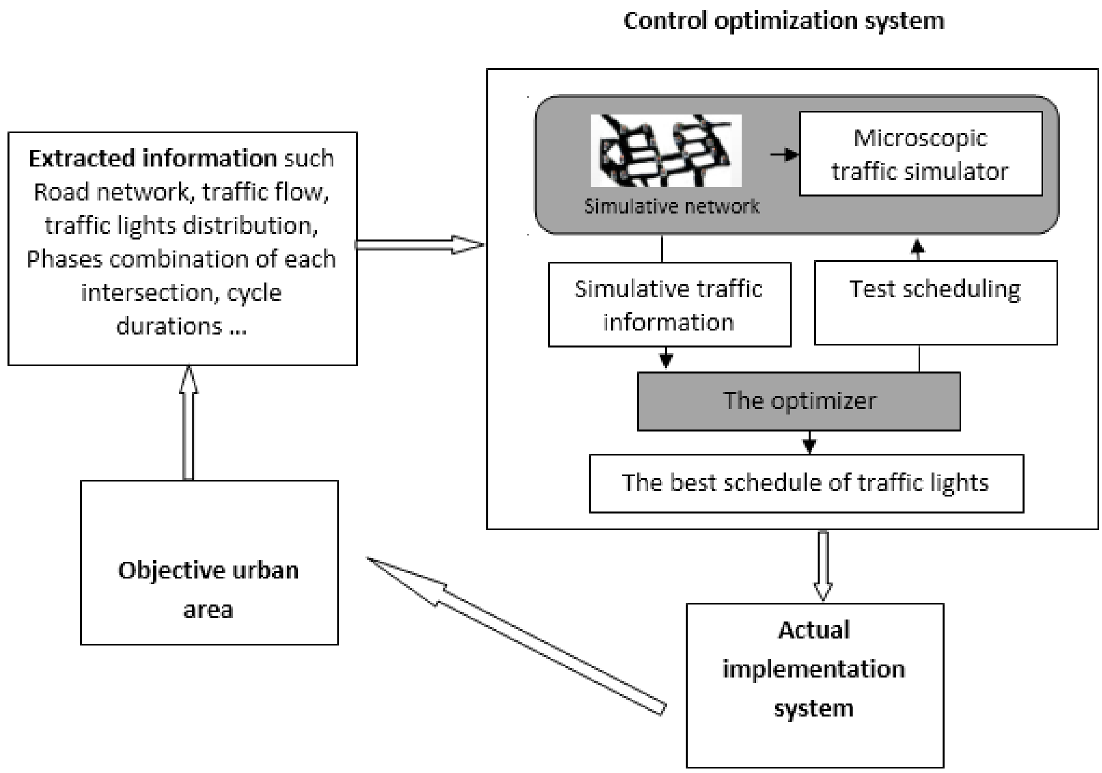


Figure 4.1: Framework of the traffic signals timing optimization (Hu et. al, 2015)

4.2 Simulator Selection

A simulator is an effective tool to gather quantitative information about the stochastic and dynamic traffic system. This study focused on the use of microscopic traffic simulator among different types of traffic simulators described previously. The reason for choosing this type of simulators is that it is more accurate than macroscopic simulators in describing the behavior of each individual vehicle-driver unit (Karakuzu & Demirci, 2010).

A simulator called SUMO was used. It is a microscopic and open source traffic simulator (Krajzewicz et al., 2012). Moreover, it can be easily interfaced to implement and evaluate the performance of the optimization algorithms. Go back to Tables 2.1 and 2.2 for more details about the features and the applications included in SUMO.

4.3 Optimization Algorithms

In this study, we have shown the comparison of performance for five global optimization algorithms. These algorithms were TLBO, Jaya, GA, PSO, and WTLBO. TLBO and Jaya are parameter-less algorithms, while GA, PSO, and WTLBO require their own specific parameters.

We chose TLBO and Jaya to optimize the duration of traffic light phases because they have been recently published, efficient, and simple algorithms (Rao, 2016b; Rao & Patel, 2011). These algorithms have been widely applied in a large number of benchmark functions and real-world applications in various engineering and scientific fields and showed effectiveness in problem-solving (Rao, 2016). However, the effectiveness and behavior of these algorithms have not yet been verified in optimizing traffic signals. Moreover, these algorithms are parameter-less and thus avoid the difficulty of tuning the parameters.

To validate the performance of TLBO and Jaya algorithms in optimizing traffic signal timing, we compared them with the most efficient algorithms among the evaluated algorithms in *Abushehab et al*. (2014) research. These algorithms were GA and PSO.

4.3.1 Genetic Algorithm

Abushehab et al. (2014) used 3 types of GA called GA type1, GA type2, and GA type 3. They concluded that GA type 3 was the most effective in solving the problem, so we used this algorithm in our research. The major three operators (selection, crossover, and mutation) of this algorithm are described as follows:

- *Selection*: The best half of population is selected as parents (μ)
- *Crossover*: every two successive parents (in order) from the selected parents are crossed to generate new two offspring and complete the other half of population (λ).

  The type of crossover used is a single point crossover where the crossover point c is randomly selected between 1 and n, where n is the number of parameters.
- *Mutation*: mutate all the parents by randomly mutating one parameter in each one.

The pseudo code and steps of GA type3 are shown in Algorithm 4.1

Algorithm 4.1: GA type 3 (Abushehab et al., 2014)

a.*Randomly generate the first population of individuals' potential solutions.*

b. Evaluate the objective function ATT, for each population record.

c. While not (number of iteration reached):

1. Select the best half chromosomes from previous generations as parents

2. Crossover between each two selected chromosomes to get two new offsprings.

3. Mutate all the parents.

4. Generate randomly the other chromosomes in the generation until a new population has been completed (Until a new population has been completed)

4.3.2 Particle Swarm Optimization Algorithm

This algorithm was explained in chapter 2. However, the operation of PSO which we selected from *Abushehab et al*. (2014) study is shown in Algorithm 4.2

Algorithm 4.2: PSO algorithm

1. $initia(\theta)$ *// initial swarm usually random*

2. for each particle $x \in \theta$:
for each dimension i
// calculate velocity according to the equation
$xv_i = w * xv_i + cp * r * (pbest_i - x_i) + cg * r * (gbest_i - x_i)$
// update particle position according to equation
$x_i = x_i + xv_i$
3. While stop criteria not reached, Go to step 02

The pseudo code and steps of TLBO, WTLBO, and Jaya algorithms can be found in Chapter 2 (see algorithm 2.1, algorithm 2.2)

4.4 Cases of the Study

The optimization algorithms were tested on three different road networks with different characteristics and different number of traffic lights to study the scalability of each algorithm. The first one was real, small in size and corresponding to the central part of Nablus city which was used by Abushehab et al. (2014). The second one was virtual, random and medium-sized. And the third one was virtual, random and large-sized. All networks were built by using traffic simulator called 'SUMO'. Besides, the homogeneity of vehicles was assumed in all tested networks. See Table 4.1 for details on each network specifications. In SUMO, a street in the network consists of nodes (the junctions that are connected together) and directed edges (the links that connect between junctions). For example, to build a simple network with 2 streets subsequent to each other, three nodes and two edges are required. Each node described by a location and an id as a reference, while each edge described by a source node id, a target node id, and an edge id as a reference.

4.4.1 Case Study 1

The basic layout of Nablus city center network is given in figure 4.2. In the peak hours, the streets and the junctions witness heavy traffic and traffic jams. This network is relatively small in size with 37 nodes and 38 edges, 8 of these intersections were signalized. Intersections had a different number of phases. However, the total number of green and red phases was 13.

Each traffic light signal may have a red, green or yellow color. All green phases are followed by a yellow phase. We assumed that the length of red or green phases was between ***10 – 60*** seconds, and the length of yellow phase was constant (3 seconds) for all traffic light signals.

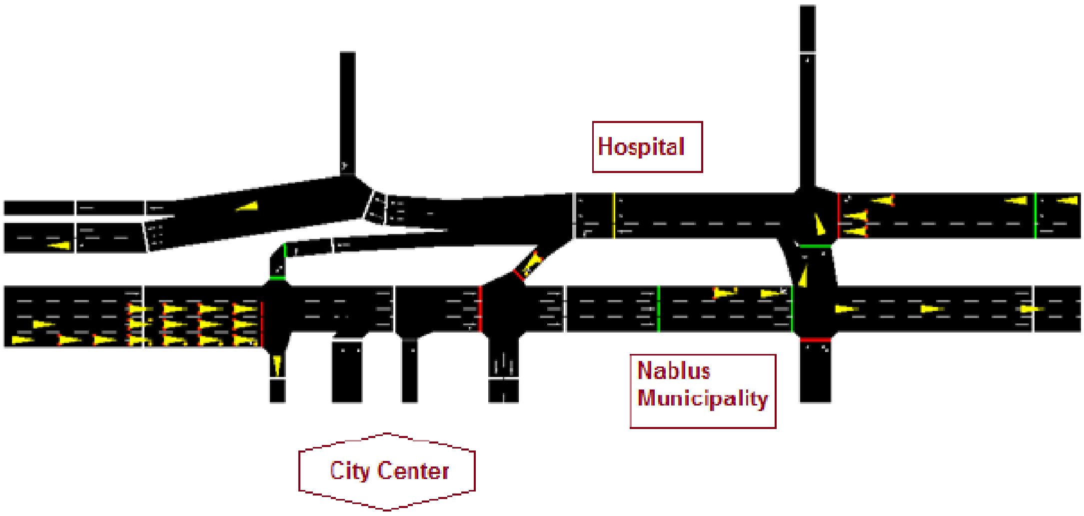


Figure 4.2: Nablus city center road network

4.4.2 Case Study 2

A virtual network was generated randomly by using NETGENERATE application which is included in SUMO simulator. This network was relatively medium-sized and it was composed of 56 nodes and 34 edges. The network contained 16 intersections controlled by traffic signals, see Figure 4.3. Intersections had a different number of phases. However, the total number of green and red phases was 34. The length of red or green phases was between ***10 – 60*** seconds and the length of yellow phase was constant (3 seconds) for all traffic light signals.

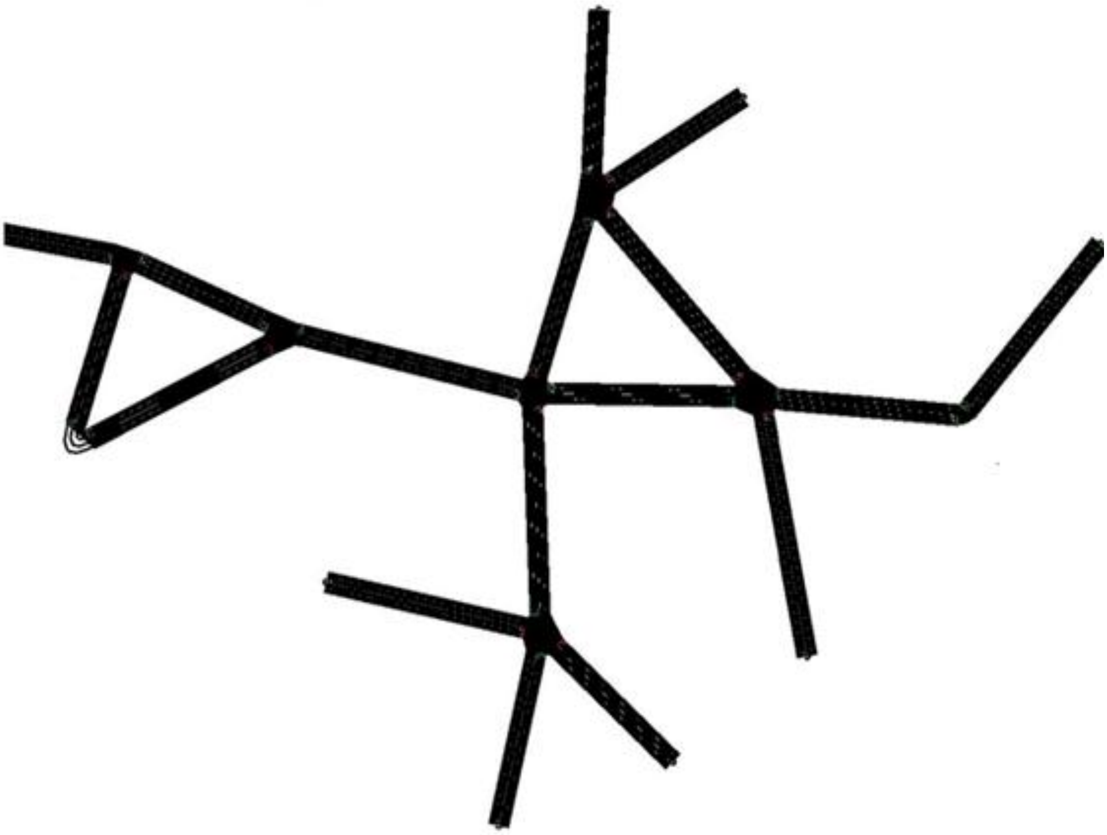


Figure 4.3: Case study 2

4.4.3 Case Study 3

A virtual network with 264 nodes and 144 edges was randomly generated by NETGENERATE application in SUMO simulator. This network is large and complex when compared to previous networks. The network contained 50 intersections controlled by traffic signals. See figure 4.4. The intersections had a different number of phases. However, the total number of green and red phases was 142. The length of red or green phases was between ***10 – 60*** seconds, and the length of yellow phase was constant (3 seconds) for all traffic light signals.

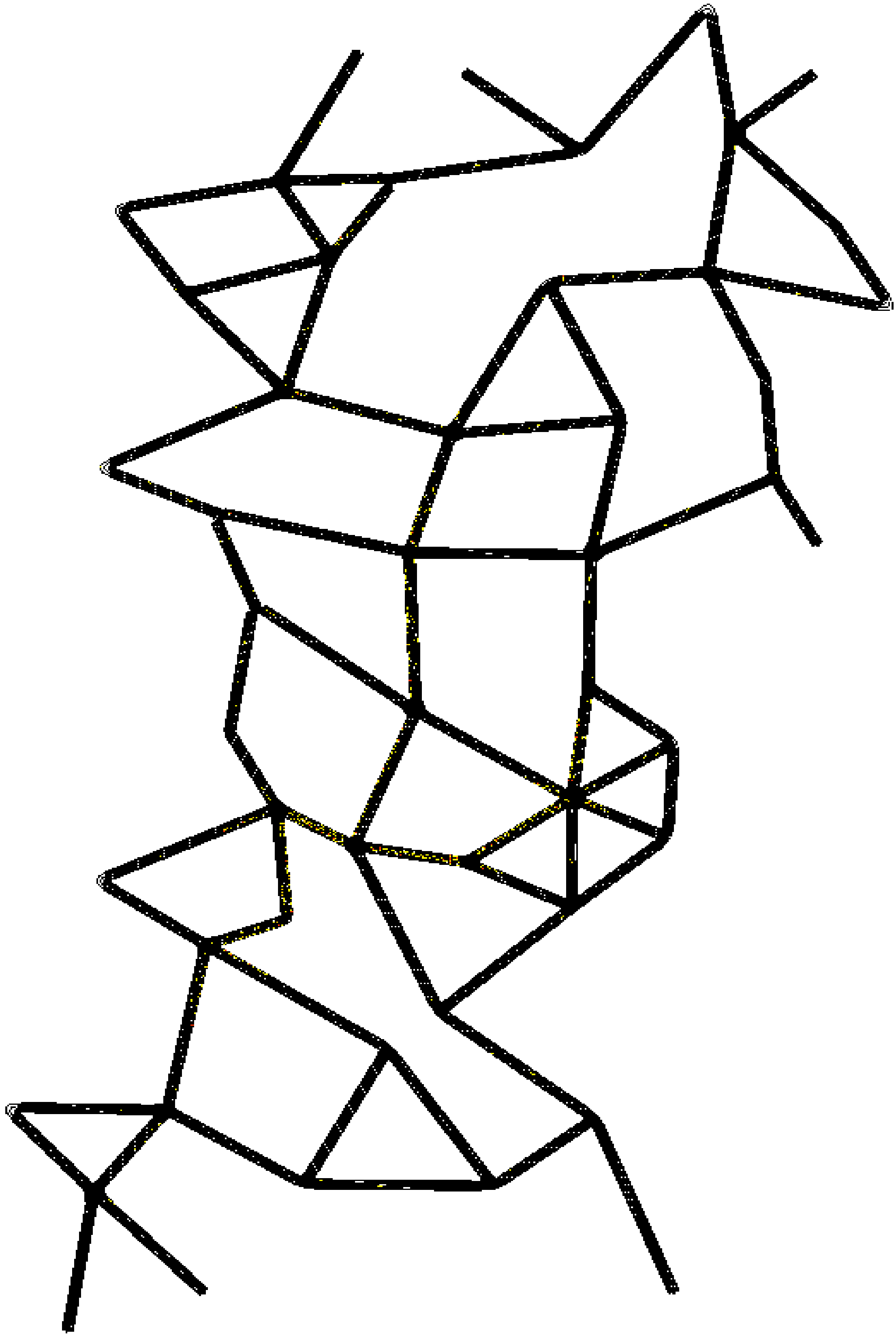


Figure 4.4: Case study 3

The main features of the three networks are summarized in Table 4.1

Table 4.1: Parameters of the case studies

network	type	Nodes (#)	edges (#)	Intersections(#) controlled by traffic lights	Total number of phases (red and green)	Loaded vehicles	Simulation time (m)
Case 1	real	37	38	8	13	1740	60
Case 2	virtual	56	34	16	34	1000	45
Case 3	virtual	264	144	50	142	3017	50

4.5 Solution Design

4.5.1 Cycle Program of Traffic Light

The following definitions need to be understood in the signal design (Garcia-Nieto et al, 2013; Hu et al., 2015):

- *Cycle*: A signal cycle is a one complete rotation through all of the phases provided.
- *Cycle time*: the needed duration to display all the phases at an intersection before returning the first phase of the cycle.
- *Phase*: the part of a cycle allocated to any combination of non-conflicting movements.

Figure 4.5 shows an example of traffic signal cycle with 4 valid phases:

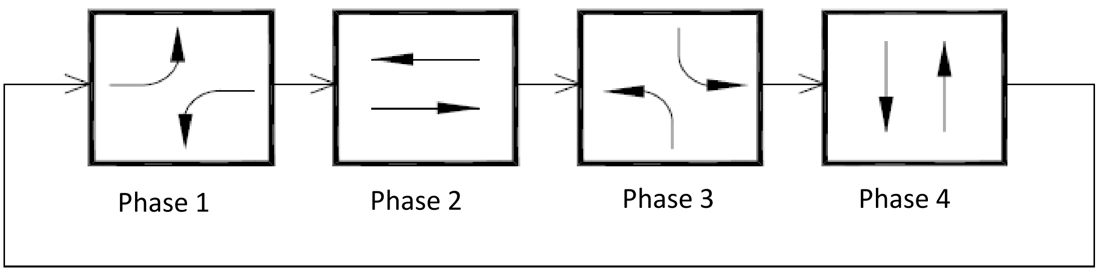
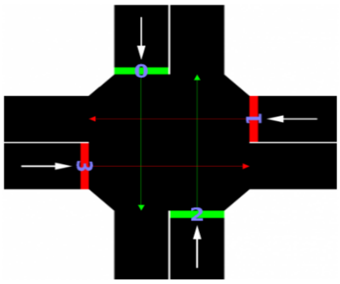


Figure 4.5: Traffic signal cycle with 4 phases

The junctions that have traffic lights control the movement of vehicles by following programs of color states and cycle durations. Each junction has its own program which synchronizes the traffic lights located at this junction, and thus ensuring that no collisions will occur between vehicles and providing safety for pedestrians.

The program at each intersection is defined by a combination of valid phases. These phases are described by duration and a set of states for the traffic lights. For example, Figure 4.6 presents a simple two-phase junction and its program generated by SUMO simulator (DLR and contributors, 2013).



(a)

```
<tlLogic id="1" type="static" programID="0" offset="0">
    <phase duration="30" state="grgr"/>
    <phase duration="3" state="yryr"/>
    <phase duration="20" state="rgrg"/>
    <phase duration="3" state="ryry"/>
</tlLogic>
```

(b)

Figure 4.6: (a) Two-phase junction, (b) Cycle program

Figure 4.6 shows that the intersection contains two main phases, and the duration of the two phases are 30 and 20 seconds. All green phases are followed by a yellow phase which is fixed and equals 3 seconds. Four traffic lights are located at the intersection to control the links 0,1,2 and 3.

Each character within a phase's state describes the state of one signal of the traffic light, where *g*, *r*, and *y* mean green, red, and yellow respectively. Each phase's state has four signals (colors). Each one of them corresponds to one of the four traffic lights located in the intersection.

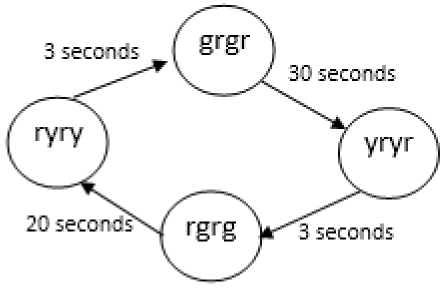


Figure 4.7: State diagram of the given two-phase junction

Figure 4.7 shows the stat diagram of the given two-phase junction system. The current state "***grgr***" means that for 30 seconds, two traffic lights (the first and the third) are green, while the other two traffic lights (the second and the fourth) are red. Then, the color states of the traffic lights are modified according to the remaining phases in a sequential manner (Garcia-Nieto et al, 2013; Hu et al., 2015).

4.5.2 Traffic Signal Optimization Model

Optimizing the traffic lights helps with facilitating the mobility in the urban traffic system and thus reducing the travel time for the vehicles. Determining the best duration of traffic signal phases problem can be modeled and formulated as an optimization problem. So, we specified the basic three elements: (1) the solution representation, (2) the objective, and (3) the evaluation function as follows (Michalewicz & Fogel, 2010):

4.5.2.1 Solution Representation

There are different types of parameters in the traffic light problem that can be optimized. The aim of this study is to optimize the cycle program (or phase durations). So, each candidate solution was represented as an n-dimensional integer vector $X=\{x_1, x_2, x_3 \ldots x_n\}$ where each element represents a phase duration (only red or green) of one state of the traffic lights involved in a given intersection. ***n*** is the total number of red or green phases of all traffic lights in all intersections (see Figure~~.~~ 4.8).

…… Intersection i Intersection i+1 …… ……

……	20	10	50	33	40	55	30	……	…….
x1	x2	x3	…..	……	……	…..	…..	xn-1	xn

Figure 4.8: Solution representation

4.5.2.2 The Objective

Our objective was to minimize the average travel time for vehicles, which leads to improve the global flow of vehicles in the urban traffic. Furthermore, minimizing the average of travel time leads to reduce the fuel consumption, and the amount of pollutants.

4.5.2.3 The Evaluation Function

In the traffic system, it is difficult to find a closed-form for the mathematical relationship between the input cycle programs and the average of travel time, so SUMO simulator was used as an evaluation function which maps each candidate solution to a real value that indicates the quality of the solution according to Eq. (4.1).

$$Fitness\ (ATT) = \frac{TT}{K} \qquad (4.1)$$

Where ***ATT*** is the Average Travel Time, ***TT*** is the total trip time for all vehicles that reached their destination during the simulation process, ***k*** is the number of these vehicles. These values are calculated from the resulting output file of SUMO.

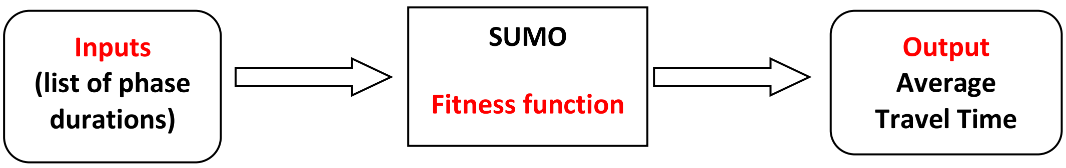


Figure 4.9: Traffic signal optimization model

As shown in Figure 4.9, the input for the traffic light optimization problem was a list of ***n*** phase durations. The output was the Average Travel Time (ATT).

The traffic light optimization problem can be formulated according to Eq. (4.2).

$$\boxed{\begin{array}{c} Minimize\ f(T) \\ \text{Subject to} \\ L \leq\ Ti\ \leq U \qquad i = 1\ \dots .\ n \end{array}} \qquad (4.2)$$

Where:
T: adjustable vector of integer values (i.e. time list of phase durations)
f: fitness function is given in equation (4.1)
L: the lower bound value
U: the upper bound value
n: the number of phases (green or red).
Ti: the i^{th} value of input vector in seconds

4.6 Experimental Setup

4.6.1 Experiment Design

The implementation of experiments in this study was based on how the simulation program (SUMO in this study) integrates with the optimization algorithm (see Figure 4.12).

4.6.1.1 SUMO Operation:

For a simulation in SUMO, at least three main XML files must be given. These input files are: network file (i.e. *name*.net.xml), routes file (i.e. *name*.rou.xml), and configuration file (i.e. *name*.sumo.cfg) where *.net.xml*, *.rou.xml*, and *.sumo.cfg* are the default suffix for network, routes, and configuration files respectively (Krajzewicz, 2010).

Network File

The network file is created from other two files by using NETCONVERT tool (see figure 4.10). These files are node files (i.e. *name.nod.xml*) which define the nodes (junctions) and their parameters such as location, type, and id. The other file is the edges file (i.e. *name*.edg.xml) which defines the directed edges that connect the nodes.

Hence, the network file describes the topology of the simulated network and contains lots of generated information such as structures within an intersection, traffic lights programs, priority, lanes, and other information.

Figure 4.10: Network file creation in SUMO

Routes:

After creating the network, the vehicles are added and routed through the edges that were defined previously. In SUMO, the vehicles have types which define their basic properties such as acceleration, deceleration, length, maximum speed, and many other attributes.

Configurations:

This file is used by SUMO to identify the input files and the output files, simulation time, and other additional settings

Simulation Output:

SUMO generates a large number of measurements where their values can be written to output files in XML- format. Some types of the available outputs are simulated detectors, values for edges or lanes, simulation (network)-based information, traffic lights-based information, and vehicle-based information.

In this study, we used the (vehicle-based information) type of output, specifically the trip information output file. This file contains aggregated information about the trip of each vehicle such as departure time, arrival time, and duration. The information was generated for each vehicle that got its destination. We used the duration (travel time) values to calculate the total travel time for the vehicles, thus finding the average of the travel time. The work of SUMO can be summarized in Figure 4.11.

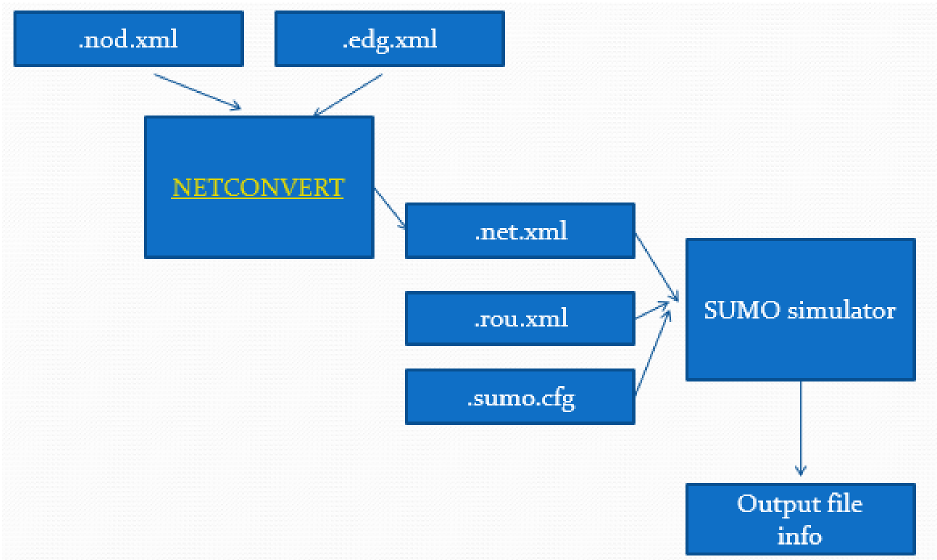


Figure 4.11: SUMO operation

4.6.1.2 Optimization Strategy

The optimization strategy for traffic signal timing consists of two main components: a microscopic simulator (SUMO), and an optimizer (see Figure. 4.12). Initially, the optimization algorithm randomly produced the population of solutions (i.e. duration of phases). The duration of phases is the decision variables of the optimization algorithm. Each solution is then written to the XML network file where each value binds to a phase duration of one state of the traffic lights. The candidate solution is then evaluated through SUMO simulator which produces the corresponding output file that includes the information about the vehicle's trip. The fitness value (i.e. average travel time) is then calculated based on the trip information file. These steps are repeated for each candidate solution. The optimizer performs its own steps to produce a new solution set based on the fitness values obtained from SUMO. The circulation process of Figure 4.12 is continued until the maximum number of iterations is reached. Therefore, the number of times the SUMO is run equals the total number of evaluations used by the optimization algorithm.

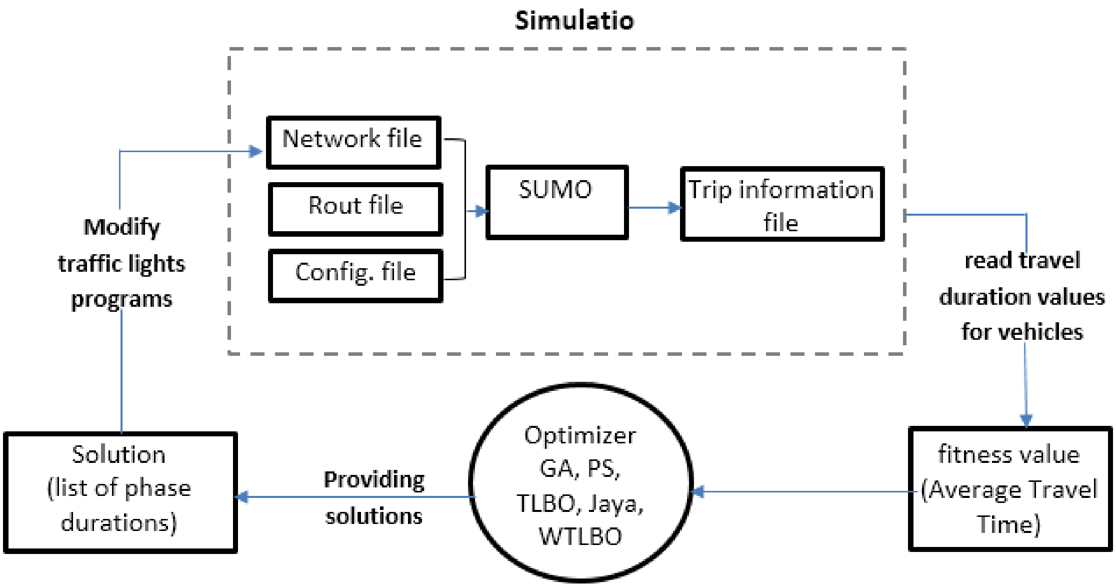


Figure 4.12: Optimization strategy for traffic signal timing

The experiment codes were implemented using c++ and python. The experiments were conducted on a server with 2 processors (Intel® Xeon® CPU E5-2650 0 @ 2.00 GHz, 32 GB memory, 64 bits windows server 2012R), in addition to 12 computers with processor: Intel® Xeon® CPU E5603@ 1.600 GHz, 12.0GB memory, 64 bits operating system at AN-Najah National University computer labs.

4.6.2 Parameters Settings

In the all experiments of this study, the specific parameters' values for each algorithm were as the following:

- ***GA settings***: Mutation Probability (MP) was 100%.
- ***PSO settings***: inertia weight (w) was 0.25, cognitive parameter (cg) was 3.5, and social component (cp) was 1.25.
- ***TLBO settings***: there are no specific parameters.
- ***WTLBO settings***: $w_{max} = 0.9$, $w_{min} = 0.1$
- ***Jaya settings***: there are no specific parameters

The parameters settings for GA and PSO were the best values determined by *Abushehab et al*. (2014) in their study. WTLBO settings were selected from *Satapathy et al*. (2013).

A common platform is required to guarantee a fair comparison between the algorithms that have been tested on different networks. Therefore, for each test site, The evaluated algorithms have been investigated using the same number of solution evaluations (SUMO simulations).

In the TLBO and WTLBO algorithms, the solution is updated and evaluated twice, in the teacher phase and the learner phase. Hence, the total number of function evaluations can be computed as in equation (4.3). While the formula which was used to count the total number of evaluations for (Jaya, GA, and PSO) algorithms is given in equation (4.4).

$$\textit{Total number of evaluations} = 2 \times \textit{population size} \times \textit{number of iterations} \qquad (4.3)$$

$$\textit{Total number of evaluations} = \textit{population size} \times \textit{number of iterations} \qquad (4.4)$$

the metaheuristic algorithms which were used are stochastic in nature. As a result, two successive runs usually do not give the same results. Hence, each algorithm was run several independent runs (with different seeds of the random number generator).

4.6.3 Statistical Analysis Methods

In order to analyze the results and investigate whether there are any statistically significant differences between the results obtained from each algorithm, we performed classic One-Way ANOVA and Tukey HSD post-hoc tests. We performed Welch's ANOVA and Games-Howell post hoc tests when the assumption of homogeneity of variances was not met (One-way ANOVA, n.d.).

Significance level (α = 0.05), normality was assumed for analysis and equal variances was verified by Leven's Test.

Levene's Test

We used Levene's test (Levene, 1960) to verify the assumption of equal variances (homogeneity of variances). The test hypothesis is defined as:

H$_0$: $All\ variances\ are\ equal$ ($\sigma_1^2=\sigma_2^2=\sigma_3^2= ...=\sigma_k^2$), k: number of groups

H$_1$: $\sigma_i^2 \neq \sigma_j^2$ for at least one pair (i, j)

The null hypothesis is rejected if P-value $\leq \alpha$

One-way ANOVA Test:

We used the one-way analysis of variance (ANOVA) to determine whether the groups of means are statistically significantly different from each other (Saunders et al., 2016).

The hypothesis is defined as:

H$_0$: $All\ means\ are\ equal$ ($\mu_1 = \mu_2 = \mu_3 = ... = \mu_k$) , k: number of groups

H$_1$: $At\ least\ one\ mean\ is\ different$

We used ANOVA test instead of multiple t-tests because every time we conduct a t-test, there is a chance for type 1 error (usually 5%) to occur. Hence, by conducting two t-tests on the same data, the chance for type 1 error will increase and so on. On the other hand, using ANOVA test ensures that Type 1 error remains at 5%, and thus, this test gives more reliable results (One-way ANOVA, n.d.).

Welch's ANOVA

It is an alternative to the classic ANOVA when the assumption of homogeneity of variances is not met. Hence, it has the most power and lowest Type 1 error rate for different-variance data. (Moder, 2010)

Post-hoc Test:

If the null hypothesis is rejected, the ANOVA does not tell us which specific groups differed. So, post hoc tests are run to confirm where the differences occurred between the groups. In this study, we used Tukey HSD and Games-Howell Post-hoc tests.

4.7 Experiments and Procedures

To answer the questions posed in chapter one, several experiments were carried out to investigate and compare the performance of five optimization algorithms on three different networks.

4.7.1 Comparing Optimization Techniques in Case study 1

The experiments in this network were divided into two phases:

4.7.1.1 Phase 1 Experiments:

We assumed that the period time for red or green light phase was between ***10 – 60*** seconds, time for yellow light phase was constant (3 seconds). To study the effect of population size, each algorithm was experimented with different population sizes of ***5, 15, 30, 50, 75*, 100, 200, 300, and *400***. The maximum number of evaluations was ***7500*** for all the tested algorithms and each algorithm was run ***20*** independent runs.

4.7.1.2 Phase 2 Experiments

In this phase, we increased the size of the solution space. The period time for red or green light phase was between ***10 – 90*** seconds and the time for the yellow light phase was constant (3 seconds). Each algorithm was experimented with different population sizes of ***5, 15, 30, 50, 75*, 100, 200,**

300 and ***400***. The maximum number of evaluations was ***7500*** for all the tested algorithms and each algorithm was run ***20*** independent runs.

4.7.2 Comparing Optimization Techniques on Case Study 2

In this case, the period time for red or green light phase was between ***10 – 60*** seconds and the time for the yellow light phase was constant (3 seconds) for all traffic light signals. To study the effect of population size, each algorithm was experimented with different population sizes of ***5, 15, 30, 50, 75*** and ***100***. The maximum number of evaluations was ***15000*** for all the tested algorithms and each algorithm was run ***20*** independent runs**.**

4.7.3 Comparing Optimization Techniques on Case Study 3

In this case, the period time for red or green light phase was between ***10 – 60*** seconds, and the time for yellow light phase was constant (3 seconds) for all traffic light signals. To study the effect of population size, each algorithm was experimented with different population sizes of ***50, 500,*** and ***1000***. the maximum number of evaluations was ***20000*** for the all the tested algorithms and each algorithm was run ***20*** independent runs**.**

In all cases, the experiments which were carried out were:

- Performance and convergence speed of basic TLBO
- Performance and convergence speed of WTLBO
- Performance and convergence speed of Jaya

- Performance and convergence speed of GA
- Performance and convergence speed of PSO
- Comparison of TLBO, WTLBO, Jaya, GA, and PSO: We compared the algorithms based on the best result of each algorithm (the best population size) obtained from previous experiments.

Table 4.2 summarizes the settings used in the experiments.

Table 4.2: Summary of experiments settings

<table>
<tr><th>Case study</th><th>Duration of phases</th><th>No. of decision variables</th><th>Solution space</th><th>Max. no. of evaluations</th><th>Specific parameters of algorithms</th></tr>
<tr><td rowspan="2">1</td><td>10-60</td><td>13</td><td>50^{13}</td><td>7500</td><td rowspan="4">GA: MP=0%.
PSO : w= 0.25,cg=3.5,cp=1.25.
TLBO: no parameters.
WTLBO: w_{max} = 0.9, w_{min} = 0.1
Jaya:no parameters</td></tr>
<tr><td>10-100</td><td>13</td><td>90^{13}</td><td>7500</td></tr>
<tr><td>2</td><td>10-60</td><td>34</td><td>50^{34}</td><td>15000</td></tr>
<tr><td>3</td><td>10-60</td><td>142</td><td>50^{142}</td><td>20000</td></tr>
</table>

4.8 Summary

In this chapter, we presented the methodology used to answer the research questions, the selected simulator, optimization algorithms, and test sites. Furthermore, it addressed the model design of TSOP. Finally, it presents the experimental setup and procedures.

5. Results and Data Analysis

5.1 Introduction

This chapter presents the simulation results of the experiments, and the comparisons between the proposed approaches and a set of well-known metaheuristic algorithms. We were first interested in analyzing the effect of common controlling parameters (i.e. population size) on the performance of each tested algorithm. Then we took the best result obtained by each algorithm and drew a comparison between the algorithms. The comparative results were presented in the form of minimum, maximum, mean, and standard deviation of the fitness values (ATT) which were obtained in 20 independent runs. The convergence speed of different algorithms was also examined.

In all experiments, PSO and GA results have not been obtained directly from the literature. We have only selected the best-recommended algorithms which evaluated by *Abushehab et. al* (2014) in the first case study. We have re-implemented them in our experiments to validate the performance of TLBO, WTLBO, and Jaya.

In all experiments, to analyze the reported results and to draw conclusions, we used descriptive and inferential statistics. In this study, we conducted classic One-Way ANOVA and Tukey HSD post-hoc tests. Whereas, we performed Welch's ANOVA and Games-Howell post hoc tests when the assumption of homogeneity of variances was not met. Significance level (α = 0.05), normality was assumed for analysis and equal variances was verified by Leven's Test.

5.2 Comparing Optimization Techniques on Case Study 1

5.2.1 Phase 1 Experiments

Table 5.1: Phase 1 experiments settings

Green or red time (s)	Yellow time (s)	Population size	evaluations
10 - 60	3	5, 15, 30, 50, 75, 100, 200, 300, 400	7500

5.2.1.1 Performance and convergence speed of basic TLBO

Table 5.2: Descriptive statistics of Basic TLBO on case study 1 with phase duration 10-60

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	64.1494	6.41178	1.43372	61.1486	67.1502	57.46	78.17
15.00	**56.6814**	1.20917	.27038	56.1155	57.2473	54.58	58.87
30.00	57.3708	2.10142	.46989	56.3874	58.3543	55.54	63.59
50.00	57.2107	1.48676	.33245	56.5148	57.9065	54.13	61.30
75.00	57.2768	1.05357	.23558	56.7838	57.7699	55.67	59.51
100.00	58.0761	.89017	.19905	57.6595	58.4927	56.81	60.27
200.00	58.8200	1.48774	.33267	58.1237	59.5163	56.33	61.53
300.00	60.0964	1.83024	.40925	59.2399	60.9530	57.91	65.92
400.00	60.5772	1.63254	.36505	59.8131	61.3412	58.17	64.47
Total	58.9176	3.36919	.25112	58.4221	59.4132	54.13	78.17

The bold value indicate best results

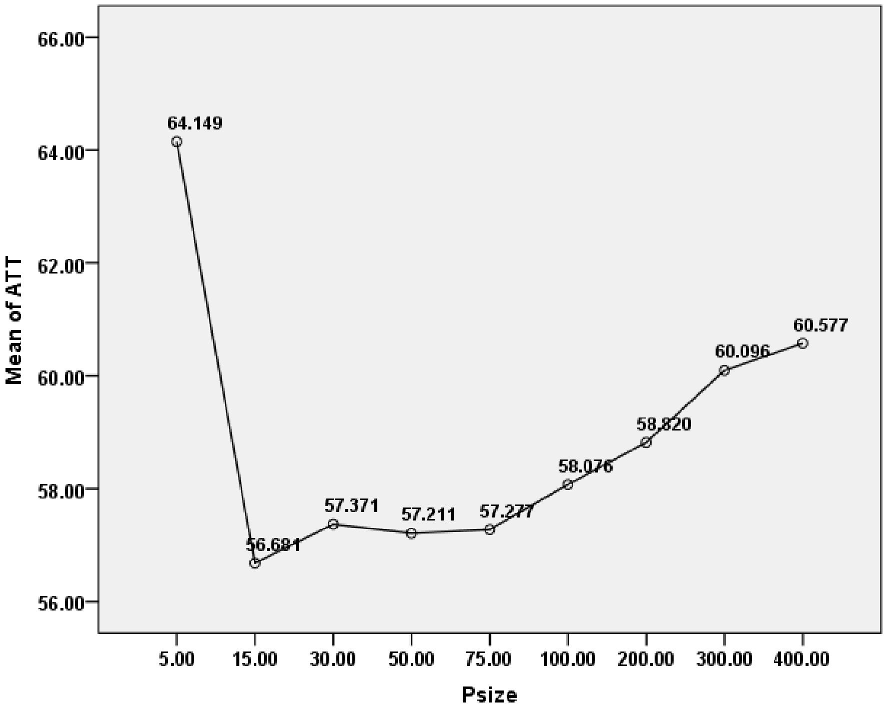


Figure 5.1: The mean results of TLBO by changing Psize on case 1 phase duration 10-60

Figure 5.2 shows the convergence of TLBO algorithm with different population sizes. The vertical axis represents the mean of fitness value (for 20 runs), and the horizontal axis represents the number of loss function evaluations. The strategy with the population size of **15** produced a better convergence rate as shown in Figure 5.2. The convergence rate was almost similar when the population size increased from 50 to 100.

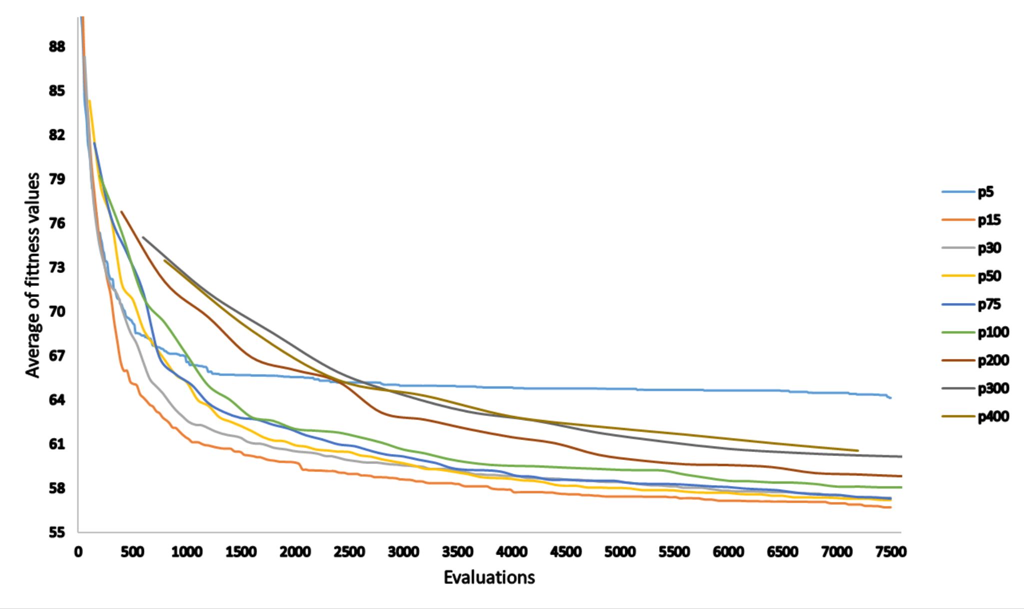


Figure 5.2: Convergence curves of TLBO by changing Psize on case 1 phase duration 10-60

The Homogeneity of variances was violated as indicated by Leven's test(*F(8,171)* = *20.364, p* < *.001*). There was a statistically significant difference between the mean result of different population sizes as determined by Welch's F test (*F(8,70.734)* = *16.562*, *p* < *.001*). Post hoc comparisons using the Games-Howell post hoc test were conducted. The results in table 5.4 revealed that there was no statically significant difference between the results of population sizes that are listed under each subset. Post hoc comparisons are listed in Appendix A, Table 5.3.

Table 5.4: Homogeneous subsets of Psize (TLBO on case 1 phase duration 10-60)

Psize	Homogeneous subsets					Significant conclusions
	1	2	3	4	5	
15.00	56.6814					15 < 100,200,300,400,5
50.00	57.2107	57.2107				
75.00	57.2768	57.2768				
30.00	57.3708	57.2768	58.0761			
100.00		58.0761	58.0761			
200.00			58.8200	58.8200		
300.00				60.0964	60.0964	
400.00					60.5772	
5.00					64.1494	

5.2.1.2 Performance and convergence speed of WTLBO

Table 5.5: Descriptive statistics of WTLBO on case study 1 with phase duration 10-60

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	74.8257	10.15676	2.27112	70.0722	79.5792	58.92	96.36
15.00	63.2486	4.81738	1.07720	60.9940	65.5032	57.06	74.07
30.00	62.4617	3.38152	.75613	60.8791	64.0443	57.90	69.71
50.00	61.9352	2.25949	.50524	60.8777	62.9926	59.24	67.36
75.00	61.8875	2.94369	.65823	60.5098	63.2651	58.27	68.57
100.00	**61.4486**	2.85035	.63736	60.1146	62.7827	57.52	70.13
200.00	62.1496	2.16817	.48482	61.1349	63.1643	58.30	66.04
300.00	63.0167	3.64534	.81512	61.3106	64.7227	58.96	73.73
400.00	64.0441	3.25096	.72694	62.5226	65.5656	57.95	70.44
Total	63.8908	5.96536	.44463	63.0135	64.7682	57.06	96.36

The bold values indicate best results

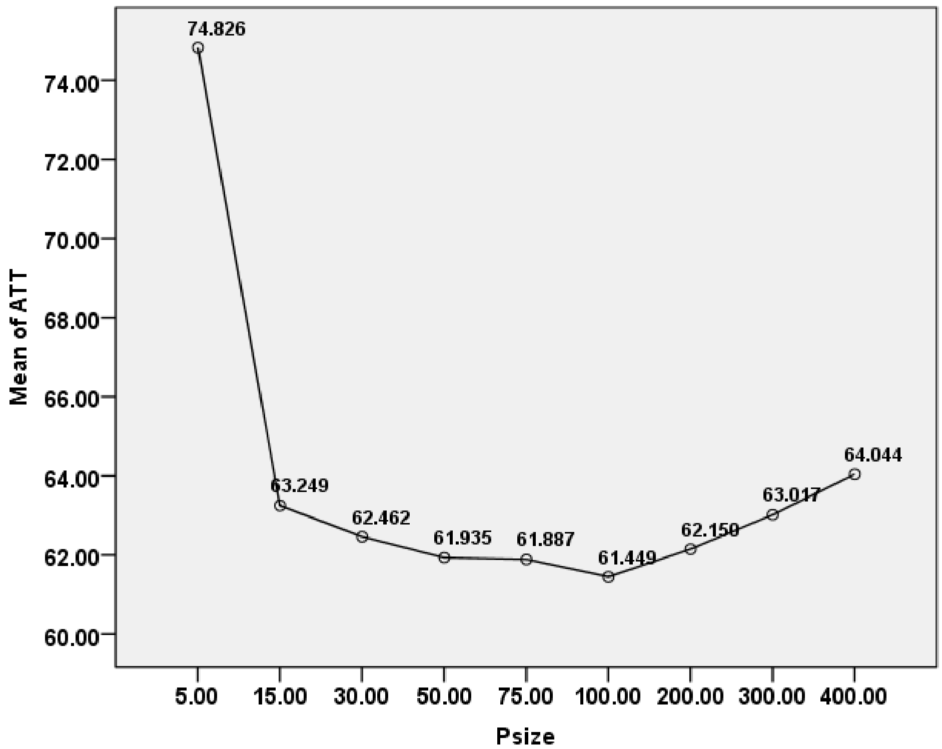


Figure 5.3: The mean results of WTLBO by changing Psize on case 1 phase duration 10-60

Table 5.5 and Fig 5.3 show that the optimal mean of fitness value seems to occur when the population size was ***100*** (mean = 61.4486). And there was no dramatic difference between the results as the population size increased from 15 to 400. In the terms of convergence rate, shown in Fig 5.4, the convergence of WTLBO algorithm with the population size of 15, 30, 50, 75, and 100 were better than the population size of 200,300,400, and 5. The convergence rate was almost similar when the population size increased from 15 to 100.

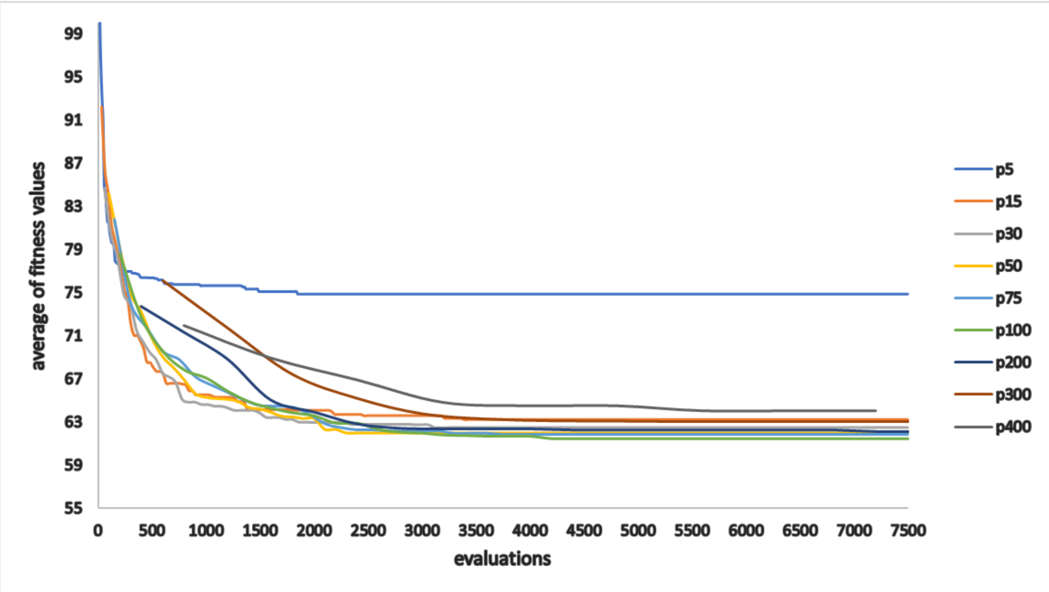


Figure 5.4: Convergence curves of WTLBO by changing Psize on case1 phase duration10-60

The Homogeneity of variances was violated as indicated by Leven's test(*F(8,171) = 7.591*, *p* < *.001*). There was a statistically significant difference between the mean result of different population sizes as determined by Welch's F test (*F(8,70.875) = 4.709*, *p* < *.001*). Post hoc comparisons that use the Games-Howell post hoc test were conducted. The results in Table 5.4 reveal that there was no statically significant difference between the results of population sizes of 15, 30, 50, 75, 100, 200, and 400.

strategy with the population size of 5 had a significantly higher result than other population sizes. See (Table 4.6 in Appendix A, table 5.7)

Table 5.7: Homogeneous subsets of Psize (WTLBO on case 1 phase duration 10-60)

Psize	Homogeneous subsets 1	Homogeneous subsets 2	Significant conclusions
15.00	61.4486		(15, 50, 75, 30, 100, 200, 300, 400) < 5
50.00	61.8875		
75.00	61.9352		
30.00	62.1496		
100.00	62.4617		
200.00	63.0167		
300.00	63.2486		
400.00	64.0441		
5.00		74.8257	

5.2.1.3 Performance and convergence speed of Jaya

Table 5.8: Descriptive statistics of Basic Jaya on case study 1 with phase duration 10-60

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
5.00	69.6357	6.87880	1.53815	66.4164	72.8551	61.72	84.82
15.00	58.2480	3.14127	.70241	56.7778	59.7181	54.02	63.49
30.00	**56.5704**	**.83032**	**.18567**	**56.1818**	**56.9590**	**55.29**	**58.57**
50.00	58.1147	1.52820	.34172	57.3995	58.8300	55.98	62.60
75.00	58.3533	1.90309	.42554	57.4626	59.2439	56.57	62.09
100.00	59.2105	2.55117	.57046	58.0166	60.4045	56.40	64.94
200.00	59.8089	2.34545	.52446	58.7112	60.9066	57.71	65.13
300.00	61.4950	2.01564	.45071	60.5516	62.4383	58.03	64.80
400.00	60.9945	2.37152	.53029	59.8846	62.1044	57.92	67.19
Total	60.2701	4.70853	.35095	59.5776	60.9627	54.02	84.82

The bold values indicate best results

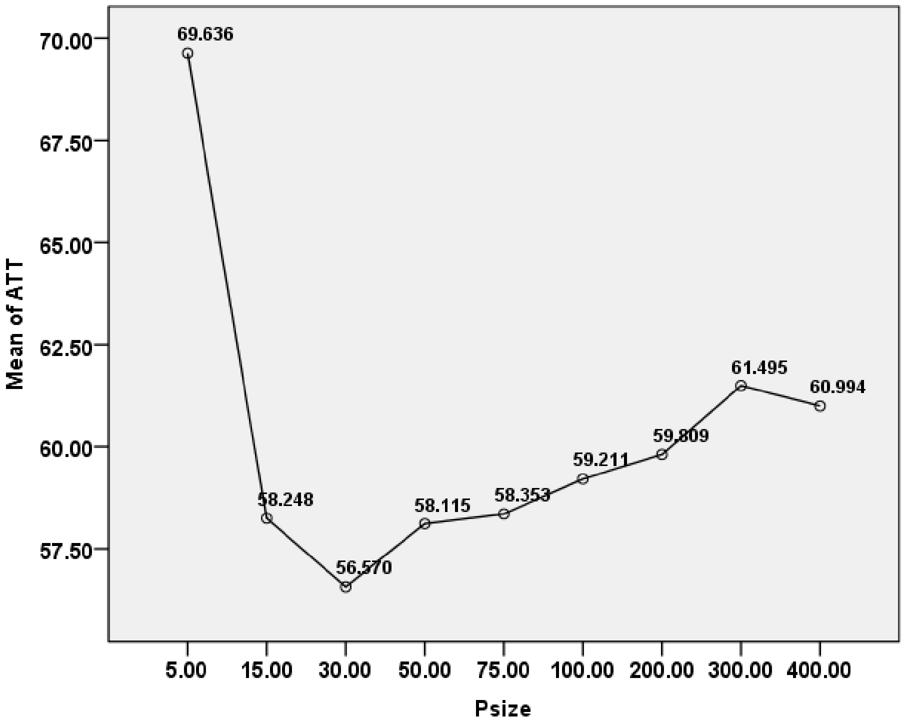


Figure 5.5. The mean results of Jaya by changing Psize on case 1 phase duration 10-60

Table 5.8 and Fig 5.5 reveal that the best result was obtained when the population size was ***30*** (mean = 56.5704). The performance of the algorithm improved as the value of population size increases from 5 to 30, and then the performance began to decline as the value of population size increases from 30 to 300. It can be seen from Fig4.6 that the strategy with the population size of **30** was faster than the other strategies. The convergence rate was almost similar to the population size of 15, 50, 75, and 100 which were better than the population size of 200,300,400 and 5.

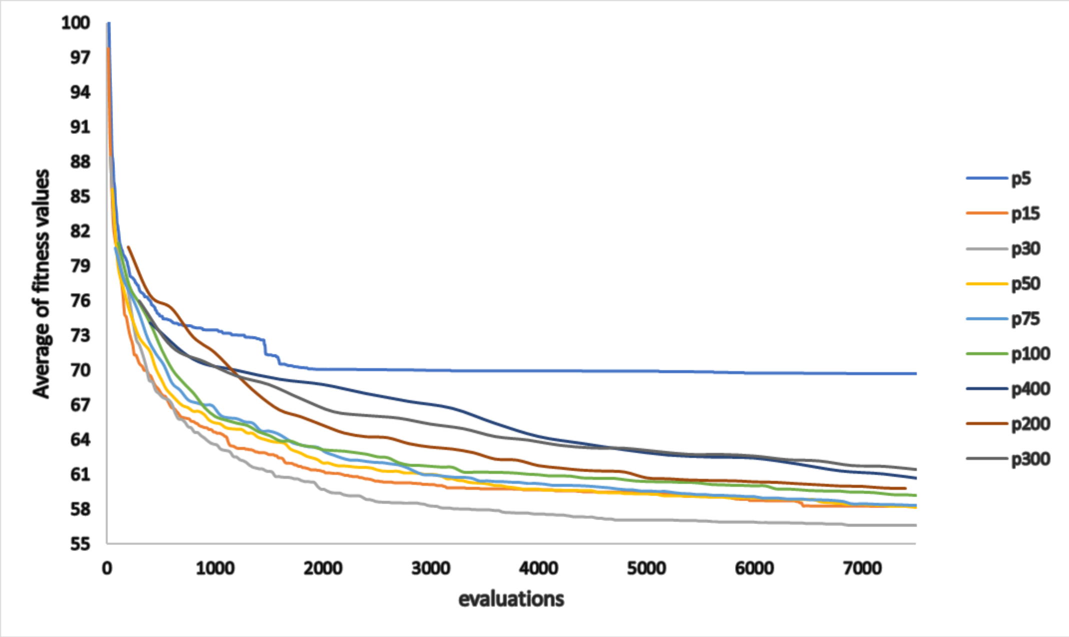


Figure 5.6: Convergence curves of Jaya by changing Psize on case 1 phase duration 10-60

Since the assumption of homogeneity of variances was violated by Leven's test(*F(8,171) = 13.258, p < .001*), we used Welch's F test which indicated that there was a statistically significant difference between the mean result of different population sizes (*F(8,69.658) = 26.410, p < .001*).

Post hoc comparisons (appendix A table 5.9) revealed that there was no statically significant difference between the results of population sizes in each subset as shown in table 5.10. Jaya algorithm with Psize of **30** had a significantly lower mean than other population sizes results except for Psize of 15. See table 5.10

Table 5.10: Homogeneous subsets of Psize (Jaya on case 1 phase duration 10-60)

Psize	Homogeneous subsets	

	1	2	3	4	Significant conclusions
30	56.5704				30 < 50,75,100,200,400,300,5
50		58.1147			50,15,75 < 400 , 300 , 5
15	58.2480	58.2480			100, 200, 300, 400 < 5
75		58.3533			
100		59.2105	59.2105		
200		59.8089	59.8089		
400			60.9945		
300			61.4950		
5.00				69.6357	

5.2.1.4 Performance and convergence speed of GA

Table 5.11: Descriptive statistics of GA on case study 1 with phase duration 10-60

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	60.6008	2.68076	.59944	59.3462	61.8555	56.63	67.34
15.00	60.3553	1.90948	.42697	59.4617	61.2490	57.64	63.13
30.00	60.6631	2.47753	.55399	59.5036	61.8226	56.47	66.12
50.00	59.2134	2.35981	.52767	58.1090	60.3179	55.47	64.07
75.00	59.6898	2.81880	.63030	58.3705	61.0090	56.46	66.94
100.00	58.9518	1.64101	.36694	58.1838	59.7198	56.44	63.11
200.0	57.9292	1.78985	.40022	57.0916	58.7669	55.10	61.40
300.0	**57.9045**	1.26747	.28341	57.3113	58.4977	55.70	60.71
400.0	58.0631	1.14783	.25666	57.5259	58.6003	54.81	59.61
Total	59.2635	2.30688	.17194	58.9242	59.6028	54.81	67.34

The bold values indicate best results

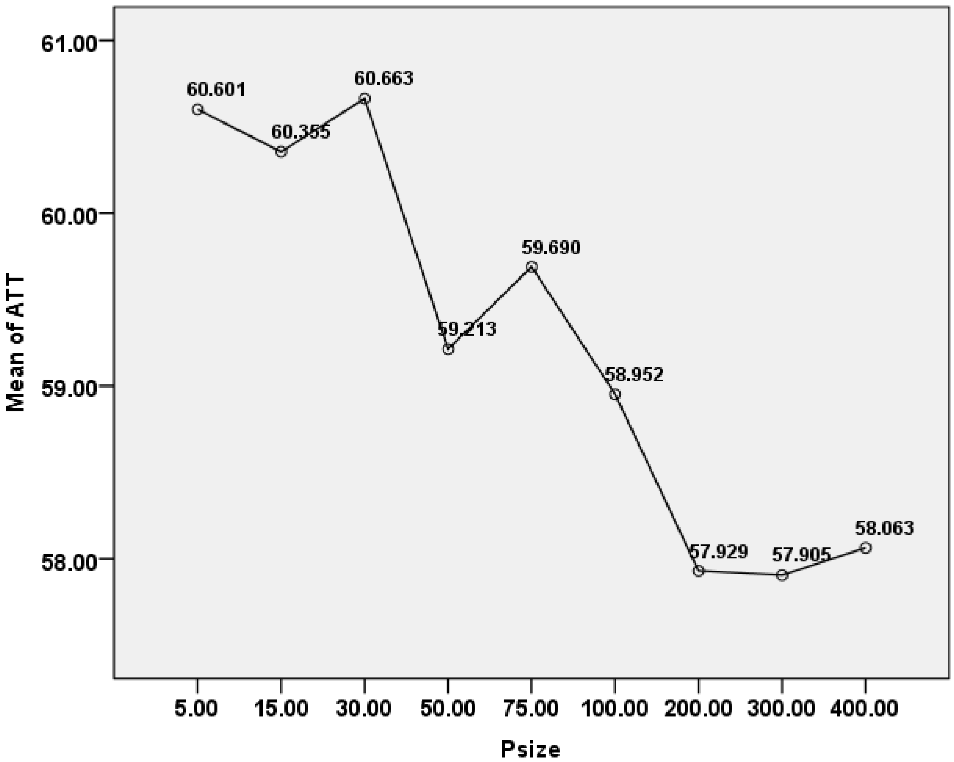


Figure 5.7: The mean results of GA by changing Psize on case 1 phase duration 10-60

Table 5.11 and Fig 5.7 reveal that the best result was obtained when the population size was ***300*** (mean = 57.9045). The algorithm with large population size values (200, 300, and 400) seems to lead to better performance than small population sizes do. It can be seen from Fig. 5.8 that the strategy with the population size of **30** was faster than other strategies. The convergence rate of the algorithm was almost similar as the population size increases from 5 to 100. During the first 2500 evaluations, the strategy with the population size of 5-100 was faster than the strategy with the population size of 200, 300, and 400. And then the speed of algorithm with the population size of 200, 300, and 400 started to improve.

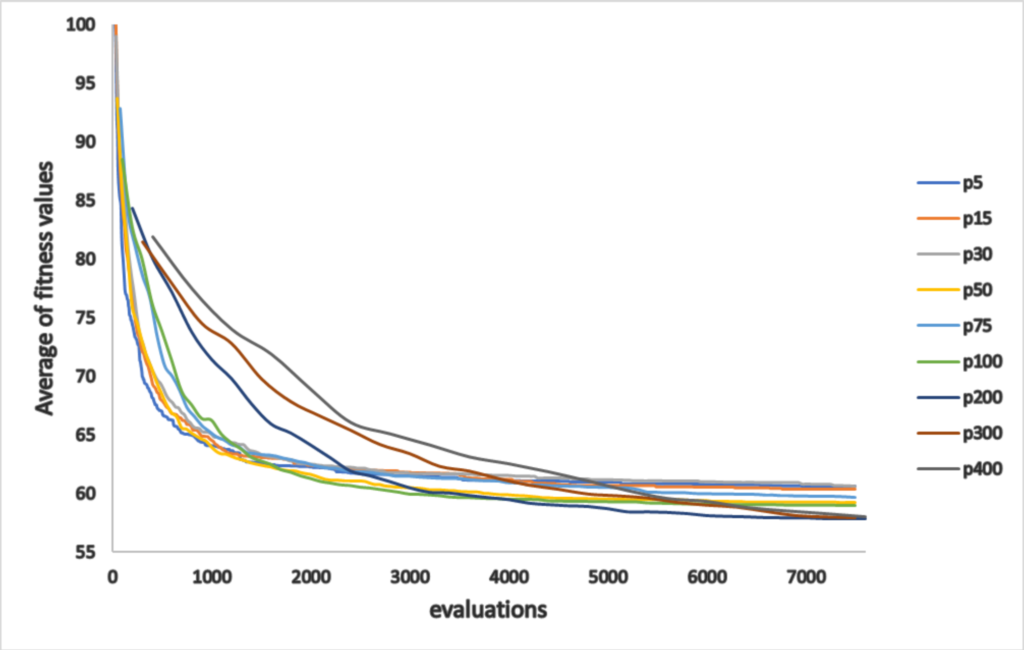


Figure 5.8: Convergence curves of GA by changing Psize on case 1 phase duration 10-60

Since the assumption of homogeneity of variances was violated by Leven's test(*F(8,171) = 2.518, p = .013 < 0.05*), we used Welch's F test which indicated that there was a statistically significant difference between the mean result of different population sizes (*F(8,70.788) = 5.890, p < .001*).

Games-Howell post hoc test revealed that there was no statically significant difference between the results of population sizes in each subset as shown in table 5.13. GA algorithm with the population size of **300, 200, and 400** had a significantly lower mean than population size of **15, 5, and 30**.

Table 5.13: Homogeneous subsets of Psize (GA on case 1 phase duration 10-60)

Psize	Homogeneous subsets 1	2	Significant conclusions
300	57.9045		**300, 200, 400 < 15, 5, 30**
200	57.9292		
400	58.0631		
100	58.9518	58.9518	
50	59.2134	59.2134	
75	59.6898	59.6898	
15		60.3553	
5		60.6008	
30		60.6631	

5.2.1.5 Performance and convergence speed of PSO

Table 5.14: Descriptive statistics of PSO on case study 1 with phase duration 10-60

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	Upper Bound	Minimum	Maximum
5.00	77.1981	11.11687	2.48581	71.9952	82.4009	57.07	102.19
15.00	69.1629	8.65479	1.93527	65.1123	73.2135	57.47	93.28
30.00	64.6863	6.77361	1.51462	61.5161	67.8564	55.84	79.26
50.00	65.1779	6.78152	1.51640	62.0041	68.3518	55.77	83.12
75.00	63.5191	4.43506	.99171	61.4434	65.5947	55.40	72.82
100.00	59.8847	3.90319	.87278	58.0579	61.7114	55.24	67.55
200.00	61.9251	4.44442	.99380	59.8451	64.0052	56.21	70.00
300.00	59.8754	2.41070	.53905	58.7471	61.0036	56.31	64.02
400.00	**59.8221**	3.34326	.74758	58.2574	61.3868	55.25	68.77
Total	64.5835	8.18423	.61002	63.3798	65.7873	55.24	102.19

The bold values indicate best results

The best result was obtained when population size was 400 (mean = 59.822). Between the population size of 5 and 30, there was a dramatic decrease in the mean. The performance of the algorithm with population sizes of 300 and 400 was almost similar (table 5.14, Fig 5.9)

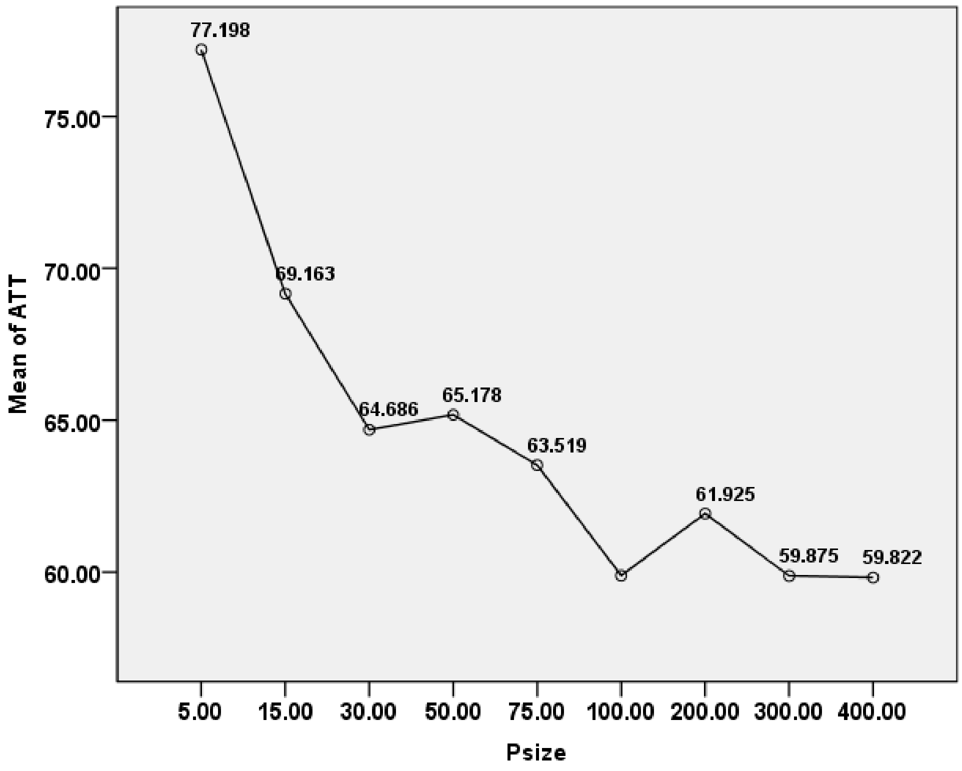


Figure5.9. The mean results of PSO by changing Psize on case1 phase duration10-60

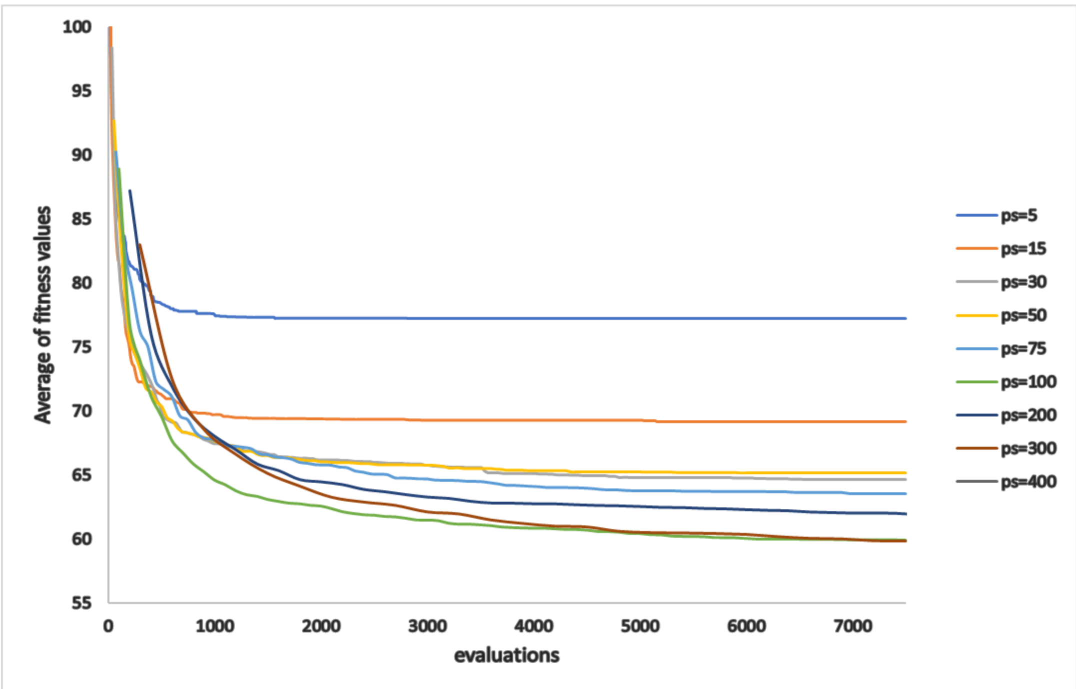


Figure 5.10: Convergence curves of PSO by changing Psize on case 1 phase duration 10-60

According to Fig 5.10, the convergence rate of the algorithm with the given population sizes can be ordered (from faster to slower) as the following: 100, 300, 200, 75, (50,30), 15, and 5 where the convergence was almost similar for the population size of 50 and 30.

Table 5.16: Homogeneous subsets of Psize (PSO on case 1 phase duration 10-60)

Psize	Homogeneous subsets			
	1	2	3	Significant conclusions
400	59.8221			**400, 300, 100, 200<15 , 5**
300	59.8754			
100	59.8847			**75, 30, 50 < 5**
200	61.9251			
75	63.5191	63.5191		
30	64.6863	64.6863		
50	65.1779	65.1779		
15		69.1629	69.1629	
5			77.1981	

Homogeneity of variances was violated as indicated by Leven's test($F(8,171) = 6.459, p < .001$). There was a statistically significant difference between the mean result of different population sizes as determined by Welch's F test ($F(8,70.389) = 10.281, p < .001$). Post hoc comparisons using the Games-Howell post hoc test were conducted. The results in Table 5.16 reveal that there was no statically significant difference between the results of population sizes that are listed under each subset. PSO algorithm with the population size of 400, 300, 100, and 200 had a significantly lower mean than the population size of 15 and 5. And the algorithm with the population size of 50, 75, and 30 had a significantly lower mean than the population size of 5. Post hoc comparisons are listed in Appendix A, Table 5.15.

Table 5.17: Summary results of statistical tests for algorithms, each with different population sizes (case 1 phase durations 10-60)

algorithm	Leven's test of homogeneity of variances				Welch F Test				One-way ANOVA	
	Leven Statistic	df1	df2	Sig.	Statistic[a]	df1	df2	Sig.	F	Sig.
TLBO	20.364	8	171	0.000	16.562	8	70.734	**0.000**	17.153	0.000
WTLBO	7.591	8	171	0.000	4.709	8	70.875	**0.000**	16.691	0.000
Jaya	13.258	8	171	0.000	26.410	8	69.658	**0.000**	30.824	0.000
GA	2.518	8	171	0.013	6.937	8	70.788	**0.000**	5.890	0.000
PS	6.459	8	171	0.000	10.281	8	70.389	**0.000**	15.885	0.000

p shown as .000, that is p < .001

5.2.1.6 Comparison of TLBO, WTLBO, Jaya, GA, and PSO

We compared the algorithms based on the best result of each algorithm (the best population size) obtained from previous experiments

Table 5.18: Comparative results of TLBO, WTLBO, Jaya, GA, and PSO case study 1 with phase duration 10-60

algorithm	Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
TLBO	**15**	56.6814	1.20917	.27038	56.1155	57.2473	54.58	58.87
WTLBO	**100**	61.4486	2.85035	.63736	60.1146	62.7827	57.52	70.13
Jaya	**30**	**56.5704**	.83032	.18567	56.1818	56.9590	55.29	58.57
GA	**300**	57.9045	1.26747	.28341	57.3113	58.4977	55.70	60.71
PSO	**400**	59.8221	3.34326	.74758	58.2574	61.3868	55.25	68.77

The bold values indicate best results

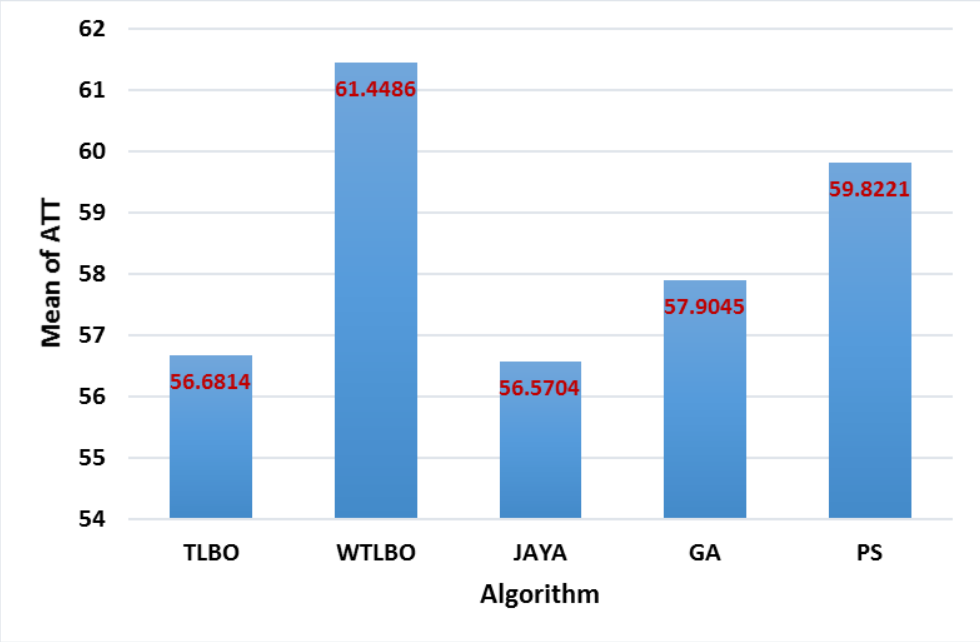


Figure 5.11: The best results of TLBO, WTLBO, Jaya, GA, PSO on case 1 phase duration 10-60

It can be seen from Table 5.18 and Fig 5.11 that the Jaya algorithm (mean = 56.57) and TLBO algorithm (mean = 56.681) were able to obtain better mean for the fitness value than other algorithms. WTLBO obtained the worst mean result (mean=61.449). The algorithms can be ordered based on the mean result (from better to worse) as follows: Jaya, TLBO, GA, PSO, WTLBO.

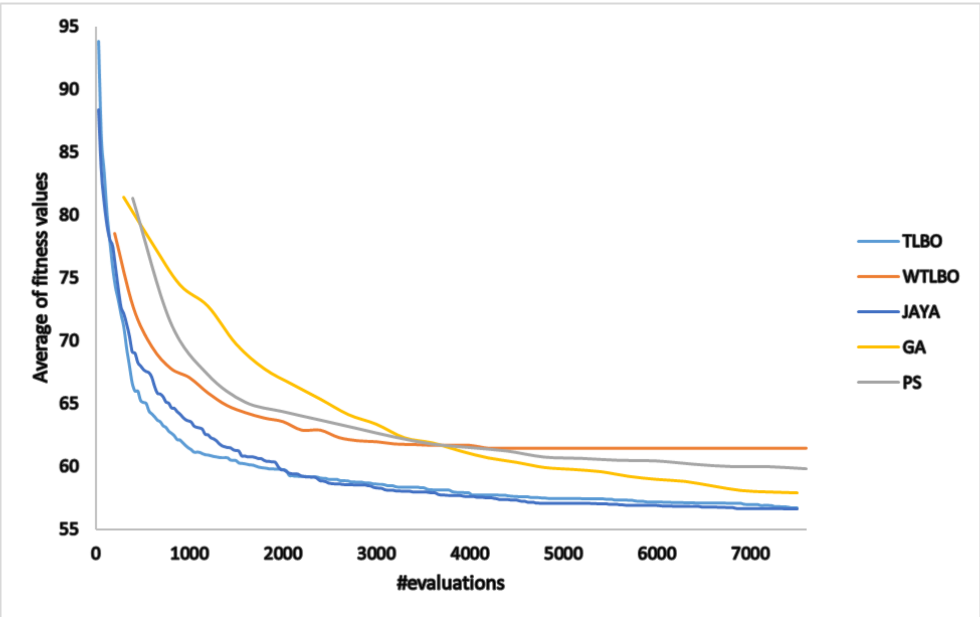


Figure 5.12: Convergence speed of TLBO, WTLBO, GA, PSO and Jaya on case study1 phase duration 10-60

The convergence rate of Jaya and TLBO algorithms was almost identical and better than the other algorithms. During the first 3700 evaluations, WTLBO was faster than GA and PSO algorithms. Then it was stuck into a local minimum and GA, PSO algorithms became faster (Fig 5.12).

Table 5.19:Statistical results for algorithms by Games-Howell post hoc test (case 1 phase duration 10-60)

Pair of comparison Algorithm I & J	Mean Difference (I-J)	95% Confidence Interval		P-value	Significance (better)
		Lower bound	Upper bound		
TLBO & WTLBO	-4.76726*	-6.7969	-2.7377	.000	**TLBO**
TLBO & Jaya	.11096	-.8340	1.0559	.997	-
TLBO & GA	-1.22315*	-2.3447	-.1016	.027	**TLBO**
TLBO & PSO	-3.14074*	-5.4836	-.7979	.005	**TLBO**
WTLBO & Jaya	4.87822*	2.9101	6.8463	.000	**Jaya**
WTLBO & GA	3.54411*	1.5027	5.5855	.000	**GA**
WTLBO & PSO	1.62653	-1.1896	4.4426	.473	-
Jaya & GA	-1.33411*	-2.3117	-.3565	.003	**Jaya**
Jaya & PSO	-3.25170*	-5.5432	-.9602	.003	**Jaya**
GA & PSO	-1.91758	-4.2703	.4351	.150	-

- : indicates that there is no significant between the compared algorithms.
****. The mean difference is significant at 0.05 level***

The Homogeneity of variances was violated as indicated by Leven's test($F(4,95) = 7.684, p < .001$). Welch's F test ($F(4,45.533) = 18.694, p < .001$) revealed that the mean values of ATT of five algorithms were not the same. Games-Howell post hoc test revealed that there were no statically significant differences between the results of (Jaya and TLBO), (GA and PSO), (PSO and WTLBO) algorithms. We can conclude that Jaya and TLBO were significantly performing better than other algorithms, and GA had a significantly lower mean than WTLBO (Table 5.19).

5.2.2 Phase 2 Experiments

Table 5.20: Phase 2 experiments settings

Green or red time (s)	Yellow time (s)	Population size	evaluations
10 - 100	3	5, 15, 30, 50, 75, 100, 200, 300, 400	7500

5.2.2.1 Performance and Convergence Speed of Basic TLBO

Table5.21: Descriptive statistics of Basic TLBO on case study 1 with phase duration 10-100

				95% Confidence Interval for Mean			
Psize	Mean	Std. Deviation	Std. Error	Lower Bound	Upper Bound	Minimum	Maximum
5.00	62.6587	7.08833	1.58500	59.3412	65.9761	56.35	81.63
15.00	56.7952	1.91279	.42771	55.9000	57.6904	53.83	62.05
30.00	**56.2996**	.93815	.20978	55.8605	56.7386	54.39	58.09
50.00	56.3679	1.24881	.27924	55.7834	56.9524	54.64	58.79
75.00	57.2853	.90972	.20342	56.8596	57.7111	55.61	59.38
100.00	57.3702	1.13565	.25394	56.8387	57.9017	55.71	59.69
200.00	58.8567	1.15321	.25786	58.3169	59.3964	56.87	61.08
300.00	59.4463	1.21290	.27121	58.8786	60.0139	57.90	62.42
400.00	62.3790	2.34748	.52491	61.2803	63.4777	58.45	66.86
Total	58.6065	3.53341	.26337	58.0868	59.1262	53.83	81.63

The bold values indicate best results

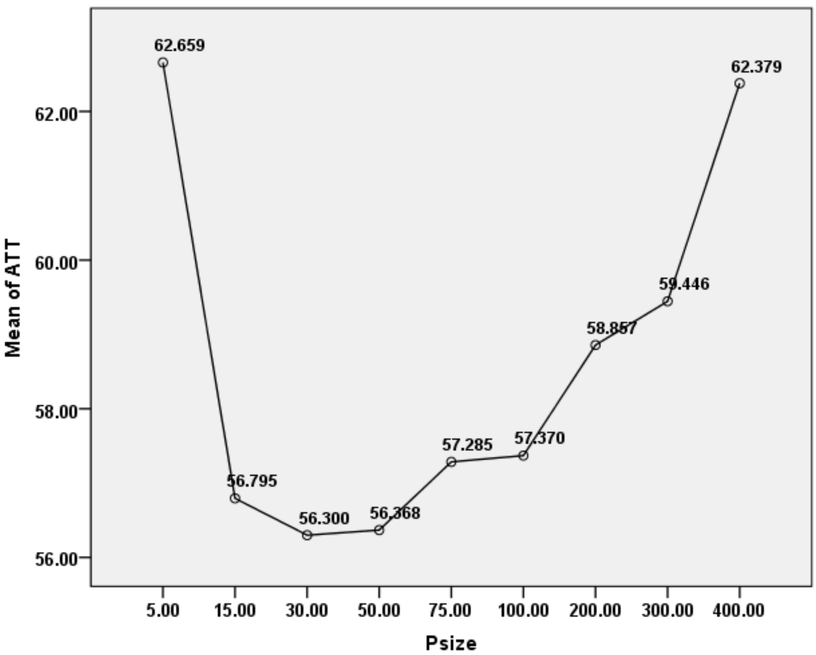


Figure 5.13: The mean results of TLBO by changing Psize on case 1 phase duration 10-100

It can be observed from Table 5.21 and Figure 5.13 that the algorithm with the population size of **30** produced the best result (*mean = 56.2996*), while the worst result was at the population size of 5 and 400. There was a dramatic fall in the mean value between population size of 5 and 15, while there was a little gradual change between 15 and 100. The strategy with the population size of **15** produced a better convergence rate as shown in figure 5.14. The convergence rate of the algorithm with the given population sizes can be ordered (from faster to slower) as the following: 15, 30, [50, 75], 100, [200,300], 5, and 400 where the convergence was almost similar to the values between brackets.

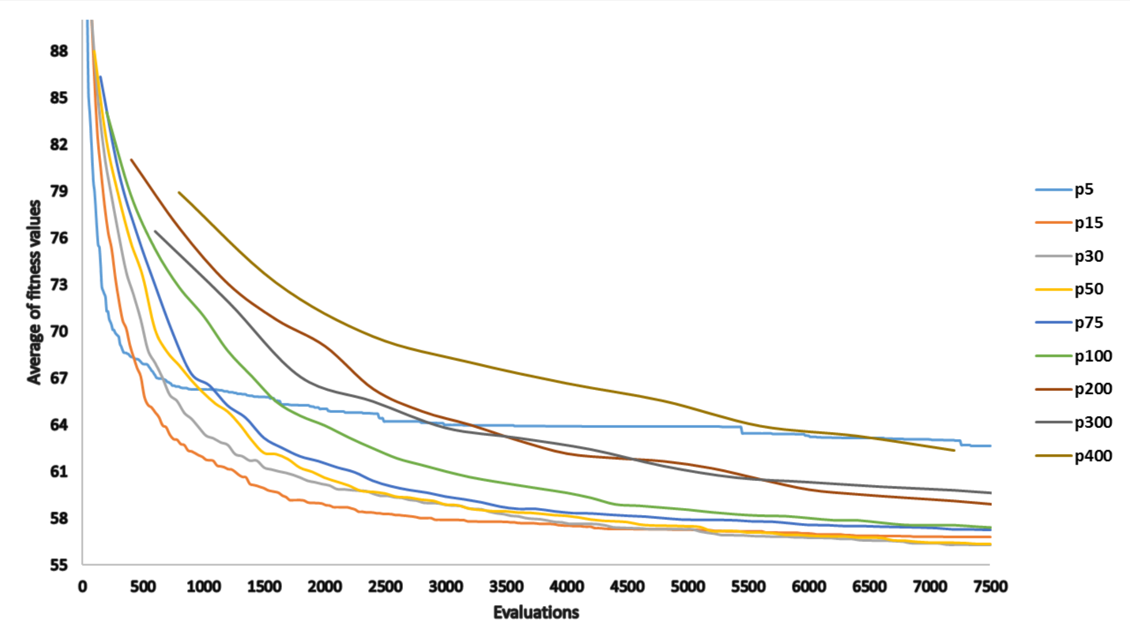


Figure 5.14: Convergence curves of TLBO by changing Psize on case1 phase duration 10-100

The Homogeneity of variances was violated as indicated by Leven's test($F(8,171) = 13.338, p < .001$). There was a statistically significant difference between the mean values as determined by Welch's F test (*F(8,70.838) = 26.711, p < .001*). Post hoc comparisons that use the Games-Howell post hoc test were conducted (Appendix A, Table 5.22). The results in table 5.23 revealed that there was no statically significant difference between the results of population sizes that are listed under each subset. The strategy with populations sizes of 15, 30, 50, 75, and 100 had a significantly lower result than populations sizes of 200, 300, 400, and 5.

Table 5.23: Homogeneous subsets of Psize (TLBO on case1 phase duration10-100)

Psize	Homogeneous subsets 1	2	3	4	Significant conclusions
30	56.2996				30< 75,100,200,300,400,5
50	56.3679	56.3679			50,15,75,100 < 200,300,400,5
15	56.7952	56.7952			200,300 < 400, 5
75		57.2853			
100		57.3702			
200			58.8567		
300			59.4463		
400				62.3790	
5				62.6587	

5.2.2.2 Performance and Convergence Speed of WTLBO

Table 5.24: Descriptive statistics of WTLBO on case study 1 with phase duration 10-100

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	Upper Bound	Minimum	Maximum
5.00	74.6681	12.59898	2.81722	68.7716	80.5646	59.67	100.74
15.00	65.1058	7.00337	1.56600	61.8281	68.3834	57.61	86.30
30.00	61.4393	2.06953	.46276	60.4708	62.4079	58.77	66.25

50.00	**60.8101**	2.23417	.49958	59.7645	61.8558	57.96	66.00
75.00	61.4900	1.66515	.37234	60.7107	62.2693	59.58	65.19
100.00	61.1128	1.77960	.39793	60.2800	61.9457	57.62	64.88
200.00	61.9175	1.68433	.37663	61.1292	62.7058	59.22	65.68
300.00	61.6889	2.41687	.54043	60.5578	62.8201	57.54	68.96
400.00	63.3597	3.31926	.74221	61.8062	64.9132	58.95	68.73
Total	63.5103	6.55773	.48878	62.5457	64.4748	57.54	100.74

The bold values indicate best results

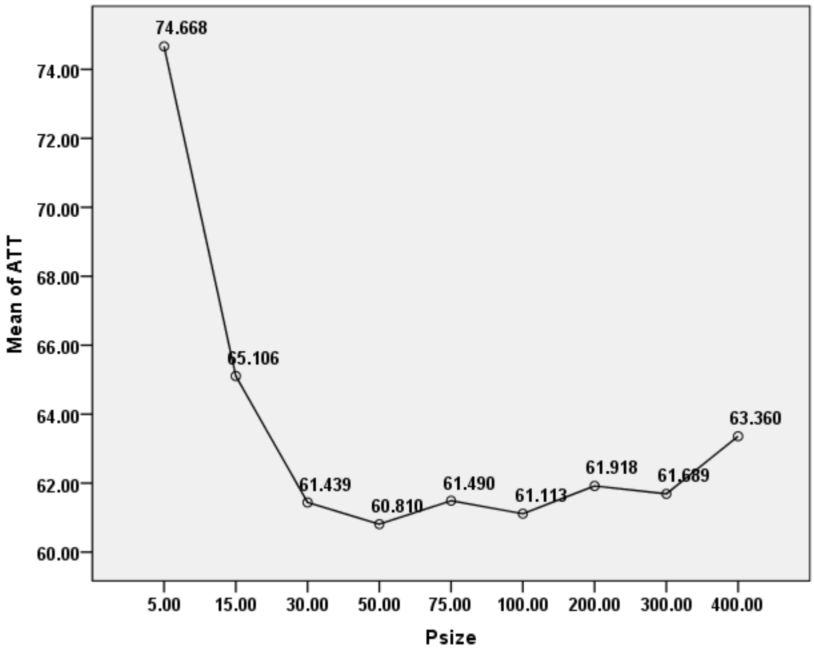


Figure 5.15: The mean results of WTLBO by changing Psize on case 1 phase duration 10-100

Table 5.24 and Fig 5.15 show that the optimal mean of fitness value seems to occur when the population size was ***50*** (mean = 60.8101). And between the population size of 5 and 30, there was a marked fall in the mean value. While there was no dramatic difference between the results as the population size increased from 30 to 400. The convergence rate of WTLBO algorithm can be ordered according to the curves of different population sizes (from faster to slower) as follows: [30, 50] , [75, 100], [200, 300], 400 15, 5 where the speed of the algorithm using the population size values between the brackets was almost identical.

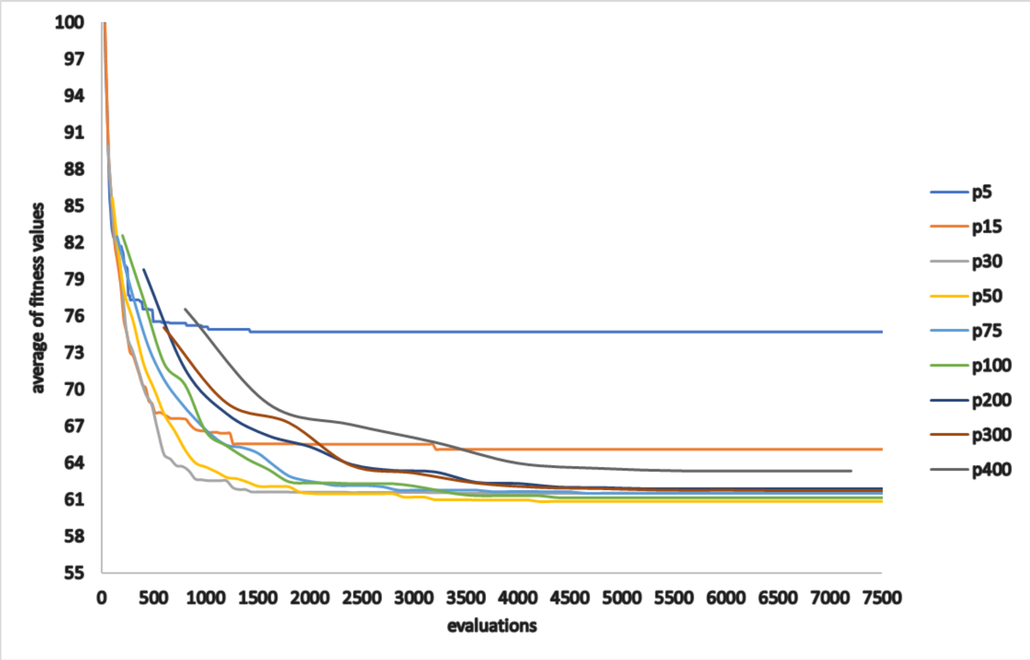


Figure 5.16: Convergence curves of WTLBO by changing Psize on case1 phase duration10-100

The Homogeneity of variances was violated as indicated by Leven's test(*F(8,171) = 30.199, p < .001*). There was a statistically significant difference between the mean result of different population sizes as determined by Welch's F test (*F(8,70.777) = 4.346, p < .001*). The results of Games-Howell post hoc test reveal that the mean values in each group in Table 5.26 were statistically equal, and the algorithm with the population size of 50,100,30,75,300,200 and 400 had a significantly lower mean than population size of 5. See (Table 5.25 in Appendix A)

Table 5.26: Homogeneous subsets of Psize (WTLBO on case 1 phase duration 10-100)

Psize	Homogeneous subsets		Significant conclusions
	1	2	
50	60.8101		50,100,30,75,300,200,400 < 5
100	61.1128		
30	61.4393		
75	61.4900		
300	61.6889		
200	61.9175		
400	63.3597		
15	65.1058	65.1058	
5		74.6681	

5.2.2.3 Performance and Convergence Speed of Jaya

Table 5.27: Descriptive statistics of Jaya on case study 1 with phase duration 10-100

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	75.0240	11.19011	2.50218	69.7868	80.2611	60.02	102.45
15.00	57.8474	4.14489	.92683	55.9075	59.7872	54.70	72.46
30.00	**57.0816**	1.85551	.41490	56.2132	57.9500	54.84	63.43
50.00	58.0467	3.61850	.80912	56.3532	59.7402	55.29	67.80
75.00	57.1236	.89116	.19927	56.7066	57.5407	55.74	58.94
100.00	58.7616	3.13406	.70080	57.2948	60.2284	55.46	67.96
200.00	59.3954	3.33158	.74496	57.8362	60.9547	56.62	69.05
300.00	61.3592	3.66131	.81869	59.6456	63.0727	56.42	70.04
400.00	61.3235	2.55967	.57236	60.1256	62.5215	56.52	67.11
Total	60.6625	7.03610	.52444	59.6277	61.6974	54.70	102.45

The bold values indicate best results

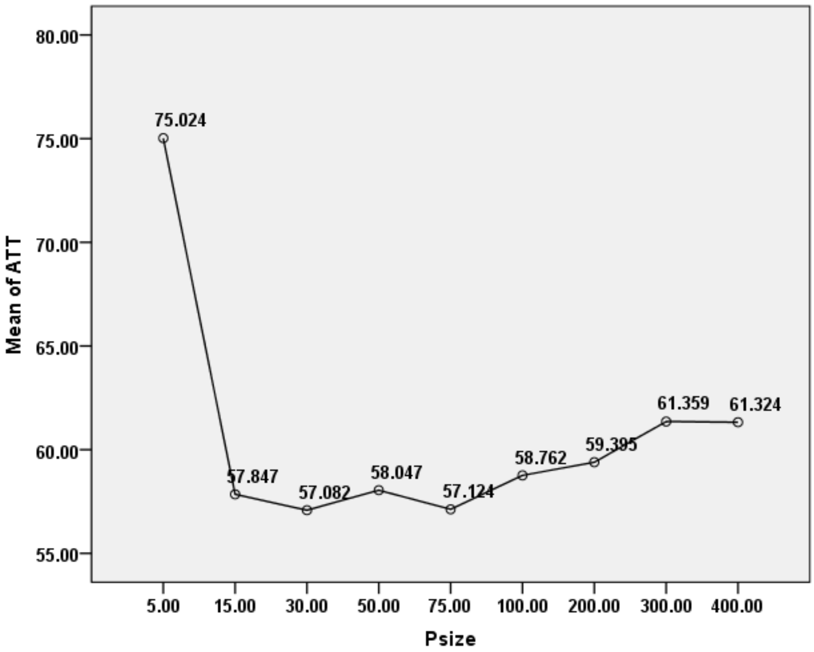


Figure 5.17: The mean results of Jaya by changing Psize on case 1 phase duration 10-100

The best mean value was obtained when the population size was ***30*** (mean = 57.0816). There was a significant fall in the mean value between the population size of 5 and 15, while the result was almost identical when the population size increased from 15 to 200 (table 5.27, fig 5.17). Fig5.18 shows that population size of 5 produced a bad convergence, while the population size of **15** and **30** yielded better convergence. The convergence rate of algorithm decreased as the population size increased from 15 to 400.

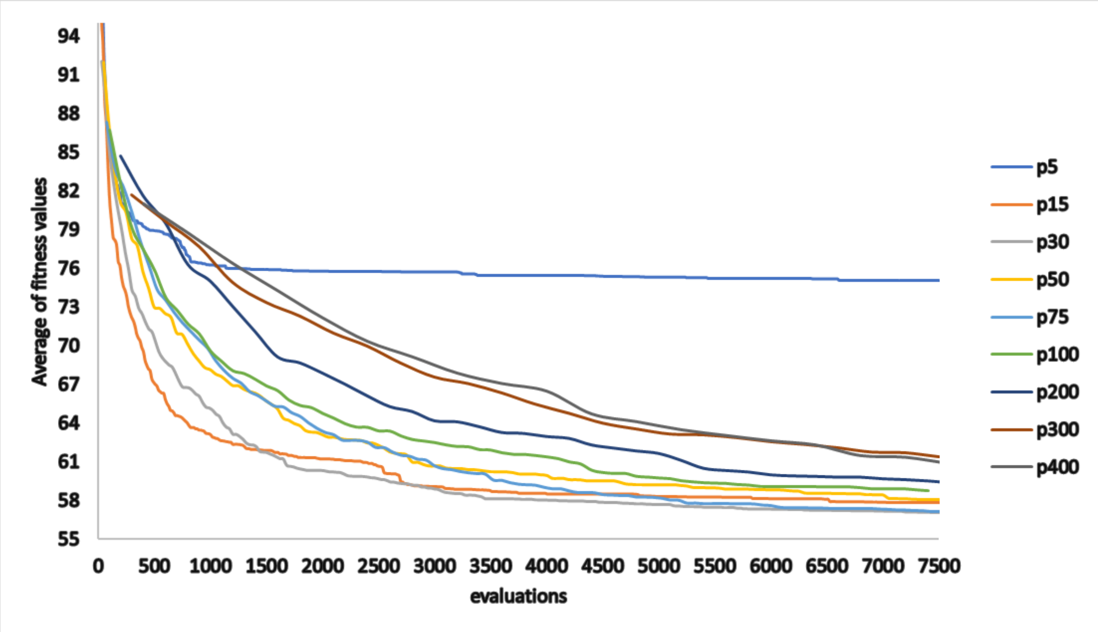


Figure 5.18: Convergence curves of Jaya by changing Psize on case 1 phase duration 10-100

The assumption of homogeneity of variances was not met by Leven's test(*F(8,171) = 6.614, p < .001*). We used Welch's F test which indicated that at least there was a pair of mean values which was significantly different. (*F(8,68.586) = 14.638, p < .001*).

Post hoc comparisons (appendix A table 5.28) reveal that there was no statically significant difference between the results of population sizes that are listed under each subset as shown in table 5.29. Jaya algorithm with the population size of **5** had significantly the highest mean, and population size of 30 and 75 had a significantly lower mean than population size of 400 and 300.

Table 5.29: Homogeneous subsets of Psize (Jaya on case 1 phase duration 10-100)

Psize	Homogeneous subsets			
	1	2	3	Significant conclusions
30	57.0816			30,75,15,50,100,200,400,300 < 5
75	57.1236			
15	57.8474	57.8474		30, 75 < 400, 300
50	58.0467	58.0467		
100	58.7616	58.7616		
200	59.3954	59.3954		
400		61.3235		
300		61.3592		
5			75.0240	

5.2.2.4 Performance and convergence speed of GA

Table 5.30: Descriptive statistics of GA on case study 1 with phase duration 10-100

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	68.7782	6.15236	1.37571	65.8988	71.6576	60.49	85.66
15.00	66.8076	5.86888	1.31232	64.0609	69.5544	56.82	81.69
30.00	64.1352	4.24336	.94884	62.1493	66.1212	56.44	70.47
50.00	64.9315	3.73804	.83585	63.1821	66.6810	59.78	74.25
75.00	61.2812	2.47472	.55336	60.1230	62.4394	57.08	66.00
100.00	62.1194	2.13445	.47728	61.1205	63.1184	57.09	65.82
200	**59.2697**	2.12045	.47415	58.2773	60.2621	55.83	62.47
300	59.5500	1.99593	.44630	58.6159	60.4841	56.95	64.13
400	60.1426	1.78435	.39899	59.3075	60.9777	57.22	63.56
Total	63.0017	4.85435	.36182	62.2877	63.7157	55.83	85.66

The bold values indicate best results

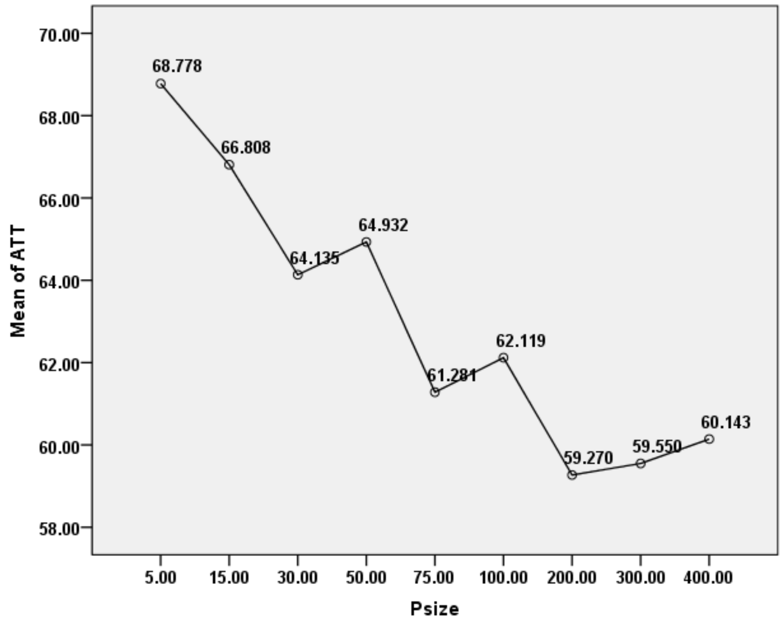


Figure 5.19: The mean results of GA by changing Psize on case 1 phase duration 10-100

The mean value gradually decreased between the population size of 5 and 30, and then increased slightly at the population size of 50, then returned to decrease at the population size of 75, then increased again at the population size of 100, then decreased to reach the best value at the population size of 200 (mean = 59.2697). The result was almost identical when the population size increased from 200 to 400. It seems that the algorithm with the large population size values (200, 300, 400) produced a better performance than the small population sizes (Table 5.30, Fig 5.19).

It can be observed from figure 5.20 that during the first evaluations, the speed of the algorithm with the population size of 200 to 400 was slow, and then improved to become better. Hence, the convergence rate can be ordered according to the curves of different population sizes (from faster to slower) as follows: 200, 300, 400, [100, 75], [50, 30], 15, 5 where the speed of the algorithm using the population size values between the brackets was almost identical.

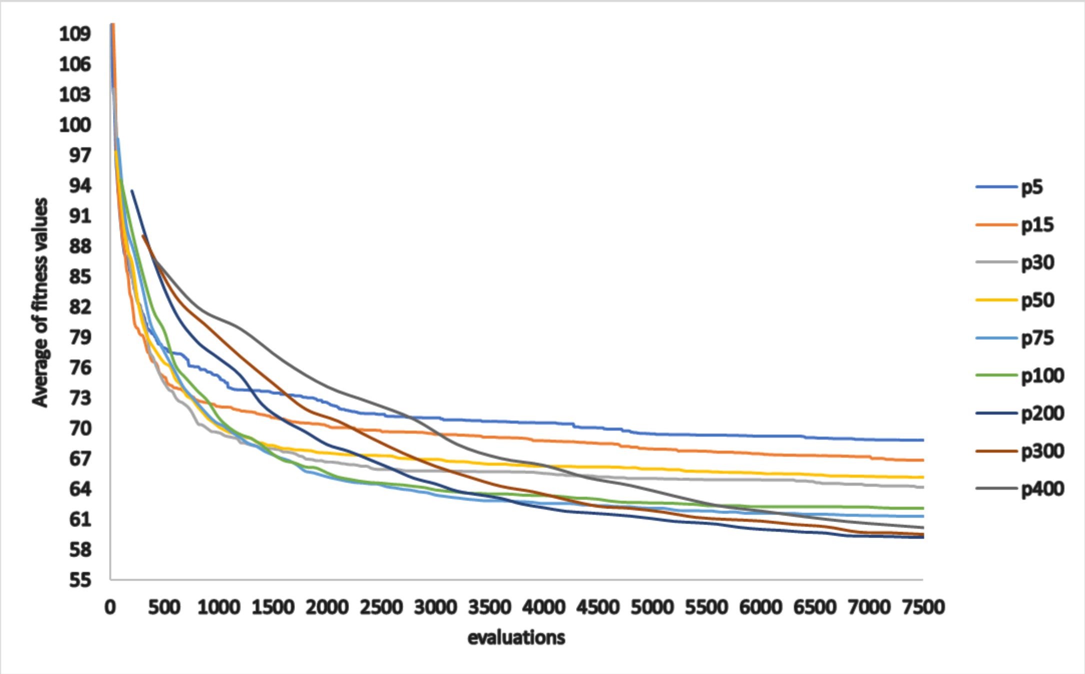


Figure 5.20: Convergence curves of GA by changing Psize on case 1 phase duration 10-100

Since the assumption of homogeneity of variances was violated by Leven's test($F(8,171) = 5.313, p < .001$), we used Welch's F test which indicated that there was a statistically significant difference between the mean result of different population sizes ($F(8,70.748) = 13.918, p < .001$).

Games-Howell post hoc test revealed that there was no statically significant difference between the results in each subset as shown in table 5.32. GA algorithm with the population size of **200** and **300** had a significantly lower mean than the population size of (**100, 30, 50, 15, 5**). And the algorithm with the population size of **400** had a significantly lower mean than the population sizes of **(30, 50, 15, 5).** The algorithm with the population size of **75** had a significantly lower mean than the population sizes of (**50, 15, 5**). Hence, there was statically significant evidence that the large population size gave better results than small size one. (see Appendix A, Table 5.31)

Table 5.32: Homogeneous subsets of Psize (GA on case 1 phase duration 10-100)

Psize	Homogeneous subsets				
	1	2	3	4	Significant conclusions
200	59.2697				
300	59.5500				**200, 300 <100, 30,50, 15, 5**
400	60.1426	60.1426			
75	61.2812	61.2812	61.2812		**400<30,50,15,5**
100		62.1194	62.1194		**75<50, 15, 5**
30			64.1352	64.1352	
50				64.9315	
15				66.8076	
5				68.7782	

5.2.2.5 Performance and Convergence Speed of PSO

Table 5.33: Descriptive statistics of PSO on case study 1 with phase duration 10-100

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	81.4132	13.47932	3.01407	75.1046	87.7217	58.83	106.72
15.00	72.1039	13.98281	3.12665	65.5597	78.6481	57.12	96.19
30.00	67.1355	8.92306	1.99526	62.9594	71.3117	56.24	84.87
50.00	66.0826	7.80040	1.74422	62.4319	69.7333	55.70	81.23
75.00	62.4027	6.04143	1.35090	59.5753	65.2302	55.75	73.06
100.00	63.9739	8.15039	1.82248	60.1594	67.7884	55.14	78.90
200	**60.5989**	4.78472	1.06990	58.3596	62.8382	56.42	70.65
300	60.9364	5.70320	1.27527	58.2672	63.6056	55.97	73.84
400	60.9439	3.93557	.88002	59.1020	62.7858	56.42	69.58
Total	66.1768	10.72875	.79967	64.5988	67.7548	55.14	106.72

The bold values indicate best results

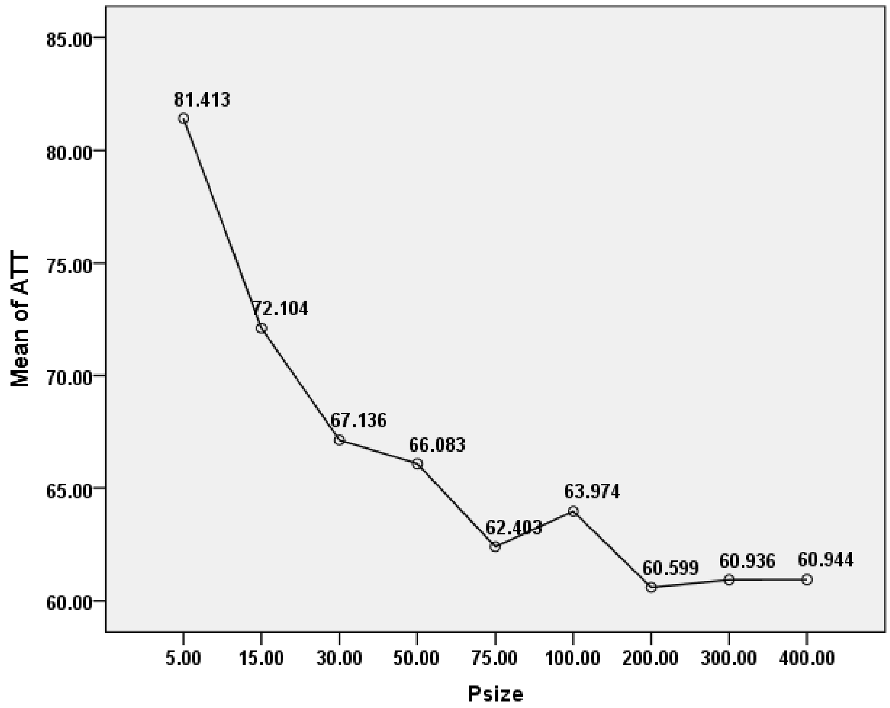


Figure 5.21: The mean results of PSO by changing Psize on case1 phase duration10-100

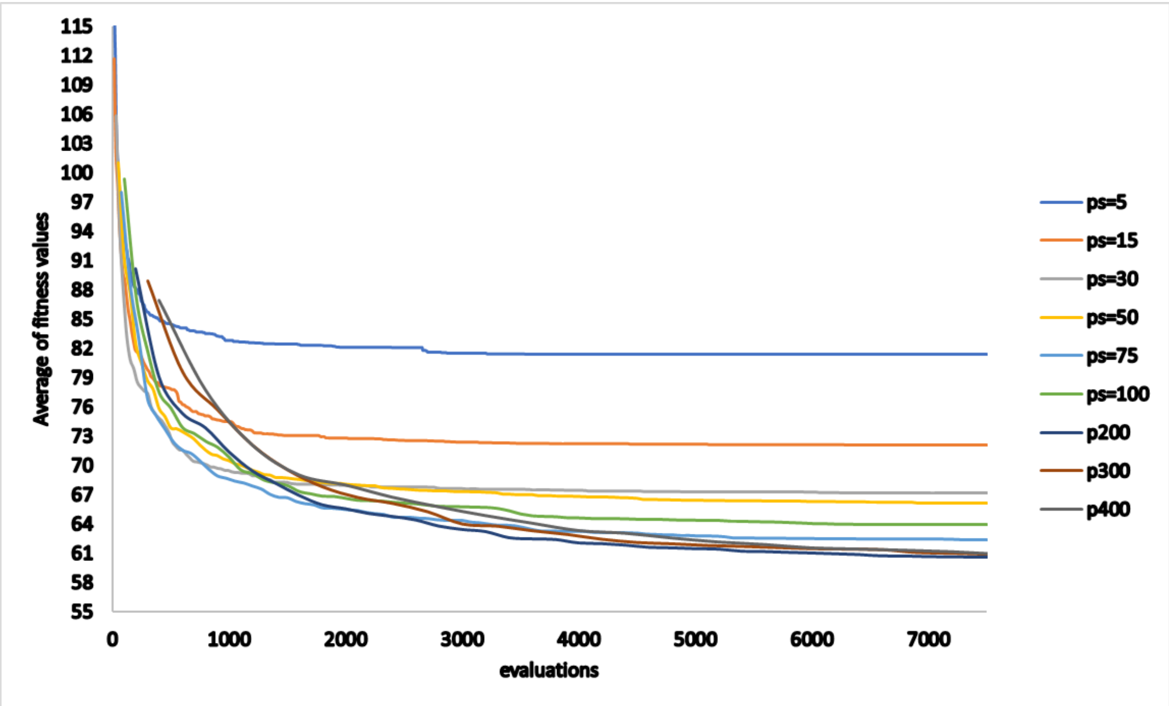


Figure 5.2: Convergence curves of PSO by changing Psize on case 1 phase duration 10-60

Table 5.33 and Fig 5.21 reveal that there was a marked decrease in the mean value between the population size of 5 and 15, and then there was a gradual decrease when the population size increased from 15 to 75. The best result was obtained when the population size was ***200*** (mean = 60.599). The performance of the algorithm with the population size of 200, 300, and 400 was almost similar.

According to Fig 5.22, the convergence rate of the algorithm with the given population sizes can be ordered (from faster to slower) as the following: 200, [300, 400], 75, 100, [50,30], 15, and 5 where the convergence speed was almost similar for the population size of (400, 300) and for (50, 30). The figure also shows that the difference in convergence speed was not great between the population sizes of 30 to 400, while the difference was obvious between 15 and 5.

Table 5.35: Homogeneous subsets of Psize (PSO on case 1 phase duration 10-100)

Psize	Homogeneous subsets			
	1	2	3	Significant conclusions
200	60.5989			**200, 300, 400 < 15 , 5**
300	60.9364			
400	60.9439			**75, 100, 50, 30 < 5**
75	62.4027	62.4027		
100	63.9739	63.9739		
50	66.0826	66.0826		
30	67.1355	67.1355		
15		72.1039	72.1039	
0			81.4132	

The homogeneity of variances was violated as indicated by Leven's test($F(8,171) = 12.763, p < .001$). Therefore, we performed Welch's F test which indicated that we strongly rejected the hypothesis (All means are equal) ($F(8,70.629) = 7.805, p < .001$). Games-Howell post hoc test was conducted (Appendix A, Table 5.34). The results in Table 5.35 reveal that there was no statically significant difference between the results of population sizes that are listed under each subset. The PSO algorithm with the population size of 200, 300, and 400 had a significantly lower mean than population size of 5 and 15. Also, the population size of 75, 100, 50, and 30 had a significantly lower mean than population size of 5.

Table 5.36: Summary results of statistical tests for algorithms, each with different population sizes (case 1 phase durations 10-100)

algorithm	Leven's test of homogeneity of variances				Welch F Test				One-way ANOVA	
	Leven Statistic	df1	df2	Sig.	Statistic[a]	df1	df2	Sig.	F	Sig.
TLBO	13.338	8	171	0.000	26.711	8	70.838	**0.000**	16.278	0.000
WTLBO	30.199	8	171	0.000	4.346	8	70.777	**0.000**	14.308	0.000
Jaya	6.614	8	171	0.000	14.638	8	68.586	**0.000**	28.321	0.000
GA	5.313	8	171	0.000	13.918	8	70.748	**0.000**	16.103	0.000
PSO	12.763	8	171	0.000	7.805	8	70.629	**0.000**	12.145	0.000

* p shown as 0.000, that is p <0 .001

From the previous results and plots, we can see how increasing the population size (i.e. n>10) provides an improvement in results. The justification for this behavior is that larger population size maintains a sufficient diversity which may improve the ability for the evaluated algorithms to explore several parts of the search space using a sufficient number of individuals and thus reducing the probability of premature convergence.

It seems clear from the results of TLBO and Jaya algorithms (as shown in Figures (5.1, 5,13, 5.5) that the continued increase in the population size (i.e. n>100) may leads to undesirable results. So, too much diversity is not always good. A possible interpretation of this behavior is that when the number of allowed evaluations is fixed for all population sizes, then increasing the population size leads to decrease the number of iterations which reduces the algorithm's power in the use of exploration and exploitation approaches (i.e. performance tends to be random), and also leads to early termination which is insufficient for convergence to acceptable solution.

On the other hand, the obtained result of GA and PSO shows a clear improvement with larger population size (i.e. n>200) (see Figures 5.7, 5.9, 5.19, 5.21). A possible reason for that is due to the parameters settings which affect the ability of GA and PSO to balance between exploration and exploitation. Besides, tuning of population size must be done in conjunction with the other specific parameters (i.e. they are inter-related) to find a proper combination of these parameters.

5.2.2.6 Comparison of TLBO, WTLBO, Jaya, GA, and PSO

We compared the algorithms based on the best result of each algorithm (the best population size) obtained from previous experiments

Table 5.37: Comparative results of TLBO, WTLBO, Jaya, GA, and PSO case study 1 with phase duration 10-100

algorithm		Mean	Std.	Std.	95% Confidence Interval for Mean	Minimum	Maximum

	Psize		Deviation	Error	Lower Bound	Upper Bound		
TLBO	**30**	**56.2996**	.93815	.20978	55.8605	56.7386	54.39	58.09
WTLBO	**50**	60.8101	2.23417	.49958	59.7645	61.8558	57.96	66.00
JAYA	**30**	57.0816	1.85551	.41490	56.2132	57.9500	54.84	63.43
GA	**200**	59.2697	2.12045	.47415	58.2773	60.2621	55.83	62.47
PSO	**200**	60.5989	4.78472	1.06990	58.3596	62.8382	56.42	70.65

The bold values indicate best results

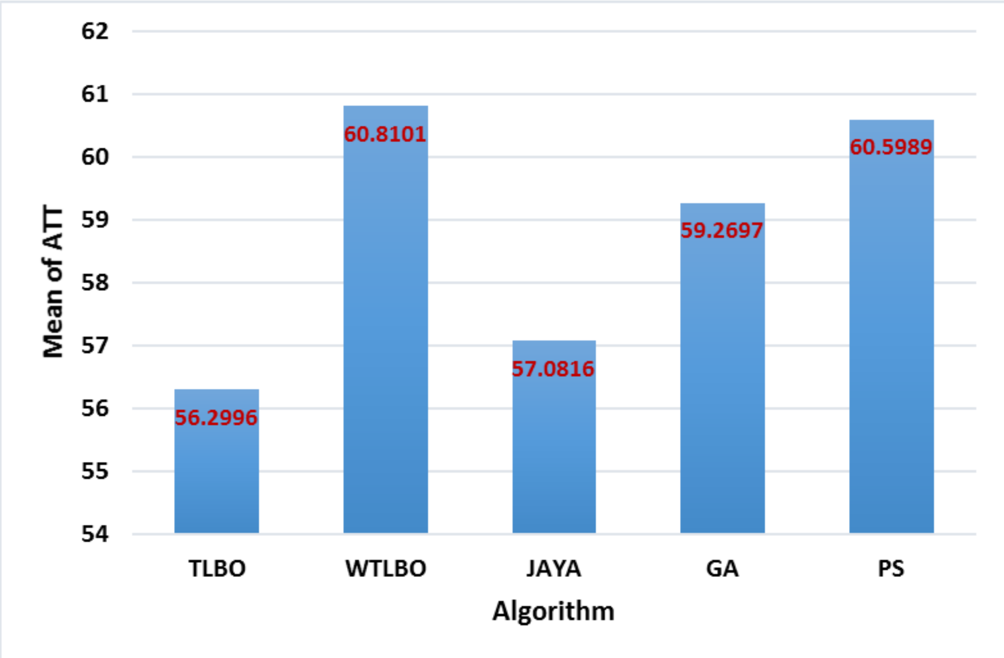


Figure 5.23: The best results of TLBO, WTLBO, Jaya, GA, PS on case 1 phase duration 10-100

Table 5.37 reveals that TLBO algorithm had obtained the best mean (56.2996), minimum (54.39), and standard deviation (0.93815) results. The algorithms can be ordered based on the mean value (from better to worse) as follows: TLBO, Jaya, GA, PSO, and WTLBO.

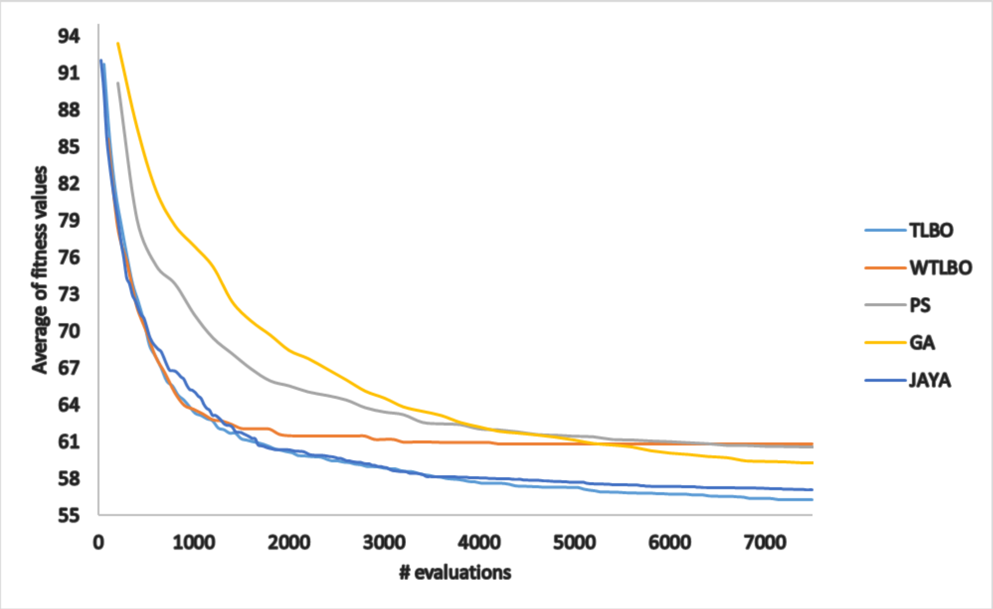


Figure 5.24: Convergence speed of TLBO, WTLBO, GA, PS and Jaya on case study1 phase duration 10-60

The convergence speed of Jaya and TLBO algorithms was almost identical and better than WTLBO, GA, and PSO algorithms. During the first evaluations, the speed of WTLBO was almost similar to the speed of TLBO and Jaya, and then it was stuck into a local minimum but remained better than PSO. GA was the slowest but gave a better solution quality than PSO and WTLBO at the end of maximum allowable evaluations. (Fig 5.24)

The homogeneity of variances was violated as indicated by Leven's test($F(4,95) = 13.310, p < .001$). Welch's F test ($F(4,44.584) = 23.706, p < .001$) revealed that the means of fitness value of five algorithms were not the same. From Games-Howell post hoc test (table 5.38) we concluded that: 1) there was no statically significant difference between the mean results of ***Jaya*** and ***TLBO***. 2) There was no statically significant difference between the mean results of ***WTLBO, GA, and PSO***. 3) both Jaya and

TLBO were statically significantly performing better than WTLBO, GA, and PSO.

Table 5.38: Statistical results for algorithms by Games-Howell post hoc test (case 1 phase duration 10-100)

Pair of comparison Algorithm I & J	Mean Difference (I-J)	95% Confidence Interval		P-value	Significance (better)
		Lower bound	Upper bound		
TLBO & WTLBO	-4.51059-*	-6.0996-	-2.9216-	.000	**TLBO**
TLBO & Jaya	-.78202-	-2.1362-	.5721	.461	-
TLBO & GA	-2.97017-*	-4.4878-	-1.4525-	.000	**TLBO**
TLBO & PSO	-4.29935-*	-7.5550-	-1.0437-	.006	**TLBO**
WTLBO & Jaya	3.72857*	1.8662	5.5909	.000	**Jaya**
WTLBO & GA	1.54042	-.4318-	3.5126	.189	-
WTLBO & PSO	.21124	-3.2383-	3.6607	1.000	-
Jaya & GA	-2.18815-*	-3.9936-	-.3827-	.011	**Jaya**
Jaya & PSO	-3.51733-*	-6.8917-	-.1430-	.038	**Jaya**
GA & PSO	-1.32918-	-4.7545-	2.0961	.786	-

- : indicates that there is no significant between the compared algorithms.
****. The mean difference is significant at 0.05 level***

5.3 Comparing Optimization Techniques on Case Study 2

Table 5.39: C~~c~~ase 2 experiments settings

Green or red time (s)	Yellow time (s)	Population size	evaluations
10 - 60	3	5, 15, 30, 50, 75, 100	15000

5.3.1 Performance and Convergence Speed of Basic TLBO

Table 5.40: Descriptive statistics of Basic TLBO on case study 2

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
5.00	116.2205	20.54194	4.59332	106.6066	125.8344	95.12	159.17
15.00	96.2170	3.53678	.79085	94.5617	97.8723	91.91	102.25
30.00	95.3710	2.89752	.64791	94.0150	96.7271	91.68	104.72
50.00	**94.7170**	1.76221	.39404	93.8922	95.5417	91.89	98.11
75.00	95.4152	2.17131	.48552	94.3990	96.4314	91.55	102.12
100.0	96.0252	2.19308	.49039	94.9988	97.0515	92.32	100.55
Total	98.9943	11.52444	1.05203	96.9112	101.0774	91.55	159.17

The bold values indicate best results

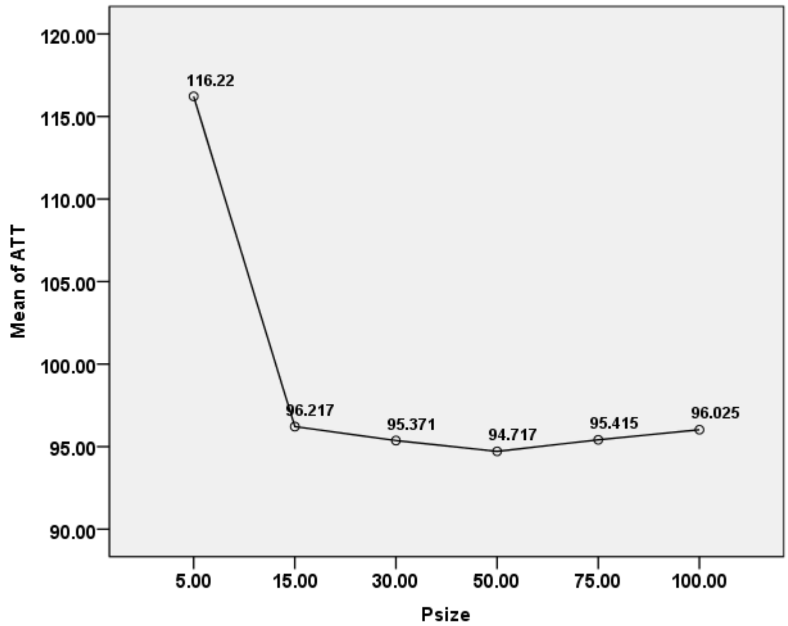


Figure 525: The mean results of TLBO by changing Psize on case 2

It can be observed from Table 45.40, and fig 5.25 that the algorithm with the population size of **50** gave the best result (mean = 94.7170). There was a dramatic fall in the mean value between population size of 5 and 15. The solution quality when the population size increased from 15 to 100 was almost the same. It was clear that the algorithm with the population size of

15 had a better convergence rate, and the algorithm with the population size of 100 didn't have a good convergence rate with respect to other population sizes (Fig 5.26).

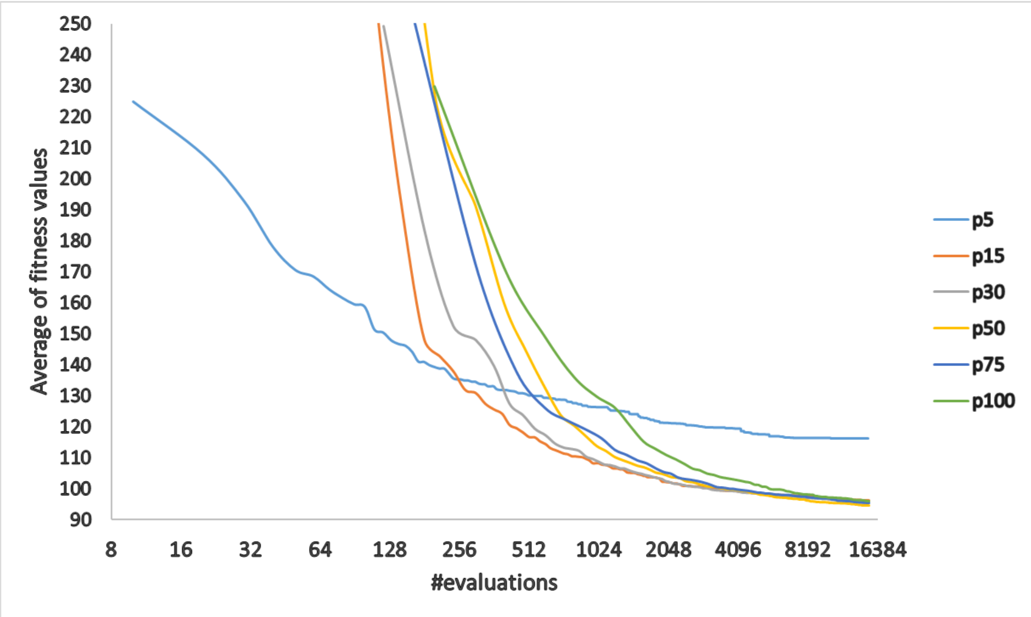


Figure 5.26: Convergence curves of TLBO by changing Psize on case2 (log scale)

Leven's test($F(5,114) = 33.593, p < .001$) indicated that the variances were statically not equal. There was a statistically significant difference between the mean values as determined by Welch's F test ($F(5,52.296) = 4.992, p = .001 < 0.05$). The Post hoc comparisons that uses the Games-Howell post hoc test were conducted (Appendix A, Table 5.41). The results in table 4.42 revealed that there was no statically significant difference between the results of the algorithm with the population sizes (50, 30, 75, 100, 15) which significantly performed better than the population size of 5.

Table 5.42: Homogeneous subsets of Psize (TLBO on case 2)

Psize	Homogeneous subsets 1	Homogeneous subsets 2	Significant conclusions
50	94.7170		**50, 30, 75, 100, 15 < 5**
30	95.3710		
75	95.4152		
100	96.0252		
15	96.2170		
5		116.2205	

5.3.2 Performance and Convergence Speed of WTLBO

Table 5.43: Descriptive statistics of WTLBO on case study 2

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean: Lower Bound	95% Confidence Interval for Mean: Upper Bound	Minimum	Maximum
5.00	133.1655	24.76415	5.53743	121.5755	144.7555	98.86	212.87
15.00	121.7906	13.94737	3.11873	115.2630	128.3181	104.07	157.36
30.00	117.7454	11.71769	2.62016	112.2614	123.2294	98.82	143.73
50.00	115.0148	6.76850	1.51348	111.8470	118.1825	105.78	129.27
75.00	**111.9968**	7.04409	1.57511	108.7001	115.2936	101.21	128.17
100.0	115.4010	12.51135	2.79762	109.5455	121.2565	102.75	160.86
Total	119.1023	15.49360	1.41437	116.3018	121.9029	98.82	212.87

The bold values indicate best results

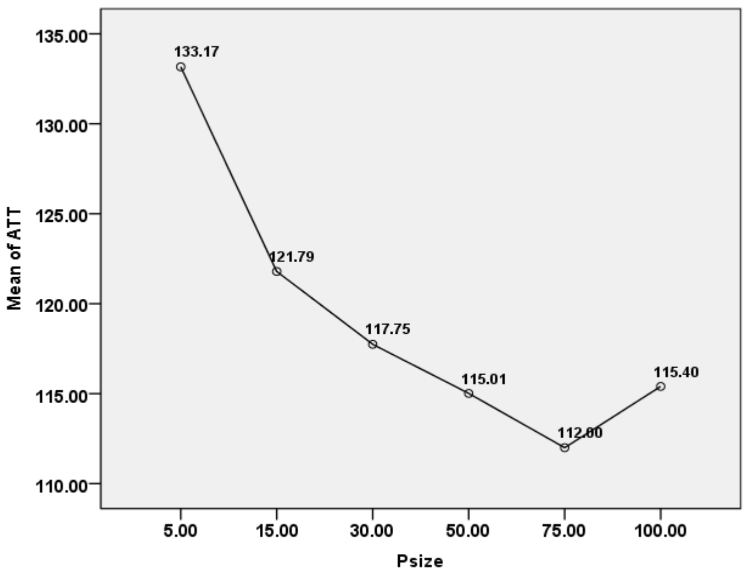


Figure 5.27: The mean results of WTLBO by changing Psize on case 1 phase duration 10-100

The mean value decreased remarkably between the population size of 5 and 15 and then decreased gradually from the population size of 15 to reach the best value at the population size of 75 (mean = 111.9968). The results of the population size of 75 and 100 were almost identical (Table 5.43, and Fig 5.27). Although the solution quality of large population size (i.e. 75) was better than the solution quality of the small population size (i.e. 15), the speed of small population size was better.

The convergence rate of WTLBO algorithm can be ordered according to the curves of different population sizes (from faster to slower) as follows: 15, [30, 5] 75, 100 where the speed of the algorithm using the population size values between the brackets was almost identical (Fig 5.28).

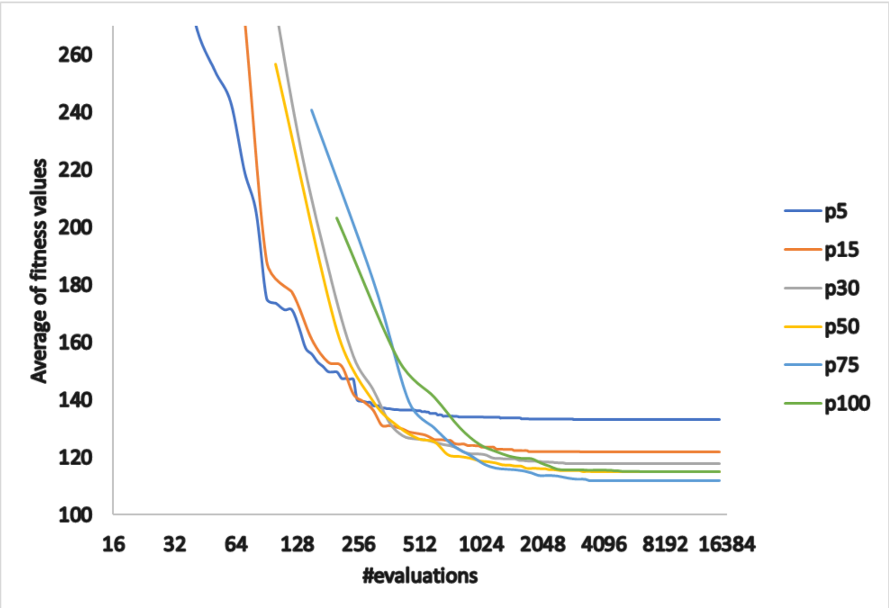


Figure 5.28: Convergence curves of WTLBO by changing Psize on case2

The homogeneity of variances was violated as indicated by Leven's test($F(5,114) = 3.619, p = .005 < 0.05$). There was a statistically significant difference between the mean result of different population sizes as determined by Welch's F test (*F(5,52.058)* = *3.810*, *p* < *.005*). The results of Games-Howell post hoc test reveals that the mean values in each group in Table 4.45 were statistically equal, and the algorithm with the population size of 75, and 50 had a significantly lower mean than population size of 5. See (Table 5.44 in Appendix A)

Table 5.45: Homogeneous subsets of Psize (WTLBO on case 2)

Psize	Homogeneous subsets		Significant conclusions
	1	2	
75	111.9968		**75, 50 < 5**
50	115.0148		
100	115.4010	115.4010	
30	117.7454	117.7454	
15	121.7906	121.7906	
5		133.1655	

5.3.3 Performance and Convergence Speed of Jaya

Table 5.46:Descriptive statistics of Jaya on case study 2

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
5.00	149.3491	43.79657	9.79321	128.8517	169.8465	110.67	262.07
15.00	100.3037	5.97972	1.33711	97.5051	103.1022	93.11	115.86
30.00	96.2366	2.58101	.57713	95.0286	97.4445	92.61	101.40
50.00	96.1719	2.48376	.55539	95.0094	97.3343	93.87	102.87
75.00	**94.9862**	1.91192	.42752	94.0913	95.8810	92.34	99.34
100.0	95.0129	2.02618	.45307	94.0646	95.9612	92.64	100.76
Total	105.3434	26.62755	2.43075	100.5302	110.1565	92.34	262.07

The bold values indicate best results

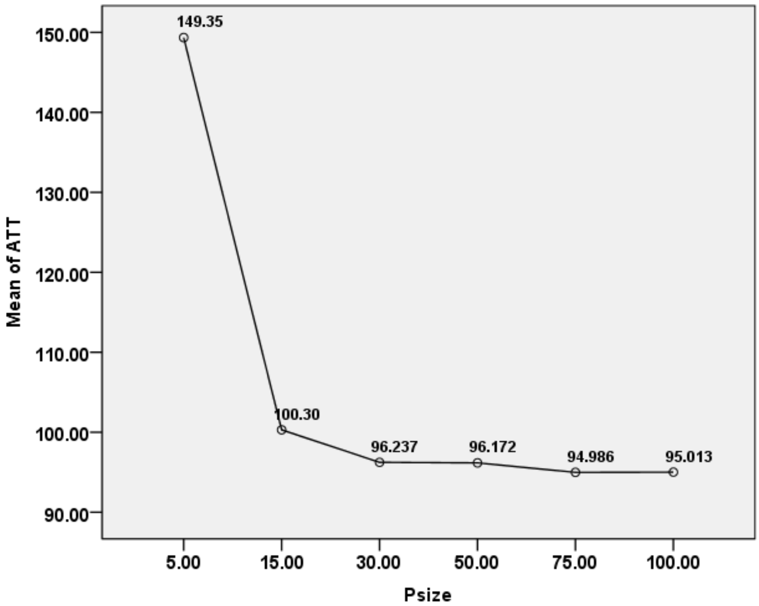


Figure 5.29: The mean results of Jaya by changing Psize on case 2

The best mean value was obtained when the population size was ***75*** (mean = 57.0816). There was a significant fall in the mean value between the population size of 5 and 15, and then the mean decreased slightly between the population size of 15 and 30. Whereas, the result was almost identical when the population size rose from 30 to 100 (table 5.46, fig 5.29).

The algorithm with the population size of 5 was the worst in terms of convergence speed. When the algorithm with the population sizes of 15-100 reached the maximum allowable evaluations (15000), it approximately converged to the same solution quality, but the algorithm with the population size of (15, 30, 50) was faster than those of (75, 100) (Fifg.5.30).

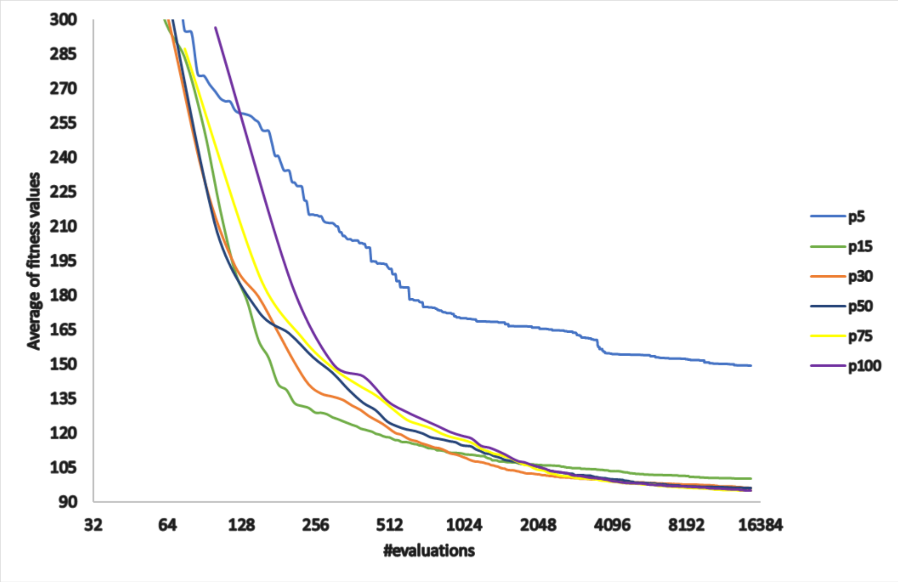


Figure 5.30: Convergence curves of Jaya by changing Psize on case 2 (log scale)

Since the assumption of homogeneity of variances was not met by Leven's test(*F(5,114) = 15.802, p < .001*), we used Welch's F test which indicated that at least there was a pair of mean values which was significantly different. (*F(5,52.160) = 9.225, p < .001*).

The Post hoc comparisons (appendix A table 5.47) reveal that there was not a statically significant difference between the means listed in each subset as shown in table 4.48. Jaya algorithm with the population size of **(75, 100)** was significantly better than the population size of 15, and the population size of (**50,30**) had a significantly lower mean than the population size of 5.

Table 5.48: Homogeneous subsets of Psize (Jaya on case 2)

Psize	Homogeneous subsets	

	1	2	3	Significant conclusions
75	94.9862			75 , 100 < 15 , 5
100	95.0129			50 , 30 , 15 < 5
50	96.1719	96.1719		
30	96.2366	96.2366		
15		100.3037		
5			149.3491	

5.3.4 Performance and Convergence Speed of GA

Table 5.49:Descriptive statistics of GA on case study 2

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
5.00	117.7317	13.16373	2.94350	111.5709	123.8925	103.96	148.15
15.00	116.1098	9.75345	2.18094	111.5450	120.6746	101.66	139.79
30.00	121.2059	15.90889	3.55734	113.7603	128.6514	105.10	161.72
50.00	109.3971	8.21609	1.83717	105.5518	113.2423	97.08	126.51
75.00	108.9029	5.39828	1.20709	106.3764	111.4293	100.02	121.96
100.00	**107.9893**	6.98175	1.56117	104.7217	111.2569	99.03	129.15
Total	113.5561	11.49327	1.04919	111.4786	115.6336	97.08	161.72

The bold values indicate best results

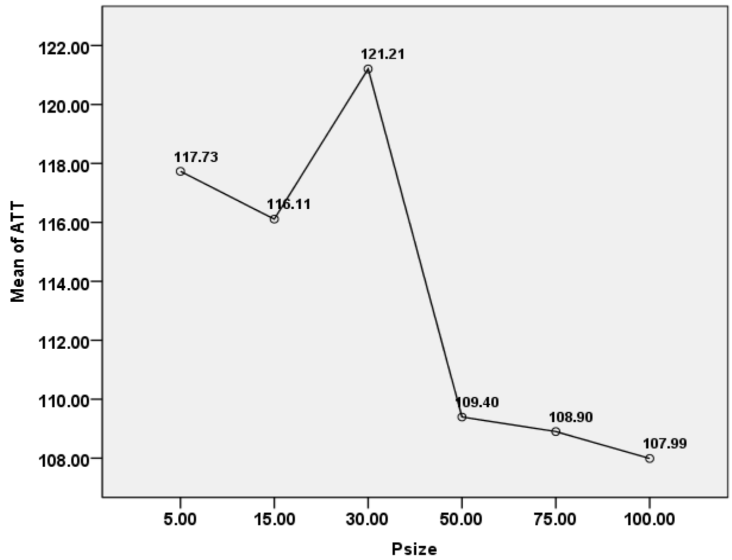


Figure 5.31: The mean results of GA by changing Psize on case 2

The mean value slightly decreased between the population size of 5 and 15, and then increased notably at the population size of 30. It returned to fall significantly between the population size of 30 and 50. It continued to decrease slightly until it reached the best value at the population size of 100 (mean = 107.9893). It seems that the algorithm with the large population size values (50, 75, 100) produced a better solution quality than the small population sizes (Table 5.49, Fig 5.31).

The algorithm with the population size of 5 was the worst in terms of convergence speed, while the algorithm with the population size of 50 was the best. It can be observed from Fig 5.32 that during the first evaluations, the algorithm with the population sizes of (15, 30) was faster than the population sizes of (75, 100), and then the opposite happened. We can conclude that the algorithm with large population size (i.e. 50-100) was faster than small population sizes (i.e. 5-30)

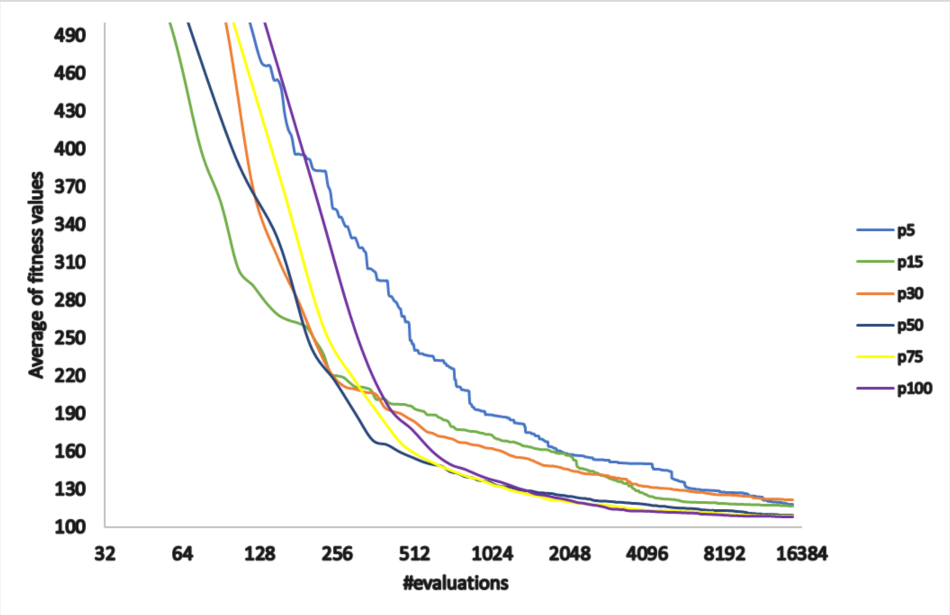


Figure 5.32: Convergence curves of GA by changing Psize on case 2

Since the assumption of homogeneity of variances was violated by Leven's test($F(5,114) = 4.247, p = .001 < 0.05$), we used Welch's F test which indicated that there was a statistically significant difference between the mean result of different population sizes (*F(5,52.221) = 5.007, p = .001 < 0.05*).

Games-Howell post hoc test revealed that there was no statically significant difference between the means in each subset as shown in table 4.51. GA algorithm with the population size of **100** had a significantly lower mean than the population size of **(15, 30)**. And the population size of **75** had a significantly lower mean than that of **(30)** (see Appendix A, Table 5.50)

Table 5.51: Homogeneous subsets of Psize (GA on case 2)

Psize	Homogeneous subsets 1	2	3	Significant conclusions
100	107.9893			**100 < 15 , 30**
75	108.9029	108.9029		**75 < 30**
50	109.3971	109.3971	109.3971	
5	117.7317	117.7317	117.7317	
15		116.1098	116.1098	
30			121.2059	

5.3.5 Performance and Convergence Speed of PSO

Table 5.52: Descriptive statistics of PSO on case study 2

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	Upper Bound	Minimum	Maximum
5.00	184.1143	70.74168	15.81832	151.0061	217.2224	117.22	417.35
15.00	130.6875	27.27303	6.09843	117.9233	143.4517	102.25	221.25

30.00	118.1745	14.53499	3.25012	111.3719	124.9770	99.72	151.76
50.00	114.5532	18.66067	4.17265	105.8197	123.2866	92.13	182.47
75.00	113.3221	9.18798	2.05450	109.0219	117.6222	99.89	138.16
100.0	**107.1917**	7.28848	1.62975	103.7805	110.6028	96.50	121.59
Total	128.0072	41.41365	3.78053	120.5213	135.4930	92.13	417.35

The bold values indicate best results

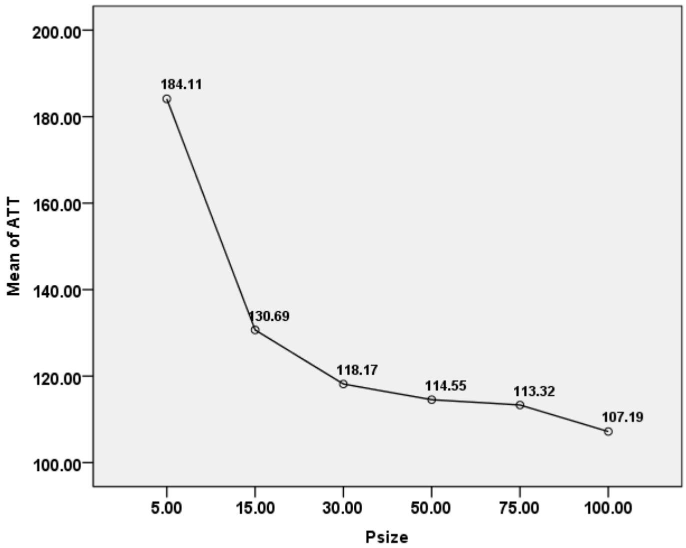


Figure 5.33: The mean results of PSO by changing Psize on case 2

Table 5.52 and Fig 5.33 reveal that the mean result decreased as the value of the population size increased from 5 to 100. Between the population size of 5 and 15, the mean value significantly decreased. Then a gradual decrease was obtained between the population size of 15 – 100. Therefore, the best result was obtained when the population size was **100** (mean = 107.1917)

According to Fig 5.34, the algorithm with population sizes of (5, 15, 30) started faster than the others. Then the order was reversed, the algorithm with the population size of (50, 75, 100) became faster than those of (5, 15). In average, the algorithm with the population size of 30 was faster than the others because it reached near to the minimum in fewer iterations, while the algorithm with the population size of 5 was the slowest because it was stuck early into a local minimum and couldn't get out of it.

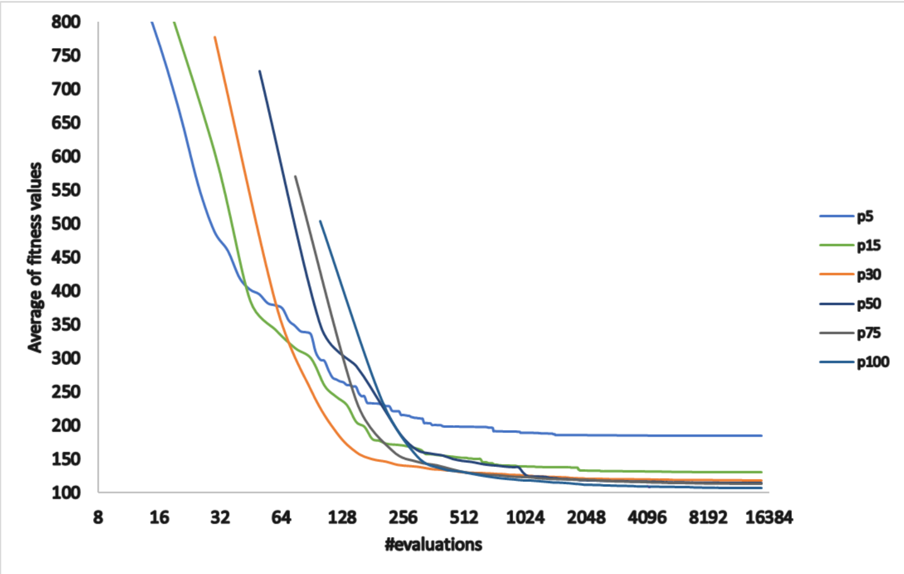


Figure 5.34: Convergence curves of PSO by changing Psize on case 2

Table 5.54:Homogeneous subsets of Psize (PSO on case 2)

Psize	Homogeneous subsets			
	1	2	3	Significant conclusions
100	107.1917			**100, 75, 50, 30, 15 < 5**
75	113.3221	113.3221		**100 < 15, 5**
50	114.5532	114.5532		
30	118.1745	118.1745		
15		130.6875		
5			184.1143	

The homogeneity of variances was violated as indicated by Leven's test($F(5,114) = 12.973, p < .001$). So, we carried out Welch's F test which indicated that we strongly rejected the hypothesis (All means are equal) ($F(5,51.164) = 8.226, p < .001$). Games-Howell post hoc test was conducted (Appendix A, Table 5.53). The results in Table 5.54 reveal that there was no statically significant difference between the results of population sizes that are listed under each subset. PSO algorithm with the population size of 100 had a significantly lower mean than the population size of 15, 5, and the population size of 5 was significantly the worst.

Table 5.55: Summary results of statistical tests for algorithms, each with different population sizes (case 2)

algorithm	Leven's test of homogeneity of variances				Welch F Test				One-way ANOVA	
	Leven Statistic	df1	df2	Sig.	Statistic[a]	df1	df2	Sig.	F	Sig.
TLBO	33.593	5	114	0.000	4.992	5	52.296	**0.001**	18.836	0.000
WTLBO	3.619	5	114	0.005	3.810	5	52.058	**0.005**	5.768	0.000
Jaya	15.802	5	114	0.000	9.225	5	52.160	**0.000**	28.479	0.000
GA	4.247	5	114	0.001	5.007	5	52.221	**0.001**	5.485	0.000
PS	12.973	5	114	0.000	8.226	5	51.164	**0.000**	15.199	0.000

* p shown as 0.000, that is p <0 .001

By observing many plots (for example see Figures 5.1, 5.7, 5.9, 5.13, 5.17, 5.22, 5.27) it is evident that for extremely small population size (i.e. p = 5) the performance of the evaluated algorithms has been the worst. A possible reason is that too small population size leads to the lack of sufficient diversity and will not provide enough exploration ability. Another valuable observation from the results of small and medium networks is that the algorithms with a very small population size (i.e. p=5) converge fast but stuck early into a local minimum because of insufficient diversity.

5.3.6 Comparison of TLBO, WTLBO, Jaya, GA, and PSO

We compared the algorithms based on the best result of each algorithm (the best population size) obtained from previous experiments.

Table 5.56: Comparative results of TLBO, WTLBO, Jaya, GA, and PSO case study 2

algorithm	Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
TLBO	50	**94.7170**	**1.76221**	**.39404**	93.8922	95.5417	**91.89**	98.11
WTLBO	75	111.9968	7.04409	1.57511	108.7001	115.2936	101.21	128.17
JAYA	75	94.9862	1.91192	.42752	94.0913	95.8810	92.34	99.34
GA	100	107.9893	6.98175	1.56117	104.7217	111.2569	99.03	129.15
PSO	100	107.1917	7.28848	1.62975	103.7805	110.6028	96.50	121.59

The bold values indicate best results

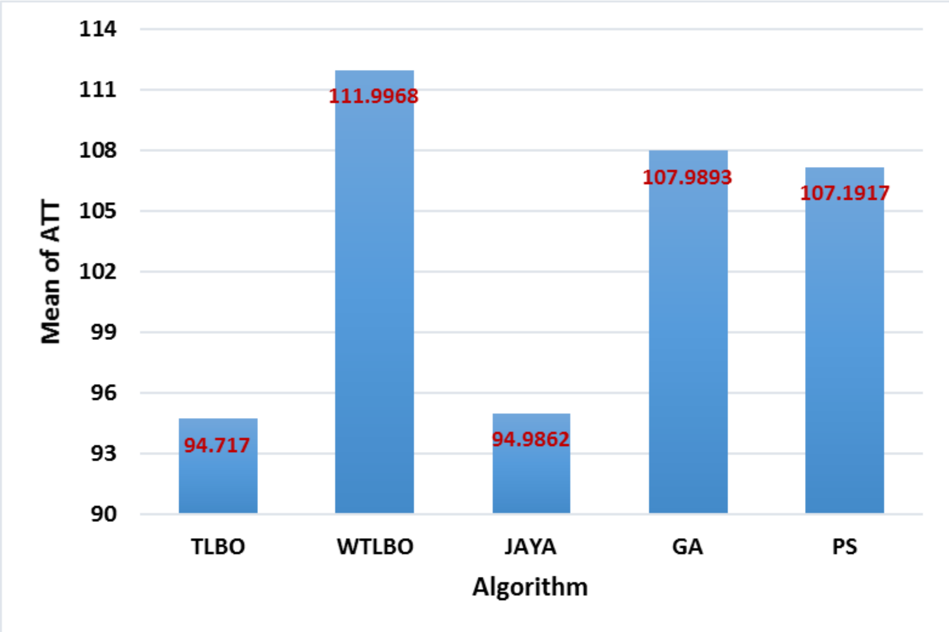


Figure 5.35: The best results of TLBO, WTLBO, Jaya, GA, PSO on case 2

Table 5.56 reveals that TLBO algorithm obtained the best mean (94.7170), minimum (91.89), and standard deviation (1.76221) results. It seems that the results of TLBO and Jaya were almost the same. The algorithms can be ordered based on the mean result (from better to worse) as follows: TLBO, Jaya, PSO, GA, and WTLBO. Moreover, the small standard deviation which was obtained by TLBO and Jaya signified that they gave a more stable performance than GA, PSO, and WTLBO.

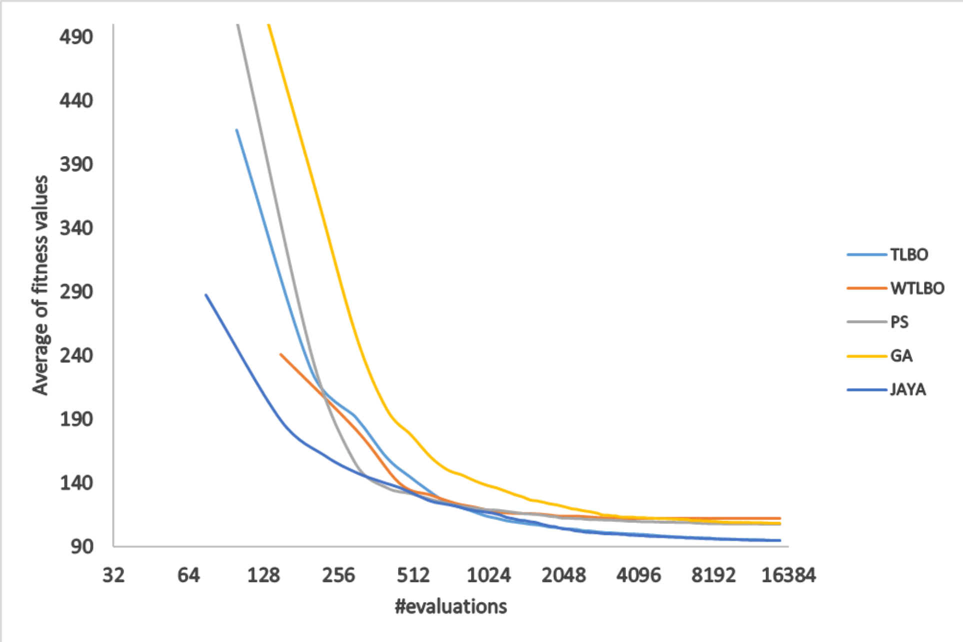


Figure 5.36: Convergence speed of TLBO, WTLBO, GA, PSO and Jaya on case study 2

The convergence speed of Jaya algorithm was the best, while GA was the worst. During the first iterations, the speed of TLBO, WTLBO, and PSO algorithms was almost the same, and then TLBO became faster. In average, the algorithms can be ordered according to the convergence velocity (from faster to slower) as follows: Jaya, TLBO, PSO, WTLBO, and GA (Fig 5.36). Moreover, the ability for TLBO and Jaya to escape the local minimum was better than PSO, WTLBO, and GA during the maximum allowable iterations.

Table 5.57: Statistical results for algorithms by Games-Howell post hoc test (case 2)

Pair of comparison Algorithm I & J	Mean Difference (I-J)	95% Confidence Interval		P-value	Significance (better)
		Lower bound	Upper bound		
TLBO & WTLBO	-17.27990*	-22.1094-	-12.4504	.000	**TLBO**
TLBO & Jaya	-.26920	-1.9344-	1.3960	.990	-
TLBO & GA	-13.27235*	-18.0608-	-8.4839	.000	**TLBO**

TLBO & PSO	-12.47470*	-17.4652-	-7.4842	.000	**TLBO**
WTLBO & Jaya	17.01070*	12.1642	21.8572	.000	**Jaya**
WTLBO & GA	4.00755	-2.3419-	10.3570	.385	-
WTLBO & PSO	4.80520	-1.6843-	11.2947	.233	-
Jaya & GA	-13.00315*	-17.8087-	-8.1976	.000	**Jaya**
Jaya & PSO	-12.20550*	-17.2124-	-7.1986	.000	**Jaya**
GA & PSO	.79765	-5.6644-	7.2597	.997	-

- : indicates that there is no significant between the compared algorithms.
****. The mean difference is significant at 0.05 level***

The homogeneity of variances was violated as indicated by Leven's test($F(4,95) = 8.384, p < .001$). Welch's F test ($F(4,44.460) = 53.892, p < .001$) revealed that the means of fitness value which were produced by the five algorithms were not the same. From Games-Howell post hoc test (table 5.57) we concluded that: 1) there was no statically significant difference between the mean results of ***Jaya*** and ***TLBO*** (p-value = 0.990). 2) There was no statically significant difference between the mean results of ***WTLBO, GA, and PS***O. 3) both Jaya and TLBO were significantly performing better than WTLBO, GA, and PSO (p-value < 0.001).

Based on the results showed in Figures 5.11, 5.23, 5.35 and the statistical analysis reported in Tables 5.19, 5.38, 5.57, it can be seen that TLBO and Jaya outperformed the other algorithms on small and medium sized networks. This can be justified by saying that the ability of TLBO and Jaya to balance between the exploration and exploitation approaches is better than WTLBO, GA, and PSO.

In TLBO, the exploitation ability is achieved in teacher phase by Eq. (2.3) which allows the learners to move towards the best learner (i.e. the teacher). It is also achieved in the learner phase by Eq. (2.5) which allows each learner to move toward another learner who has a better knowledge. The exploration approach is employed in the learner phase by Eq. (2.5), the learner moves away from another randomly selected learner who has a worse knowledge. The obtained results demonstrate the high ability of TLBO to employ exploration and exploitation approaches.

The victorious nature of Jaya algorithm is proved by the obtained results. It always allows the modified solution to move towards the best solution and moves away from the worst solution by Eq. (2.6).

Moreover, the performance of WTLBO, GA, and PSO is sensitive to the parameters settings which majorly contributing to the diversification and intensification of the search space. So, it can easily say that the recommended parameters for these algorithms not work well enough and need to be properly tuned for each case study. Besides, the performance of GA is mainly influenced by the selection, mutation, and crossover operators. The selection method used in this study, which is based on selecting the best half of the population, may have negative results. It is therefore useful to take advantage of other selection methods such as roulette wheel, tournament, and rank-based.

5.4 Comparing Optimization Techniques on Case Study 3

Table 5.58: Case 3 experiments settings

Green or red time (s)	Yellow time (s)	Population size	evaluations
10 - 60	3	50, 500, 1000	20000

5.4.1 Performance and convergence speed of basic TLBO

Table 5.59: Descriptive statistics of Basic TLBO on case study 3

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
50	**162.7186**	1.33168	.29777	162.0953	163.3418	**159.36**	164.84
500	165.3769	1.68883	.37763	164.5865	166.1672	163.18	168.88
1000	167.5066	1.73150	.38718	166.6963	168.3170	164.04	171.36
Total	165.2007	2.52123	.32549	164.5494	165.8520	159.36	171.36

The bold values indicate best results

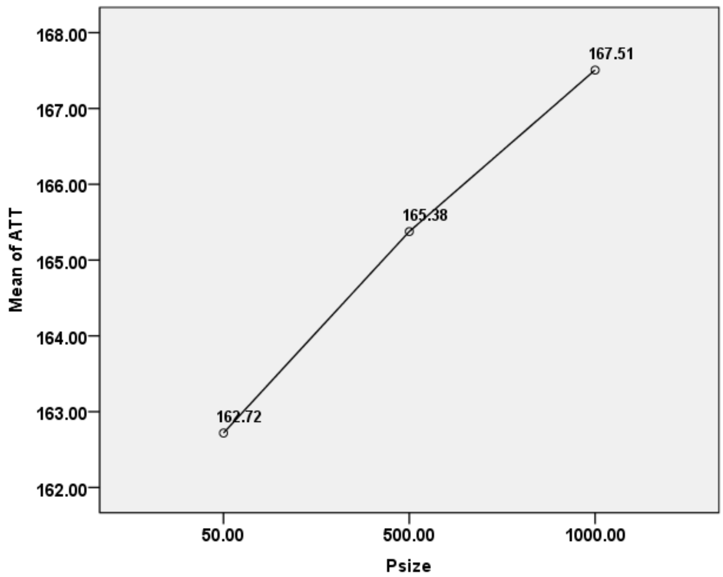


Figure 5.37: The mean results of TLBO by changing Psize on case 3

It can be observed from Table 5.59, and fig 5.37 that the optimal mean of fitness value seems to occur when the population size was 50 (mean = 162.7186). The average value grew when the population size increased from 50 to 1000. It was clear that the algorithm with the population size of 50 had a better convergence rate, while the algorithm with the population size of 1000 was the worst (Fig 5.38).

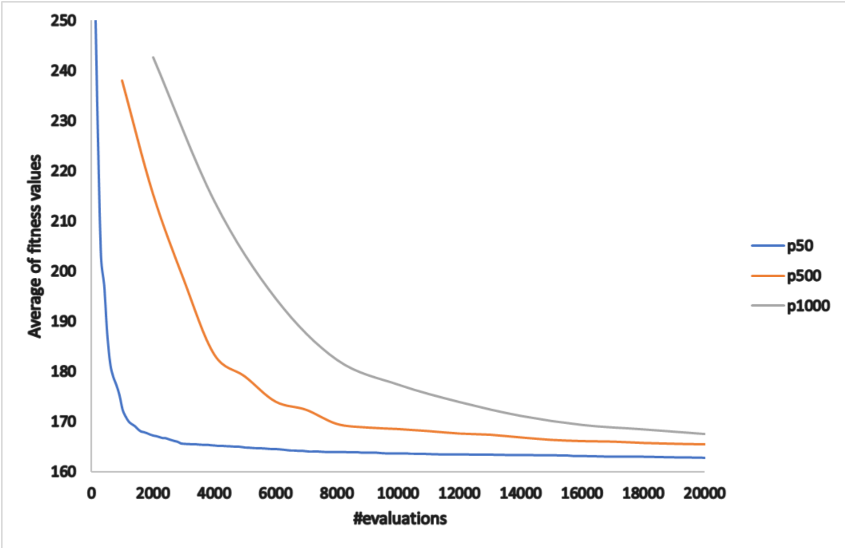


Figure.5.38: Convergence curves of TLBO by changing Psize on case 3

The assumption of homogeneity of variances was met by Leven's test(*F(2,57) = 0.602, p =0.551 > 0.05*). There was a statically significant difference between groups as determined by one-way ANOVA (*F(2,57) = 45.291, p < .001*). A Tukey post hoc test revealed that the mean of fitness values was statically significantly lower when the population size was 50 compared to the population size of 500 (p<0.001) and the population size of 1000 (p<0.001). Also, the algorithm with the population size of 5000 had a significantly lower mean than that of 1000 (p<0.001) (Appendix A, Table 5.60)

5.4.2 Performance and Convergence Speed of WTLBO

Table 5.61: Descriptive statistics of WTLBO on case study 3

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
50	169.6188	1.71639	.38380	168.8155	170.4221	167.20	173.22

500	**169.1707**	1.58167	.35367	168.4305	169.9109	**166.66**	172.64
1000	169.2355	1.21712	.27216	168.6658	169.8051	167.60	171.68
Total	169.3417	1.50704	.19456	168.9523	169.7310	166.66	173.22

The bold values indicate best results

Changing the values of the population size as shown in table 5.61 did not affect the solution quality of WTLBO algorithm, and the results appeared to be close.

The algorithm with the given population sizes reached almost the same minimum. But, the algorithm with the population size of 50 was faster than the those of 500 and 1000, while the algorithm with the population size of 1000 was the slowest. In average, the fitness value (ATT) dropped rapidly and then the algorithm stuck early into a local minimum without being able to get out of it (see Fig 5.39).

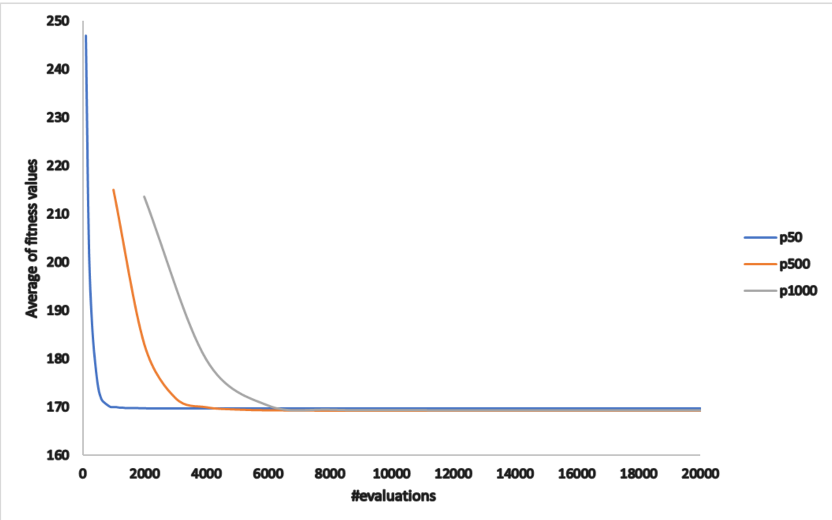


Figure 45.39: Convergence curves of WTLBO by changing Psize on case2

One-way ANOVA revealed that the differences between the means were not statically significant (*F(2,57) = 0.508, p =**0.604**> 0.05*).

5.4.3 Performance and Convergence Speed of Jaya

Table 5.62: Descriptive statistics of Jaya on case study 3

Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
50	**171.3808**	5.73861	1.28319	168.6951	174.0666	**164.57**	186.41
500	177.6305	4.13263	.92408	175.6963	179.5646	169.64	185.73
1000	183.2182	5.06116	1.13171	180.8495	185.5869	171.49	191.75

Total	177.4098	6.93746	.89562	175.6177	179.2020	164.57	191.75

The bold values indicate best results

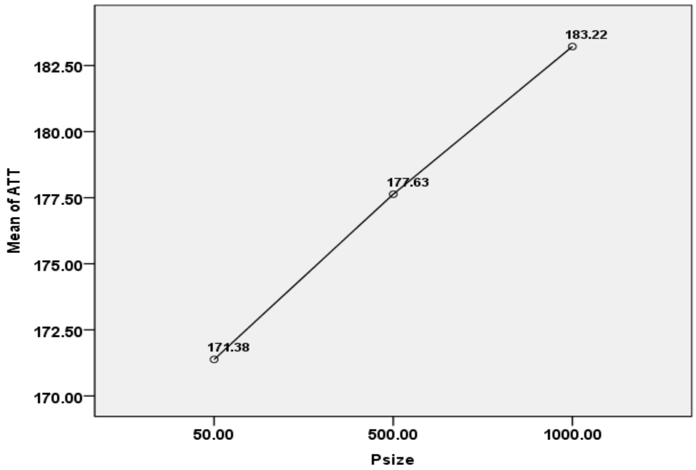


Figure 5.40: The mean results of Jaya by changing Psize on case 3

The best mean of ATT seemed to occur when the population size was ***50*** (mean = 171.3808). The population sizes can be ordered based on the obtained results (from best to worst) as follows: 50, 500, and 1000 (Table 5.62, Fig 5.40). It was clear that the algorithm with the population size of **50** had a better convergence rate, while the algorithm with the population size of 1000 was the worst (Fig 5.41).

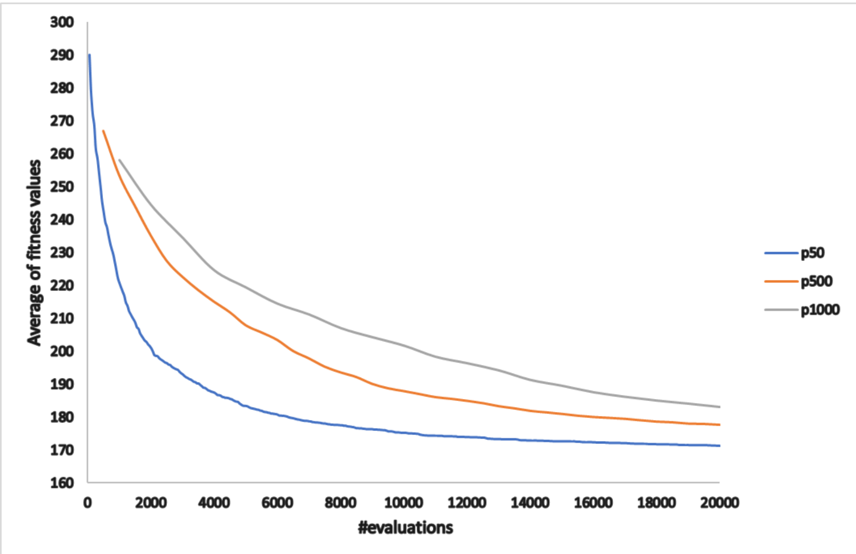


Figure 5.41: Convergence curves of Jaya by changing Psize on case 3

The assumption of homogeneity of variances was met by Leven's test($F(2,57) = 0.449, p = 0.641 > 0.05$). So we used one-way ANOVA which indicated that the differences between the group means were statically significant ($F(2,57) = 27.822, p < .001$). A Tukey post hoc test revealed that the mean of fitness values was statically significantly lower when the population size was 50 when compared to the population size of 500 (p=0.001) and the population size of 1000 (p<0.001). Furthermore, the algorithm with the population size of 5000 had a significantly lower mean than the population size of 1000 (p=0.002) (Appendix A, Table 5.63)

5.4.4 Performance and Convergence Speed of GA

Table 5.64: Descriptive statistics of GA on case study 3

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
50	**186.9638**	7.69043	1.71963	183.3646	190.5631	177.59	203.82
100	190.3240	2.44366	.54642	189.1804	191.4677	185.78	194.78
1000	206.8579	2.81503	.62946	205.5404	208.1754	200.58	212.82
Total	194.7153	10.02028	1.29361	192.1267	197.3038	177.59	212.82

The bold values indicate best results

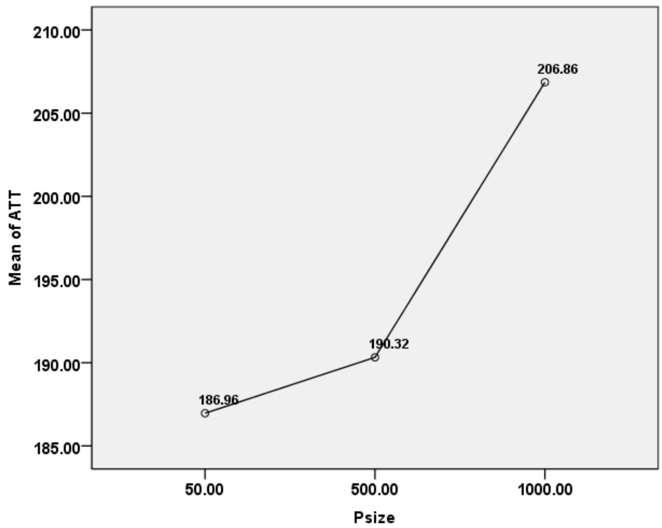


Figure 5.42: The mean results of GA by changing Psize on case 3

The mean value rose slightly when the population size increased from 50 to 500, and then increased again remarkably at the population size of 1000. The algorithm produced a better solution quality (mean = 186.9638) when the population size was **50** (Table 5.64, Fig 5.42). In terms of convergence speed, the population sizes can be ordered based on the curves as shown in fig 4.43 (from best to worst) as follows: 50, 500, and 1000. We can conclude that the algorithm with small population sizes (i.e. 50) gave a better solution quality and convergence speed than the algorithm with large population size (i.e. 1000).

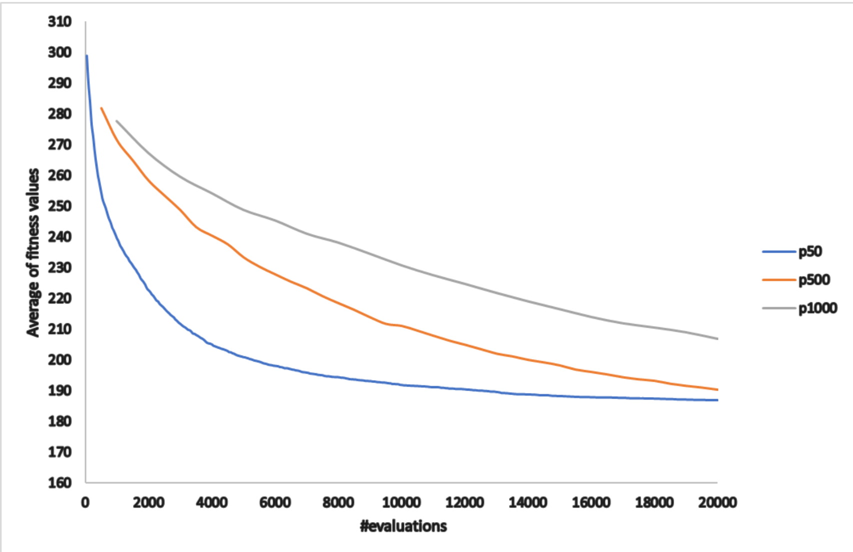


Figure 5.43: Convergence curves of GA by changing Psize on case 3

Since the assumption of homogeneity of variances was violated by Leven's test ($F(2,57) = 13.124, p < .001$), we used Welch's F test which indicated that the differences between the group means were statically significant ($F(2,34.731) = 210.201, p < .001$).

Games-Howell post hoc test revealed that the mean of ATT was statically significantly lower when the population size was 50 ($p<0.001$) and 500 ($p<0.001$) when compared to the population size of 1000. There was no statically significant difference between the population size of 50 and 500 means ($p=0.173$) (Appendix A, Table 5.65)

5.4.5 Performance and Convergence Speed of PSO

Table 5.66 Descriptive statistics of PSO on case study 3

	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
50	209.6569	13.31831	2.97806	203.4238	215.8901	185.68	235.42
500	202.1364	12.26783	2.74317	196.3949	207.8780	**179.57**	224.15
1000	**201.2848**	8.35225	1.86762	197.3759	205.1938	185.27	217.67
Total	204.3594	11.93504	1.54081	201.2763	207.4426	179.57	235.42

The bold values indicate best results

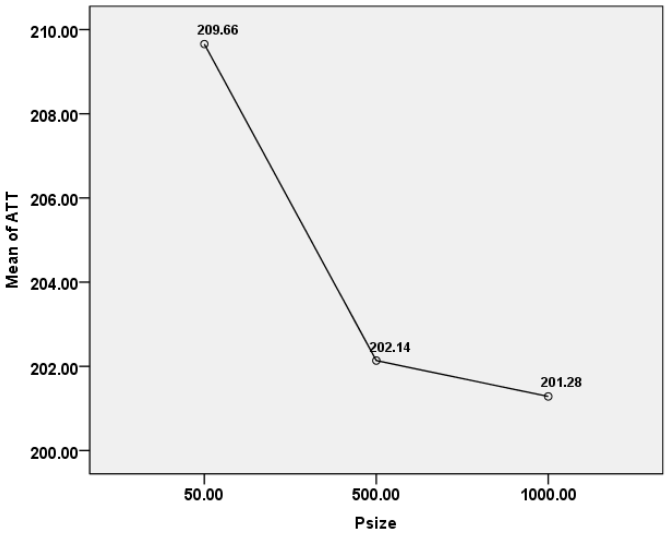


Figure 5.44: The mean results of PSO by changing Psize on case 3

Table 5.66 and Fig 5.44 reveal that the mean of ATT fell remarkably as the value of the population size increased from 50 to 500. And, there was a slight drop between the population sizes of 500 and 1000. The best result was obtained when the population size was **1000** (mean = 201.2848)

The performance of PSO differs from the other algorithms in the large network. It performed better with very large population size. Although the large population size maintains a high level of diversity, it is not recommended to be very large because it is resource consuming and the convergence may not be achieved during the allowable evaluations. However, PSO behavior can be explained by the choice of inappropriate parameters values for the problem at hand. PSO parameters (w, cp, and cg) must be properly tuned due to balance between exploration and exploitation. A smaller value of *w* assists the local exploitation, while a larger value of *w* encourages the global exploration. It is therefore advisable to try other methods for tuning the inertia weight parameter.

According to Fig 5.45, the algorithm with the population size of 50 started faster than the others. Then the order was reversed, the algorithm with the population size of (500, 1000) became faster than that of 50. In average, the algorithm with the population size of 500 and 1000 can reach almost the same minimum solution but in fewer iterations when the population size was 500, while the algorithm with the population size of 50 stuck early into a local minimum and couldn't get out of it.

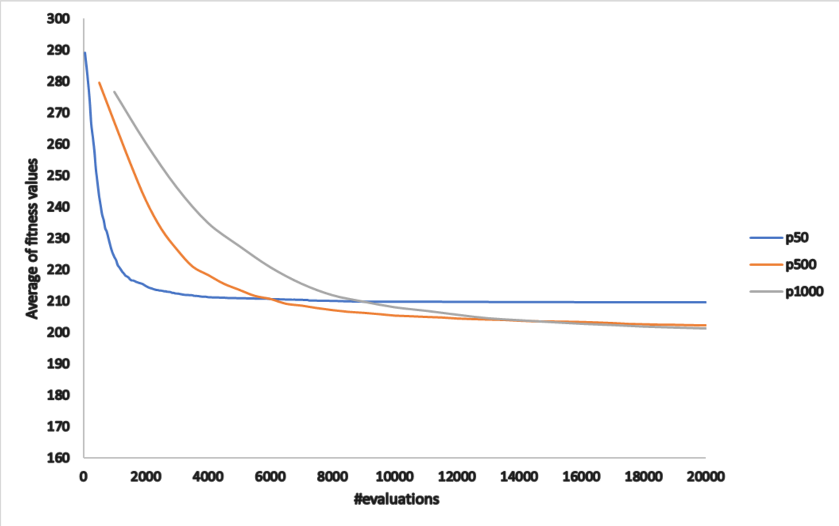


Figure 5.45: Convergence curves of PSO by changing Psize on case 3

The assumption of homogeneity of variances was met by Leven's test(*F(2,57) = 1.502, p =0.231 > 0.05*), so we used one-way ANOVA which indicated that the differences between the means were not statically significant (*F(2,57) = 3.203, p .064*). So, we accepted the hypothesis (the means are equal).

Table 5.67: Summary results of statistical tests for algorithms, each with different population sizes (case 3)

algorithm	Leven's test of homogeneity of variances				Welch F Test				One-way ANOVA	
	Leven Statistic	df1	df2	Sig.	Statistic[a]	df1	df2	Sig.	F	Sig.
TLBO	0.602	2	57	0.551	49.119	2	37.407	0.000	45.291	**0.000**
WTLBO	0.895	2	57	0.414	0.431	2	37.109	0.653	0.508	**0.604**
Jaya	0.449	2	57	0.641	23.565	2	37.288	0.000	27.822	**0.000**
GA	13.124	2	57	0.000	210.201	2	34.731	**0.000**	93.167	0.000
PSO	1.502	2	57	0.231	2.897	2	36.262	0.068	3.203	**0.048**

* p shown as 0.000, that is p <0 .001
* Shaded cells to distinguish the used test

From the results obtained in Figures (5.37, 5.40, 5.42) we see that TLBO, Jaya, and GA performed better with a relatively small population size (i.e. n=50). this finding proves that the use of a very large population size (e.g. n>500) is not favored even in high-dimensional problems.

5.4.6 Comparison of TLBO, WTLBO, Jaya, GA, and PSO

Table 5.68: Comparative results of *TLBO, WTLBO, Jaya, GA, and PS*O case study3

algorithm	Psize	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
TLBO	**50**	**162.7186**	**1.33168**	.29777	162.0953	163.3418	**159.36**	164.84
WTLBO	**50**	169.6188	1.71639	.38380	168.8155	170.4221	167.20	173.22
JAYA	**50**	171.3808	5.73861	1.28319	168.6951	174.0666	164.57	186.41
GA	**50**	186.9638	7.69043	1.71963	183.3646	190.5631	177.59	203.82
PSO	**1000**	201.2848	8.35225	1.86762	197.3759	205.1938	185.27	217.67

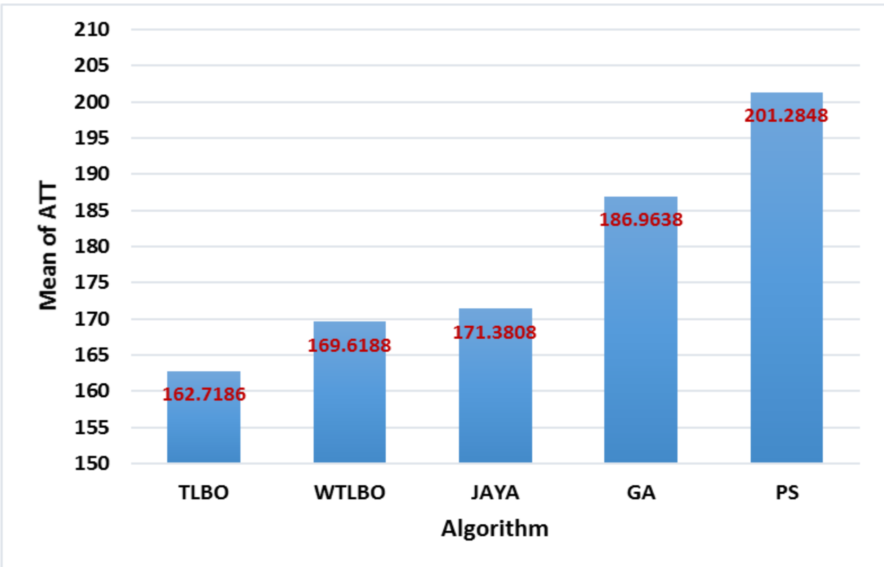


Figure 5.46: The best results of TLBO, WTLBO, Jaya, GA, PSO on case 3

Table 5.68 reveals that TLBO algorithm obtained the best mean (162.7186), min (159.36), and standard deviation (1.33168) results. Thus it gave a more stable performance than other algorithms. It seems that WTLBO and Jaya gave almost the same average, but with a WTLBO preference in terms of stability (std. = 1.71639). The algorithms can be ordered based on the mean result (from better to worse) as follows: TLBO, WTLBO, Jaya, GA, and PSO.

In addition, TLBO, WTLBO, Jaya, and GA algorithms performed better when the population size was small (i.e. 50), while PSO algorithm performed better when the population size was large (i.e. 500, 1000).

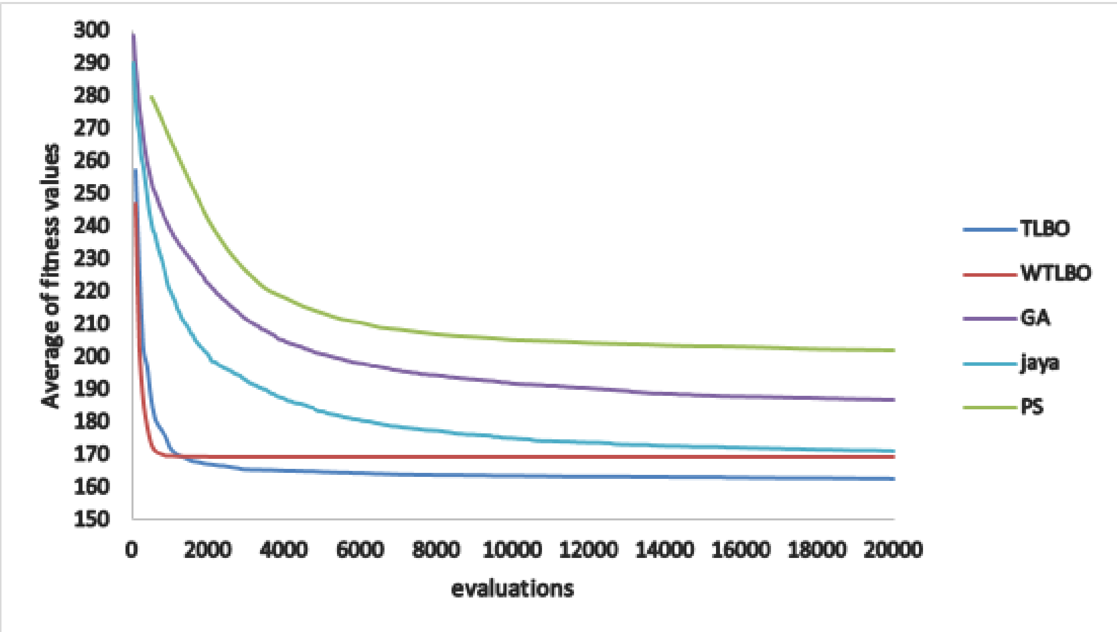


Figure 5.47: Convergence speed of TLBO, WTLBO, GA, PSO and Jaya on case study 2

When comparing the convergence speed for the algorithms, we found that TLBO was the best, then WTLBO, Jaya, GA, PSO respectively

(Fig 5.47). Moreover, Jaya algorithm reached nearly the same result as WTLBO algorithm, but with fewer iterations for WTLBO.

Table 5.69: Statistical results for algorithms by Games-Howell post hoc test (case 3)

Pair of comparison Algorithm I & J	Mean Difference (I-J)	95% Confidence Interval		P-value	Significance (better)
		Lower bound	Upper bound		
TLBO & WTLBO	-6.90025*	-8.2952-	-5.5053	.000	**TLBO**
TLBO & Jaya	-8.66230*	-12.5859-	-4.7387	.000	**TLBO**
TLBO & GA	-24.24527*	-29.4643-	-19.026	.000	**TLBO**
TLBO & PSO	-38.56630*	-44.2264-	-32.906	.000	**TLBO**
WTLBO & Jaya	-1.76205	-5.7303-	2.2062	.685	-
WTLBO & GA	-17.34502*	-22.5964-	-12.0936	.000	**WTLBO**
WTLBO & PSO	-31.66605*	-37.3558-	-25.9763	.000	**WTLBO**
Jaya & GA	-15.58297*	-21.7504-	-9.4156-	.000	**Jaya**
Jaya & PSO	-29.90400*	-36.4324-	-23.3756	.000	**Jaya**
GA & PSO	-14.32103*	-21.5920-	-7.0501-	.000	**GA**

- : indicates that there is no significant between the compared algorithms.
****. The mean difference is significant at 0.05 level***

The homogeneity of variances was violated as indicated by Leven's test($F(4,95) = 11.091, p < .001$). Welch's F test indicated that the differences between the means were statically significant (*F(4,43.845) = 174.574, p < .001*).

From Games-Howell post hoc test (table 5.69) we concluded that: 1) there was no statically significant difference between the results of ***Jaya*** and ***WTLBO*** (p-value 0.685). 2) TLBO had a significantly lower mean than WTLBO (p-value<0.001), Jaya (p-value<0.001), GA (p-value <0.001), and PSO (p-value<0.001). 3) Both WTLBO and Jaya were significantly performing better than GA, and PSO (p-value < 0.001). 4) GA had a significantly lower mean than PSO (p-value<0.001).

The obtained results demonstrate the efficiency of TLBO on the large network as well as the small and medium size networks. This confirms the high ability of TLBO to achieve exploration and exploitation approaches, and therefore produce an acceptable solution in a reasonable amount of time.

Moreover, there is a marked improvement in the performance of WTLBO toward the large network. Jaya performed better than WTLBO in the small and medium size networks, while, for the large network Jaya and WTLBO had significantly the same result. These results confirm how the use of weight parameter (w) enhanced the performance of WTLBO on large network. The value of (w) is linearly reduced from (w_{max}) to (w_{min}) by Eq. (2.11). So, during the first iterations, large value of w encourages the exploration approach, and then decreasing to assist the local exploitation approach. However, the poor performance of WTLBO for the small and medium size networks proves that it is certainly beneficial to tune (w_{max} and w_{min}) parameters based on the problem at hand. It seems that the recommended values of w_{max} and w_{min} have a negative effect in the small and medium size networks.

TLBO, Jaya, and PSO are swarm-based algorithms. There is a great similarity in the basic equations of these algorithms, especially Jaya and PSO. However, PSO showed poor performance in the small, medium, and large networks. This finding can be justified by saying that the improper tuning of parameters leads to undesirable solution quality. So, these parameters need to be correctly tuned to suit each new problem.

5.5 Summary

In this chapter, five optimization algorithms were investigated on three different networks by simulation tool, SUMO. Firstly, the effect of common controlling parameters (i.e. population size) was tested to prove that they influence the effectiveness of the algorithm. To guarantee a fair comparison, all algorithms were tested with the same population sizes by keeping the same number of solution evaluations on each test site. From the simulation results, it can be known that the population size affects both the solution quality and the convergence speed of the algorithm.

Moreover, in the small and medium-sized networks, the effectiveness of TLBO and Jaya algorithms was better when the population size was small and its value ranged between 15-100. While GA and PSO performed better when the population size was large and its value ranged between 200-400. The value of the population size did not significantly affect the performance of WTLBO except for the value less than 10. In the large network, the best result of TLBO, Jaya, WTLBO, and GA was obtained when the population size was small (i.e. 50), while PSO performed better with large population size (i.e. 1000).

Secondly, a comparison was made between the optimum results obtained by each algorithm. Table 5.70 reports the summary of simulation results for cases studies in the form of descriptive and inferential statistics based on the average measure.

Table 5.70: Ccomparative results of all study cases in the form of descriptive and inferential statistics

Case study	**Case 1 (phase1)**			**Case1 (phase2)**			**Case 2**			**Case 3**		
Solution space	**1.2207e+22**			**2.5419e+25**			**5.8208e+57**			**3.5873e+239**		
	Descriptive	**Inferential**		**Descriptive**	**Inferential**		**Descriptive**	**Inferential**		**Descriptive**	**Inferential**	
		P-value	Sig.		P-value	Sig.		P-value	Sig.		P-value	Sig.
TLBO & WTLBO	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**
TLBO & GA	**TLBO**	.027	**TLBO**	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**
TLBO & PSO	**TLBO**	.005	**TLBO**	**TLBO**	.006	**TLBO**	**TLBO**	.000	**TLBO**	**TLBO**	.000	**TLBO**
TLBO & Jaya	Jaya	.997	-	**TLBO**	.461	-	**TLBO**	.990	-	**TLBO**	.000	**TLBO**
WTLBO & GA	**GA**	.000	**GA**	GA	.189	-	GA	.385	-	**WTLBO**	.000	**WTLBO**
WTLBO & PSO	PSO	.473	-	PSO	1.000	-	PSO	.233	-	**WTLBO**	.000	**WTLBO**
WTLBO & Jaya	Jaya	.000	**Jaya**	Jaya	.000	**Jaya**	Jaya	.000	**Jaya**	**WTLBO**	.685	-
GA & PSO	GA	.150	-	GA	.786	-	PSO	.997	-	GA	.000	**GA**

GA & Jaya	Jaya	.003	**Jaya**	Jaya	.011	**Jaya**	Jaya	.000	**Jaya**	Jaya	.000	**Jaya**
PSO & Jaya	Jaya	.003	**Jaya**	Jaya	.038	**Jaya**	Jaya	.000	**Jaya**	Jaya	.000	**Jaya**

We can conclude that for case study 1 (phase 1) TLBO and Jaya were significantly performing better than other algorithms. GA algorithm was significantly better than WTLBO. There was no significant difference between GA & PSO, and between PSO & WTLBO. The results of phase 2 was similar to the phase 1 except that there was no significant difference between GA and WTLBO.

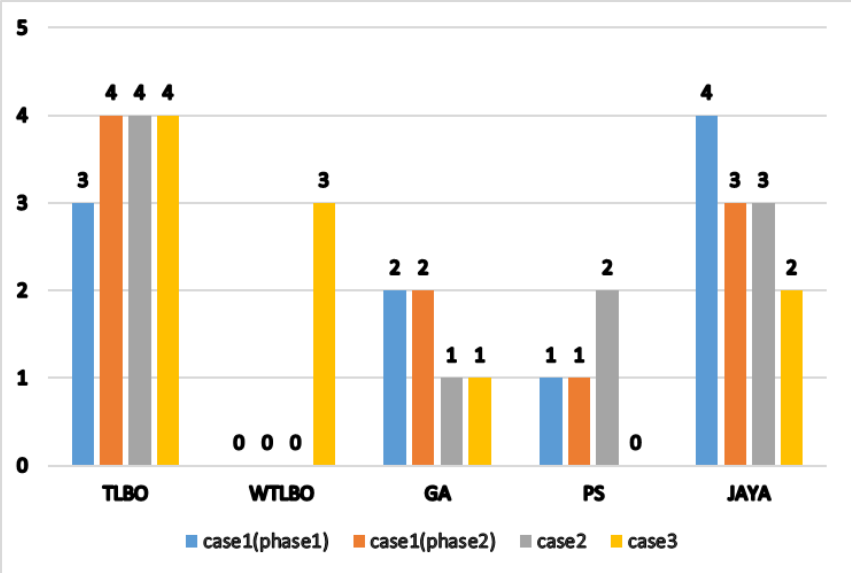
For case study 2, there was no statically significant difference between the mean results of ***Jaya*** and ***TLBO***. There was not a statically significant difference between the mean results of ***WTLBO, GA, and PS***O. While, both Jaya and TLBO were significantly performing better than WTLBO, GA, and PSO.

In the last case syudy, TLBO outperformed the other considering algorithms. The performance of WTLBO improved compared its performance in the small and medium networks, both WTLBO and Jaya were significantly performing better than GA, and PSO. Hence, PSO was the worst.

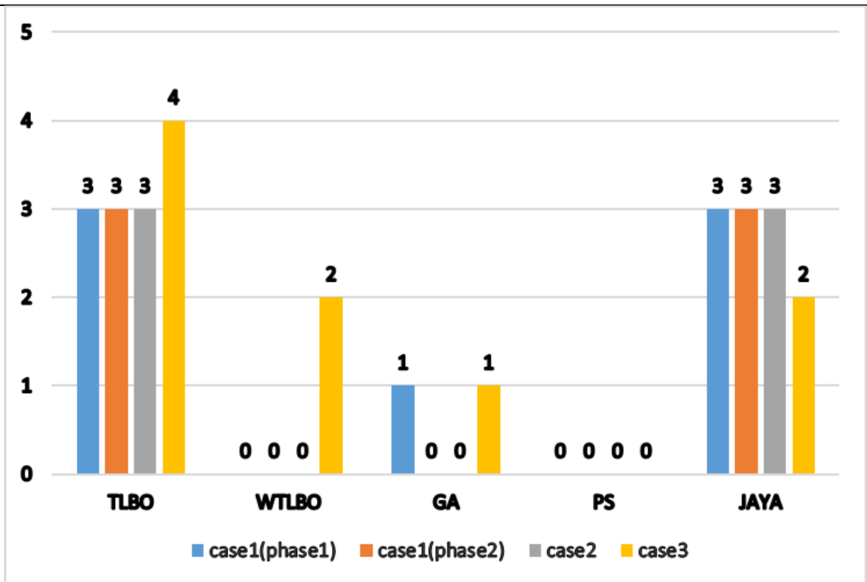
Table 5.71 shows the number of times for which the mean result obtained by the algorithm is better or comparable to the other considered algorithms. The graphical comparison of TLBO, Jaya, WTLBO, GA and PSO algorithms is shown in figure 5.48 and figure 5.49.

Table 5.71: The ability of each algorithm to find a better mean solution

Case study / Solution space	Descriptive statistics					Inferential statistics				
	Case 1 (phase1) 1.2207e+22	**Case1(phase2) 2.5419e+25**	**Case2 5.8208e+57**	**Case3 3.5873e+23 9**	**total**	**Case 1 (phase1) 1.2207e+22**	**Case1(phase2) 2.5419e+25**	**Case2 5.8208e+57**	**Case3 3.5873e+23 9**	**total**
TLBO	3	4	4	4	15	3	3	3	4	13
WTLBO	0	0	0	3	3	0	0	0	2	2
GA	2	2	1	1	6	1	0	0	1	2
PSO	1	1	2	0	4	0	0	0	0	0
Jaya	4	3	3	2	12	3	3	3	2	11



(a)



(b)

Figure 5.48: The ability of each algorithm to outperform other algorithms for the study cases (a) descriptive statistic -(b) inferential statistic

It can be observed from table 5.71 and the graphical comparison (Figure 5.48 and Figure 5.49) that TLBO was significantly performing better in 13 cases. Jaya was able to outperform in 11 cases. On the other hand, WTLBO, GA, and PSO were less competitive: GA and WTLBO were significantly better in 2 cases, while PSO was unable to outperform any other algorithm significantly.

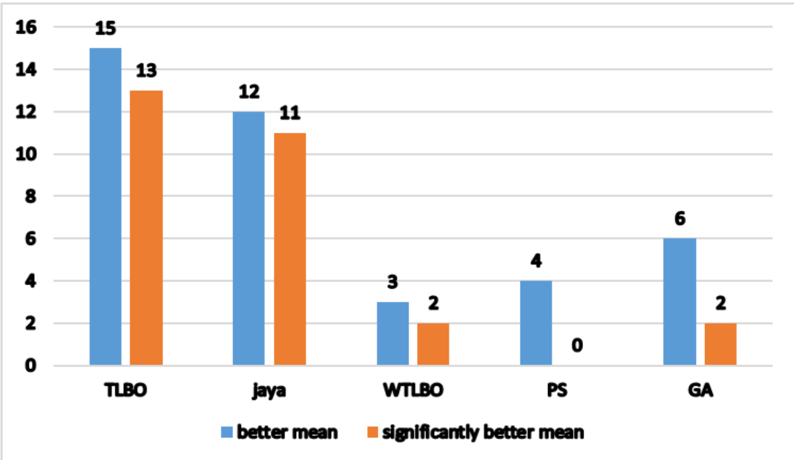
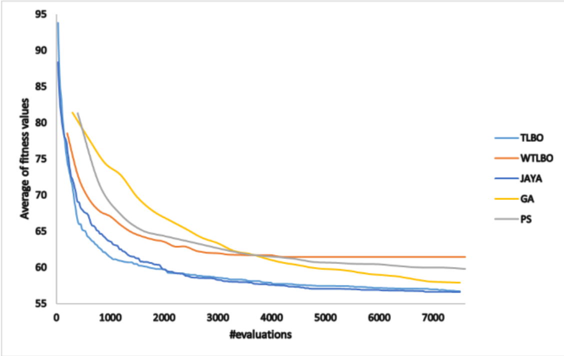


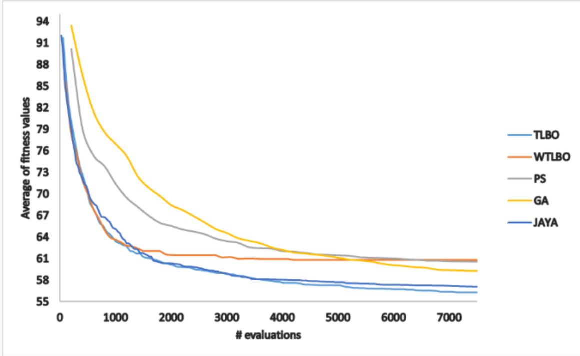
Figure 5.49: The total number of times each algorithm was able to outperform others

When we compared the convergence speed for the algorithms, we found that the convergence of TLBO and Jaya was alike in the small network, and both algorithms were faster than WTLBO, GA, and PSO. While TLBO became faster than Jaya in the medium and large networks. On the other hand, GA and PSO were less competitive. In the small and medium-sized network, GA and PSO reached nearly the same result but PSO was faster. While in the large network, GA gave a better solution quality in fewer evaluations.

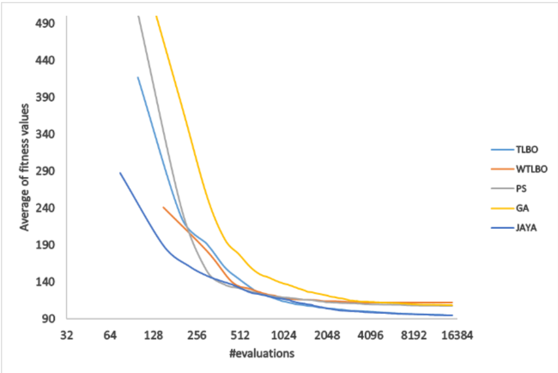
In all cases, WTLBO started faster than GA and PSO but it stuck early into a local minimum. (see fig 5.50)



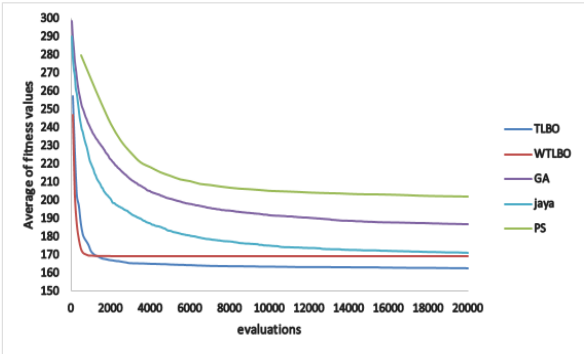
(a)



(b)



(c)



(d)

Figure 5.50: Convergence speed of TLBO, WTLBO, GA, PSO and Jaya algorithms (a) Case 1 phase 1(b) Case 1 phase 2 (c) Case 3 (d) Case 4

6. Conclusions and Discussion

6.1 Overview

The simulation results and data analysis of the study were presented in details in the previous chapter. This chapter summarizes the conclusions and recommendations. Also, it explores the limitations of the study and the directions for future research.

6.2 Summary

Nowadays, most urbanized cities suffer from congestion, which leads to air pollution, increased fuel consumptions, wasting time, and other economic problems. It is therefore imperative to find innovative and effective solutions to reduce this problem. One of the most effective solutions is the development of smart traffic light control so that traffic signals are self-adjustable.

Researchers made several efforts for developing approaches to solve TSOP. These efforts included mathematical optimization models, simulation-based approaches, and metaheuristic techniques. They were introduced in the literature review chapter.

In the optimizer of traffic control system, the selected optimization technique plays an important role in determining the effectiveness of the proposed approach, so metaheuristic optimization algorithms have become popular in the field of traffic signal timing. In this thesis, a simulation-based strategy was used which based on the integration between SUMO microscopic simulator and a proper metaheuristic optimization algorithm. Five algorithms of global optimization techniques, namely TLBO, Jaya, WTLBO, GA, and PSO were tested on a real network and two virtual networks having different characteristics to find the optimal or near optimal timing for traffic signals.

The literature heavily depends on GA and a few other traditional optimization techniques. This work shows that there are other algorithms such as TLBO and Jaya which have not yet been verified in optimizing traffic signals timing, and could give better results.

6.3 Conclusions

The hypotheses were verified and the questions of the study were answered.

Firstly, the common controlling parameters (i.e. population size) influence both the solution quality and the convergence speed of the algorithm. Hence, each algorithm requires the proper tuning of the population size parameter for its operation. The results show that:

- In the small and medium-sized networks, TLBO and Jaya algorithms with smaller population size (15-100) produced better solution quality and better speed than that with higher population size (>100) for the same number of evaluations. While in general, GA and PSO performed better with higher population size (200 – 400).

- In the large network, the best result of TLBO, Jaya, WTLBO, and GA was obtained when the population size was small (i.e. 50), while PSO performed better with large population size (i.e. 1000).
- For the WTLBO, there are no large variations in the performance of the algorithm associated with the population size.

In the literature, there were some studies that analyzed the effect of different population size on the performance of optimization algorithms. *Chen et al* (2012), *Mora-Melia et al* (2017), *Roeva et al* (2014) showed that a large population size may not always prove useful and more efficient than small populations in finding the best solutions. On the other hand, *Rao and Patel* (2012) concluded that higher population size gave better results than smaller population size. Other researchers developed a rule for the recommended populations size (Storn, n.d).

The results of this search showed that the population size is a dependent parameter, and there is no general or exact rule to calculate it. It depends on many parameters such as the problem structure, number of dimensions, search space, and the number and the adjustable parameters of the algorithm. It also depends on the algorithm used. So, the population size parameter must be tuned to balance the ability of the algorithm to diversify and intensify.

Secondly, the performance comparisons were done with the parameter-less algorithms like TLBO, Jaya and other techniques that require their specific parameters, like WTLBO, GA, PSO. From the results analysis, it is evident that:

- Overall, parameter-less algorithms were significantly performing better in finding a better solution in less computation time than other considered algorithms.
- The performance of TLBO and Jaya was alike in small and medium-sized networks. While TLBO performed and scaled well toward the more complex network.
- In the small and medium-sized network, the solution of GA and PSO was significantly the same but in terms of convergence speed, PSO was faster. While GA performed and scaled better than PSO toward the more complex network.
- The performance of WTLBO improved in the large network compared to the small and medium-sized networks. Moreover, in all cases studies, WTLBO suffered from premature convergence and stuck early into a local minimum. Therefore, the internal parameters need to be tuned.

The effectiveness of optimization techniques is sensitive to the common and specific parameters which majorly contributing to the diversification and intensification of the search space, and hence affect the solution quality and the convergence speed remarkably. As a result, these parameters need to be guessed based on previous experience or tuned to suit each new problem. However, this process is difficult, time-consuming, and it may lead to a wrong optimal solution. Finding the optimal parameters is considered as an optimization problem itself.

The five algorithms we evaluated showed much variance in performance. GA, PSO, and WTLBO have their own specific parameters in addition to the common controlling parameters, while TLBO and Jaya do not require any specific parameters, they only require the tuning of common parameters (i.e population size and number of generations).

Abushehab et. Al (2014) used a benchmark function called (*Rastrigin*) to find the best choice of internal parameters. But, he did not prove the relationship between *Rastrigin* benchmark function and the structure of traffic signal timing problem for the case study. In addition, the optimization of traffic signals is a hard and complex problem due to the complexity and stochastic behavior of the road network system. This is why it is impossible to predict neither the problem structure nor the optimal solution. For these reasons, TLBO and Jaya algorithms which are free of parameters are suitable to optimize traffic signals timing in a reasonable computation time.

Thirdly, the performance of the algorithms varies depending on the size and the characteristics of the network to be resolved. Hence, if the algorithm performs well in solving a problem, it does not have to be good at solving another problem. However, TLBO algorithm shows stability, consistency, and scalability in all cases studies.

Above all, the main objective of this study is not to prove that TLBO and Jaya algorithms are the “best” algorithms among the other algorithms. Actually, there is no algorithm capable of solving all problems better than others (no-free-lunch theorem). It all depends on the problem to be solved. Rather, this study raises the awareness of TLBO and Jaya among the researchers working in the field of traffic signals timing optimization.

6.4 Limitations of the Study

This research has some limitations, which could be summarized as follows:

- SUMO simulator consumes a long time to evaluate the candidate solutions during the optimization processes. For example, in case study 1 each run takes about 3 hours to finish the 7500 allowable evaluations. This made it difficult to explore each algorithm with higher populations size. In addition, the performance of some algorithms may change as the number of maximum allowable evaluations increases and this need further research.
- The findings of the research cannot be generalized. Statistical analyses were based on a small sample size (20 independent runs).

6.5 Future research

An additional search could be developed from this study. The following points can be studied in the future:

- In addition to the experiments carried out in this study, more simulation experiments on other cases are still needed to test the capability of generalizing our results.
- Explore the parameter-less algorithms (TLBO and Jaya) in calibration the input parameters of the microscopic simulation model to minimize the error between the observed real data and the simulation output (Abdalhaq&Abu Baker, 2014). Then optimize the traffic signals timing for the well-tuned test site (Schneeberger & Park,2003).
- Propose a modified TLBO and Jaya algorithms with self-adaptive population size parameter, so that they are free of internal and common parameters.
- Investigate the effect of elitism concept on the performance of tested algorithms in TSOP.
- Propose a hybrid algorithm (TLBO-PSO, TLBO-GA, TLBO-local search) to improve the exploration and exploitation capacity.
- Exploring other types of metaheuristic techniques to optimize traffic signals timing and compare them with TLBO and Jaya. This helps

with determining the most consistent algorithm for solving the search problem.

- Optimize the traffic light setting with respect to several objectives simultaneously (eg: delay time, noise emission, fuel consumption) using multi-objective TLBO.

References

A

Abdalhaq B., Abu Baker M. (2014). **Using Meta Heuristic Algorithms to Improve Traffic Simulation**, Journal of Algorithms and Optimization, Vol. 2 1ss. 4, PP. 110-128.

Abushehab, R. K., Abdalhaq, B. K., & Sartawi, B. (2014). **Genetic vs. particle swarm optimization techniques for traffic light signals timing**. 2014 6th International Conference on Computer Science and Information Technology (CSIT). doi:10.1109/csit.2014.6805975

Adacher, L. (2012). **A Global Optimization Approach to Solve the Traffic Signal Synchronization Problem**. Procedia - Social and Behavioral Sciences, 54, 1270-1277. doi:10.1016/j.sbspro.2012.09.841

Affenzeller, M., Wagner, S., & Winkler, S. (2008). **Evolutionary Systems Identification: New Algorithmic Concepts and Applications**. INTECH Open Access Publisher.ch. 2, pp. 29–48

Akcelik, R., (March 1981). **Traffic signals: capacity and timing analysis**. Research report ARR no. 123.Melbourne, VIC, Australia: Australian Road Research Board.

Alkhraisat, H., & Rashaideh, H. (2016). Dynamic Inertia Weight Particle Swarm Optimization for Solving Nonogram Puzzles. International Journal of Advanced Computer Science and Applications, 7(10). doi:10.14569/ijacsa.2016.071037

Álvarez, P., & Hadi, M. A. (2014). **Evaluating the effectiveness of signal timing optimization based on microscopie simulation**. Obrasy proyectos, (16), 85-94. doi:10.4067/s0718-28132014000200006

Arasomwan, M. A., & Adewumi, A. O. (2013). On the Performance of Linear Decreasing Inertia Weight Particle Swarm Optimization for Global Optimization. The Scientific World Journal, 2013, 1-12. doi:10.1155/2013/860289

Arockia, S. (2017). **Nature Inspired Metaheuristic Algorithms**. International Research Journal of Engineering and Technology (IRJET), 04(10). Retrieved from https://irjet.net/

Aljaafreh A., & Al-Oudat N. (2014): **Optimized Timing Parameters for Real-Time Adaptive Traffic Signal Controller**, UKSim-AMSS 16th International Conference on Computer Modelling and Simulation

B

Barceló, J., & Casas, J. (2006). **Stochastic Heuristic Dynamic Assignment Based on AIMSUN Microscopic Traffic Simulator**. Transportation Research Record: Journal of the Transportation Research Board, 1964, 70-80. doi:10.3141/1964-09

Bazzan, C., & Anna L. (2007). **Traffic as a Complex System: Four Challenges for Computer Science and Engineering** ………..

Boyd, S. P., & Vandenberghe, L. (2015). **Convex optimization**. Cambridge: Cambridge University Press.

Burghout W (2004) **Hybrid microscopic-mesoscopic traffic simulation**. Unpublished doctoral thesis, KTH Royal Institute of Technology, Stockholm.

Busch, Fritz & Kruse, G. (2001). **MOTION for SITRAFFIC - a modern approach to urban traffic control**. 61 - 64. 10.1109/ITSC.2001.948630.

Beheshti, Z., & Shamsuddin, S. M. (2013). **A review of population-based meta-heuristic algorithm**. International Journal of Advances in Soft Computing and Its Applications. 5. 1-35.

C

Ceder, A., & Reshetnik, I. (2001). **An algorithm to minimize queues at signalized intersections**. Journal of the Operational Research Society, 52(6), 615-622. doi:10.1057/palgrave.jors.2601138

Ceylan H. and Ceylan H., 2012. **A Hybrid Harmony Search and TRANSYT Hill Climbing Algorithm for Signalized Stochastic Equilibrium Transportation Networks**. Transportation Research Part C: Emerging Technologies, 25, 152–167.

Chen, T., Tang, K., Chen, G., & Yao, X. (2012). **A large population size can be unhelpful in evolutionary algorithms**. Theoretical Computer Science, 436, 54-70. doi:10.1016/j.tcs.2011.02.016

Chong, E. K., & Zak, S. H. (2013). **An introduction to optimization. Hoboken**, NJ: Wiley.

Chowdhury D., Santen L., & Schadschneider A. (2000). "**Statistical physics of vehicular traffic and some related systems**", Physics Reports, vol. 329, iss. 4-6, pp. 199-329, ISSN 0370-1573.

D

Damay, N. (2015). **Multiple-objective optimization of traffic light using a genetic algorithm and a microscopic traffic simulator** (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-168413

DAns, G., & Gazis, D. (1975). **Optimal control of oversaturated store-and-forward transportation networks**. Transport Sci; 10: 1–19.

Dellorco, M., Baskan, O., & Marinelli, M. (2013). **A Harmony Search Algorithm approach for optimizing traffic signal timings**. PROMET - Traffic&Transportation, 25(4). doi:10.7307/ptt.v25i4.979

Denos C. Gazis, 1967. **Mathematical Theory of Automobile Traffic**. Science, 157(3789), 273-281.

DLR and contributors (2013): SUMO homepage. http://sumo.sourceforge.net/

E

Ehlert, A., Schneck, A., & Chanchareon, N. (2017). **Junction parameter calibration for mesoscopic simulation in Vissim**. Transportation Research Procedia, 21, 216-226. doi:10.1016/j.trpro.2017.03.091

F

FHWA, CORSIM 6.0, **User's Guide and Reference Manual**, December 2006

G

Gao, K., Zhang, Y., Sadollah, A., & Su, R. (2016). **Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search**. Applied Soft Computing, 48, 359-372. doi:10.1016/j.asoc.2016.07.029

Garcia-Nieto, J., Olivera, A. C., & Alba, E. (2013). **Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization**. IEEE Transactions on Evolutionary Computation, 17(6), 823-839. doi:10.1109/tevc.2013.2260755

Gartner, N. (1990). Opac: **Strategy for Demand-Responsive Decentralized Traffic Signal Control**. Control, Computers, Communications in Transportation, 241-244. doi:10.1016/b978-0-08-037025-5.50041-4

Gawron Ch.(1998): **Simulation-Based Traffic Assignment: Computing User Equilibria in Large Street Networks**. Inaugural Dissertation, Universität zu Köln, Germany,

Gazis, D. C. (1964). **Optimum control of a system of oversaturated intersections**.

Goldberg, D. E., & Deb, K. (1991). **A Comparative Analysis of Selection Schemes Used in Genetic Algorithms**. Foundations of Genetic Algorithms, 69-93. doi:10.1016/b978-0-08-050684-5.50008-2

H

Hale, D. (2005), **Traffic Network Study Tool – TRANSYT-7F**, United States Version. McTrans Center in the University of Florida, Gainesville.

Hewage, K.N., Ruwanpura, J.Y., (2004). **Optimization of traffic signal light timing using simulation**. In: WSC ’04: Proceedings of the 36th Conference on Winter Simulation, Winter Simulation Conference, pp. 1428–1436.

Holland, J.H. (1975). **Adaption in Natural and Artificial Systems**. University of Michigan Press.

Hoogendoorn SP.,& Bovy PHL, (2001). **State-of-the-art of vehicular traffic flow modeling**. J. Syst. Control. Eng.. 215. 283-303. 10.1243/0959651011541120.)

Hu, W., Wang, H., Yan, L., & Du, B. (2015). **A swarm intelligent method for traffic light scheduling: application to real urban traffic networks**. Applied Intelligence, 44(1), 208-231. doi:10.1007/s10489-015-0701-y

Husch, D., and Albeck, J.(2006), **Synchro Studio 7 User’s Guide**. Trafficware, Ltd.

J

Jiajia He, Zai’en Hou, 2012. **Ant Colony Algorithm for Traffic Signal Timing Optimization**. Advances in Engineering Software, 43(1), 14-18.

Jiao, P., & Sun, T. (2014). **Multiobjective Traffic Signal Control Model for Intersection Based on Dynamic Turning Movements**

Estimation. Mathematical Problems in Engineering, 2014, 1-8. doi:10.1155/2014/608194

Jiao, P., Li, Z., Liu, M., Li, D., & Li, Y. (2015). **Real-time traffic signal optimization model based on average delay time per person**. Advances in Mechanical Engineering, 7(10), 168781401561350. doi:10.1177/1687814015613500

Jebari, K. (2013). **Selection Methods for Genetic Algorithms**. International Journal of Emerging Sciences. 3. 333-344.

K

Kaman Science Corporation, 1996. **CORSIM User Manual Version 1.01**. ISBN: Colorado 80933-7463.

Karakuzu, C., Demirci, O., (2010). **Fuzzy logic based smart traffic light simulator design and hardware implementation**. Appl. Soft Comput. 10 (1), 66–73.

Kaur T, Agrawal S (2014) **Adaptive traffic lights based on hybrid of neural network and genetic algorithm for reduced traffic congestion**. In: Recent Advances in Engineering and Computational Sciences (RAECS). IEEE, pp 1–5

Kennedy, J.; Eberhart, R. (1995). "**Particle Swarm Optimization**". Proceedings of IEEE International Conference on Neural Networks. **IV**. pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

Kotushevski G. and Hawick K. A. (2009). "**A review of traffic simulation software**". Technical Report CSTN-095, Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand.

Krajzewicz, D. (2010). **Traffic Simulation with SUMO – Simulation of Urban Mobility**. Fundamentals of Traffic Simulation International Series in Operations Research & Management Science, 269-293. doi:10.1007/978-1-4419-6142-6_7

Krajzewicz, D. (2014). **SUMO in Scientific Literature**, 2002–2012. Simulation of Urban Mobility Lecture Notes in Computer Science, 160-174. doi:10.1007/978-3-662-45079-6_12

Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L. (2012): "**Recent Development and Applications of SUMO - Simulation of Urban Mobility**". In: International

Journal on Advances in Systems and Measurements, 5 (3&4), pp.128-138. ISSN 1942-261x

Krajzewicz, D., Hartinger, M., Hertkorn, G., Nicolay, E., Rössel, C., Ringel, J., Wagner, P. (2004):**Recent Extensions to the open source Traffic Simulation SUMO**. In: Proceedings of the 10th World Conference on Transport Research (on CD), WCTR04 - 10th World Conference on Transport Research, Istanbul (Turkey), 2004-07-04 - 2004-07-08

Krajzewicz, D., Hertkorn, G., Rössel, C., & Wagner, P. (2002). "**SUMO (Simulation of Urban MObility); An open-Source Traffic Simulation**".In: Proceedings of the 4th Middle EastSymposium on Simulation and Modelling (MESM2002),Edited by: A.~Al-Akaidi, pp. 183 - 187, SCS EuropeanPublishing House, ISBN 90-77039-09-0

Krauss, S. (1998). **Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics** (Unpublished doctoral dissertation).

Kumar, A., & Sing, RR. (2017). **Traffic congestion and possible solution in urban transport system**, 4th International Conference on Emerging Trends in Engineering Technology, Science and Management,603-607.

L

Levene, H. (1960). **In Contributions to Probability and Statistics**: Essays in Honor of Harold Hotelling, I. Olkin et al. eds., Stanford University Press, pp. 278-292.

Li, NY. (2010). **Signal control optimization models with the consideration of both pedestrians and vehicles**. Master's Thesis, Beijing Jiaotong University, Beijing, China.

Liang, Z., Chen, H., Liang, Z., Qin, C., Zhou, Y., & Fang, W. (2017). **The design and optimization of overlapping phase combination for isolated intersection**. 2017 4th International Conference on Transportation Information and Safety (ICTIS). doi:10.1109/ictis.2017.8047917

Lieberman, E., Chang, J., & Prassas, E. (2000). **Formulation of Real-Time Control Policy for Oversaturated Arterials**. Transportation Research Record: Journal of the Transportation Research Board, 1727, 77-88. doi:10.3141/1727-10

Lim, G., Kang, J.J., Hong, Y., (2001). **The optimization of traffic signal light using artificial intelligence**. In: FUZZ-IEEE, pp. 1279–1282.

López-Neri, E., Ramírez-Treviño, A., & López-Mellado, E. (2010). **A modeling framework for urban traffic systems microscopic simulation**. Simulation Modelling Practice and Theory,18(8), 1145-1161. doi:10.1016/j.simpat.2009.09.007

Lowrie, P. (1982). **The Sydney coordinated adaptive control system— principles, methodology, algorithms**. New York: IEE Conference Publication.

Lu, M. E. (2015). **Dynamic network-wide traffic signal optimization** (Unpublished doctoral dissertation). Braunschweig

Luke, S. (2013). **Essentials of metaheuristics**. Lulu.com.

M

Maciejewski, M.(2010). **A comparison of microscopic traffic flow simulation systems for an urban area**. Transport Problems : an International Scientific Journal. 5. (REF FOR SUMO2)

Mafarja, M., & Mirjalili, S. (2018). **Whale optimization approaches for wrapper feature selection**. Applied Soft Computing, 62, 441-453. doi:10.1016/j.asoc. 2017.11.006

Mahut, M., (2001). **A multi-lane extension of the Space-Time Queue Model of Traffic Dynamics**. TRISTAN IV, Azores Islands

Mccrea, J., & Moutari, S. (2010). A hybrid macroscopic-based model for traffic flow in road networks. European Journal of Operational Research, 207(2), 676-684. doi:10.1016/j.ejor.2010.05.018

Michalopoulos, PG., & Stephanopoulos, G.(1978), **Optimal control of oversaturated intersections: theoretical and practical considerations**. Traffic Eng Control; 19: 216–221.

Mihăiţă, A. S., Dupont, L., & Camargo, M. (2018). Multi-objective traffic signal optimization using 3D mesoscopic simulation and evolutionary algorithms. Simulation Modelling Practice and Theory, 86, 120-138. doi:10.1016/j.simpat.2018.05.005

Miller, A. J. (1963). **Settings for fixed-cycle traffic signals**. J Oper Res Soc; 14: 373–386.

Mirchandani, P., & Head, L. (2001). **A real-time traffic signal control system: architecture, algorithms, and analysis**. Transportation Research Part C: Emerging Technologies, 9(6), 415-432. doi:10.1016/s0968-090x(00)00047-4

Mitsakis, E., Aifadopoulou, G., Grau, J. M., Chrysohoou, E., & Morfoulaki, M. (2014). Combination Of Macroscopic And Microscopic Transport Simulation Models: Use Case In Cyprus. International Journal For Traffic And Transport Engineering, 4(2), 220-233. doi:10.7708/ijtte.2014.4(2).08

Moder, K. (2010). **Alternatives to F-Test in One Way ANOVA in case of heterogeneity of variances (a simulation study)**. Psychological Test and Assessment Modeling, Volume 52,2010 (4), 343-353

Moghimi, B., Kalantari, N., & Kamga, C. (2018). **Simultaneous Optimization of Signal Timing and Capacity Improvement in Urban Transportation Networks Using Simulated Annealing**. arXiv preprint arXiv:1802.01717.

Mora-Melià, D., Martínez-Solano, F. J., Iglesias-Rey, P. L., & Gutiérrez-Bahamondes, J. H. (2017). **Population Size Influence on the Efficiency of Evolutionary Algorithms to Design Water Networks**. Procedia Engineering, 186, 341-348. doi:10.1016/j.proeng.2017.03.209

Mehboob, U., Qadir, J., Ali, S., & Vasilakos, A. (2016). **Genetic algorithms in wireless networking: Techniques, applications, and issues**. Soft Computing, 20(6), 2467-2501. doi:10.1007/s00500-016-2070-9

N

Neumüller, C., Wagner, S. (2011). **Parameter Meta-optimization of Metaheuristic Optimization Algorithms**. Master's Thesis, Hagenberg, Austria.

Neumüller, C., Wagner, S., Kronberger, G., & Affenzeller, M. (2012). **Parameter Meta-optimization of Metaheuristic Optimization Algorithms**. Computer Aided Systems Theory – EUROCAST 2011 Lecture Notes in Computer Science, 367-374. doi:10.1007/978-3-642-27549-4_47

O

Olstam, J. J., & Tapani, A. (2004). **Comparison of car-following models**. Linköping: Swedish National Road and Transport Research Institute.

Ozbay, K., Mudigonda, S., & Bartin, B. (2005). **Development and calibration of an integrated freeway and toll plaza model for new jersey turnpike using paramics microscopic simulation tool**. Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.doi:10.1109/itsc.2005.1520216

One-way ANOVA. (n.d.). Retrieved November 23, 2017, from https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php

P

Pandey, H. M. (2016). **Jaya a novel optimization algorithm: What, how and why?**6th International Conference - Cloud System and Big Data Engineering (Confluence). doi:10.1109/confluence.2016.7508215

Park, B. & Kamarajugadda, A. (2007). **Development and Evaluation of a Stochastic Traffic Signal Optimization Method**. International Journal of Sustainable Transportation, 1(3), 193-207. doi:10.1080/15568310600737568

Park, B., & Yun, I. (2006). **Evaluation of Stochastic Optimization Methods of Traffic Signal Control Settings for Coordinated Actuated Signal Systems**, Transportation Research Board 85th Annual Meeting, Washington DC, United States, 2006-1-22 to 2006-1-26.

PASSER V. Texas Transportation Institute, College Station, Texas, 2002.

Pattberg, B. (n.d.). **SUMO – Simulation of Urban MObility**. Retrieved April 03, 2018, from http://sumo.dlr.de/

PTV AG (2015), **VISSIM Version 8.0**, User Manual, August 2015.

Pandey, H. M. (2016). **Jaya a novel optimization algorithm: What, how and why?** 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence). doi:10.1109/confluence.2016.7508215

R

Rahmani, S., Mousavi, S., & Kamali, M. (2011). **Modeling of road-traffic noise with the use of genetic algorithm**. Applied Soft Computing, 11(1), 1008-1013. doi:10.1016/j.asoc.2010.01.022

Rao, R. V. (2016b). **Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems**. International Journal of Industrial Engineering Computations, 19-34. doi:10.5267/j.ijiec.2015.8.004

Rao, R. V. (2016a). **Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems**. Decision Science Letters, 1-30. doi:10.5267/j.dsl.2015.9.003

Rao, R. V., & Patel, V. (2012). **An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems**. International Journal of Industrial Engineering Computations, 3(4), 535-560. doi:10.5267/j.ijiec.2012.03.007

Rao, R., Rai, D., Ramkumar, J., & Balic, J. (2016). **A new multi-objective Jaya algorithm for optimization of modern machining processes**. Advances in Production Engineering & Management, 11(4), 271-286. doi:10.14743/apem2016.4.226

Rao, R., Savsani, V., & Vakharia, D. (2011). **Teaching–learning-based optimization: A novel method for constrained mechanical design optimizationproblems**. Computer-AidedDesign,43(3), 303-315. doi:10.1016/j.cad.2010.12.015

Rao, R., Savsani, V., & Vakharia, D. (2012). **Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems**. Information Sciences,183(1), 1-15. doi:10.1016/j.ins.2011.08.006

Rao, R.V. (2015). **Teaching Learning Based Optimization and its Engineering Applications**. London: Springer-Verlag.

Rao, R.V., Savsani, V.J., & Vakharia, D.P. (2011). **Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems**. Computer-Aided Design,43, 303–315.

Rao, R.V., Savsani, V.J., &Balic, J. (2012b). **Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems**. Engineering Optimization, 44(12), 1447-1462.

Ratrout, N. T., & Reza, I. (2014). **Comparison of Optimal Signal Plans by Synchro & TRANSYT-7F Using PARAMICS – A Case Study**. Procedia Computer Science, 32, 372-379. doi:10.1016/j.procs.2014.05.437

Renfrew, D., & Yu, H. (2009). **Traffic signal control with ant colony optimization**: a thesis(Unpublished master's thesis).

Robertson, D., & Bretherton, R. (1991). **Optimizing networks of traffic signals in real time-the SCOOT method**. IEEE Transactions on Vehicular Technology, 40(1), 11-15. doi:10.1109/25.69966

Roeva, O., Fidanova, S., & Paprzycki, M. (2014). **Population Size Influence on the Genetic and Ant Algorithms Performance in Case of Cultivation Process Modeling**. Recent Advances in Computational Optimization Studies in Computational Intelligence, 107-120. doi:10.1007/978-3-319-12631-9_7

Rothlauf, F. (2013). **Design of modern heuristics: Principles and application**. Berlin: Springer.

Rouphail, N. M., Park, B., & Sacks, J. (2000). **Direct Signal Timing Optimization: Strategy Development and Results**. In XI Pan American Conference in Traffic and Transportation Engineering, Brazil, November 19-23,2000.

S

Sabar, N. R., Kieu, L. M., Chung, E., Tsubota, T., & Almeida, P. E. (2017). A memetic algorithm for real world multi-intersection traffic signal optimisation problems. Engineering Applications of Artificial Intelligence, 63, 45-53. doi:10.1016/j.engappai.2017.04.021

Sastry K., Goldberg D., and Kendall G., (2005): "**Genetic algorithms," in Search methodologies**, pp. 97–125, Springer

Satapathy, S. C., Naik, A., & Parvathi, K. (2013). **Weighted Teaching-Learning-Based Optimization for Global Function Optimization**. Applied Mathematics, 04(03), 429-439. doi:10.4236/am.2013.43064

Schneeberger, J. D., Park B. (2003). **Evaluation of traffic signal-timing optimization methods using a stochastic and microscopic simulation program** (Unpublished doctoral dissertation). Retrieved from http://cts.virginia.edu/docs/UVACTS-5-0-4.pdf

Storn, R. (n.d.). **On the usage of differential evolution for function optimization**. Proceedings of North American Fuzzy Information Processing. doi:10.1109/nafips.1996.534789

T

Talbi, E. (2009). **Metaheuristics: From design to implementation**. Hoboken, NJ: John Wiley & Sons.

Traffic signals: Capacity and timing analysis. (1981). Transportation Research Part A: General,15(6), 505. doi:10.1016/0191-2607(81)90135-7

trafficware. **Synchro studio user guide**. U, U.

Torres-Jiménez, J., & Pavón, J. (2014). **Applications of metaheuristics in real-life problems**. Progress in Artificial Intelligence, 2(4), 175-176. doi:10.1007/s13748-014-0051-8

W

Warberg, A., Larsen, J., & Jørgesen, R. M. (2008). **Green wave traffic optimization - a survey**. Kgs. Lyngby: Informatics and Mathematical Modeling, Technical University of Denmark.

Webster, F. V. (1958). **Traffic signal settings**, Road Research technical Paper No.39, HMSO, London, 1958.

Welch's ANOVA: Definition, Assumptions. (n.d.). Retrieved February 24, 2018, from http://www.statisticshowto.com/welchs-anova/

Wolpert, D., & Macready, W. (1997). **No free lunch theorems for optimization**. IEEE Transactions on Evolutionary Computation, 1(1), 67-82. doi:10.1109/4235.585893

X

Xie, X., Smith, S. F., Chen, T., & Barlow, G. J. (2014). **Real-time traffic control for sustainable urban living**. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). doi:10.1109/itsc.2014.6957964

Y

Yun I., Park B. (2006): **Application of Stochastic Optimization Method for an Urban Corridor**, Proceedings of the 2006 Winter Simulation Conference, 1-4244-0501-7/06/$20.00 ©2006 IEEE

Yang, X. (2010). **Nature-inspired metaheuristic algorithms**. Frome (U.K.): Luniver Press.

Z

Zhao, Y., Fang, W., & Qing, C. (2016). **A PSO based Signal Timing Optimization Approach of Phase Combination**. Proceedings of the 2016 International Conference on Civil, Transportation and Environment. doi:10.2991/iccte-16.2016.54

Zou, Feng, et al. "**Teaching–learning-Based optimization with learning experience of other learners and its application**." Applied Soft Computing, vol. 37, 2015, pp. 725–736., doi:10.1016/j.asoc.2015.08.047.

Appendices

Appendix A: post hoc comparisons tables

Table 4.3 Statistical results for TLBO by Games-Howell post hoc test (case 1 phase duration 10-60)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound	
5.00	15.00	7.46800*	1.45899	.001	2.4271	12.5089	15
	30.00	6.77854*	1.50876	.004	1.6307	11.9264	30
	50.00	6.93874*	1.47176	.003	1.8716	12.0059	50
	75.00	6.87256*	1.45294	.003	1.8438	11.9013	75
	100.00	6.07327*	1.44747	.011	1.0553	11.0912	100
	200.00	5.32940*	1.47181	.034	.2621	10.3967	200
	300.00	4.05296	1.49098	.198	-1.0553-	9.1612	-
	400.00	3.57221	1.47946	.325	-1.5112-	8.6556	-
15	30.00	-.68946-	.54213	.932	-2.4973-	1.1184	-
	50.00	-.52927-	.42852	.943	-1.9410-	.8824	-
	75.00	-.59545-	.35861	.766	-1.7753-	.5844	-

	100.00	-1.39474-*	.33574	.006	-2.5038-	-.2857-	15
	200.00	-2.13860-*	.42869	.000	-3.5509-	-.7263-	15
	300.00	-3.41504-*	.49050	.000	-5.0415-	-1.7886-	15
	400.00	-3.89579-*	.45427	.000	-5.3961-	-2.3955-	15
30	50.00	.16019	.57560	1.000	-1.7437-	2.0641	-
	75.00	.09401	.52564	1.000	-1.6696-	1.8576	-
	100.00	-.70528-	.51031	.894	-2.4299-	1.0194	-
	200.00	-1.44914-	.57573	.259	-3.3534-	.4551	-
	300.00	-2.72558-*	.62313	.003	-4.7757-	-.6754-	30
	400.00	-3.20633-*	.59503	.000	-5.1688-	-1.2438-	30
50	75.00	-.06618-	.40746	1.000	-1.4138-	1.2815	-
	100.00	-.86547-	.38748	.410	-2.1554-	.4245	-
	200.00	-1.60934-*	.47031	.036	-3.1550-	-.0636-	50
	300.00	-2.88578-*	.52727	.000	-4.6229-	-1.1487-	50
	400.00	-3.36653-*	.49374	.000	-4.9900-	-1.7430-	50
75	100.00	-.79929-	.30842	.225	-1.8145-	.2160	-
	200.00	-1.54316-*	.40764	.015	-2.8914-	-.1949-	75
	300.00	-2.81960-*	.47222	.000	-4.3943-	-1.2449-	75
	400.00	-3.30035-*	.43447	.000	-4.7423-	-1.8584-	75
100	200.00	-.74387-	.38767	.607	-2.0345-	.5468	-
	300.00	-2.02031-*	.45509	.004	-3.5492-	-.4914-	100
	400.00	-2.50106-*	.41579	.000	-3.8909-	-1.1112-	100
200	300.00	-1.27644-	.52741	.304	-3.0140-	.4611	-
	400.00	-1.75719-*	.49389	.026	-3.3812-	-.1332-	200
300	400.00	-.48075-	.54840	.993	-2.2844-	1.3229	-
*. The mean difference is significant at the 0.05 level.							

Table 4.6 Statistical results for WTLBO by Games-Howell post hoc test (case 1 phase duration 10-60)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
					95% Confidence Interval		
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	Lower Bound	Upper Bound	
5.00	15.00	11.57711*	2.51363	.002	3.1230	20.0313	15
	30.00	12.36396*	2.39368	.001	4.2009	20.5270	30
	50.00	12.89051*	2.32664	.000	4.8736	20.9074	50
	75.00	12.93822*	2.36458	.000	4.8403	21.0362	75
	100.00	13.37704*	2.35886	.000	5.2916	21.4625	100
	200.00	12.67607*	2.32229	.001	4.6682	20.6840	200
	300.00	11.80902*	2.41297	.002	3.6016	20.0165	300
	400.00	10.78158*	2.38462	.004	2.6391	18.9241	400
15	30.00	.78685	1.31609	1.000	-3.5675-	5.1412	-
	50.00	1.31341	1.18980	.969	-2.6902-	5.3170	-
	75.00	1.36112	1.26239	.973	-2.8379-	5.5601	-
	100.00	1.79993	1.25163	.874	-2.3690-	5.9688	-
	200.00	1.09896	1.18127	.989	-2.8831-	5.0810	-
	300.00	.23192	1.35085	1.000	-4.2267-	4.6905	-
	400.00	-.79553-	1.29954	.999	-5.1012-	3.5101	-
30	50.00	.52656	.90940	1.000	-2.4876-	3.5407	-
	75.00	.57427	1.00250	1.000	-2.7241-	3.8726	-

	100.00	1.01308	.98892	.981	-2.2424-	4.2686	-
	200.00	.31211	.89821	1.000	-2.6697-	3.2939	-
	300.00	-.55493-	1.11183	1.000	-4.2102-	3.1003	-
	400.00	-1.58238-	1.04889	.845	-5.0299-	1.8651	-
50	75.00	.04771	.82978	1.000	-2.6899-	2.7853	-
	100.00	.48652	.81332	1.000	-2.1945-	3.1676	-
	200.00	-.21444-	.70022	1.000	-2.5160-	2.0871	-
	300.00	-1.08149-	.95900	.965	-4.2695-	2.1065	-
	400.00	-2.10894-	.88527	.325	-5.0389-	.8210	-
75	100.00	.43881	.91624	1.000	-2.5726-	3.4502	-
	200.00	-.26215-	.81750	1.000	-2.9626-	2.4383	-
	300.00	-1.12920-	1.04771	.974	-4.5814-	2.3230	-
	400.00	-2.15665-	.98067	.427	-5.3815-	1.0682	-
100	200.00	-.70097-	.80080	.993	-3.3437-	1.9417	-
	300.00	-1.56801-	1.03472	.841	-4.9801-	1.8441	-
	400.00	-2.59546-	.96678	.188	-5.7759-	.5850	-
200	300.00	-.86704-	.94841	.990	-4.0253-	2.2912	-
	400.00	-1.89449-	.87378	.448	-4.7908-	1.0018	-
300	400.00	-1.02745-	1.09218	.989	-4.6196-	2.5647	-
*. The mean difference is significant at the 0.05 level.							

Table 4.9 Statistical results for Jaya by Games-Howell post hoc test (case 1 phase duration 10-60)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	11.38775*	1.69094	.000	5.6912	17.0843	15
	30.00	13.06530*	1.54931	.000	7.6889	18.4417	30
	50.00	11.52099*	1.57565	.000	6.0916	16.9503	50
	75.00	11.28246*	1.59593	.000	5.8101	16.7549	75
	100.00	10.42518*	1.64052	.000	4.8518	15.9986	100
	200.00	9.82678*	1.62510	.000	4.2892	15.3643	200
	300.00	8.14075*	1.60282	.001	2.6533	13.6282	300
	400.00	8.64126*	1.62699	.001	3.0994	14.1831	400
15	30.00	1.67755	.72653	.379	-.8166-	4.1717	-
	50.00	.13324	.78112	1.000	-2.4910-	2.7575	-
	75.00	-.10529-	.82126	1.000	-2.8380-	2.6274	-
	100.00	-.96257-	.90488	.976	-3.9437-	2.0186	-
	200.00	-1.56097-	.87660	.694	-4.4554-	1.3335	-
	300.00	-3.24700-*	.83458	.012	-6.0175-	-.4765-	15
	400.00	-2.74649-	.88011	.076	-5.6515-	.1586	15

30	50.00	-1.54432-*	.38890	.011	-2.8445-	-.2442-	30
	75.00	-1.78284-*	.46428	.017	-3.3500-	-.2157-	30
	100.00	-2.64012-*	.59991	.005	-4.6875-	-.5928-	30
	200.00	-3.23852-*	.55635	.000	-5.1318-	-1.3452-	30
	300.00	-4.92455-*	.48746	.000	-6.5738-	-3.2753-	30
	400.00	-4.42405-*	.56185	.000	-6.3368-	-2.5113-	30
50	75.00	-.23853-	.54576	1.000	-2.0370-	1.5600	-
	100.00	-1.09580-	.66498	.772	-3.3095-	1.1179	-
	200.00	-1.69420-	.62596	.185	-3.7709-	.3825	-
	300.00	-3.38023-*	.56561	.000	-5.2470-	-1.5135-	50
	400.00	-2.87973-*	.63085	.002	-4.9736-	-.7859-	50
75	100.00	-.85728-	.71170	.950	-3.2073-	1.4927	-
	200.00	-1.45568-	.67538	.454	-3.6808-	.7694	-
	300.00	-3.14171-*	.61986	.000	-5.1793-	-1.1041-	75
	400.00	-2.64120-*	.67992	.011	-4.8818-	-.4006-	75
100	200.00	-.59840-	.77491	.997	-3.1462-	1.9494	-
	300.00	-2.28443-	.72702	.072	-4.6812-	.1123	-
	400.00	-1.78393-	.77886	.373	-4.3445-	.7766	-
200	300.00	-1.68603-	.69152	.294	-3.9617-	.5896	-
	400.00	-1.18553-	.74583	.804	-3.6367-	1.2657	-
300	400.00	.50051	.69595	.998	-1.7902-	2.7912	-
*. The mean difference is significant at the 0.05 level.							

Table 4.12 Statistical results for GA by Games-Howell post hoc test (case 1 phase duration 10-60)

Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	.24553	.73595	1.000	-2.1882-	2.6792	
	30.00	-.06226-	.81623	1.000	-2.7458-	2.6213	
	50.00	1.38740	.79860	.720	-1.2397-	4.0145	
	75.00	.91109	.86983	.978	-1.9481-	3.7702	
	100.00	1.64905	.70283	.347	-.6886-	3.9867	
	200.00	2.67160*	.72076	.019	.2826	5.0606	
	300.00	2.69630*	.66306	.009	.4659	4.9267	
	400.00	2.53776*	.65207	.015	.3350	4.7405	
15	30.00	-.30779-	.69944	1.000	-2.6151-	1.9996	
	50.00	1.14188	.67878	.753	-1.0946-	3.3783	
	75.00	.66556	.76131	.993	-1.8564-	3.1876	
	100.00	1.40353	.56298	.268	-.4492-	3.2562	
	200.00	2.42607*	.58522	.005	.5023	4.3499	

	300.00	2.45078*	.51247	.001	.7518	4.1498	
	400.00	2.29223*	.49818	.002	.6340	3.9504	
30	50.00	1.44966	.76508	.621	-1.0651-	3.9645	
	75.00	.97335	.83916	.960	-1.7872-	3.7339	
	100.00	1.71131	.66450	.235	-.4918-	3.9145	
	200.00	2.73386*	.68344	.009	.4749	4.9928	
	300.00	2.75856*	.62228	.004	.6726	4.8445	
	400.00	2.60002*	.61056	.006	.5444	4.6556	
50	75.00	-.47631-	.82202	1.000	-3.1827-	2.2301	
	100.00	.26165	.64271	1.000	-1.8655-	2.3888	
	200.00	1.28420	.66228	.593	-.9016-	3.4699	
	300.00	1.30890	.59896	.440	-.6946-	3.3124	
	400.00	1.15036	.58678	.581	-.8210-	3.1217	
75	100.00	.73797	.72933	.982	-1.6930-	3.1689	
	200.00	1.76051	.74663	.340	-.7191-	4.2401	
	300.00	1.78522	.69109	.240	-.5445-	4.1150	
	400.00	1.62667	.68056	.331	-.6770-	3.9304	
100	200.00	1.02255	.54298	.629	-.7627-	2.8078	
	300.00	1.04725	.46365	.392	-.4822-	2.5767	
	400.00	.88871	.44780	.564	-.5930-	2.3704	
200	300.00	.02470	.49041	1.000	-1.5973-	1.6467	
	400.00	-.13384-	.47545	1.000	-1.7122-	1.4445	
300	400.00	-.15854-	.38236	1.000	-1.4159-	1.0988	
*. The mean difference is significant at the 0.05 level.							

Table 4.15 Statistical results for PS by Games-Howell post hoc test (case 1 phase duration 10-60)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
5.00	15.00	8.03516	3.15032	.243	-2.3545-	18.4248	-
	30.00	12.51179*	2.91090	.004	2.8282	22.1953	30
	50.00	12.02014*	2.91182	.007	2.3340	21.7063	50
	75.00	13.67899*	2.67633	.001	4.6121	22.7459	75
	100.00	17.31338*	2.63458	.000	8.3451	26.2816	100
	200.00	15.27293*	2.67710	.000	6.2042	24.3417	200
	300.00	17.32271*	2.54358	.000	8.5542	26.0912	300
	400.00	17.37594*	2.59579	.000	8.4956	26.2563	400
15	30.00	4.47662	2.45751	.668	-3.6270-	12.5803	-
	50.00	3.98498	2.45860	.787	-4.1220-	12.0920	-
	75.00	5.64382	2.17457	.232	-1.6450-	12.9327	-
	100.00	9.27822*	2.12297	.005	2.1224	16.4341	100
	200.00	7.23777*	2.17553	.048	.0536-	14.5291	200
	300.00	9.28755*	2.00894	.003	2.4002	16.1749	300
	400.00	9.34078*	2.07464	.004	2.3034	16.3781	400
30	50.00	-.49164-	2.14326	1.000	-7.5355-	6.5522	-
	75.00	1.16720	1.81041	.999	-4.8379-	7.1723	-
	100.00	4.80160	1.74809	.174	-1.0275-	10.6307	-
	200.00	2.76114	1.81156	.836	-3.2473-	8.7696	-

	300.00	4.81093	1.60769	.116	-.6591-	10.2810	-
	400.00	4.86416	1.68907	.137	-.8068-	10.5351	-
50	75.00	1.65884	1.81189	.990	-4.3515-	7.6692	-
	100.00	5.29324	1.74963	.099	-.5413-	11.1278	-
	200.00	3.25278	1.81304	.685	-2.7608-	9.2664	-
	300.00	5.30257	1.60936	.063	-.1734-	10.7785	-
	400.00	5.35580	1.69066	.076	-.3208-	11.0324	-
75	100.00	3.63440	1.32107	.165	-.7114-	7.9802	-
	200.00	1.59394	1.40397	.965	-3.0203-	6.2081	-
	300.00	3.64372	1.12874	.065	-.1298-	7.4172	-
	400.00	3.69696	1.24192	.105	-.4025-	7.7964	-
100	200.00	-2.04045-	1.32264	.828	-6.3916-	2.3106	-
	300.00	.00933	1.02583	1.000	-3.4013-	3.4200	-
	400.00	.06256	1.14918	1.000	-3.7194-	3.8445	-
200	300.00	2.04978	1.13058	.674	-1.7302-	5.8298	-
	400.00	2.10301	1.24359	.748	-2.0023-	6.2083	-
300	400.00	.05323	.92165	1.000	-2.9933-	3.0998	-
*. The mean difference is significant at the 0.05 level.							

Table 4.22 Statistical results for TLBO by Games-Howell post hoc test (case 1 phase duration 10-100)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
5.00	15	5.86348*	.86134	.000	3.1572	8.5698	15
	30	6.35911*	.86134	.000	3.6528	9.0654	30
	50	6.29076*	.86134	.000	3.5845	8.9971	50
	75	5.37332*	.86134	.000	2.6670	8.0796	75
	100	5.28847*	.86134	.000	2.5822	7.9948	100
	200	3.80201*	.86134	.001	1.0957	6.5083	200
	300	3.21241*	.86134	.008	.5061	5.9187	300
	400	.27967	.86134	1.000	-2.4266-	2.9860	-
15	30	.49563	.47639	.978	-1.1042-	2.0955	-
	50	.42728	.51080	.995	-1.2672-	2.1218	-
	75	-.49015-	.47362	.979	-2.0829-	1.1026	-
	100	-.57500-	.49742	.960	-2.2315-	1.0815	-
	200	-2.06147-*	.49943	.007	-3.7236-	-.3993-	15
	300	-2.65107-*	.50645	.000	-4.3331-	-.9690-	15
	400	-5.58381-*	.67711	.000	-7.8144-	-3.3532-	15
30	50	-.06835-	.34926	1.000	-1.2214-	1.0847	-
	75	-.98579-*	.29221	.040	-1.9462-	-.0254-	30
	100	-1.07064-	.32938	.055	-2.1554-	.0141	30
	200	-2.55710-*	.33242	.000	-3.6522-	-1.4620-	30
	300	-3.14670-*	.34287	.000	-4.2777-	-2.0157-	30
	400	-6.07944-*	.56528	.000	-7.9944-	-4.1645-	30
50	75	-.91743-	.34548	.201	-2.0590-	.2242	-
	100	-1.00228-	.37744	.199	-2.2434-	.2388	-

	200	-2.48875-*	.38009	.000	-3.7384-	-1.2391-	50
	300	-3.07834-*	.38927	.000	-4.3578-	-1.7989-	50
	400	-6.01109-*	.59457	.000	-8.0007-	-4.0215-	50
75	100	-.08485-	.32537	1.000	-1.1571-	.9874	-
	200	-1.57131-*	.32844	.001	-2.6541-	-.4885-	75
	300	-2.16091-*	.33902	.000	-3.2802-	-1.0417-	75
	400	-5.09365-*	.56295	.000	-7.0030-	-3.1843-	75
100	200	-1.48647-*	.36191	.006	-2.6759-	-.2970-	100
	300	-2.07606-*	.37154	.000	-3.2974-	-.8547-	100
	400	-5.00880-*	.58311	.000	-6.9683-	-3.0493-	100
200	300	-.58960-	.37423	.812	-1.8197-	.6405	-
	400	-3.52234-*	.58483	.000	-5.4863-	-1.5584-	200
300	400	-2.93274-*	.59084	.001	-4.9124-	-.9531-	300
*. The mean difference is significant at the 0.05 level.							

Table 4.25 Statistical results for WTLBO by Games-Howell post hoc test (case 1 phase duration 10-100)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
5.00	15	9.56231	3.22321	.113	-1.2026-	20.3272	-
	30	13.22874*	2.85497	.004	3.3474	23.1100	30
	50	13.85794*	2.86117	.002	3.9643	23.7516	50
	75	13.17805*	2.84172	.004	3.3229	23.0332	75
	100	13.55524*	2.84518	.003	3.6933	23.4172	100
	200	12.75057*	2.84228	.006	2.8943	22.6068	200
	300	12.97914*	2.86858	.005	3.0705	22.8878	300
	400	11.30838*	2.91335	.019	1.3063	21.3105	400
15	30	3.66642	1.63294	.413	-1.9228-	9.2557	-
	50	4.29562	1.64376	.235	-1.3177-	9.9090	-
	75	3.61574	1.60966	.414	-1.9234-	9.1548	-
	100	3.99292	1.61577	.298	-1.5591-	9.5449	-
	200	3.18826	1.61065	.572	-2.3529-	8.7294	-
	300	3.41682	1.65663	.519	-2.2258-	9.0595	-
	400	1.74606	1.73298	.982	-4.0827-	7.5748	-
30	50	.62920	.68097	.990	-1.6096-	2.8680	-
	75	-.05069-	.59396	1.000	-2.0079-	1.9065	-
	100	.32650	.61032	1.000	-1.6820-	2.3350	-
	200	-.47817-	.59665	.996	-2.4438-	1.4874	-
	300	-.24960-	.71148	1.000	-2.5911-	2.0919	-
	400	-1.92036-	.87466	.432	-4.8273-	.9866	-
50	75	-.67989-	.62307	.972	-2.7373-	1.3775	-
	100	-.30270-	.63869	1.000	-2.4078-	1.8024	-
	200	-1.10737-	.62564	.700	-3.1726-	.9578	-
	300	-.87880-	.73596	.953	-3.2984-	1.5408	-
	400	-2.54956-	.89468	.139	-5.5141-	.4150	-
75	100	.37719	.54496	.999	-1.4143-	2.1687	-
	200	-.42748-	.52961	.996	-2.1681-	1.3131	-
	300	-.19891-	.65628	1.000	-2.3716-	1.9738	-
	400	-1.86967-	.83037	.402	-4.6556-	.9162	-

100	200	-.80467-	.54790	.863	-2.6057-	.9963	-
	300	-.57610-	.67113	.994	-2.7930-	1.6408	-
	400	-2.24686-	.84215	.203	-5.0640-	.5703	-
200	300	.22857	.65872	1.000	-1.9514-	2.4085	-
	400	-1.44219-	.83230	.722	-4.2332-	1.3488	-
300	400	-1.67076-	.91812	.669	-4.7046-	1.3631	-
*. The mean difference is significant at the 0.05 level.							

Table 4.28 Statistical results for Jaya by Games-Howell post hoc test (case 1 phase duration 10-100)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
5.00	15	17.17662*	2.66832	.000	8.1111	26.2422	15
	30	17.94240*	2.53635	.000	9.1648	26.7200	30
	50	16.97731*	2.62975	.000	8.0007	25.9539	50
	75	17.90033*	2.51011	.000	9.1739	26.6267	75
	100	16.26239*	2.59847	.000	7.3552	25.1696	100
	200	15.62854*	2.61073	.000	6.6944	24.5626	200
	300	13.66480*	2.63272	.001	4.6815	22.6481	300
	400	13.70043*	2.56681	.001	4.8607	22.5402	400
15	30	.76578	1.01546	.997	-2.6581-	4.1896	-
	50	-.19930-	1.23032	1.000	-4.2470-	3.8484	-
	75	.72371	.94801	.997	-2.5449-	3.9923	-
	100	-.91422-	1.16195	.997	-4.7494-	2.9210	-
	200	-1.54808-	1.18911	.924	-5.4666-	2.3704	-
	300	-3.51182-	1.23664	.138	-7.5796-	.5560	-
	400	-3.47619-	1.08931	.068	-7.0979-	.1456	-
30	50	-.96508-	.90930	.975	-4.0128-	2.0827	-
	75	-.04207-	.46028	1.000	-1.5893-	1.5051	-
	100	-1.68001-	.81441	.515	-4.3926-	1.0325	-
	200	-2.31386-	.85271	.186	-5.1615-	.5338	-
	300	-4.27760-*	.91783	.002	-7.3555-	-1.1997-	30
	400	-4.24197-*	.70692	.000	-6.5783-	-1.9056-	30
50	75	.92301	.83330	.967	-1.9423-	3.7884	-
	100	-.71492-	1.07042	.999	-4.2370-	2.8072	-
	200	-1.34877-	1.09984	.945	-4.9648-	2.2673	-
	300	-3.31252-	1.15106	.127	-7.0955-	.4705	-
	400	-3.27689-	.99110	.050	-6.5551-	.0013	-
75	100	-1.63793-	.72858	.411	-4.1344-	.8585	-
	200	-2.27179-	.77115	.132	-4.9183-	.3748	-
	300	-4.23553-*	.84260	.001	-7.1336-	-1.3375-	75
	400	-4.19990-*	.60606	.000	-6.2636-	-2.1362-	75
100	200	-.63385-	1.02278	.999	-3.9960-	2.7283	-
	300	-2.59760-	1.07767	.308	-6.1443-	.9491	-
	400	-2.56197-	.90483	.141	-5.5426-	.4186	-
200	300	-1.96374-	1.10690	.698	-5.6035-	1.6760	-
	400	-1.92811-	.93945	.520	-5.0275-	1.1713	-
300	400	.03563	.99893	1.000	-3.2698-	3.3411	-
*. The mean difference is significant at the 0.05 level.							

Table 4.31 Statistical results for GA by Games-Howell post hoc test (case 1 phase duration 10-100)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound	
5.00	15	1.97053	1.90125	.980	-4.2788-	8.2199	-
	30	4.64293	1.67119	.160	-.8897-	10.1755	-
	50	3.84662	1.60973	.324	-1.5090-	9.2023	-
	75	7.49695*	1.48283	.001	2.4750	12.5189	75
	100	6.65872*	1.45615	.003	1.6998	11.6176	100
	200	9.50845*	1.45513	.000	4.5519	14.4650	200
	300	9.22817*	1.44629	.000	4.2918	14.1646	300
	400	8.63558*	1.43240	.000	3.7302	13.5410	400
15	30	2.67240	1.61941	.771	-2.6801-	8.0249	-
	50	1.87609	1.55590	.949	-3.2905-	7.0427	-
	75	5.52642*	1.42422	.016	.7122	10.3406	75
	100	4.68819	1.39642	.055	-.0593-	9.4357	100
	200	7.53792*	1.39535	.000	2.7929	12.2829	200
	300	7.25764*	1.38614	.001	2.5340	11.9813	300
	400	6.66505*	1.37163	.002	1.9741	11.3560	400
30	50	-.79631-	1.26450	.999	-4.9559-	3.3633	-
	75	2.85402	1.09842	.228	-.8068-	6.5149	-
	100	2.01579	1.06212	.621	-1.5472-	5.5788	-
	200	4.86552*	1.06072	.002	1.3062	8.4249	200
	300	4.58524*	1.04857	.004	1.0573	8.1132	300
	400	3.99265*	1.02932	.016	.5129	7.4724	400
50	75	3.65033*	1.00243	.023	.3267	6.9740	75
	100	2.81210	.96252	.123	-.3988-	6.0230	-
	200	5.66183*	.96097	.000	2.4552	8.8684	200
	300	5.38155*	.94754	.000	2.2113	8.5518	300
	400	4.78896*	.92620	.001	1.6747	7.9032	400
75	100	-.83823-	.73076	.962	-3.2429-	1.5664	-
	200	2.01150	.72872	.162	-.3867-	4.4097	-
	300	1.73122	.71091	.296	-.6113-	4.0737	-
	400	1.13863	.68221	.760	-1.1164-	3.3937	-
100	200	2.84973*	.67276	.004	.6387	5.0608	200
	300	2.56945*	.65344	.009	.4214	4.7175	300
	400	1.97686	.62208	.066	-.0714-	4.0251	-
200	300	-.28028-	.65115	1.000	-2.4208-	1.8602	-
	400	-.87287-	.61969	.887	-2.9129-	1.1672	-
300	400	-.59259-	.59865	.985	-2.5615-	1.3763	-
*. The mean difference is significant at the 0.05 level.							

Table 4.34 Statistical results for PS by Games-Howell post hoc test (case 1 phase duration 10-100)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound	
5.00	15	9.30925	4.34287	.461	-4.9649-	23.5833	-
	30	14.27761*	3.61465	.010	2.2929	26.2623	30
	50	15.33051*	3.48237	.003	3.7204	26.9406	50

	75	19.01041*	3.30296	.000	7.8742	30.1466	75
	100	17.43926*	3.52222	.001	5.7183	29.1602	100
	200	20.81425*	3.19833	.000	9.9312	31.6973	200
	300	20.47677*	3.27276	.000	9.4157	31.5379	300
	400	20.46929*	3.13991	.000	9.7183	31.2203	400
15	30	4.96836	3.70904	.911	-7.3470-	17.2837	-
	50	6.02126	3.58026	.752	-5.9343-	17.9768	-
	75	9.70117	3.40601	.149	-1.8003-	21.2026	-
	100	8.13001	3.61903	.403	-3.9319-	20.1920	-
	200	11.50500*	3.30464	.043	.2458	22.7642	200
	300	11.16752*	3.37673	.059	.2621-	22.5971	300
	400	11.16004*	3.24814	.049	.0273	22.2928	400
30	50	1.05290	2.65016	1.000	-7.6659-	9.7717	-
	75	4.73280	2.40956	.577	-3.2496-	12.7152	-
	100	3.16165	2.70231	.958	-5.7238-	12.0471	-
	200	6.53664	2.26401	.134	-1.0367-	14.1100	-
	300	6.19916	2.36799	.218	-1.6629-	14.0612	-
	400	6.19168	2.18071	.151	-1.1658-	13.5491	-
50	75	3.67990	2.20619	.761	-3.5970-	10.9568	-
	100	2.10875	2.52265	.995	-6.1829-	10.4004	-
	200	5.48374	2.04621	.196	-1.3214-	12.2889	-
	300	5.14626	2.16070	.325	-1.9928-	12.2853	-
	400	5.13878	1.95365	.219	-1.4141-	11.6917	-
75	100	-1.57115-	2.26857	.999	-9.0634-	5.9210	-
	200	1.80383	1.72326	.978	-3.8769-	7.4846	-
	300	1.46635	1.85776	.997	-4.6404-	7.5731	-
	400	1.45887	1.61226	.991	-3.8901-	6.8078	-
100	200	3.37499	2.11332	.799	-3.6664-	10.4164	-
	300	3.03751	2.22436	.903	-4.3226-	10.3976	-
	400	3.03003	2.02383	.847	-3.7714-	9.8315	-
200	300	-.33748-	1.66463	1.000	-5.8179-	5.1429	-
	400	-.34496-	1.38532	1.000	-4.9076-	4.2177	-
300	400	-.00748-	1.54944	1.000	-5.1369-	5.1219	-
*. The mean difference is significant at the 0.05 level.							

Table 4.41 Statistical results for TLBO by Games-Howell post hoc test (case 2)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	20.00350*	4.66090	.004	5.3621	34.6449	15
	30.00	20.84945*	4.63879	.003	6.2507	35.4482	30
	50.00	21.50355*	4.61019	.002	6.9589	36.0482	50
	75.00	20.80530*	4.61891	.003	6.2443	35.3663	75
	100.00	20.19535*	4.61942	.004	5.6333	34.7574	100
15	30.00	.84595	1.02236	.960	-2.2272-	3.9191	-
	50.00	1.50005	.88358	.545	-1.2008-	4.2009	-
	75.00	.80180	.92799	.952	-2.0118-	3.6154	-
	100.00	.19185	.93055	1.000	-2.6284-	3.0121	-
30	50.00	.65410	.75832	.953	-1.6458-	2.9540	-
	75.00	-.04415-	.80964	1.000	-2.4829-	2.3946	-

	100.00	-.65410-	.81257	.965	-3.1011-	1.7929	-
50	75.00	-.69825-	.62530	.871	-2.5782-	1.1817	-
	100.00	-1.30820-	.62909	.320	-3.2000-	.5836	-
75	100.00	-.60995-	.69008	.948	-2.6802-	1.4603	-
*. The mean difference is significant at the 0.05 level.							

Table 4.44 Statistical results for WTLBO by Games-Howell post hoc test (case 2)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	11.37495	6.35528	.487	-7.9573-	30.7072	-
	30.00	15.42010	6.12604	.154	-3.3440-	34.1842	-
	50.00	18.15075*	5.74054	.046	.2552	36.0463	50
	75.00	21.16865*	5.75709	.015	3.2385	39.0988	75
	100.00	17.76450	6.20402	.076	-1.1892-	36.7182	-
15	30.00	4.04515	4.07329	.917	-8.1933-	16.2836	-
	50.00	6.77580	3.46657	.393	-3.8317-	17.3833	-
	75.00	9.79370	3.49391	.086	-.8806-	20.4680	-
	100.00	6.38955	4.18965	.651	-6.1868-	18.9659	-
30	50.00	2.73065	3.02586	.943	-6.4647-	11.9260	-
	75.00	5.74855	3.05715	.432	-3.5277-	15.0248	-
	100.00	2.34440	3.83300	.990	-9.1570-	13.8458	-
50	75.00	3.01790	2.18440	.737	-3.5357-	9.5715	-
	100.00	-.38625-	3.18077	1.000	-10.0773-	9.3048	-
75	100.00	-3.40415-	3.21055	.893	-13.1705-	6.3622	-
*. The mean difference is significant at the 0.05 level.							

Table 4.47 Statistical results for Jaya by Games-Howell post hoc test (case 2)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	49.04545*	9.88407	.001	17.9317	80.1592	15
	30.00	53.11255*	9.81020	.000	22.1378	84.0873	30
	50.00	53.17725*	9.80895	.000	22.2048	84.1497	50
	75.00	54.36295*	9.80254	.000	23.4024	85.3235	75
	100.00	54.33620*	9.80369	.000	23.3735	85.2989	100
15	30.00	4.06710	1.45634	.091	-.4095-	8.5437	-
	50.00	4.13180	1.44786	.081	-.3251-	8.5887	-
	75.00	5.31750*	1.40379	.011	.9590	9.6760	75
	100.00	5.29075*	1.41178	.012	.9149	9.6666	100
30	50.00	.06470	.80096	1.000	-2.3383-	2.4677	-
	75.00	1.25040	.71823	.515	-.9137-	3.4145	-

	100.00	1.22365	.73372	.561	-.9839-	3.4312	-
50	75.00	1.18570	.70087	.546	-.9240-	3.2954	-
	100.00	1.15895	.71675	.593	-.9957-	3.3136	-
75	100.00	-.02675-	.62293	1.000	-1.8958-	1.8423	-
*. The mean difference is significant at the 0.05 level.							

Table 4.50 Statistical results for GA by Games-Howell post hoc test (case 2)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	1.62190	3.66342	.998	-9.4164-	12.6602	-
	30.00	-3.47415-	4.61723	.974	-17.3507-	10.4024	-
	50.00	8.33465	3.46978	.186	-2.1789-	18.8482	-
	75.00	8.82885	3.18139	.095	-.9689-	18.6266	-
	100.00	9.74240	3.33188	.066	-.4170-	19.9018	-
15	30.00	-5.09605-	4.17266	.823	-17.7477-	7.5556	-
	50.00	6.71275	2.85161	.199	-1.8547-	15.2802	-
	75.00	7.20695	2.49270	.070	-.3808-	14.7947	-
	100.00	8.12050*	2.68211	.049	.0310	16.2100	100
30	50.00	11.80880	4.00373	.063	-.4121-	24.0297	-
	75.00	12.30300*	3.75656	.034	.6600	23.9460	75
	100.00	13.21655*	3.88483	.024	1.2826	25.1505	100
50	75.00	.49420	2.19824	1.000	-6.1543-	7.1427	-
	100.00	1.40775	2.41090	.992	-5.8346-	8.6501	-
75	100.00	.91355	1.97340	.997	-5.0260-	6.8531	-
*. The mean difference is significant at the 0.05 level.							

Table 4.53 Statistical results for PS by Games-Howell post hoc test (case 2)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
5.00	15.00	53.42675*	16.95318	.043	1.1010	105.7525	15
	30.00	65.93980*	16.14876	.006	15.3264	116.5532	30
	50.00	69.56110*	16.35941	.004	18.5213	120.6009	50
	75.00	70.79220*	15.95118	.003	20.5627	121.0217	75
	100.00	76.92260*	15.90206	.001	26.7860	127.0592	100
15	30.00	12.51305	6.91044	.475	-8.5539-	33.5800	-
	50.00	16.13435	7.38931	.272	-6.1840-	38.4527	-
	75.00	17.36545	6.43520	.114	-2.5840-	37.3149	-
	100.00	23.49585*	6.31245	.013	3.8073	43.1844	100
30	50.00	3.62130	5.28908	.983	-12.2949-	19.5375	-

	75.00	4.85240	3.84503	.803	-6.7927-	16.4975	-
	100.00	10.98280	3.63585	.054	-.1283-	22.0939	-
50	75.00	1.23110	4.65102	1.000	-12.9926-	15.4548	-
	100.00	7.36150	4.47964	.579	-6.4586-	21.1816	-
75	100.00	6.13040	2.62241	.206	-1.7578-	14.0186	-
*. The mean difference is significant at the 0.05 level.							

Table 4.60 Statistical results for TLBO by Tukey HSD post hoc test (case 3)

Multiple Comparisons							
Dependent Variable: ATT							
Tukey HSD							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
50.00	500	-2.65830*	.50410	.000	-3.8714	-1.4452-	50
	1000	-4.78810*	.50410	.000	-6.0012	-3.5750-	50
500	1000	-2.12980*	.50410	.000	-3.3429	-.9167-	500
*. The mean difference is significant at the 0.05 level.							

Table 4.63 Statistical results for Jaya by Tukey HSD post hoc test (case 3)

Multiple Comparisons							
Dependent Variable: ATT							
Tukey HSD							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
50.00	500	-6.24960-*	1.58772	.001	-10.0703-	-2.4289-	50
	1000	-11.83735-*	1.58772	.000	-15.6581-	-8.0166-	50
500	1000	-5.58775-*	1.58772	.002	-9.4085-	-1.7670-	500
*. The mean difference is significant at the 0.05 level.							

Table 4.65 Statistical results for GA by Games-Howell post hoc test (case 3)

Multiple Comparisons							
Dependent Variable: ATT							
Games-Howell							
(I) Psize	(J) Psize	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound	
50.00	500	-3.36022	1.80436	.173	-7.8817-	1.1612	-
	1000	-19.89406*	1.83122	.000	-24.4671-	-15.3210-	50
500	1000	-16.53384*	.83354	.000	-18.5683-	-14.4994-	500
*. The mean difference is significant at the 0.05 level.							

Appendix B: Algorithms

Algorithm 1: TLBO

Begin

Initialize *N* (number of learners) and *D* (number of dimensions);

Initialize learners and evaluate them;

while (stopping condition is not met);

{Teacher Phase}

Choose the best learner as $X_{Teacher}$ and calculate the mean X_{Mean} of all learners;

for all learners

TF = round (1 + rand (0,1));

Update all learners according to Eq:

$$X^{new} = X^{old} + rand(0,1)\,[X_{teacher} - TF * X_{mean}]$$

end for

Evaluated the new learners;

Accept the new solutions if it is better than the old one;

{Learner Phase}

for all learners

Randomly select another learner Xj that is different from it Xi;

IF X_i is better than X_j , e.g if ($X_i < X_j$)

$X_i^{new} = X_i^{old} + rand(0,1)\,(X_i - X_j)$

Else

$X_i^{new} = X_i^{old} + rand(0,1)\,(X_j - X_i)$

end for

Accept the new solution if it is better than the old one;

Update the teacher and the mean;

end while

end

Algorithm 2: Jaya algorithm

S1 Initialize

$PS \leftarrow Population\_size$

$NDV \leftarrow Number\_of\_Design\_Variables$

$TER\_COD \leftarrow Termination\_Condition$

S2 Until the termination condition not satisfied, Repeat S3 to S5

S3 Evaluate the best and worst solution

Set $best \leftarrow Best\_Solution\_Population$

Set $worst \leftarrow Worst\_Solution\_Population$

S4 Modify the solution

$X_{j,k}^{new} = X_{j,k} + r1_j \left(X_{j,best} - \left|X_{j,k}\right|\right) - r2_j \left(X_{j,worst} - \left|X_{j,k}\right|\right)$

S5 if (solution corresponding to $X_{j,k,i}^{new}$ better than that correspnding to $X_{j,k,i}$)

Update the previous solution

Else

No update in the previous solution

S6 Display the optimum result

Where:

- $X_{j,k,}$: the value of the j^{th} variable for the k^{th} candidate during the i^{th} iteration
- $X_{j,\ best}$: the value of the j^{th} for the best candidate solution.
- $X_{j,worst:}$: the value of the j^{th} for the worst candidate solution.
- r: random number in the range [0,1].

Algorithm 3: GA algorithm

1: $P \leftarrow generate\ initial\ population$
2: evaluate P
3: **while** termination criteria not met **do**
4: $P_{selected} \leftarrow select\ solution\ from\ P$
5: $P_{offspring} \leftarrow recombine\ individuals\ from\ P_{selected}$
6: Mutate some $P_{offspring}$
7: Evaluate $P_{offspring}$
8: $P \leftarrow P_{offspring}$
9: **end while**
10: return P_0 (best solution)

Algorithm 2.4: PSO algorithm

1. $initia(\theta)$ *// initial swarm usually random*

2. for each particle $x \in \theta$:

for each dimension i

// calculate velocity

$$xv_i = w * xv_i + cp * r * (pbest_i - x_i) + cg * r\ (gbest_i - x_i)$$

// update particle position according to equation)

$$x_i = x_i + xv_i$$

3. While stop criteria not reached, Go to step 02

where:

xi: the i^{th} dimension of particle position

xv_i: the i^{th} dimension of the velocity component

r: a uniformly distributed random real number within [0, 1].

$pbest_i$: particle best value found so far of dimension i

$gbest_i$: global best value found so far of dimension i

w, cp, cg: tunable parameters. w (inertia weight), cp (weight of local information), cg (weight of global information)

جامعة النجاح الوطنية

كلية الدراسات العليا

تحسين توقيت إشارات المرور باستخدام خوارزميات التحسين التخمينية الخالية من المعلمات

إعداد

ثائر أحمد درويش ظاهر

إشراف

د. بكر عبد الحق

قدمت هذه الأطروحة استكمالاً لمتطلبات الحصول على درجة الماجستير في الحوسبة المتقدمة بكلية الدراسات العليا في جامعة النجاح الوطنية في نابلس، فلسطين

2018

تحسين توقيت إشارات المرور باستخدام خوارزميات التحسين التخمينية الخالية من المعلمات

إعداد
ثائر أحمد ظاهر
إشراف
د. بكر عبد الحق

الملخص

يمثل الازدحام المروري تحديًا شائعًا في المدن الحضرية، لذلك يتم استخدام العديد من الطرق لحلها. الحل القوي الذي يمكن أن يحل مشكلة الازدحام هو عن طريق تطوير نظام التحكم في حركة المرور في الوقت الحقيقي الفوري باستخدام تقنية التحسين الأمثل لتقليل التأخير الكلي للمركبات أو زيادة الإنتاجية. اقترح الباحثون العديد من نماذج المحاكاة واستخدموا تقنيات متنوعة لتحسين توقيت إشارات المرور.

الغرض من هذا البحث هو تقييم ومقارنة تقنيات التحسين التخمينية لتحسين مدة المراحل الضوئية (الخضراء أو الحمراء) من أجل تقليل وقت السفر للمركبات. في هذا العمل، تم تطبيق خوارزميات جديدة لا تحتوي على معلمات محددة (عديمة المعلمات) تسمى خوارزمية التحسين القائمة على التعلم والتعليم وخوارزمية Jaya لحل مشكلة تحسين إشارات المرور. لم يتم تطبيق هذه الخوارزميات على المشكلة المدروسة حتى الآن. تم مقارنة أداء خوارزمية TLBO و Jaya بثلاث خوارزميات تحتوي على بعض المعلمات التي يجب تعيينها مثل الخوارزمية الجينية GA وخوارزمية سرب الجسيمات PSO وخوارزمية WTLBO. تم أيضًا دراسة تأثير معلمات التحكم الشائعة (على سبيل المثال، حجم السكان) على فعالية كل خوارزمية.

تم استخدام نظام محاكاة عشوائي يدعى SUMO (محاكاة الحركة الحضرية) كأداة لتنفيذ وتقييم سرعة الاداء والتقارب لكل خوارزمية. حيث تم محاكاة ثلاثة شبكات طرق بأحجام مختلفة:

صغيرة ومتوسطة وكبيرة تحتوي على 13، 34، 141 مرحلة ضوئية على التوالي لدراسة قابلية التوسع للخوارزميات المختبرة.

بعد إجراء العديد من التجارب، أظهرت المقارنات والمناقشات أن خوارزمية TLBO و Jaya لهما أداء أفضل الخوارزميات الأخرى للشبكات الصغيرة والمتوسطة. علاوة على ذلك، حققت خوارزمية TLBO أفضل أداء وقابلية للتوسع في الشبكة الكبيرة. لذا، فإن خوارزمية TLBO هي تقنية فعالة وقوية وقابلة للتوسع لإيجاد الحل الأمثل لتوقيت إشارات المرور