HYBRID META-HEURISTIC ALGORITHMS FOR SOLVING VEHICLE ROUTING PROBLEMS

SARAH HAZWANI BINTI ADNAN

UNIVERSITI SAINS MALAYSIA

2021

HYBRID META-HEURISTIC ALGORITHMS FOR SOLVING VEHICLE ROUTING PROBLEMS

by

SARAH HAZWANI BINTI ADNAN

Thesis submitted in fulfilment of the requirements for the degree of Master of Science

November 2021

ACKNOWLEDGEMENTS

بسم الله الرحمن الرحيم

In the Name of Allah, the Most Beneficent, the Most Merciful

Praise to The Almighty Allah for giving me time, strength of knowledge and wisdom, throughout this study.

First and foremost, I would like to acknowledge MOHE through the Fundamental Research Grant Scheme (FRGS): 203 for providing financial support for this study.

I am truly blessed to have my parents, Adnan Mustafa and Suhaida Abdul Hamid, as my pillar of strength and for patiently waiting for the completion and successful output from this study. Thank you for always believe in me and encourage me to stay strong and to dream big. To my brothers and sister, Safuan Hadi, Safiah Hanis and Safiuddin Hilmee, thank you for the encouragement, time and support given.

I am deeply indebted to my supervisors, Dr Ng Theam Foo, Dr Wang Shir Li and Dr Haidi Ibrahim for all of the professional guidance given, endless knowledge sharing, constructive criticism, as well as for the advices on the changes, ideas and amendments made in this study. My heartiest gratitude to my hardworking supervisors for every continuous advice and support given which enable me to remain on the right path during the long and arduous course of this memorable and challenging journey.

These acknowledgements would not be completed without mentioning my supportive study mates, Siti Fairuz Mohd Radzi, Dr Azzah Said, Fatin Nabilla Ariffin, Dr Norhidayati Nadiha, Dr Siti Nor Ain Mayan, and Nor Syazana Zamree. Thank you for all of the support and patience everyone endures for me throughout the years. I would like to forward my appreciation to the Centre for Global Sustainability Studies (CGSS) directors and staffs, for help and supports given. May all of the individuals mentioned above be blessed with constant love, health and forever bless.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS

φ	Random variable, independently generated within uniform distribution between [0,1]
c_1	Cognitive learning factors (acceleration factor)
c_2	Social learning factor (acceleration factor)
g_{best}	Global best particle
g_{max}	Maximum generation
G	Generation
N	Swarm / Population Size
Np	Number of populations
p_{best}	Local best particle
T_{max}	Maximum iteration
v_{max}	Maximum velocity
v_{min}	Minimum velocity
w	Weight
w_{start}	Initial weight
w_{end}	Final weight

LIST OF ABBREVIATIONS

ADE	Adaptive Differential Evolution
BBDE	Barebones Differential Evolution
BBOB	Black Box Optimization Problem
CEA	Co-Evolutionary Algorithm
CEC	Congress on Evolutionary Computation
CLPSO	Comprehensive Learning Particle Swarm Optimizer
CVRP	Capacitated Vehicle Routing Problem
CVRPTW	Capacitated Vehicle Routing Problem with Time Windows
DCPSO	Discrete Continuous PSO
DE	Differential Evolution
DEPSO	Hybrid of Differential Evolution and Particle Swarm Optimization
GA	Genetic Algorithm
GECCO	Genetic and Evolutionary Computation
EA	Evolutionary Algorithm
EAs	Evolutionary Algorithms
EC	Evolutionary Computation
EPSDE	Ensemble of Mutation Strategies and Parameters in DE
FADE	Fuzzy adaptive Differential Evolution
FCPSO	Fully Connected Particle Swarm Optimization
FDPPTW	Flexible Delivery and Pickup Problem with Time Window
FIPS	Fully Informed Particle Swarm
FIPSaDE	Fully Informed Particle Swarm with Self-adaptive Differential Evolution
FIPSaDE-LS	FIPSaDE with Local Search
FIPSaDE- NOX	FIPSaDE with Novel Ordered Crossover
HPSO-DE	Hybridizing Particle Swarm Optimization with Differential Evolution
IADE	Improved Adaptive Differential Evolution
IPSO	Isolated Particle Swarm Optimization
JADE	Adaptive Differential Evolution with Optional External Archive
jDE	Adaptive Differential Evolution
LMDP	Last Mile Distribution Problem
LS	Local Search

MERCY	Medical Relief Society Malaysia
MMOP	Multimodal Multiobjective Problem
neo-DRM-SD	Disaster Risk Model – Sustainable Development
NOX	Novel Ordered Crossover
NP-hard	Non-deterministic polynomial-time hard
OED	Orthogonal Experimental Design
OLPSO	Orthogonal Learning Particle Swarm Optimization
PF	Pareto Front
PSs	Pareto Optimal Sets
PSO	Particle Swarm Optimization
PSOJADE	Particle Swarm Optimization with Adaptive Differential Evolution with Optional External Archive
SaDE	Self- adaptive Differential Evolution
SaDEPS	Self-adaptive Differential Evolution hybrid Particle Swarm
SAPSO	Self-adaptive Particle Swarm Optimization
SI	Swarm intelligence
SIPSO	Selectively-informed Particle Swarm Optimization
SLPSO	Self-adaptive Learning based Particle Swarm Optimization
TDRM	Total Disaster Risk Management
TDVRP	Time Dependent Vehicle Routing Problem
UNISDR	United Nations International Strategy for Disaster Reduction
VND	Variable Neighborhood Descent
VRP	Vehicle Routing Problem
VRPSPD	Vehicle Routing Problem with Simultaneous Pickup and Delivery Problem
VRPTW	Vehicle Routing Problem with Time Windows
VRP-SPDTW	Vehicle Routing Problem with Simultaneous Pickup and Deliveries with Time Windows

LIST OF APPENDICES

ALGORITMA META-HEURISTIK HIBRID DALAM PENYELESAIAN MASALAH PENGHALAAN KENDERAAN

ABSTRAK

Dalam usaha untuk menangani kerugian atau kehilangan yang diakibatkan daripada bencana yang dialami oleh masyarakat setempat, penjadualan dan penghalaan logistik yang cekap dan efisien amatlah diperlukan. Seandainya semua parameter yang diperlukan telah diketahui terlebih dahulu, maka proses penjadualan dan penghalaan logistik tidak akan dibebani masalah pengurusan masa dan masalah pengurusan tekanan yang berkaitan walaupun menghadapi kesan negatif yang boleh berpunca daripada kecelakaan yang berlaku yang seterusnya membantu dalam melestarikan sumber yang tersedia. Sehubungan itu, sistem penjadualan dan penghalaan logistik yang strategik amatlah dituntut untuk mengatasi pembatasan masalah ini. Lantaran, kajian ini mencadangkan pendekatan secara heuristik yang menggabungkan salah satu variasi dari algorithm Pengoptimuman Kerumunan Zarah (*Particle Swarm Optimization (*PSO*)*) iaitu *Fully Informed Particle Swarm Optimization (*FIPS*)* dan *Self-adaptive Differential Evolution (*SaDE*)* Pembangunan system penjadualan logistik yang sistematik ini bukan sahaja akan mengurangkan pembaziran dari sudut masa dan kos, tetapi juga akan memudahkan operasi pengagihan logistik dan membantu dalam menguruskan dan mencapai kelestarian sumber dalam jangka masa yang panjang. Secara ringkasnya, pembangunan sistem sebegini mampu mendatangkan faedah dalam pengurusan risiko bencana. Kebolehupayaan algoritma ini disahkan dengan menandingi empat algoritma lain yang terkenal dalam ujian menggunakan 25 masalah penanda aras untuk menyelesaikan masalah pengoptimuman berasaskan parameter nombor nyata. Keputusan pengiraan juga menunjukkan bahawa algoritma ini mampu memberikan kadar penumpuan yang lebih pantas untuk kebanyakan masalah penanda aras dan kadar keberkesanan yang makin bertambah apabila dimensi masalah bertambah besar. Algoritma tersebut dipertingkatkan lagi dengan kaedah tambahan dalam menyelesaikan masalah penghalaan kenderaan dengan kapasiti dan tetingkap masa (CVRPTW*)*, iaitu FIPSaDE berserta carian setempat (FIPSaDE-LS) dan FIPSaDE bersama pindah silang novel bertertib (FIPSaDE-NOX). Variasi yang ditambahbaik ini dinilai keupayaanya dengan

pembandingan terhadap keputusan terkenal daripada penanda aras Solomon. Algoritma FIPSaDE, FIPSaDE-LS dan FIPSaDE-NOX mempunyai mekanisma yang efektif dan amat cekap dalam mengadaptasi kebolehan untuk menjelajah dan mengeksploitasi setiap zarah untuk menyelesaikan pelbagai masalah. Keputusan penanda aras dan keputusan simulasi menunjukkan kebolehan kesemua variasi FIPSaDE yang dibangunkan dimana kesemuanya berkebolehan dalam menyelesaikan pelbagai jenis masalah yang diberikan. Algoritma FIPSaDE dan dua variasi yang diperkenalkan menjanjikan keputusan yang baik dalam menyelesaikan masalah berasaskan parameter nombor nyata dan jugak dalam masalah CVRPTW. Algoritma-algoritma yang anjal ini menunjukkan potensi yang boleh diperluas bukan hanya dalam permasalahan penghalaan kenderaan, tetapi jugak dalam masalah penjadualan yang lain.

HYBRID META-HEURISTIC ALGORITHMS FOR SOLVING VEHICLE ROUTING PROBLEMS

ABSTRACT

In order to lessen the damages or loss receive during disaster by the affected community, proper and efficient scheduling and routing of fleets is required. Considering all negative consequences that would emerge from the calamitous event, arbitrary fleets scheduling, and routing process is burdened by time efficiency and pressure handling; except all the parameter needed is known beforehand which further helps in the sustainability of resources. Accordingly, a strategic fleet scheduling and routing system is demanded to overcome this limitation. Therefore, a heuristic approach which integrates Fully Informed Particle Swarm (FIPS) with Self-Adaptive Differential Evolution (SaDE) is proposed, namely FIPSaDE. The development of a systematic fleet scheduling system will not only minimize the possible wastage in terms of time and cost; but will also smooth the operations of logistic distribution and helps in managing and achieving long-term resource sustainability. In short, the development of such system will be beneficial in disaster risk management. The strength of the algorithm is validated using 25 comprehensive benchmark problems from the literature against four known algorithms in solving real-based parameter optimization problem. The computational results also demonstrate that for most of the benchmark functions, the algorithms show faster convergence and are increasingly efficient as the problem dimensions get larger. The algorithm is further enhanced with additional method in solving a variant of Vehicle Routing Problem, which is the Capacitated Vehicle Routing Problem with time window constraints or CVRPTW, namely, FIPSaDE with Local Search (FIPSaDE-LS) and FIPSaDE with Novel Ordered Crossover (FIPSaDE-NOX). The extended variants of proposed algorithm are assessed against best known results from Solomon Benchmark. FIPSaDE algorithm together with FIPSaDE-LS and FIPSaDE-NOX are effective and efficient mechanisms to adaptively adjust the exploration and exploitation strengths of each particle in combating cases with different characteristics. The benchmark test result and simulation results statistically affirm the effectiveness and efficiency of all three variants of FIPSaDE in tackling different problems introduced. FIPSaDE algorithm

and its two variants have shown promising results on real-parameter benchmark problems and on CVRPTW problems. The algorithms are flexible and have the potential to be extended to not only vehicle routing problems, but also to cope with other scheduling problems.

CHAPTER 1

INTRODUCTION

1.1 Introduction

"Disaster response planning and prevention, or preparedness, are performed when all is sane and quiet, and decisions are made in a rational, carefully considered manner."

Aho (2012)

Disaster risk management is about the preparedness of a community, city, or region during disaster in terms its safety and security levels. According to the terminology specified by United Nations International Strategy For Disaster Reduction (UNISDR, 2009), disaster risk management refers to the systematic process of manipulating resources available in order to lessen the unfavorable impacts of hazards and the possibility of disaster. One of the key issues in disaster risk management is distribution of resources to the affected areas. A strategic planning process is in demand to orderly handle resources distribution.

Immediate response and related decision are very crucial in relieving, controlling, and alleviating the disaster impacts. Thus, a disaster risk management model (Figure 1.1) had been proposed (Ibrahim et al., 2013) to help us understand stages or processes happen before disaster, during disaster, and after disaster. Another disaster risk management model is used by MERCY (Figure 1.2) had a quite similarity with the former model, which the former is much more details than the latter. The Total Disaster Risk Management (TDRM) model (Figure 1.2) approach is adopted by MERCY Malaysia on 2005 to further support its involvement in being a full-fledged humanitarian organization, where previously only act on medical relief (MERCY Malaysia, 2015).

TDRM model applies disaster risk management process to all five phases in the disaster management cycle, from emergency response to recovery activities, to prevention/mitigation process, and preparedness/readiness. Emergency response phase is critical as it involves life-saving phase; ensures any rescue efforts, medical assistance required, and evacuation plan are taking place when a disaster occurs. Then, rehabilitation and reconstruction efforts take place in Recovery phase. Prevention or Mitigation phase focuses on efforts needed to prevent or mitigate any damages done when a disaster happened. The TDRM model ended with Disaster Preparedness phase to helps minimize the impact of disaster. Adverse impact can be effectively reduced if proper and adequate planning is placed in all five phases of the disaster management cycle (MERCY Malaysia, 2015).

According to Ibrahim et al. (2013), risk reduction and resilience enhancement processes are given equal importance as presented in the neo-DRM-SD Model (Disaster Risk Model – Sustainable Development) (Figure 1.1). The model prompts strategic intervention at the risk level to reduced manifold risks posed by sustainable development challenges to levels that are manageable by people and planet through mitigation and adaptation process. The proposed approach assists in providing no-regret measures while magnify the efforts needed on mitigation challenges involving policy, finance, and mind-set changes.

It is important to note that there are four independent variables that act as pillars for the neo-DRM-SD model, which are Prevention (Prev), Preparedness (Prep), Response (Resp) and Recovery (Reco). There are many definitions to sustainable development, where the most common used is "*development that meets the needs of the present without compromising the ability of future generations to meet their own needs*" (Ibrahim et al. 2013). Two key concepts that need to be given priority to when determining the best decision for sustainability, are the concept of needs and the idea of limitations imposed by the state of technology and social organization on the environment's ability in order to cater for the needs of present and future time. Sustainable development or sustainability is an important concept for 21st century stating that it is important to have development while preserving the natural resources and protecting the planet while advancing the economic growth.

According to Thomas and Kopczak (2005), humanitarian logistics is defined as "*process of planning, implementing and controlling the efficient, cost-effective flow and storage of goods and material, as well as related information, from the point of origin to the point of consumption for the purpose of alleviating the suffering of vulnerable people*". There are several reasons that encompass the important of humanitarian logistics as the central for disaster management, which is the effectiveness and speed of response is crucial for major humanitarian program.

Besides that, by considering the needs of transportation and supplies during the logistics process, it can be one of the most expensive parts in the relief operation. Post-event learning could also be analyzed by looking through the repository of data from the logistics activities. As logistics data reflects all aspects of relief operation, including during preceding disaster until post-disaster recovery period, data retrieved from the situations is crucial to the performance of both current and future planning and operations.

Figure 1.1 Neo Disaster Risk Management for Sustainable Development (neo-DRM-SD) (Ibrahim, Koshy, & Asrar, 2013)

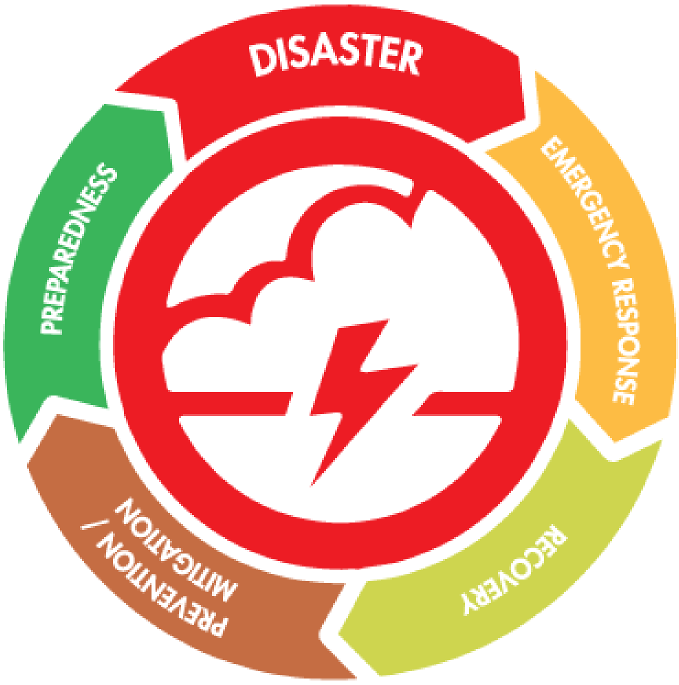

Figure 1.2 Total Disaster Risk Management (TDRM) Model (MERCY Malaysia, 2015)

Lack of recognition of the importance of logistics as most of the funds received are mostly allocated for the use of front-door services, such as the provision of food, water, shelter, sanitation and more. These are the common problems face in the field of humanitarian logistics as discovered by Thomas and Kopczak (2005):

1. Lack of professional staff involve in the disaster management process.
2. Inadequate use of tracking and tracing medium or the use of technology.
3. Lack of institutional learning on handling relief efforts.
4. Limited collaboration in order to gain fund for logistics needs.

Proper management of logistics routing has the opportunity in increasing its contribution and helps further the effectiveness of disaster risk management.

This thesis explores the problem and strengths faced by Self-Adaptive Differential Evolution and Particle Swarm Optimization in the area of Vehicle Routing Problems.

The output of this work comes in the form of a hybridized model of a Self-Adaptive Differential Algorithm and Fully Informed Particle Swarm; a variant of Particle Swarm Optimization, called the Fully Informed Particle Swarm with Self-adaptive Differential Evolution (FIPSaDE).

This chapter is structured as follows: A brief overview of the importance of fleet scheduling and FIPSaDE in Section 1.2. Section 1.3 explained the problems intended to be solved by this research. Research questions dealt in this research are listed in Section 1.4, objectives of this research in Section 1.5, motivation of this research in Section 1.6 and then the scope of this study in Section 1.7. Finally, brief summary on structure for each chapter is written in Section 1.8.

1.2 Background of Study

Proper and efficient scheduling of fleets is required in order to lessen the damages or loss received by the affected community during natural disasters such as earthquake and flood (Liu et al., 2016). Considering all negative consequences that would emerge from the calamitous event, arbitrary fleet scheduling process is burdened by time efficiency and pressure handling; except all the parameters needed are known beforehand (Korošec & Papa, 2013).

During the occurrence of natural disasters such as flood, inappropriate scheduling of emergency logistic such as food, medicine and humanitarian services is important to minimize the victim’s sufferings. The inability of providing medical supply to the affected area in time may also increase the spreading of certain diseases, which will worsen the situation. Fleet arrangement significantly contributes to the operational effectiveness of emergency logistic during the flood disaster. Unfortunately, optimizing the fleet scheduling is complicated as it is matched by uncertainty and complexity of the flood problem. Planning the emergency logistic scheduling arbitrarily without considering all possibly circumstances in advance will cause a delay in dispatching emergency commodities to the affected areas. Therefore, a decision-making tool that generates the fleet schedule effectively in order to optimize the use of the available fleets at the minimum cost is needed.

1.3 Problem Statement

In handling emergency situation, limited resources are expected as either resources might be unavailable, damaged, or slow. Considering this limitation issue, the scheduling process should be performed as optimal as possible to cater the losses occurred in the emergency situation. Scientific and proper allocation of resources can seriously reduce the level of emergency seriousness by properly handling the limited resources. The Vehicle Routing Problem or VRP has been used to modelled the emergency relief situations as covered in various papers (Afshar & Haghani, 2012; Liu et al., 2016; Mguis et al., 2012; Tlili et al., 2017)

VRPs falls under scheduling problem and is a very complex NP-hard (non-deterministic polynomial-time hard) problem to be solved without the help of optimization techniques. VRP is one of the most important processes required in reducing the cost of scheduling work during emergency situations such as floods, earthquake and more (Liu et al., 2016; Mguis et al., 2012).

VRP has always been a rather complicated task especially for large scale scheduling problems. In fact, logistic resources and logistics demand are dynamic as they differ across both time and area. The change in demand is met by the appropriate dispatching of available or might be limited resources. As the routing problems are full of uncertainty, the effect of any substance in the scheduling resources (e.g., different demand, unwanted changes on the number of customer or demand, changing or switching route) may affect any interconnected resources and constraints imposed on the routing system.

The problem includes optimization algorithms and lists of mathematical formula which is difficult to solve by manual calculation which often involves the use of large number of variables and restrictions, and excessive computation time (Barcos et al., 2010). Evolutionary algorithms offer calculation method designed to solve various problem cases including the optimization and scheduling problems (Peng et al., 2020).

Many optimization methods have been applied when solving the VRP. The techniques range from judgement of expertise to powerful mathematical programming methods. The judgement of expertise depends solely upon human expertise, previous knowledge of the expertise and knowledge of the system. The most applied approaches for this problem can be classified into three groups that are from mathematical optimization methods (e.g., linear programming, non-linear programming, dynamic programming, integer, and mixed integer programming), heuristic method (e.g., Saving algorithm, Two phase algorithms and Tour Construction Heuristics) and meta-heuristic methods (e.g., Nearest Neighbor Algorithm, Generic Algorithm, Differential Evolution, Particle Swarm Optimization). The use of meta-heuristics method to find approximate solution to large instances problems become a practical consideration in dealing with large instance problems in a reasonable time (Bozorgi-Amiri et al., 2012).

Additionally, some problems require different parameter settings according to different problems characteristics. These optimization problems are varied in terms of complexity. Some problems might require an algorithm with high exploration capability when the solution space increase exponentially according to problem dimension. Problem characteristics may also change with the increase in the problem dimension. For instance, the changes on problem characteristics may turns unimodal problems into multimodal problem with high dimension.

Therefore, in order to address the aforementioned NP-hard problem instances, an improved version of evolutionary algorithm with the ability to adaptively choose the suitable parameter settings to suit the problem is investigated. Fully Informed Particle Swarm or FIPS, is selected as the algorithm that will be used to strengthen the ability of exploration particle for the developed algorithm. A combination between FIPS and SaDE or Self-adaptive Differential Evolution is reckoned as both algorithm will contemplate each other by FIPS focusing in enhancing the exploration ability (Jordan, Helwig, & Wanka, 2008; Łukasik & Kowalski, 2014) while SaDE (Brest et al., 2006) will be focusing in exploiting the particle to navigate through the swarm space. Heighten ability in exploration and exploitation strategy for the improved algorithm is expected.

Moreover, two variants of the proposed algorithm have been integrated with local search method to further improve its performance in dealing with VRP.

1.4 Research Questions

1. What is the element needed in creating an evolutionary algorithm suited for VRPs?

2. How to achieve the cost reduction in scheduling based on evolutionary approaches in order to minimize the vehicle routing scheduling problem?

3. How to computationally model the heuristic and stochastic models in scheduling system?

1.5 Research Objectives

This thesis aims to alleviate the aforementioned issues by developing a hybrid evolutionary algorithm for solving the VRPs. The objectives of this research are presented as follows:

1. To evaluate the effectiveness and performances of hybridization FIPS and SaDE for solving the optimization problem.

2. To develop a fleet scheduling system or a self-adapting stochastic programming model in the evolutionary-based scheduling system that will minimize the possible wastage of distance and cost.

1.6 Significance of the Study

The distribution of emergency commodities relies heavily on scheduling vehicle routing. Inappropriate scheduling and routing of fleet will cause delay in the delivery of commodities to the disaster areas. Whereas proper routing of logistic resources could significantly contribute to the emergency operational effectiveness.

Arbitrary routing based on trial-and-error, intuition or experiences is inadequate in order to maximize resources available and minimize undesirable wastage of time and cost. The proposed system with improved version of algorithm gather in this research could possibly smooth out emergency logistic and is beneficial in disaster risk management.

1.7 Scope of the Study

The scope of this research focused on the design and development of the hybrid evolutionary algorithms. This research will also focus on managing resources in VRP by implementing a vehicle-scheduling model. The enhanced variants of hybrid algorithms created will solve the single objective global minimization problems and multi-criteria routing problems. Specifically, an optimize scheduling which combines evolutionary algorithm and swarm intelligence optimization will be implemented intended to gain the best solution to minimize undesirable wastage of time and cost.

The proposed algorithm is tested on a total of 25 benchmarks functions with different types of problems to evaluate the performance of the algorithm. Each of the benchmark problem is created specially to evaluate certain properties of algorithm which are useful in verifying the viability of fundamental concepts introduced into the proposed evolutionary algorithm. A set of Solomon problem instances are specifically designed to imitate varies properties of VRP. These problem instances will also be used in investigating and evaluating the feasibility of the proposed algorithms. The proposed algorithm is also tested on a real data scenario taken from the Kelantan's flood management database. A Java UI will be specifically designed to examine the implementation of proposed algorithm towards benchmark functions and also real data scenario collected. The solution will then be presented in image format for better visualization.

The proposed algorithms alongside with numerous published evolutionary algorithm variants are coded and tested using Java programming language and Eclipse software with Intel® Core™ i5-7200U CPU @ 2.50GHz and 4GB RAM environment.

1.8 Organization of Thesis

This thesis consists of five chapters and the organization of the thesis is discussed as below. In Chapter 2, a literature review on the Evolutionary algorithm is discussed in details. The literature further reviewed regarding Differential Evolution (DE), variants of DE, Particle Swarm Optimization (PSO) and variants of PSO, for which the proposed algorithm is based on the variants of these two algorithms. The aim of this chapter is to identify the motivation that leads to the objective of the research.

Chapter 3 introduced the proposed algorithm in solving CVRPTW, also known as a Capacitated Vehicle Routing Problem with Time Windows, where the number of delivery vehicle is fixed beforehand, demands is known beforehand and serviced by a single commodity from a common depot at minimum transit cost. The routing problem is also bounded by time constraints. This study focus on hybridizing FIPS algorithm with SaDE algorithm and the details of the hybridization are described in this chapter. The related computational results and the discussion of the hybridizing implementation is presented in the later part of chapter 3.

Chapter 4 provides an experimental study to identify the competitive nature of the developed algorithm in solving real-based parameter optimization problem and integer-based problem. In addition, the developed FIPSaDE is evaluated on a set of known benchmark functions for VRP related problem. The proposed algorithms are also compared with known algorithm for real-based parameter optimization test and known best results for CVRPTW test case.

Chapter 5 concludes the thesis and summarizes the research objectives addressed in the thesis. Recommendation and future work improvement are also provided in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents the literature review of Evolutionary Algorithm (EA) and Vehicle Routing Problem (VRP). The chapter starts with the introduction to EA in Section 2.1, followed by the discussion on the related algorithm under evolutionary algorithm from Section 2.2 to Section 2.5. Then, the next subsection discusses the benchmark for optimization problem. Section 2.7 explores on the literature review on VRP and Section 2.8 explores on the benchmark test used for VRP with Time Window (VRPTW) problem. Summary for Chapter 2 is presented in Section 2.9.

2.2 Evolutionary Algorithm (EA)

Inspired by natural phenomena, evolution processes and collective behaviors of groups of animals, EA and swarm intelligence (SI) have become more popular each year in computer intelligence discipline (Ma et al., 2019). Both types of artificial intelligence had been used in wide area of real-life problems; optimal power flow (Y. Chen et al., 2016), imaging problem (He & Huang, 2016; Kang et al., 2013), medical related problem (Gómez et al., 2016; S. P. Singh et al., 2016), manufacturing (Hu et al., 2012), routing (Fallah et al., 2019; Kechmane et al., 2018), and scheduling (Akjiratikarl et al., 2007; Yusof et al., 2011) and more. EAs are referred to evolutionary computation (EC) algorithm (Slowik & Kwasnicka, 2020) and simulate the rule of living organism such as following the rule of natural selection, mutation, recombination and survival of the fittest.

Common idea for EA is that it works in a group of a population of individuals, where they are coded in binary, real number of composite data structured (Pytel, 2020). The population is processed in several loops, called as generation, until the population able to reached the optimum value, the minimum or maximum value of fitness function assigned to each individual. The fitness function describes the ability of every individual in the population to be chosen for the next generation by following the survival of the fittest rule.

In order to find the best optimum value intended by algorithm, there are two main processes manipulated by EAs which are exploration process and exploitation process. Exploration process will search for a new region in the solution space where an optimum value may exist. Exploitation process is a process of searching within previously visited neighborhood. It is important to keep both exploitation and exploration process in balance as both processes can help in finding the optimum value.

Most of real-life problems can be represented in nonlinear or linear optimization problems by varying in number of decision variables used. However, it is quite difficult to represent problems in linear programming considering the constraints face for each problem. In last decades, as stated by Thakur, Meghwani, and Jalota (2014), finding the global optimum solution for nonlinear optimization problems has become an active research area. Many techniques had been formulated to solve this kind of problem in which can be categorized into two groups, which are deterministic and stochastic.

According to Thakur et al. (2014), deterministic technique is difficult to implement and it depends on priori information about the objective function as the technique is local optimization technique. Search technique for this type of method relies heavily on initial guess solution and information about the problems space.

There are a number of techniques for this group that may be solve the problem by transforming the problem into unconstrained problem such as Quadratic Penalty technique, Augmented Lagrangian method (Singh, 2017), etc. Stochastic technique refers to methods such as Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), etc. where the algorithm can give different result for each run test carried out (Slowik & Kwasnicka, 2020).

EAs such as GA, DE, and Evolutionary Programming have been widely used in numeral diverse areas (Čičková & Števo, 2010; Shan et al., 2014; Shen & Wang, 2016) and scheduling problems is one of the kind (Shan et al., 2014). However, some of these algorithms alone may have drawback issues or limitations (Chase et al., 2010).

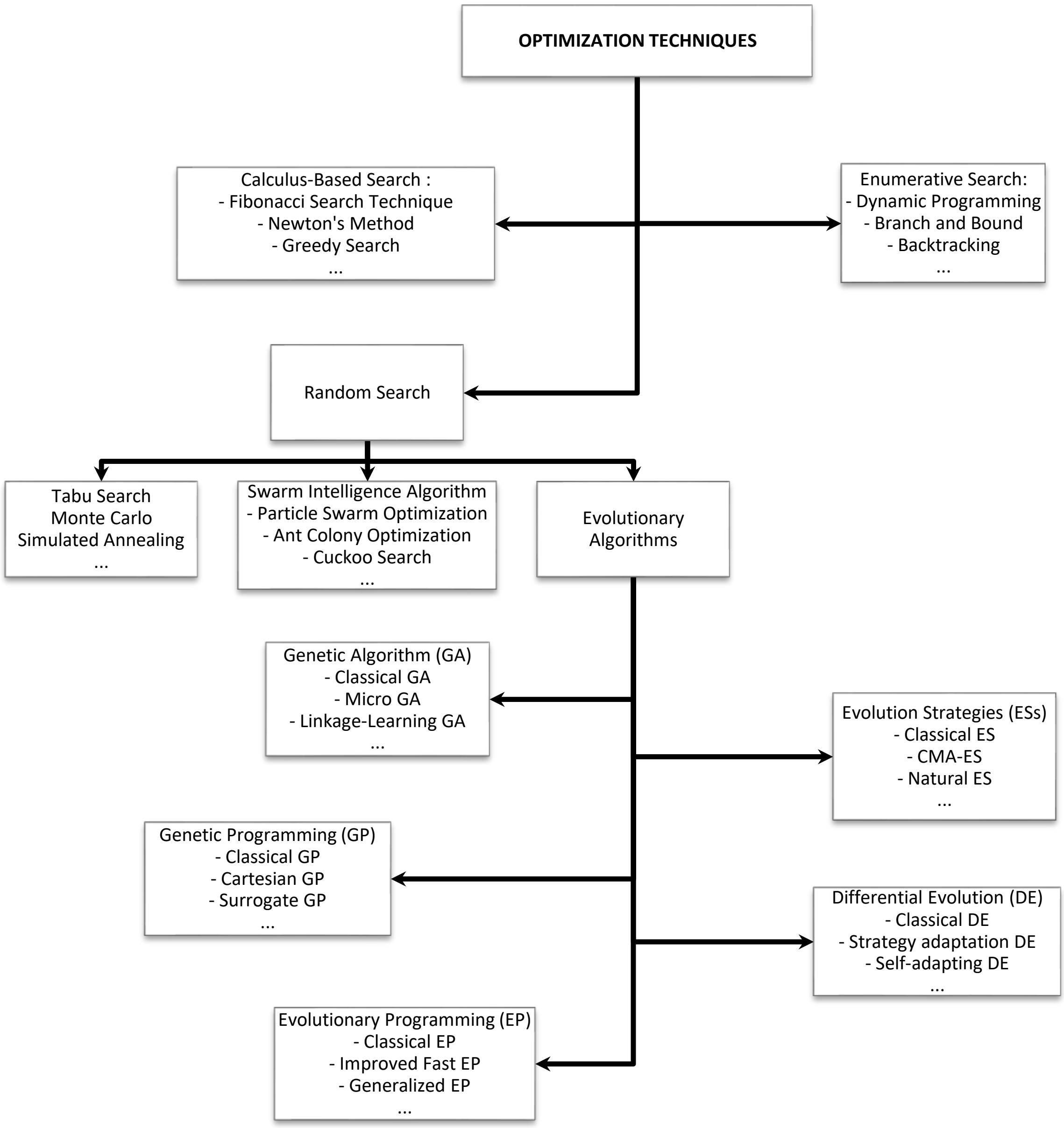

Figure 2.1 Taxonomy of nature-inspired methods (Slowik & Kwasnicka, 2020)

2.3 Differential Evolution (DE)

Differential Evolution, a vector population based stochastic optimization method, is a simple with a remarkable performance (Ardia et al., 2011) and powerful evolutionary algorithm (Yan et al., 2011) for solving global optimization problems (Yang et al., 2013). DE had been introduced by Storn & Price (1995) in order to overcome GA time consuming drawback (Leboucher et al., 2016). It is known to perform better in solving intricate global optimization problems and is brief in process (Shan et al., 2014).

DE uses similar operators as its predecessor algorithm, GA, which are crossover, mutation, and selection. The main difference between GA and DE is the mutation phase that makes DE becomes self-adaptive and improves the selection phase process. GA relies on crossover phase while DE relies on mutation phase. Mutation scheme in DE helps makes DE more self-adaptive in which all solutions have the chances in becoming a parent without depending on their fitness value (Karaboğa & Okdem, 2014).

Three advantages of DE are the ability to find the true global minimum regardless of the initial parameter values, fast convergence and using a few control parameter (Karaboğa & Okdem, 2014; Yang et al., 2013). DE relies in its solution in handling starting initial point, where multiple starting point is randomly chosen during sampling of objective function (Price et al., 2005). The main step of DE is summarized in Figure 2.2.

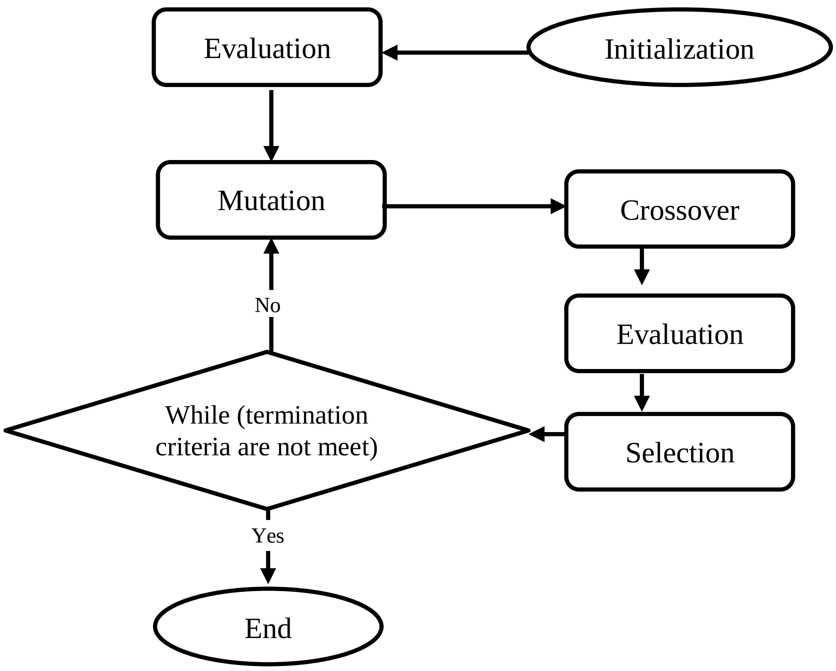

Figure 2.2 The DE algorithm

According to the original DE algorithm (Price et al., 2005), DE solution is represented in vector type. Each vector is indexed with a number from 0 to Np, which stands for number of populations. Generation is represent by G while an individual is represented as a D-dimensional vector (Brest & Maučec, 2011). New points are developed with the scaled difference of two randomly selected population vectors, results in a multiple random new point. To produce a trial vector, $u0$**,** DE will next add the scaled, random vector difference to a third randomly selected population vector. During selection stage, the trial vector is set to compete with population vector of the same index.

This process of choosing the next best vector is repeated until all Np population vectors have competed against a randomly generated trial vector. The result of this process is a new generation where the survivors from previous Np pairwise competition are chosen as the parents in the next generation. Both Algorithm 2.1 and Algorithm 2.2 briefly explain the process of basic DE. More details on the main process of differential evolution by (Price, Storn, & Lampinen, 2005) is explained from Subsection 2.3.1 to subsection 2.3.4.

```
Input: Randomly initialized position and velocity of
particles
-----------------------------------------------------
Begin
Initialize population;
Evaluate fitness;
     For i = 0 to max-iteration
     Do
          Begin
               Create Difference-offspring;
               Evaluate fitness;
               If an offspring is better than its parent;
                    Then, replace parent by offspring in
                    next gen;
               End if;
     End for;
End;
-----------------------------------------------------
Output: Position of appropriate global optima, x*
```

Algorithm 2.1 Pseudocode for simplified form of DE according to (Price et al., 2005)

```
…
while (convergence criterion not yet met) {
      // v_i defines the mutated vector
      // x_i defines a vector of the current vector population
      // u_i defines a vector of the trial vector
      // Np defines number of population
      for (i = 0; i < Np; i++) {
            // value for r1,r2 and r3 are randomly selected
            // from 1 until Np
            r1= random(Np);
            r2= random(Np);
            r3= random(Np);
            v_i = x_r3 + F * (x_r1 - x_r2);
            if (f(v_i) ≤ f(x_i))
                  u_i = v_i;
            else
                  u_i = x_i;
      }
} end while
…
```

Algorithm 2.2 Pseudocode for mutation process of DE according to (Price et al., 2005)

2.3.1 Initialization

The success of an EA depends on its ability of choosing suitable parameter settings (Qin et al., 2009). There are two types of parameter updating rules in an adaptive EA, which are absolute and empirical values (Angeline, 1995; Qin et al., 2009). Absolute updating rules pre-specify how the parameter modification would be made while empirical updating rules evolve the parameter values according to the competition inherent in EAs (Angeline, 1995).

Current population is symbolized by P_x as in Equation (2.1) and composed by vectors $x_{i,g}$ as in Equation (2.2) that have already been accepted as either initial points or by comparison with another vectors.

$$P_{x,g} = (x_{i,g}), \qquad i = 0,1,\dots,N_P - 1, \qquad g = 0,1,\dots,g_{max}, \tag{2.1}$$

$$x_{i,g} = (x_{j,i,g}), \qquad j = 0,1,\dots,D-1 \tag{2.2}$$

In working with arrays and modular arithmetic (Neale, 2011), the starting number is generally 0. Index i refers to population index, assigned to each vector that runs from 0 to $N_P - 1$. While index g refers to generation of the vector and runs from 0 to g_{max}. As for index j, it is assigned to parameter within vectors in which runs from 0 until D. The D parameter influences the optimization of objective function as it is referring to the value of variables used in defining the objective function.

Upper and lower bounds for each vector's parameter is prescribed before initializing of population. The bounds are represented by b_L and b_U respectively. After initializing bounds, a random number of real value is generated for each parameter within prescribed range. For example, if the initial value, g, is 4, the equation for the random number generator is presented as in Equation (2.3).

$$x_{j,i,4} = random_j(0,1).\left(b_{j,U} - b_{j,L}\right) + b_{j,L} \tag{2.3}$$

$random_j$ represents random number generator for j^{th} parameter which returns a uniformly distributed random number from the range of 0 to 1. This process results in a list of vectors filled with random numbers that are confined by previous upper and lower bound limit.

Once initialized, randomly chosen vectors are mutated to produce an intermediary population, $P_{v,g}$, of Np mutant vectors, $v_{i,g}$ as presented in Equation (2.4). The mutation process will be further explained as in sub-section 2.3.2.

$$P_{v,g} = (v_{i,g}) = (v_{j,i,g}), \qquad j = 0,1,\dots,D-1, \qquad i = 0,1,\dots,Np-1, \qquad g = 0,1,\dots,g_{max}, \tag{2.4}$$

Next, the mutated vector is then recombined with each vector in the current population in order to produce a trial population, $P_{u,g}$, presented as in Equation ((2.5)), of Np trial vectors, $u_{i,g}$. Each individual in the population will take turn being selected as a parent vector in each mutation iteration.

$$P_{u,g} = (u_{i,g}) = (u_{j,i,g}), \qquad j = 0,1,\dots,D-1, \qquad i = 0,1,\dots,N_P-1, \qquad g = 0,1,\dots,g_{max}, \tag{2.5}$$

On recombination or crossover process, a single array holding both trial population and mutant population is produced. The explanation on crossover process will be explained as in sub section 2.3.3.

2.3.2 Mutation

Mutation is introduced to expand search space. The mutation process starts with choosing a parent vector from the initial population of N_P size. Target vector is selected in a way that both target vector and parent vector are not the same vector from the population. Next, two vectors are selected randomly in which all of the selected individuals are not equal to each other such that i $\neq$ r1 $\neq$ r2 $\neq$ r3 where i is the index for parent vector, $r1$ is the base vector, $r2$ and $r3$ are randomly selected vectors.

Index $r1$ can be determined in multiple ways. One of the ideas in choosing the target vector is by choosing random vector index that has different value from parent vector i. $r2$ and $r3$ are also randomly selected once per mutant and both must have different value from base and target vector index. A new mutant vector can be created by using one of the mutation strategies as in Table 2.1.

Equation ((2.6)) presents the formation of $DE/x/y/z$ needed to create mutant vector by combining three different randomly chosen vectors, The x position index specifies the vector to be mutated, y is the number of difference vectors used and z denotes the crossover scheme. x can be in the form of $'rand'$ which is randomly chosen population vector or $'best'$ where the best vector is taken from current population.

The most used strategies for y position index are by using 1 or 2 difference vectors. Hence, using $r2$ and $r3$ for one difference vector and $r2, r3, r4$ and $r5$ for two difference vectors. The most useful strategies are $DE/rand/1$, $DE/best/1$, $DE/current-to-best/1$, $DE/best/2$ and $DE/rand/2$ (Brest & Maučec, 2011).

$$v_{i,g} = x_{r1,g} + F \cdot \left(x_{r2,g} - x_{r3,g}\right) \tag{2.6}$$

Scale factor F is a mutation scale value with a positive real number that is employed to control evolution rate of population. Effective value of F is seldom greater than 1.0 and has no upper limit.

Table 2.1 List of mutation strategies in Differential Evolution

No	**Mutation Strategies**	**Equation**
1	$DE/rand/1/bin$	Use good solutions immediately $v_i = x_{r1} + F_1(x_{r2} - x_{r3})$
2	$DE/best/1/bin$	Always start from best $v_i = x_{best} + F_1(x_{r2} - x_{r3})$
3	$DE/rand-to-best/1/bin$	Include movement to best (analogy with PSO) $v_i = x_{r1} + F_1(x_{r2} - x_{r3}) + F_2(x_{best} - x_{r1})$
4	$DE/current-to-best/1/bin$	Use good solutions immediately $v_i = x_i + F_1(x_{r2} - x_{r3}) + F_2(x_{best} - x_i)$
5	$DE/best/2/bin$	Always start from best but use two different vectors $v_i = x_{r1} + F_1(x_{r2} - x_{r3} + x_{r4} - x_{r5})$
6	$DE/rand/2/bin$	Add two different vectors instead of one $v_i = x_{best} + F_1(x_{r2} - x_{r3} + x_{r4} + x_{r5})$

2.3.3 Crossover or Recombination

Recombination, also known as discrete recombination, crossover and uniform crossover is implemented in order to reuse previously successful individuals. The crossover is formulated as in Equation ((2.7)), together with rules in processing the crossover value.

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if(rand_j(0,1) \leq Cr \; or \; j = j_{rand} \\ x_{j,i,g} & otherwise \end{cases} \tag{2.7}$$

Cr is a crossover parameter with a user-defined value bounded between 0 and 1. The bounds are set according to the nature of problem (Brest et al., 2006). The value represents crossover probability or represents the probability of creating trial vector parameter from the mutant vector (Brest et al., 2006) As both trial parameter from mutant and from previous vector are stored as a single array, the source that contributes to those values can be determined by comparing Cr to the output of a uniform random number generator.

If the random number generated is less than or equal to Cr , then the value for trial parameter is then taken from mutant vector, $v_{i,g}$. If the random number is bigger than Cr, then the value will be taken from vector $x_{i,g}$ Index j_{rand} represents randomly chosen integer within the range from 1 to Np. The value of the control parameter may sometimes fall out of bounds. During this instance, the value is either reset to bound value or use the out of bound value without any changes (Brest et al., 2006).

2.3.4 Selection

Selection process in Differential Evolution mimics survival-of-the-fittest. For example, a better vector having an equal or lower objective value (for minimization problem) has a higher chance to retain in the population for the next generation, as shown in Equation ((2.8)). After new population for new generation has been formed, the process of mutation, recombination and selection will be repeated until certain termination criteria are satisfied.

$$u_{i,g+1} = \begin{cases} v_{i,g} & if \; f(v_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & otherwise \end{cases} \tag{2.8}$$

2.4 Self-Adaptive Differential Algorithm

Control parameter is not considered as evolving object on earliest EA algorithm as it is included in external fixed parameter. Later on, after a number of literatures done, it was realized that these parameter should be altered in evolution process in order to achieve optimal convergence demanded (Brest et al., 2006). In order to manipulate control parameters, two major forms or the control ideas are introduced, which are parameter tuning and parameter control.

Parameter tuning is an approach to find good values of the control parameter before running the algorithm. As for parameter control, it is an approach in which the parameters are changed during the run time of DE. Pseudocode for Self-adaptive Differential Evolution or SaDE is summarized in Algorithm 2.3.

```
Set the generation counter, G = 0
Randomly initialize a population of Np individuals
Evaluate the population
While stopping criterion is not satisfied
     Do
          Calculate strategy probability.
          Update the Success and Failure Memory
          Assign trial vector generation strategy and
          parameter to each target vector
          Generate a new population (each trial vector is
          generated according to associated trial vector
          generation strategy and parameter and in Section
          3.2
          Randomly initialize trial vector U within search
          space if any variable is outside its boundaries
          Select the best vector and parameter
          Update the Success and Failure Memory
          Increment generation count, G = G + 1
End While
```

Algorithm 2.3 Pseudocode for original SaDE algorithm

SaDE is a variant of DE that can obtain better solutions compared to classical DE algorithm (Zhalechian et al., 2017). SaDE had been employed in various optimization problems. In SaDE, 2 out of 3 critical control parameters of DE, F and Cr, are tuned for DE improvement. Np is not favoured as chosen parameter as the value is not sensitive to the efficiency and robustness of DE algorithm (Brest et al., 2006).

According to research by Qin and Suganthan (2005), Np is retained as a user-specified parameter, while F is set between (0,2] with a normal distribution of mean 0.5 and standard deviation of 0.3 for different individuals in the current population. The range between (0,2] is chosen instead of normally used (0,1] as a means to keep both small F values for local search and high F values for global search for generating potential good mutant vector. As for the control parameter Cr, proper value choice will lead to good performance under several learning strategies while a wrong value choice will lead to deteriorate performance even under several learning strategies. Therefore, the authors decide to dynamically adapt Cr value to the suitable range by assembling previous learning experience within a certain generation.

Initially, Cr will be normally distributed in a range with mean of Cr_m and standard deviation, std, is 0.1. Starting Cr_m value is set at 0.5. The Cr values will remain for several generations and a new value with similar normal distribution will be generated in another generation. In every generation, better Cr values associated with trial vector that managed to enter next generation are recorded. The mean for normal distribution of Cr according to all recorded values corresponding to successful trial vectors during the period is recalculated. The above procedure is repeated until a suitable Cr value range for the problem is found.

The record of successful Cr values is not kept during recalculation of normal distribution mean in order to avoid possible inappropriate long-term accumulation effects. After a specified number of generations, a local search named Quasi-Newton method, is introduced to speed up the convergence speed. The method is implemented as the prespecified maximum function evaluation (MAX_FEs) are too small to reach the acquired level accuracy.

Brest et al. (2006) introduced jDE, another self-adaptive Differential Evolution algorithm based on the self-adapting control parameter mechanism. F and Cr are set as random numbers within certain ranges or set as the values of latest generation created. jDE demonstrated better than or almost the same result compared to traditional DE algorithm. Zaharie (2003) proposed ADE, an adaptive DE algorithm with a variable population where a new parameter was introduced to control the rate of population variance. Liu and Lampinen (2005) developed FADE, a fuzzy adaptive DE algorithm which implemented fuzzy logic to manipulate F and Cr values. FADE could converge faster than traditional DE when tested on problem with high dimensionality.

In Ensemble of Mutation Strategies and Parameters in DE (EPSDE), a pool of distinct mutation strategies along with a pool of values for each control parameter coexist through the evolution process in an adaptive algorithm (Vasconcelos Segundo et al., 2017) created by (Mallipeddi et al., 2011). Each member in the population is randomly assigned with a mutation strategy and associated parameter values.

If the generated trial vector can produce better solution compared to the target vector, then the mutation strategy and parameter values for the particular member are retained with trial vector. Thus, it will be chosen to become the parent chromosome in the next generation. If the generated target vector is better than the developed trial vector, then the mutation strategy and associate parameter values assigned to the target vector will be reinitialized with a new mutation strategy and parameter values taken from the respective pools or taken from the successful combination stored. The retained method or the reinitialized method able to leads the vector into an increase probability of choosing better offspring with better combination of mutation strategy and the associated control parameters for future generations.

Jiang et al. (2013) present IADE, an improved adaptive DE algorithm with a simple structure that could automatically adjust the control parameters according to the fitness values during optimization process. In IADE, there is no strict rule when choosing F and Cr and the value range for both parameters are varied according to different optimization problems used.

The fitness values of the subsequent previous steps are an important factor in creating the next F and Cr values. An exponential function to map F and Cr in the range from $[0,1]$ to $[0.5,1]$ is used as the exponential function has a smooth characteristic compared to linear function, with arbitrary order derivatives. The expression used to