HYBRIDIZING BAT ALGORITHM WITH LOCAL DISCRETE SEARCH FOR DYNAMIC TRAVELING SALESMAN PROBLEM

WONG YUAN LING

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITI MALAYSIA SABAH
2015
DECLARATION

I hereby declare that this thesis, submitted to Universiti Malaysia Sabah as partial fulfillment of the requirements for the degree of bachelor computer science, has not been submitted to other university for any degree. I also certify that the works described herein is entirely my own, except for the quotations and summaries sources of which have been duly acknowledged.

This thesis might be available within the university library and maybe photocopied or loaned to other libraries for the purpose of consultation.

01 July 2015

________________________
WONG YUAN LING

CERTIFIED BY

________________________
ASSOCIATE PROFESSOR DR. JASON TEO TZE WI
SUPERVISOR
ACKNOWLEDGMENT

First and foremost, I would like to thank my supervisor, Associate Professor Dr. Jason Teo Tze Wi for his patience, constructive guidance and motivation that lead to the completion of this thesis. Not forgetting, to my examiner and panel, Dr. Lau Hui Keng and Dr. Norazlina Khamis for their critical views and ideas about my project.

Additionally, I would like to thank my family and friends for their unconditional support, help and insights.

Last but not least, an utmost thanks and gratitude to God for His grace.
ABSTRACT

Dynamic optimization problems have become increasingly popular to solve in computational science. As real world problems are becoming more complex day by day, dynamic optimization problems have become an alternative platform for the application of heuristics methods. A higher class of heuristics, called metaheuristics has been a popular choice in solving optimization problems because of their ability to adapt to uncertainties and is not biased to any specific problem. One of the well-known classes of metaheuristics is swarm intelligence, inspired by the behavior of animal swarm in the nature. Ant Colony System and Particle Swarm Optimization are two instances of swarm intelligence. Bat algorithm on the other hand is one of the latest swarm intelligence proposed. It has been applied to many continuous and discrete problem domains. In this research, we will attempt to apply hybrid bat algorithm with local search to solve a well-established dynamic combinatorial problem, which is the dynamic traveling salesman problem (DTSP). The experiments included in this thesis are the parameter tuning of the bat algorithm parameters and then using those parameters to determine the optimal settings. The settings are then used in the bat algorithm framework to compare with other metaheuristics such as ACO and ACO with local search. As we have proposed two variants of the bat algorithm, we found that from the experiments, the second proposed variant which is the bat algorithm with natural frequency performs better compared to the bat algorithm with the original proposed frequency across all benchmarks and dynamic test cases. However, the proposed algorithms were still unable to outperform the conventional ACO and hybridized ACO algorithms. Nevertheless, their performances are enhanced by the hybridization of the 2-Opt local search.
ABSTRAK

Masalah pengoptimuman dinamik merupakan masalah yang semakin popular untuk diselesaikan di dalam bidang pengkomputeran sains. Masalah dunia sebenar yang semakin kompleks telah menjadikan masalah pengkomputeran sebagai platform alternatif untuk pengaplikasian kaedah heuristik. Kaedah heuristik yang lebih sofistikated, juga dikenali sebagai metaheuristik merupakan pilihan yang popular untuk menyelesaikan masalah pengoptimuman dinamik disebabkan kemampuan mereka untuk mengadaptasi kepada ketidaktentuan dan tidak menyebelahi mana-mana masalah yang spesifik. Salah satu klas metaheuristik yang terkenal adalah kecerdasan kawanan, yang diinspirasikan daripada perlakuan kawanan haiwan semula jadi. Antara kecerdasan kawanan yang telah dicadangkan adalah sistem koloni semut dan pengoptimuman kawan an partikel. Algoritma kelawar pula merupakan salah satu kecerdasan kawanan yang telah dicadangkan baru-baru ini. Ia telah diaplikasikan di dalam pelbagai masalah domain diskret dan lanjutan. Di dalam kajian ini, kami akan cuba untuk mengaplikasi algoritma kelawar dengan strategi carian tempatan untuk menyelesaikan masalah kombinatorik dinamik iaitu masalah perjalanan jurujual dinamik. Eksperimen yang telah dijalankan termasuklah pengoptimuman parameter algoritma kelawar dan perbandingan antara nilai optima yang diperolehi dari algoritma kelawar yang dicadangkan, sistem koloni semut dan sistem koloni semut yang telah dihibridasikan dengan strategi carian tempatan. Kita telah mencadangkan dua jenis algoritma kelawar dan hasil daripada eksperimen eksperimen yang dilakukan, kita mendapati bahawa algoritma jenis kedua mempunyai nilai optima yang lebih baik untuk setiap kes perjalanan. Walaupun begitu, prestasi algoritma yang dicadangkan kurang memberangangkan kerana nilai optima yang diperoleh kurang jika dibandingkan dengan algoritma semut and versi semut yang dihibridasikan dengan carian tempatan. Namun, ini membuktikan bahawa sesuatu algoritma itu dapat dipertingkatkan dari segi efisien melalui pernambahan strategi carian tempatan.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>ii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>iv</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>iv-viii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix-x</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xi-xii</td>
</tr>
<tr>
<td>GLOSSARY OF RESEARCH KEY TERMS AND PHRASES</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>xiv-xviii</td>
</tr>
<tr>
<td>LIST OF MATHEMATICAL SYMBOLS</td>
<td>xix</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problem Background</td>
<td>1-2</td>
</tr>
<tr>
<td>1.2.1 Definition of TSP</td>
<td>2-3</td>
</tr>
<tr>
<td>1.2.2 Definition of DTSP</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Problem Statement</td>
<td>3-4</td>
</tr>
<tr>
<td>1.4 Hypothesis</td>
<td>4</td>
</tr>
</tbody>
</table>
1.5 Objectives of Research ........................................................................................................ 4
1.6 Organization of Report ......................................................................................................... 5-6

CHAPTER 2: LITERATURE REVIEW

2.1 Overview .......................................................................................................................... 7
2.2 Optimization Problems and it Forms .................................................................................. 8-9
2.3 TSP and DTSP Previous Experimental Results ......................................................................9-13
2.4 Bat Algorithm in Literature Review .................................................................................. 13
    2.4.1 Problem Domain Solved by BA ........................................................................ 13-15
    2.4.2 Different Techniques of BA .................................................................................... 16
    2.4.3 BA in Static TSP .................................................................................................... 16
2.5 Critical Summary ..............................................................................................................17-25
    2.5.1 Open Research Questions ....................................................................................... 25-26
2.6 Conclusion .......................................................................................................................... 27

CHAPTER 3: METHODOLOGY

3.1 Overview .......................................................................................................................... 28
3.2 Development Process Model ............................................................................................ 22-29
    3.2.1 Basic Problem Coding and Determine Research Scope Phase .................................. 29
    3.2.2 Integrate and Release Planning Phase ....................................................................... 30
    3.2.3 Iteration and Test Phase .......................................................................................... 30
    3.2.4 Deploy Proper Version of Code Phase ......................................................................... 30
3.3 Software and Hardware Requirement .................................................................................. 30
3.4 Basic Bat Algorithm ......................................................................................................... 31-32
    3.4.1 Local Search used in hybrid bat algorithm .............................................................. 33-34
3.5 Performance Benchmark of DTSP .................................................................................... 34-35
6.2 Experiment 1: Performance of BA-I and BA-II in static TSP environment

   Opposing to DABA ................................................................. 61-64

6.2.1 Experiment 2.1 Using Fixed Size of Candidate List of 5 ................................. 65-73

6.2.2 Experiment 2.2 Using Fixed Size of Candidate List of 10 .............................. 73-74


CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion .................................................................................. 86-87

7.2 Future Work ................................................................................ 87

REFERENCES
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.0: Existing Research on Solving DTSP</td>
<td>18-19</td>
</tr>
<tr>
<td>Table 2.1: Summary of Bat Algorithm in Literature Review</td>
<td>20-22</td>
</tr>
<tr>
<td>Table 3.1: The Tunable Parameters of Bat Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>Table 3.2: Test Cases and Their Dynamic Value</td>
<td>37</td>
</tr>
<tr>
<td>Table 4.1: The Tunable Parameters of Bat Algorithm</td>
<td>45</td>
</tr>
<tr>
<td>Table 6.1: The Algorithms Involved in the Experiments</td>
<td>60</td>
</tr>
<tr>
<td>Table 6.2: TSP Instances Involved in the Experiments</td>
<td>61</td>
</tr>
<tr>
<td>Table 6.3: Experiment One Settings</td>
<td>61-62</td>
</tr>
<tr>
<td>Table 6.4: Best Solutions found by DABA and BA-I</td>
<td>62</td>
</tr>
<tr>
<td>Table 6.5: Experiment Two Settings</td>
<td>63-64</td>
</tr>
<tr>
<td>Table 6.6: Experiment Result 2.1 Using Table 6.1 Settings (Low Parameter Settings)</td>
<td>65-66</td>
</tr>
<tr>
<td>Table 6.7: Experiment Result 2.1 Using Table 6.1 Settings (Default Parameter Settings)</td>
<td>66-68</td>
</tr>
<tr>
<td>Table 6.8: Experiment Result 2.1 Using Table 6.1 Settings (High Parameter Settings)</td>
<td>68-70</td>
</tr>
</tbody>
</table>
Table 6.9: Comparison of BA-II Obtained From Table 6.6, 6.7 and 6.8  
70-72

Table 7.0: Experiment Results 2.2 Based On Experiment Result 2.1 (Low Parameters Setting)  
73-74

Table 7.1: Experiment Three Settings  
75

Table 7.2: Experiment Three Result 2.1 Low Parameters Settings of BA (High Parameter Settings)  
76-82

Table 7.3: Summary of Error Performance for Experiment Three  
83-84

Table 7.4: Summary of Overall Error Performance for Experiment Three  
85
<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Proposed taxonomies</td>
<td>8</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>The adapted extreme programming</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>The Flowchart of Proposed Bat Algorithm</td>
<td>44</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Generation of Random Object for Swapping</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Generation of Random Environment Over Time</td>
<td>47</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Generation of Reappearing Environment Over Time</td>
<td>47</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Generation of Varying Environment with Different Speed of Change and Degree of Change</td>
<td>48</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Evaluating 2-Opt operation</td>
<td>48</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Improvement of 2-Opt Tour Length</td>
<td>49</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Initialization of Integer Type (Matrices)</td>
<td>49</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Distances in the Form of Matrices</td>
<td>50</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Matrices Representing Distances</td>
<td>50</td>
</tr>
<tr>
<td>Figure 6.0</td>
<td>Calculation of Euclidean Distance</td>
<td>50</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>Calculation of Distance Matrix</td>
<td>51</td>
</tr>
</tbody>
</table>
Figure 6.2: Struct Initialization

Figure 6.3: Allocating Bat structure

Figure 6.4: Allocating Rate Emission, Velocity and Candidate List

Figure 6.5: Bat Frequency

Figure 6.6: Velocity

Figure 6.7: First Rate Emission Initialization

Figure 6.8: The Movement of Bat after Cities in Candidate List are Traversed

Figure 6.9: An Exception Handling for Updating Method

Figure 7.0: The Rate Emission and Velocity Probability along Traversed Edge

Figure 7.1: Continuations from Figure 3.1

Figure 7.2: Fitness Evaluation of a Tour

Figure 7.3: The Second Rate Emission (Global Best Bat)

Figure 7.4: Swapping Two Cities for 2-Opt

Figure 7.5: Evaluating 2-Opt Operation

Figure 7.6: Improvement Tour Length in 2-Opt

Figure 7.7: Optimizing Final Tour with 2-Opt
**GLOSSARY OF RESEARCH KEY TERMS AND PHRASES**

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark problem</td>
<td>A standardized and specific problem for assessing the performance of algorithms</td>
</tr>
<tr>
<td>Combinatorial</td>
<td>Related to finite discrete domain variables</td>
</tr>
<tr>
<td>Continuous</td>
<td>Related to varying continuously real-valued variables</td>
</tr>
<tr>
<td>Domain</td>
<td>Defines the scope where the problem occurs</td>
</tr>
<tr>
<td>Dynamic</td>
<td>A consistent change in state over a period of time</td>
</tr>
<tr>
<td>Evolutionary Computation</td>
<td>Subfield of artificial intelligence that involves combinatorial and continuous optimization problems</td>
</tr>
<tr>
<td>Non-polynomial</td>
<td>Is the set of decision problems where the “yes” instances can be accepted in polynomial time by a deterministic Turing machine</td>
</tr>
<tr>
<td>Optimization problem</td>
<td>Problem of finding best solution from all feasible solutions</td>
</tr>
<tr>
<td>Problem instances</td>
<td>A set of computational problems that usually falls under the same category</td>
</tr>
<tr>
<td>Static</td>
<td>A fixed and unchanged state over a period of time</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

ABC  -  Artificial Bee Colony
ACO  -  Ant Colony Optimization
ACS  -  Ant Colony System
AES  -  Agent-based Evolutionary Search
ALS  -  Annealing with Learning Scheme
APF  -  Annealed Particle Filter
ASTP-GA  -  Asymmetric Traveling Salesman Genetic Algorithm
BA  -  Bat Algorithm
BAM  -  Bat Algorithm with Mutation
BAST  -  Bat Algorithm based on Scheduling Tool
BBA  -  Binary Bat Algorithm
BBO  -  Biogeograph-based Optimization
CBA  -  Chaotic Bat Algorithm
CS - Cuckoo Search

CS-GRASP - Cuckoo Search with Greedy Randomized Adaptive Search Procedure

CVRP - Capacitated Vehicle Routing Problem

CWS - Clark and Wright Savings

DABA - Directed Artificial Bat Algorithm

DBA - Discrete Bat Algorithm

DBG - Dynamic Benchmark Generator

DCOP - Dynamic Combinatorial Optimization Problem

DE - Differential Evolution

DIOEA - Dynamic Inver-over Evolutionary Algorithm

DOP - Dynamic Optimization Problem

DPSO - Dynamic Particle Swarm Optimization

DTSP - Dynamic Traveling Salesman Problem

EC - Evolutionary Computing
FA - Firefly Algorithm

FLBA - Fuzzy Logic Bat Algorithm

GA - Genetic Algorithm

GRASP - Greedy Randomized Adaptive Search Procedure

GSA - Gravitational Search Algorithm

HBA - Hybrid Bat Algorithm

HBA-PR - Hybridized Bat Algorithm with Path Relinking

HS - Harmony Search

IBA - Improved Bat Algorithm

IEEE - Institute of Electrical and Electronics Engineers

ILS - Iterated Local Search

MMRE - Mean Magnitude of Relative Error

MOBA - Multi-Objective Bat Algorithm

NP - Non-deterministic Polynomial time
PBIL - Population-based Incremental Learning

PF - Particle Filter

PGA - Parallel Genetic Algorithm

PSO - Particle Swarm Optimization

RFD - River Formation Dynamic

SA - Simulated Annealing

SD - Standard Deviation

SGA - Simple Genetic Algorithm

SOP - Static Optimization Problem

SSPF - Scatter Search Particle Filter

SR-GCWS - Simulation via Routing in the Generalized Clark and Wright Savings

STSP-GA - Symmetric Traveling Salesman Problem Genetic Algorithm

TLBO - Teaching and Learning Based Optimization

TS - Tabu Search
TSP - Traveling Salesman Problem

TSPLIB - Library of sample instances for TSP (and related) instances

XOR - Exclusive OR
LIST OF MATHEMATICAL SYMBOLS

\( \Sigma \) - Sum of

\( \neq \) - Not equal to

\( = \) - Equal to

\( > \) - Greater than

\( < \) - Less than

\( \leq \) - Less greater than or equal to

\( \in \) - Belong to

\( \mathcal{O} \) - Big O notation

\( \approx \) - Rounded to

\( \% \) - Percentage of
CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter will examine the research topic which is hybridizing bat algorithm with discrete local search for solving dynamic traveling salesman problem (DTSP) from different aspects. The aspects include introduction to the problem background, problem statement, hypothesis, objectives and scope of research. In problem background, we will briefly explain the Traveling Salesman Problem (TSP), DTSP and why they remain difficult to solve till these days. TSP and DTSP will also be defined in this section. In problem statement section, we will briefly justify the use of metaheuristics method especially the bat algorithm in solving DTSP. For hypothesis, we will state the expected outcome for implementing bat algorithm with local discrete search in DTSP. We will further state the objectives and the scope for the research before settling on the organization of this report.

1.2 Problem Background

There are many core open problems still exist in the computational complexity theory that remain challengingly unsolved. One of the problems is non-polynomial (NP) hard problems where no efficient algorithm is known as they cannot be solved in polynomial time. In this context, Traveling Salesman Problem (TSP) would be the most famous case of NP hard problem. Till these days, TSP remains the most intensively studied problem in computational mathematics. The main objective for TSP can be described as simple as finding the shortest possible route while traveling exactly once across a set of cities that consists of certain distances and eventually returning to the starting city with minimal total distance. The solution
space for TSP problem can be extremely large depending on the number of cities. In other words, it increases exponentially with the number of cities. Variants of TSP have also existed which the most crucial being the Dynamic Traveling Salesman Problem (DTSP). DTSP is basically the generalization of classic TSP where changes can be introduced by adding or deleting cities, swapping location of cities or changing the value of pairwise distances. After changes are introduced, the salesman needs to re-plan his route. Many real world optimization problems are dynamic optimization problems (DOPs) where changes may occur over time regarding the objective function, decision variable, constraints and others. Therefore, DTSP is one of the useful benchmark problems that can be used to emulate the real dynamic traffic scenario. For example, exchangeable cities and traffic factors where the global optimum value is unknown during the environment.

1.2.1 Definition of Traveling Salesman Problem (TSP)

In order to create an algorithm framework for DTSP, we have to define TSP first. It can be described as follows: Given a collection of cities and pairwise distances of the cities, we need to find the shortest path that starts from one city and visits each of the other cities once and only once before returning to the starting city. The main objective is to minimize the sum of distances used to visit the entire tour. Usually the problem is represented by a fully connected weighted graph \( G = (N, A) \), where \( N = \{0, \ldots, n\} \) is a set of nodes and \( A = \{(i, j) : i \neq j\} \) is a set of arcs. The collection of cities is represented by the set \( N \) and the connections between them by the set \( A \). Each connection \((i, j)\) is associated with a non-negative value \( d_{ij} \) which represents the distance between cities \( i \) and \( j \). TSP can be formally described as follows:

\[
f(x) = \min \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} \psi_{ij}
\]
Subject to:

\[ \psi_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is used in the tour} \\ 0, & \text{otherwise} \end{cases} \]

Where \( \psi_{ij} \in \{0, 1\} \), \( n \) is the number of the cities and \( d_{ij} \) is the distance between city \( i \) and \( j \).
Each city \( i \in N \) has a location defined by \((x, y)\) and each connection \((i, j) \in A\) is associated with a non-negative distance \( d_{ij} \).

1.2.2 Definition of DTSP

In this paper, the DTSP functions as the platform to change any static instances for conventional TSP into a dynamic environment where the fitness landscape remains unchanged but the algorithm will be inclined to be shifted to search for different parts of fitness landscape between time intervals. Since the encoding of the problems changes, the DTSP can be defined as:

\[ D(t) = \{d_{ij}(t)\}_{n(t) \times n(t)} \]

Where \( d_{ij}(t) \) is the cost from city \( i \) and \( j \), \( t \) is the real world time. The number of cities \( n(t) \) and the cost matrix are time dependent.

1.3 Problem Statement

Traditional method such as brute force search is infeasible to be applied to solve TSP. Therefore, the use of heuristics and exact methods are widely accepted instead to solve TSP. A more versatile heuristics, which is the metaheuristics has gained popularity in recent years due to its flexibility which can be used as the black-box approach. Nevertheless, applying metaheuristics method can solve TSP or DTSP more efficiently in terms of the quality of the optimal solutions. Through this research, we will implement a relatively new metaheuristics, bat
algorithm with local search to solve DTSP, considering its potential in various fields of optimization problems.

1.4 Hypothesis

Hybridization of bat algorithm with local search strategy will improve the chances of finding a good quality of optimal solution. The quality of the optimal solution in this context shall be based on comparison of optimal solutions obtained in the hybridized bat algorithms and other metaheuristics methods.

1.5 Objectives of Research

a. To design and implement proposed an improved metaheuristic algorithm which is the hybridized bat algorithm with local discrete search to solve DTSP.

b. To deploy and test hybridized bat algorithm with local discrete search to solve DTSP.

c. To verify the performance of hybridized bat algorithm with local discrete search by direct comparison with other heuristics and metaheuristics for solving DTSP.

1.6 Scope of Research

Our research will focus on the application of recently developed nature inspired algorithm which is bat algorithm, which will be hybridized with local discrete search in solving traveling salesman problem with a dynamic environment (as the benchmark problem) only and not solving other optimization problems. In order to demonstrate the effectiveness of the algorithm, several well known problem instances have been selected from the IEEE competition website in regard of the benchmark problem (Mavrovouniotis, Li, Yang and Yao, 2013). The problem instances consist of seven cities, namely eil51, kroA100, kroA150, kroA200, lin318, and pr439. In between testing and performing the algorithm to solve DTSP, some research might
also be conducted on the tuning of the parameters in the bat algorithm to improve its overall performance. The tuning of parameters shall be aimed in refining the rate of the convergence for the bat algorithm. Hopefully after some parameter tuning or new algorithm integration, it will perform better in solving DTSP. The newly created algorithm will be evaluated against other metaheuristics algorithms which are ant colony optimization, basic bat algorithm and ant colony optimization with 2-Opt local search.

1.7 Organization of Report

This report contains three chapters and they are organized in the following manner; chapter one will introduce the overview about TSP and DTSP in problem background and problem statement. Following problem statement, we will propose the hypothesis for this research. We will also determine our objectives for this research which also serves as the research motivation. This chapter ends with scope of research.

Chapter two explores the importance of investigation in both combinatorial and continuous optimization problem. The combinatorial problem will be further specified into TSP that operates in static and dynamic environment. We will briefly describe the metaheuristics used to solve TSP and DTSP. This chapter also highlights the main metaheuristics method used in this research, which is the bat algorithm in terms of its previous related work and its application in both theoretical and real-world scenario. Finally, for critical summary, we will summarize the existing work related to TSP and DTSP, justify the weakness and the strength of bat algorithm, how bat algorithm is adapted to solve discrete optimization problem and where it is most beneficial to apply to. This chapter concludes on the proposal of some open research questions.

Chapter three explains the methodology used. The first part describes the basic bat algorithm; the rest describes the potential local search algorithms to be integrated into the bat algorithm. This chapter also defines the performance benchmark in terms of experimental setup, statistical testing, parameter and the tuning of algorithm, the determination and evaluation of dynamism and the metric of TSP.
Chapter four describes the system design and analysis. This chapter will elaborate on the design of proposed hybridized bat algorithms, together with the modification of bat algorithm formulas and the solution representation definition. The chapter ends with the structures of proposed bat algorithms.

Chapter five shows the code snippets of the proposed algorithms corresponding to explanations in chapter four.

Chapter six focuses on the testing and results of the proposed algorithms based on the statistical testing criteria performance explained in chapter three.

Chapter seven concludes the report by summarizing previous chapters and findings. Moreover, the future works regarding the proposed algorithms are revealed.