Heterogeneous Ant Colony Optimisation Methods and their Application to the Travelling Salesman and PCB Drilling Problems

By

Ahamed Fayeez Tuani Ibrahim

Computer Science Department,

University of Exeter

2020

Supervised by Professor Edward Keedwell & Dr Matthew Collett

Declaration

Submitted by Ahamed Fayeez Tuani Ibrahim, to the University of Exeter, as a thesis for the degree of Doctor of Philosophy in Computer Science.

This thesis is available for the Library use on the understanding that it is copyright material and that no quotation from this thesis may be published without proper acknowledgment.

I certify that all material in this thesis that is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signature) ....

Abstract

Ant Colony Optimization (ACO) is an optimization algorithm that is inspired by the foraging behaviour of real ants in locating and transporting food source to their nest. It is designed as a population-based metaheuristic and have been successfully implemented on various NP-hard problems such as the well-known Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP) and many more. However, majority of the studies in ACO focused on homogeneous artificial ants although animal behaviour researchers suggest that real ants exhibit heterogeneous behaviour thus improving the overall efficiency of the ant colonies. Equally important is that most, if not all, optimization algorithms require proper parameter tuning to achieve optimal performance. However, it is well-known that parameters are problem-dependant as different problems or even different instances have different optimal parameter settings. Parameter tuning through the testing of parameter combinations is a computationally expensive procedure that is infeasible on large-scale real-world problems. One method to mitigate this is to introduce heterogeneity by initializing the artificial agents with individual parameters rather than colony level parameters. This allows the algorithm to either actively or passively discover good parameter settings during the search. The approach undertaken in this study is to randomly initialize the ants from both uniform and Gaussian distribution respectively within a predefined range of values. The approach taken in this study is one of biological plausibility for ants with similar roles, but differing behavioural traits, which are being drawn from a mathematical distribution. This study also introduces an adaptive approach to the heterogeneous ant colony population that evolves the alpha and beta controlling parameters for ACO to locate near-optimal solutions. The adaptive approach is able to modify the exploitation and exploration characteristics of the algorithm during the search to reflect the dynamic nature of search. An empirical analysis of the proposed algorithm tested on a range of Travelling Salesman Problem (TSP) instances shows that the approach has better algorithmic performance when compared against state-of-the-art algorithms from the literature.

In the name of the Almighty God, the Most Gracious & The Most Merciful

Acknowledgments

First, I would like to express my gratitude to the Almighty God for giving me the strength and the courage to undertake and complete this study.

I would like to take this opportunity to thank my wife, Norhazlina Maidin for her support and sacrifices that she made along this beautiful but winding journey to the completion of this thesis. I am forever grateful and indebted to my parents and siblings for their unwavering support, care, and words of encouragement that pushed me until the last mile. I also would like to thank my parents-in-law for being understanding and supportive all these years. To my little sweethearts, Marissa and Aqeel, thank you for your love, support, and patience in me completing this thesis.

It has been a wonderful experience doing my Ph.D. at the University of Exeter hence I would like to acknowledge the people who have contributed to the success of this work. I will always be grateful to Professor Edward Keedwell for his expertise and patience in supervision that were crucial in the completion of this thesis. I would like to thank him for his encouragement and guidance in allowing me to grow as a researcher. One could not ask for any friendlier or supportive supervisor than him. I would like to sincerely thank my second supervisor, Dr. Matthew Collett for his time and inputs from the biological perception as well as his expert advice in analysing the results. I also would like to thank Dr. Antony Galton for being my mentor throughout my Ph.D.

I would like to express my appreciation to my employers, UTeM and Ministry of Higher Education (MoHE), Malaysia for the study leave and the financial support. Finally, I would like to thank everyone who may have contributed either directly or indirectly to the completion of this thesis.

Table of Contents

Declaration ii
Abstract iii
Acknowledgment v
Table of Contents vi
List of Tables ix
List of Figures xiii
List of Symbols xix
List of Acronyms xxi

1 Introduction 22

1.1 Overview 22
1.2 Motivation 24
1.3 Research Questions & Hypotheses 26
1.4 Aims & Objectives 27
1.5 Major Contributions 28
1.6 Thesis Organization 28
1.7 List of Publication 29

2 Literature Review 31

2.1 Emergent Behaviour 32
2.2 Real ants to artificial ants 34
2.3 Ant Colony Optimization 36
2.4 General Framework of ACO 38
2.5 Conventional ACO 40
2.6 Parameter Settings in ACO 45
2.7 Genetic Algorithm 47
2.8 Combinatorial Optimization Problem 48
2.9 Performance Metrics for Optimization Algorithm 50
2.9.1 Optimum-based Performance Metrics 50
2.9.2 Behaviour-based Performance Metrics 52
2.10 Heterogeneity 54

2.10.1 Heterogeneity in PSO 54
2.10.2 Heterogeneity in ACO 56
2.10.2.1 Individual-level Heterogeneity 56
2.10.2.2 Multi-colony Heterogeneity 61
2.10.3 Hybrid ACO 63
2.11 Concluding Remarks 66

3 Heterogeneous Ant Colony Optimization via a Uniform Distribution 68

3.1 Motivation 69
3.2 Formulations of the Heterogeneous Approach 70
3.2.1 Design of the Proposed Algorithm 72
3.3 Experimental Setup, Results & Discussion for Heterogeneous Ant System (HAS) 75
3.3.1 Experimental Setup 75
3.3.2 Verifying the Base Algorithm 75
3.3.3 Parameter Exploration of HAS 77
3.3.4 Comparison against the Base Algorithm 90
3.3.5 Heterogeneous Ants' Distribution Analysis (HAS) 95
3.4 Experimental Setup, Results & Discussion for Heterogeneous Max-Min Ant System (HMMAS) 101
3.4.1 Parameter Exploration of HMMAS 101
3.4.2 Comparison against the Base Algorithm (HMMAS) 106
3.4.3 Heterogeneous Ants' Distribution Analysis 109
3.5 Concluding Remarks 114

4 Heterogeneous Ant Colony Optimization using a Gaussian Distribution 116

4.1 Motivation 117
4.2 The Framework 118
4.2.1 Exploring Gaussian Properties 120
4.3 Experimental Setup, Results & Discussion 124

4.3.1 Experimental Setup ... 124
4.3.2 Comparison against the Base Algorithm & HMMAS ... 125
4.3.3 Printed Circuit Board (PCB) Holes Drilling Problem ... 128
4.3.4 GHMMAS Ants' Distribution Analysis ... 132
4.4 Concluding Remarks ... 136

5 A Heterogeneous Adaptive Max-Min Ant Algorithm System for Traveling Salesman Problem ... 138

5.1 Motivation ... 139
5.2 Heterogeneous Adaptive MMAS - The Framework ... 140
5.2.1 Greedy Ants Initialization ... 144
5.2.2 Adaptive Interval ... 148
5.2.3 Mutation ... 150
5.2.4 Local Search ... 154
5.3 Experimental Setup ... 154
5.4 Results & Discussion ... 156
5.4.1 Overall Results ... 156
5.4.2 Parameter Adaptation ... 160
5.4.3 Algorithm's Behaviour ... 165
5.5 Conclusion ... 167

6 Conclusion & Future Work ... 170

6.1 Significance of the Research & Contributions ... 171
6.1.1 HAS & HMMAS ... 171
6.1.2 GHMMAS ... 172
6.1.3 HAACO ... 173
6.2 Limitations of The Proposed Work ... 173
6.3 Conclusion ... 173
6.3 Future Work ... 174

References ... 176

List of Tables

2.1 Successful Conventional ACO 40
2.2 Summary of individual-level heterogeneity 58
3.1 Developed AS, MMAS vs Original AS, MMAS. Results show the average of the best cost. (Note: Average of 15 trials) 75
3.2 Results from parameter exploration of Heterogeneous Ant System (HAS) where α is uniformly distributed between 0 and 1 and 0 and 2 while β distribution varies. Algorithm tested on oliver30.tsp, eil51.tsp and eil101.tsp. Results represent the average best cost of 15 trials while values in bold represent the best in each category 78
3.3 p-value of post-hoc test using the Bonferroni correction method for HAS tested on oliver30.tsp (Opt=423.74). p-values in bold indicate the significant difference between the algorithms in comparison 80
3.4 p-value of post hoc test using the Bonferroni correction method for HAS with different range of α and β values tested on eil51.tsp (Opt=426). p-values in bold indicate significant difference between the algorithms in comparison 84
3.5 p-value of post hoc test using the Bonferroni correction method for HAS with different range of α and β values tested on eil101.tsp (Opt=629). p-values in bold indicate significant difference between the algorithms in comparison 87
3.6 Best, average and worst cost comparison between AS & HAS for several TSP instances for 25 trials with stopping criterion was set at 10 000 iterations. The optimum for the TSP instances are 426 (eil51.tsp), 21282 (kroA100.tsp), 15780(d198.tsp) and 42029(lin318.tsp). Results in bold represent best in each category 92

3.7 Results of parameter exploration of Heterogeneous Max-Min Ant System (HMMAS) where α is uniformly distributed between 0 and 1 as well as 0 and 2 while β distribution varies. The heterogeneous algorithm tested on eil51.tsp and eil101.tsp. Results represent the average best cost out of 15 trials while values in bold represent the best in each category 102

3.8 Best, average and worst cost comparison between HMMAS & MMAS tested on several TSP instances for 25 trials with stopping criterion of 10 000 iterations. The optimum for the TSP instances are 426 (eil51.tsp), 21282 (kroA100.tsp), 15780(d198.tsp) and 42029(lin318.tsp). Results in bold represent best in each category. 106

4.1 Results of GHMMAS with variation in distribution spread (standard deviation (s.d)) of α and β tested on eil51.tsp (Opt = 426) and eil101.tsp (Opt = 629). Values in bold are the best in each category. Results are of 15 trials, each trial = 1 000 iterations 121

4.2 Comparison between GHMMAS, HMMAS & MMAS tested on several TSP instances for 20 trials with stopping criterion of 10 000 iterations. The optimum for the TSP instances are 21282 (kroA100.tsp), 15780 (d198.tsp), 80369 (pr226.tsp) and 42029 (lin318.tsp). Results in bold represent best in each category 125

4.3 Results of Wilcoxon rank-sum test conducted on GHMMAS, HMMAS & MMAS based on solutions found in each of the 20 trials. Results in bold represent statistical significance 126

4.4 Best cost and average comparison between GHMMAS and MMPAS tested on several PCB instances for 20 trials with stopping criterion of 2 000 iterations. Results in bold represent best in each category.. 129

5.1 Performance comparison of the heterogeneous adaptive approach with a different number of iterations for greedy ants to explore the search space (-5 indicates 5 iterations and -10 indicates 10 iterations for greedy ants to explore) 145

5.2 Comparison the average best cost of heterogeneous adaptive approach with different uniform mutation range tested on several TSP instances where the results in bold indicate best in each category. Results are of 20 trials, each trial = 1000 iterations 151

5.3 Comparison the average best cost of heterogeneous adaptive approach with different standard deviation for the Gaussian mutation operator tested on several TSP instances where the results in bold indicate best in each category. Results are of 20 trials, each trial = 1000 iterations 151

5.4 Performance comparison of heterogeneous adaptive approach with uniform and Gaussian mutation operator respectively tested on several TSP instances. Results are of 20 trials, each trial = 1 000 iterations and those in bold indicates the best of each category 152

5.5 TSP instances, size of the problem, and its known optimum 154

5.6 Parameter settings as in the proposed approach 154

5.7 Fitness solution comparison of HAACO against several other approaches tested on several TSP instances. The result in bold indicates the best of each category....................................... 156

5.8 Ranking comparison of HAACO against several other approaches tested on several TSP instances based on the fitness solution in Table 5.7. The result in bold indicates the best of each category 156

5.9 Comparison of HAACO against other approaches based on average fitness solution. The result in bold indicates the best of each category 157

5.10 Ranking comparison of HAACO based on the average fitness solution in Table 5.9. The result in bold indicates the best of each category 158

5.11 Fitness solution comparison of HAACO against several other approaches tested on several TSP instances based on the lowest worst fitness solution over 20 trials. The result in bold indicates the best of each category 158

5.12 Ranking comparison of HAACO based on the average fitness solution in Table 5.11. The result in bold indicates the best of each category 159

List of Figures

2.1 Metaheuristic categories 32
2.2 Double bridge experimental setup (a) Same length branches (b) Different length branches 34
2.3 Example of a fully connected graph 36
2.4 Example of boxplots with same median but different quartile values 51
3.1 Example of the initial population of a single trial drawn from (a) uniform distribution and (b) normal distribution 70
3.2 Flowchart of the proposed heterogeneous approach 71
3.3 The principle of heterogeneity in ACO 73
3.4 Ant System (AS) behaviour for different α-β combinations 77
3.5 Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against oliver30.tsp. *(Note: 0_1 or 0_2 indicates the range of α values for heterogeneous colony while 0_5 to 4_5 indicates the range of β values)* 79
3.6 Confidence intervals of each of HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on oliver30.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap 81
3.7 Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-5 to 4-5 indicates the range of β values)* 82
3.8 Confidence interval of each of the HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combinations when tested on eil51.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap 85
3.9 Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against eil101.tsp. *(Note: 0-*

1 or 0-2 indicates the range of α values for heterogeneous colony while 0-5 to 4-5 indicates the range of β values) 86

3.10 Confidence intervals of each of the HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combinations when tested on eil101.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap 88

3.11 Boxplots illustrating the performance comparison of the heterogeneous approach against that of AS in terms of the best cost of 25 trials tested on several TSP instances. The blue line in each figure represents the optimum of that particular instance 91

3.12 Convergence plot of HAS and AS represented by the average best cost of 25 trials over 10 000 iterations 94

3.13 Histogram representing Alpha & Beta of iteration-best ants for HAS (eil51.tsp) 95

3.14 Histogram representing Alpha & Beta of iteration-best ants for HAS (kroA100.tsp) 97

3.15 Histogram representing Alpha & Beta of iteration-best ants for HAS (d198.tsp) 98

3.16 Histogram representing Alpha & Beta of iteration-best ants for HAS (lin318.tsp) 99

3.17 Boxplots representing 15 trials of different HMMAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-2 to 2-3 indicates the range of β values)* 102

3.18 Confidence intervals of each of the HMMAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on eil51.tsp 103

3.19 Boxplots representing 15 trials of different HMMAS parameter settings as shown in X-axis tested against eil101.tsp. *(Note:*

0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-2 to 2-3 indicates the range of β values) 104

3.20 Confidence intervals of each of the HMMAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on eil101.tsp 104

3.21 Boxplots representing the best cost of 25 trials of both MMAS and HMMAS tested on several TSP instances 108

3.22 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on eil51.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS 109

3.23 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on kroA100.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS... 110

3.24 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on d198.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS 111

3.25 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on lin318.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS... 112

4.1 Example of the initial population drawn from normal distribution consisting of 25 different populations representing 25 trials 117

4.2 Boxplot representing 15 trials of different GHMMAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: Ex: 0.1, 0.1 indicate the standard deviation for mean α and β)* 122

4.3 Boxplot representing 15 trials of different GHMMAS parameter settings as shown in the X-axis tested against eil101.tsp. *(Note: Ex: 0.1, 0.1 indicate the standard deviation for mean α and β)* 122

4.4 Boxplots representing the best cost of 20 trials of MMAS, HMMAS, and GHMMAS tested on several TSP instances 127
4.5 The optimized tour (2597) found by GHMMAS when tested on a280.tsp 129
4.6 The optimized tour (3943) found by GHMMAS when tested on tsp225.tsp 130
4.7 The optimized tour (51966) found by GHMMAS when tested on pcb442.tsp 130
4.8 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on kroA100.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations o1 GHMMAS 131
4.9 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on d198.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS 132
4.10 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on pr226.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS 133
4.11 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on lin318.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS 133
4.12 Standard deviations of fitness solutions found by the ants for every iteration (a) kroA100.tsp, (b) d198.tsp, (c) pr226.tsp and (d) lin318.tsp 135
5.1 The iterative procedure of the proposed approach 142
5.2 Ant's fitness solution representation in every iteration 143
5.3 Boxplots representing the best cost of heterogeneous adaptive approach with different initial greedy ants run tested on ch150.tsp (20 trials) 145

5.4 Boxplots representing the best cost of heterogeneous adaptive approach with different initial greedy ants run tested on kroA200.tsp (20 trials)....... 145

5.5 Pheromone distributions of the heterogeneous adaptive ants (eil76.tsp)....... 146

5.6 Boxplots representing best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on eil101.tsp 148

5.7 Boxplots representing best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on ch150.tsp....... 148

5.8 Boxplots representing best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on d198.tsp....... 149

5.9 Best cost plot of a single trial of HAACO tested on eil76.tsp 160

5.10 Parameter convergence of HAACO tested on st70.tsp 162

5.11 Parameter convergence of HAACO tested on lin105.tsp 163

5.12 Parameter convergence of HAACO tested on ch150.tsp 164

5.13 Analysis of HAACO tested on st70.tsp....... 165

5.14 Analysis of HAACO tested on lin105.tsp 165

5.15 Analysis of HAACO tested on ch150.tsp 166

List of Symbols

Item	*Meaning*
α	*ACO parameter that controls the relative importance of pheromone.*
α^k	*ACO parameter that controls the relative importance of pheromone for each individual ant.*
β	*ACO parameter that controls the relative importance of heuristic.*
β^k	*ACO parameter that controls the relative importance of heuristic for each individual ant.*
τ_{ij}	*Amount of pheromone on path i to j.*
$[\tau_{min}, \tau_{max}]$	*Range of pheromone values on the path*
$\Delta\tau_{ij}^{best}$	*Pheromone amount deposited by global-best or iteration-best ant*
τ_0	*Initial pheromone of the landscape.*
η_{ij}	*Heuristic value between path i to j.*
d_{ij}	*Distance between node i to node j.*
P_{ij}^k	*Probabilistic rule of ant k choosing node j from i.*
ρ	*Evaporation rate.*
N_i^k	*Feasible nodes for ant k to choose while at node i.*
I	*All possible nodes from node i.*
L_k	*Tour length of ant k*
L_{gb}	*Tour length of global best ant*
L_{best}	*Tour lenth of iteration-best or global-best*

P_{best}	*Parameter introduced to describe convergence*
m	*Number of ants*
n	*Number of cities for TSP dataset*
L_{nn}	*Tour length of TSP instance using nearest neighbour algorithm.*
Q	*Constant*
q	*Variable randomly drawn with uniform probability from 0 to 1.*
$q0$	*Constant used in ACS*
σ	*Standard Deviation*
AI	*Adaptive Interval*

List of Acronyms

ACO – Ant Colony Optimization

AI – Artificial Intelligence

SI – Swarm Intelligence

PSO – Particle Swarm Optimization

TSP – Travelling Salesman Problem

PCB – Printed Circuit Board

AS – Ant System

MMAS – Max-Min Ant System

ACS – Ant Colony System

EAS – Elitist Ant System

ASrank/RAS – Rank-based Ant System

Ant Q – Ant System based on Q-learning

HAS – Heterogeneous Ant System

HMMAS – Heterogeneous Max-Min Ant System

MMPAS – Max-Min Paraconsistent Ant System

GHMMAS – Gaussian Heterogeneous Max-Min Ant System

GA – Genetic Algorithm

HAACO – Heterogeneous Adaptive Ant Colony Optimization

ABC – Artificial Bee Colony

ES – Evolution Strategy

EP – Evolutionary Programming

GP – Genetic Programming

GRASP – Greedy Randomized Adaptive Search Procedure

VNS – Variable Neighbourhood Search

Chapter 1 Introduction

This chapter introduces the themes of this study and discusses the motivation for the work. Thereafter, the research questions and hypotheses concerning the study are highlighted followed by the aims and objectives of this study. Next, this chapter presents the scope of the research work and concludes with an overview of the dissertation as per-chapter basis.

1.1 Overview

Nature consists of various biological systems which in turn consist of populations of simple, individual agents that demonstrate decentralized, self-organized behaviour. Interestingly, these intelligent agents, which adhere to a set of basic rules, are able to communicate directly with each other or indirectly via the environment without the need for centralized control. Despite the individual agents being simple in its behaviour, together they form a very highly structured organization that allows them to collectively solve complex problems as well as being robust to adverse conditions. As an example, collectively, a colony of ants is able to accomplish nest construction or foraging for food where a single ant might fail.

The collective behaviour of the individual agents in the biological systems has inspired a collection of computational algorithms known as swarm intelligence used to solve complex problems. These algorithms consist of artificial agents that co-operate collectively without any centralized control while solving problems in various fields such as optimization, big data analysis, robotics and many more. It is a class of metaheuristics that takes inspiration from the behaviour of animals. Metaheuristics can be considered as a higher level trial-and-error method to find or locate quality solutions but without a guarantee that an optimal solution can be found [1]. The idea of the metaheuristics approach in solving complex problems is to explore and exploit the search landscape iteratively, effectively, and efficiently. The near-optimal solutions of the metaheuristics approach are considered good enough given the trade-off between solution quality versus time taken for the solution to be found. Some of the well-known metaheuristics algorithms are simulated annealing, particle swarm optimization (PSO), genetic algorithm, tabu search, ant colony optimization (ACO) and many more.

ACO is a population-based metaheuristic which is stochastic in nature and designed to construct solutions iteratively, also known as a constructive method [2] in order to solve combinatorial optimization problems. ACO algorithms are largely inspired by the foraging behaviour of the Argentine ants [3]. The basic concept is founded on the pheromone laying mechanism of the real ants while locating and transporting the food from the source to the nest via the shortest path. The algorithm consists of a colony of artificial ants that cooperatively explores and exploits the search landscape by constructing solutions to the optimization problems. The ants then exchange information regarding the solution's quality via artificial pheromone deposition and an evaporation mechanism. This mechanism, which is an indirect communication medium called 'stigmergy', allows an individual ant to alter the environment and thus acts as a stimuli for the colony of ants [4]. In the solution construction phase, each individual ant uses two important variables to guide them towards good solutions which are the problem-specific heuristic information and the feedback from other ants via the stigmergic information. These concepts act as the fundamental framework for most ACO algorithms [3][5][6][7].

In recent years, many ACO variants have been developed and successfully applied to various problems such as routing [8], scheduling [9], image processing, assembly line and many more. This shows that ACO is one of the most promising algorithms in swarm intelligence due to its robustness in solving various problems. An extensive review of past research and recent trends in ACO can be found in [10].

1.2 Motivation

It is well known that the behaviour and performance of an ACO algorithm strongly depend on the parameters initialized during the start-up [11][12][13][14]. Dorigo et al [6] analysed and summarised three categories of parameter values which are the good parameters, poor parameters that will not cause stagnation and lastly, poor parameters which will lead the colony to stagnation behaviour. This suggestion has acted as a guide for many ACO algorithms where the parameter values are set during initialization and kept constant throughout the search process. However, various studies and analyses both empirically and theoretically have shown that the optimal parameter settings are very much dependant on the problem being solved, the problem instances or even a particular stage of the search process [15][16][17][18][19][20]. As an example, parameter values in job shop scheduling [9] did not corroborate to any of the parameter suggestions by Dorigo and Stützle [3, p. 71].

Generally, parameter tuning may enhance the performance of the algorithm if tuned carefully. However, it is trivial and computationally expensive as it requires a considerable amount of time and processing power. In addition to this, a deep understanding of the algorithm’s behaviour and the problem being tackled is also important during parameter tuning. On top of that, the trial-and-error method is practically ineffective because it is a computationally exhaustive process to tune the parameters for every problem or problem instance. The tuning of the parameter values before the optimization process does not guarantee optimal performance in the ACO algorithms [21]. In essence, little research has been reported on parameter tuning in ACO [22].

Another drawback of the ACO algorithm is the exploration-exploitation conundrum where exploration is the ability of the algorithm to continue to search in the unexplored area of the search landscape while exploitation is the ability of the algorithm to perturb the solutions found in order to locate better solutions. The algorithm will achieve sub-optimal performance if it spends too much time on the exploration phase while if an algorithm performs too much exploitation, then the algorithm is more likely to get stuck in local optima due to premature convergence to sub-optimal solutions. Hence, a proper balance is required between exploration and exploitation in order to achieve a highly successful, efficient and robust algorithm [10][23][24].

Lack of population diversity is a key reason for premature convergence to local optima, especially in ACO algorithms. As most of the ACO algorithms deploy a homogeneous concept, where all ants in the colony have similar ‘behavioural traits’, the algorithm is unable to escape from this phenomenon due to stagnation behaviour of which all ants construct the same tours repeatedly. It is also down to the nature of the ACO algorithm that is unable to switch between the exploration and exploitation phase hence stuck in local optima.

All the aforementioned problems are based on static problems while dynamic problems where the search environment change over time poses different challenges to ACO researchers [25]. As this study only focuses on the static environment, the drawback of the ACO algorithm in a dynamic environment will not be explained here but is explored in [26]. As most of the ACO algorithms with the aforementioned drawbacks deploy a homogeneous population, one of the possible approaches to overcome the problems is to maintain diversity in a population-based algorithm such as ACO by implementing a heterogeneous single population approach where the ants are initialized with individual ‘behavioural traits’. This will allow the algorithm to switch between exploration and exploitation as the search progresses due to the inclusion of explorative and exploitative ants. The proposed framework will then be able to promote a self-adaptive approach by taking advantage of the specific strengths of each individual ant in different stages of the search process. Both of these approaches will be able to mitigate the aforementioned drawbacks of the ACO algorithm by removing the need for tedious optimal parameter tuning process and create a more robust and scalable ACO algorithm.

Lastly, heterogeneity is omnipresent in nature. Several biological studies have shown that real ant colonies are in fact heterogeneous where the ants are known to have individual ‘behavioural traits’ or personalities [27][28][29][30]. The ant colonies with higher variation between nest members are more productive and more efficient in nest maintenance and division of labour [28]. In another study, animal behaviour scientists have also found that the ant colonies do have individual personalities similar to that of humans where the colony consists of ants with different levels of aggressive behaviour [31]. In conclusion, instead of heterogeneity, conventional ACO algorithms deploy the homogeneous concept mainly due to the algorithmic simplicity in implementation. Therefore, the heterogeneous approach, which is proven to be effective from the biological aspect point of view, will be explored and its effectiveness will be analysed in this study.

1.3 Research questions & Hypotheses

This research aims to address the following questions:

1. How will the heterogeneous approach best be implemented and what degree of improvement is possible over base algorithm(s)?

2. What is/are the most suitable parameter value(s) that can be manipulated in order to introduce the heterogeneous approach in ACO?

3. How can an effective feedback mechanism to the self-adaptive approach be provided so that valuable information of the search landscape is taken into account during its decision-making?

4. How can the performance of the proposed approach be measured in terms of exploration-exploitation?

5. Can the process of evolution be used to explore the space of algorithm meta-parameters and lead to improved performance?

The hypothesis is that with heterogeneity, a combination of ants that are more inclined towards the exploration of the search space while other ants in the colony exploit the best path found creates a balance in the search process. This is due to the behaviours of the ants of which are randomly initialized either to be more inclined towards exploration or exploitation. In addition to that, the diversity preservation introduced by the unique biases towards the pheromone trail and local heuristics for each ant this algorithm helps balance the exploration-exploitation, increases robustness with respect to parameter settings and reduces the number of algorithm parameters that need to be set.

1.4 Aims & Objectives

The main aim of this research work is to study, explore and propose a single colony heterogeneous ACO framework that is robust to parameter settings with improved overall performance that is able to avoid getting stuck in local optima. In addition to that, the algorithm must be able to balance exploration and exploitation that indirectly increases the overall performance. The focus is on exploring and investigating the heterogeneous approach in ACO and how this method can alleviate the tedious parameter tuning procedure while able to achieve competitive solutions across different problem sizes with a simple to implement approach but still able to produce competitive results, if not the best when compared against other approaches. The following objectives are established to achieve these aims:

1. To develop a heterogeneous ACO that is able to improve on the performance of the base algorithms.
2. To explore the ranges of meta-parameters and their effect on the performance of the algorithm.
3. To explore the distribution-type within meta-parameter settings in the heterogeneous approach.
4. To propose a self-adaptive parameter adaptation of heterogeneous ACO that is able to locate and converge to instance-optimal parameter settings.
5. To apply the method to Travelling Salesman and Printed Circuit Board drilling problem instances.
6. To evaluate the performance of the proposed approach using appropriate measurement indicators on general problem instances such as TSP.
7. To compare the above methods with state-of-the-art methods taken from the literature.

1.5 Major Contributions

The thesis contributes to the field of ACO by introducing heterogeneity in ACO framework which is a novel algorithmic approach. The following are the contributions of the thesis that support the framework and to achieve the afore-mentioned aim and objectives:

1. The initial study of the thesis revolves around initializing the population of ants randomly from a uniform (Chapter 3) and Gaussian (Chapter 4) distribution within a pre-defined range of values. The variety of 'behavioural traits' introduced by the heterogeneous approach maintains the population's diversity hence the improvement in the performance of the algorithm compared to traditional ACO. The increased robustness and sensitivity to distinct parameters highlights the advantages of the proposed approach.
2. A self-adaptive approach consisting a hybrid of heterogeneous ACO with Genetic Algorithm (GA) was proposed in Chapter 5. The approach allows the algorithm to adapt the parameters as the search progresses, automatically adjust its search strategy by alternating between exploration and exploitation whenever required and autonomously locate instance-optimal parameter settings. The collective intelligence of the heterogeneous population also enables the ants to explore both the fitness and parameter landscape simultaneously hence more feasible than the time-consuming task of fine-tuning the parameters.

1.6 Thesis Organization

The remainder of this thesis is structured as follows.

Chapter 2 presents the biological aspects of real ants and heterogeneity in nature followed by a comparison between real ants and artificial ants. Then, a more detailed explanation of the conventional ACO algorithms as well as the parameter settings in ACO were explored. The general performance metrics used in an optimization especially in ACO is discussed in this chapter along with previous work in ACO in relation to heterogeneity.

Chapter 3 discusses in detail the implementation and analysis of the heterogeneous Ant System (HAS) and heterogeneous Max-Min Ant System (HMMAS). A comparison of the proposed approach against that of the base algorithm applied to several TSP instances is also presented in this chapter.

The heterogeneous approach is extended in Chapter 4 by introducing a heterogeneous Max-Min Ant System randomly drawn from a Gaussian distribution with a predefined mean and standard deviation known as GHMMAS. The chapter highlights the improved performance of GHMMAS over HMMAS and MMAS when tested on several medium-sized TSP instances. In addition to that, GHMMAS is applied to the PCB holes drilling problem and compared against MMPAS, a heterogeneous algorithm based on recruitment learning of ants.

Chapter 5 introduces and discusses a hybrid algorithm between ACO and GA with the implementation of heterogeneous adaptive Max-Min Ant System termed HAACO. The algorithm adapts the α and β parameters of the ACO population over time in order to achieve instance-optimized parameter settings. The approach is tested on several TSP instances and compared against two state-of-the-art hybrid approaches that suggest the proposed approach has a better performance in most of the TSP instances applied.

The thesis concludes by summarizing the heterogeneous approach as well as the results presented along with the significance of the research work in Chapter 6. Lastly, a future direction for the heterogeneity approach is suggested.

1.7 List of Publication

Part of the contributions in this study have appeared in a number of peer-reviewed international conferences while a revised journal paper has been resubmitted to Applied Soft Computing. The list of publications are as follows:

1. A. T. I. Fayeez, E. Keedwell, and M. Collett, “H-ACO: A Heterogeneous Ant Colony Optimisation Approach with Application to the Travelling Salesman Problem,” in *EA2017*, M. (Eds. . Lutton, E., Legrand, P., Parrend, P., Monmarché, N., Schoenauer, Ed. Paris, France: Springer International Publishing, 2017, pp. 149–162.
2. A. T. I. Fayeez, E. Keedwell, and M. Collett. “Investigating Behavioural Diversity via Gaussian Heterogeneous Ant Colony Optimization for Combinatorial Optimization Problems”, In Proceedings of the 2nd International Conference on Advances in Artificial Intelligence (ICAAI 2018). Association for Computing Machinery, New York, NY, USA, 2018, pp. 46–50. DOI:https://doi.org/10.1145/3292448.3292459.
3. A. T. I. Fayeez, E. Keedwell, and M. Collett. “Heterogeneous Adaptive Ant Colony Optimization with 3-Opt Local Search for Travelling Salesman Problem” – submitted after revision to Applied Soft Computing.

Chapter 2 Literature Review

This chapter presents the concept of emergent behaviour in nature followed by the idea of ACO from the viewpoint of real ants to artificial ants. A review of the conventional ACO algorithms as well as the parameter adaptation techniques mainly used in ACO are elaborated by defining the general framework of ACO and parameter setting in ACO. Apart from ACO, Genetic Algorithm (GA) is also discussed briefly in this chapter that provides an insight into the framework that is used in Chapter 5. Meanwhile, Traveling Salesman Problem (TSP), which is a well-known combinatorial optimization problem, is applied in this study to gauge the performance of the proposed approaches hence briefly discussed in this chapter. In addition to this, several performance metrics typically used in determining the efficiency of an optimization algorithm are discussed such as the branching factor that plays an important role in detecting stagnation behaviour in ACO. Lastly, a brief explanation of homogeneous and heterogeneous behaviours of social insects at the individual level and multi-colony level will be concluding this chapter as this acts as an inspiration for this thesis.

2.1 Emergent Behaviour

Nature has always fascinated researchers especially in how social insects such as ant colonies, a swarm of bees, flock of birds or school of fish, which are normally considered as less-intelligent and simple agents, effectively and efficiently solve complex problems although individually each agent has limited ability. Interestingly, these agents exhibit no central control but collectively and co-operatively able to solve the problem in a self-organized manner. These social insects usually display social coherence although the individual behaviour of each agent suggests the presence of stochasticity. This is known as the emergent behaviour which is a result of multiple synergistic interactions between agents or with the search landscape that indirectly creates this complex behaviour. Generally, emergent behaviour can be observed in various natural phenomena from biological to physics domains. This creates a synergistic effect on the swarm while tackling the problem in hand. The behaviour allows the researchers to study and develop algorithms under a field of study aptly named the nature-inspired metaheuristic algorithms. The word 'Meta' refers to high-level methodology and heuristics means finding new strategies in solving a problem. Therefore, metaheuristics refer to the complex level strategies developed to address complex level problems. A brief description of metaheuristic methods is given in Figure 2.1 which illustrates the different methodologies. Swarm intelligence [32], which is within the artificial intelligence field, consists of several computational methods inspired by the intelligent behaviours of a group of individuals, who without any form of centralized control, managed to collectively solve complex problems. The swarm intelligence algorithms are based on the local interactions of relatively simple agents that collectively achieve certain goals globally. The local interactions between these artificial agents and the environment lead to emergent behaviour that enables the individual agents to solve various NP-hard optimization problems. Prominent examples of swarm intelligence algorithms are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) although many others exist. Most swarm intelligence algorithms deploy homogeneous individuals due to simplicity in implementation thus lower the possibility of programming errors.

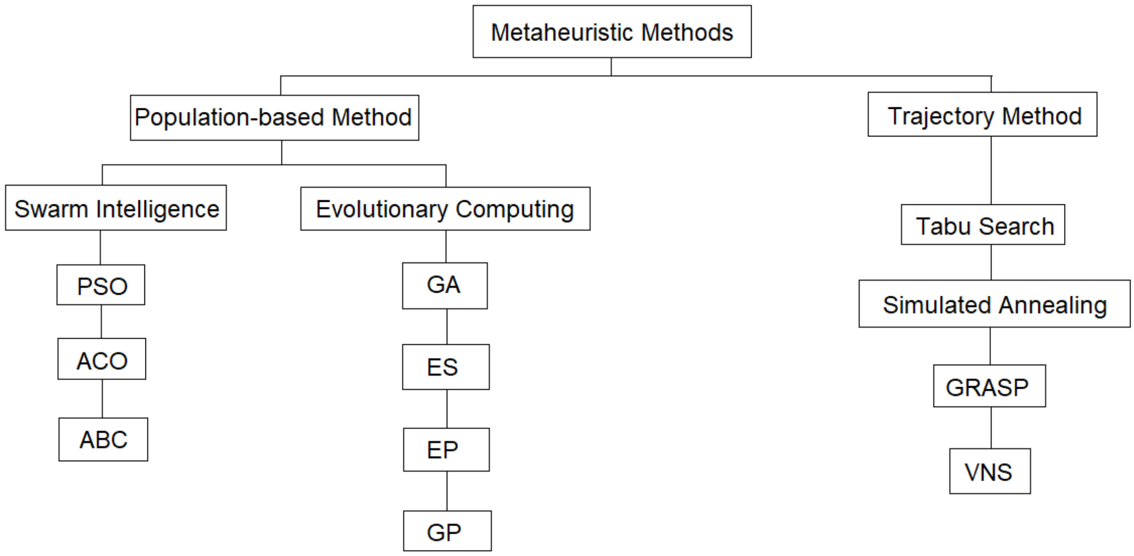


Figure 2.1: Metaheuristic categories

However, it is also possible to initialize the swarm as a heterogeneous colony which consists of artificial agents with individual ‘behavioural traits’. As an example, some insects such as ant colonies exhibit heterogeneous behaviour where the individuals may differ in morphological characteristics as well as their function in a colony. For that instance, a soldier ant might be stronger compared to normal worker ant while worker ants might have different job scopes such as nest maintenance, foraging for food and many more. The individual ‘behavioural trait’ contributes to the emergent, colony-level behaviour in the ant colonies that in turn allows the colony to self-organize and collectively solve problems. Correspondingly, the heterogeneous concept can also be implemented in swarm intelligence algorithms that can create a diverse population of agents with their own perspective while tackling the search landscape [3] [4].

Furthermore, the adaptive nature of social insects is also considered as an emergent behaviour where in this case, ant colonies can intelligently adapt to the change in internal or external environments by adapting the number of workers engaging in different tasks. In the state of emergency such as attack to the nest, the ant colonies can mobilize the workers to focus on the urgent matter before returning to the normal state. This is achievable due to the threshold in real-ants relating to decision-making without the need for central control [33]. Interestingly, the adaptation can also be advantageous to the performance of the swarm intelligence algorithms by enabling the algorithms to learn and adapt to the current state of the search process concerning stagnation or exploration-exploitation phases hence creating a robust, flexible and scalable system.

2.2 Real ants to artificial ants

Ants are considered social insects in the form of colonies that can be in the range of several dozen individual ants to highly organized colonies. These colonies occupy very large territories and vary in behaviours, caste, or job tasks. Ant colonies communicate between individuals and this behaviour is similar to human societies. Therefore, ant behaviours have become an inspiration and a popular subject of research especially their ability to solve complex problems such as foraging for food. Foraging ants travel distances from their nest searching for food and scent trails help them to manage their way back to their nest. In humid conditions and dangerous terrains, foraging ants may die due to dehydration thus its ability to be able to find the shortest path back to the nest can greatly reduce the risk. Ants find their way back to the nest via incorporating the pheromone trail laying with environmental learning. Pheromone laying is a process of stigmergy that relates to indirect communication between ants. The ants deposit the pheromone on the ground while walking to and from a food source. The ants tend to find the shortest path towards a food source hence the shorter path will have a higher concentration of pheromone. Therefore, other ants will follow the path where the pheromone concentration is higher thus enabling ants to transport food faster and in an efficient way. Foraging in ant colonies can be divided into two which are the purely individual foraging, a process of wandering for food for own’s consumption and foraging with recruitment which is a process of forager ants recruiting fellow ants from its colony to bring back the food to their nest in foraging with recruitment.

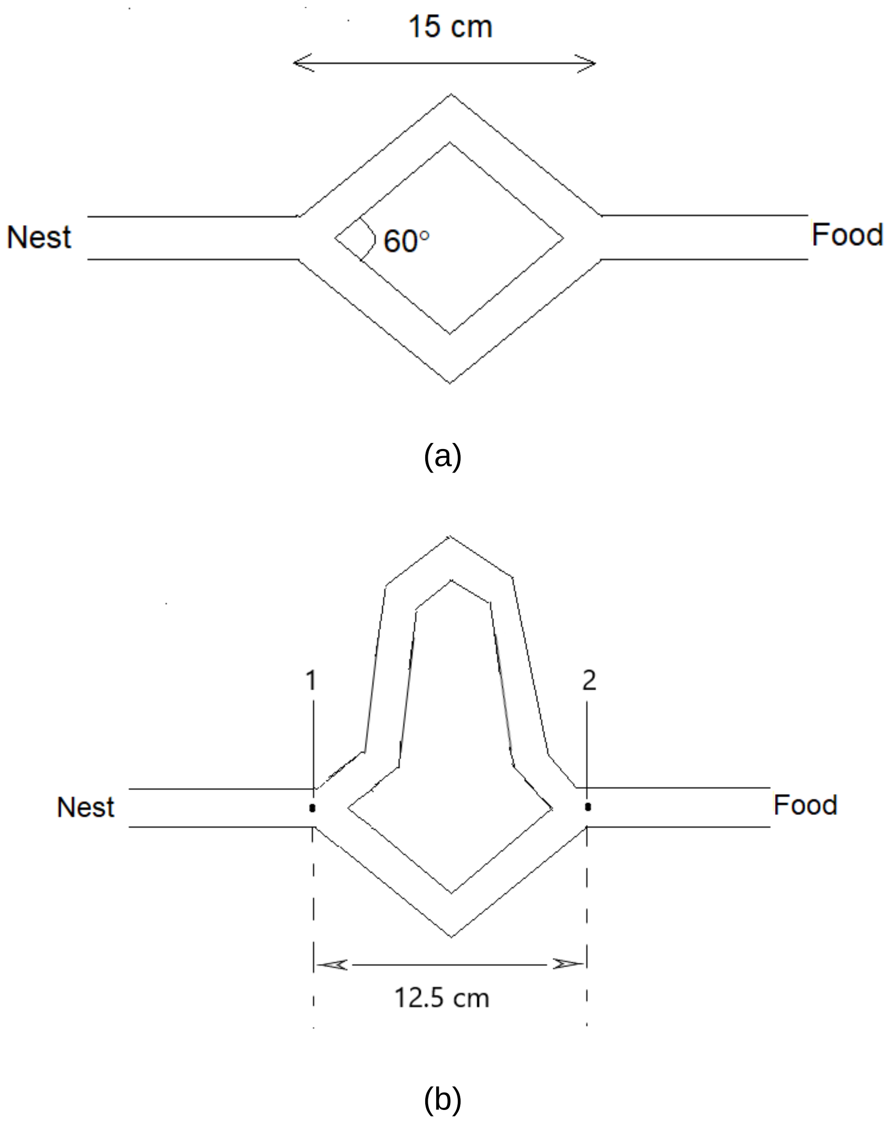


Figure 2.2: Double bridge experimental setup (a) Same length branches [34] (b) Different length branches [35].

The nature of real ants is that they will wander randomly unless pheromone trails are found on the ground and highly likely to follow one while reinforcing the trail in the process. Pheromone is a chemical substance that is deposited by ants while they travel along a certain path i.e the path from nest to the food source and back to the nest while transporting the food. Deneubourg et al [34] conducted the double bridge experiment with equal length branches from nest to the food source to study the pheromone laying concept of the ants. A colony of Argentine ants was used in the experiment where the ants were connected to food source via equal lengths of a double bridge as shown in Figure 2.2a. They concluded that at the beginning of the experiment, ants randomly explore both paths towards the food source and lay pheromone while doing so. However, the ants converge onto one particular path after some time due to the random probabilities thus causing one path to have higher pheromone concentration compared to the other. In addition to this, another variant of the double bridge experiment was conducted by Goss et al [35] with one path longer than the other as shown in Figure 2.2 (b). The experimental result shows that the shorter path leads the ants faster from nest to food source thus receiving a higher amount of pheromone thus increasing the probability of other ants choosing the shorter path. This breakthrough is the inspiration for the implementation of ACO algorithms such as Ant System (AS), Rank-based Ant System (AS_{rank}) and many more.

2.3 Ant Colony Optimization

ACO is an optimization algorithm that is inspired by the foraging behaviour of real ants from nest to food source and back to the nest. This foraging behaviour is the underlying concept of interactions between agents in ACO that conforms to the emergent behaviour method. The artificial agents communicate with each other by replicating the pheromone deposition mechanism of real ants. The ants use this information to act as a guide to the food source without requiring any physical communication between them. Therefore, the basic principle of an ACO algorithm is to find the optimized path in a connected graph of a problem via indirect communication using artificial pheromone deposition.

The conclusion from the double bridge experiments suggests that autocatalytic behaviour can be used to design a system with artificial ants to solve various optimization problems. This is due to the fact that the ants are able to locate the shortest path collectively even though they are simple agents individually. Interestingly, the real ants also show exploratory and exploitative behaviours respectively where initially the ants explore both short and long paths and later converge to the shortest path after several iterations [8]. Additionally, the ants can adapt to the changes in the pheromone landscape due to pheromone evaporation and the process of transporting food from the food source to the nest which is an iterative process. It can be seen that the deposition of the pheromone trail on the path traversed can be considered as a memorization process as well as this allows the ant to mark the path they follow while foraging. All these properties support the idea of designing an optimization algorithm based on the foraging behaviour of real ants.

In order to fulfil the aforementioned idea, any problem to be solved must be converted to a fully connected graph with several edges and nodes connecting them as shown in Figure 2.3. This is followed by allowing the ants to walk from the nodes (in this case, the nest) toward the destination (food source) as in the forward mode and travel back to the nest as in the backward mode. The artificial ants can “smell” the artificial pheromone on the path and also deposit pheromone as they travel back to the nest as shown in the backward mode. The pheromone intensity is inversely proportional to the solution found where the shorter the path, the more pheromone deposited by the ant that traversed the path. Hence, the amount of pheromone deposited depends on the quality of the solution.

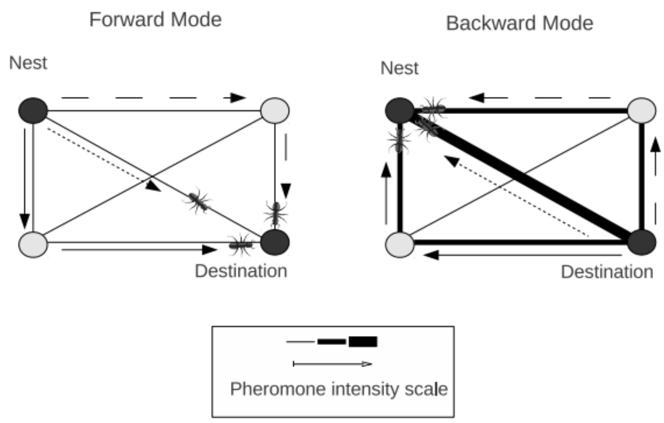


Figure 2.3: Example of a fully connected graph [36]

This may look simple and straightforward but the complexity increases when the fully connected graph increases in size with hundreds or thousands of possible paths. If the artificial ants depict the same behaviour of the real ants without any modification, then there are chances of the artificial ants getting trapped in an infinite loop thus reinforcing the same tour repeatedly. Therefore, artificial ants are allowed to remember the nodes they have visited thus preventing the ants to revisit a node twice. This at the same time allows the ants to evaluate the length of the path found as well as re-trace the path for pheromone deposition.

In addition to the pheromone trail, the artificial ants also make use of heuristic information in their decision-making. The heuristic information contains the difference of each node in the fully connected graph. Therefore, the heuristic value is inversely proportional to the distance between each node. Traveling salesman problem (discussed later) contains the distance information which allows the algorithm to pre-compute the heuristic information. Lastly, pheromone evaporation does exist in the ACO framework except that the evaporation rate is much faster compared to that in real ants. Pheromone evaporation, usually on all paths, is implemented to encourage exploration of new search areas while also preventing the algorithm from premature convergence to local optima.

2.4 General Framework of ACO

ACO is specially designed to solve NP-hard combinatorial optimization problems [10]. The general ACO framework consists of four main phases with one optional phase as shown in Algorithm 2.1. Once the problem instance has been loaded into the algorithm and the main parameters are initialized, a population of ants construct their solutions and pheromone trails are updated until the stopping criterion is met. The additional step is applying the local search procedure which is usually used to improve the solution found. This step is optional as it is best used especially when to solve large instances as it can further improve the solution found by performing neighbourhood search. Each ant starts with an empty solution and constructs the solution by adding nodes or components that it has traversed until all nodes have been visited. The choice of the node to be visited is based on the probabilistic rule that consists of pheromone trail and heuristic information. Each information component has a coefficient to create a bias toward pheromone or heuristic during the decision-making. Each variant of ACO has its own choice of the coefficient values.

Begin

Load the problem instance;

Initialization;

While termination criterion not met **do**

Ants construct solution;

Local search procedure (optional);

Pheromone trail update;

End

End

Algorithm 2.1: ACO General Framework

The pheromone trail update procedure allows the reinforcement of pheromone on good solutions while bad solutions will see a reduction in the pheromone values. This is to mimic the autocatalytic behaviour from real ants to artificial ants because of the higher the pheromone on the components of good solutions, the higher the probability of ants choosing that particular component while constructing their solution. Firstly, the pheromone is evaporated on all the trails globally to prevent stagnation followed by pheromone deposition on the trail of the good solution. The amount of pheromone deposited is proportional to the quality of the solution found. Lastly, the local search procedure is an algorithm that is usually used to improve on the solution found and it is optional for researchers to implement this procedure. However, it must be noted that function evaluations from the local search procedure should be taken into account together with the main function in determining the total function evaluation count.

2.5 Conventional ACO

ACO falls under the category of constructive heuristic because of its nature in solving a problem by starting with an empty solution and constructively extends the solution until a complete solution is achieved. ACO is well suited for combinatorial optimization problems due to this procedure [37]. Table 2.1 shows the successful conventional ACO algorithms where the first ant algorithm was introduced by Dorigo et al [38] as a stochastic search algorithm for the well-known traveling salesman problem (TSP). Three key aspects of AS are the ant cycle, ant density and ant quantity where the difference between these variants of AS algorithms is that ant density and ant quantity's pheromone are updated after every move to the adjacent city but ant cycle's pheromone is updated after the candidate solution is built. On top of that, the amount of pheromone deposited in the ant cycle corresponds with the quality of its solution where the better the solution, the higher the amount of pheromone deposited in the tour (See equation 3). Encouraging results in the ant cycle as compared to ant density and ant quantity caused the latter algorithms to be abandoned. AS was developed with the aim of tackling the Travelling Salesman Problem (TSP) which is a combinatorial optimization problem that itself has attracted extensive research [39]. AS starts pheromone initialization where all the edges in the graph were given a certain amount of initial pheromone, τ_0. Then, the artificial ants randomly deployed on cities and begin to construct their tour iteratively. The ants maintain a *tabu list* which contains the cities it has visited thus preventing a city to be visited twice therefore the only unvisited city can be chosen in the next iteration. This is done by using a probabilistic state transition rule also known as random-proportional rule (P_{ij}^k) in which ants prefer to move to cities which are connected by short edges with a high amount of pheromone [38] as shown in equation 1 where τ_{ij} and η_{ij} are pheromone trail intensity on edge (*i, j*) and heuristic information of edge (*i, j*) respectively.

Table 2.1: Successful Conventional ACO

Num	Year	Algorithm	Author
1	1991	Ant System (A.S)	Dorigo et al [40][41][6]
2	1992	Elitist Ant System	Dorigo et al [6]
3	1995	Ant Q	Gambardella et al [42]
4	1997	Ant Colony System (ACS)	Gambardella et al [7]
5	1996	Max-Min A.S	Stützle et al [43][5]
6	1999	Rank-based A.S	Bullnheimer et al [44]

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} \text{ if } j \in N_i^k \quad (1)$$

The heuristic information of TSP is calculated by using the equation $1/d_{ij}$ where d_{ij} is the distance between city *i* and *j*. α and β are two relative parameters that determine the weight of the pheromone trail and heuristic information. N_i^k are the unvisited cities of ant *k* when it is at city *i*. Once all the ants have built their solution, the pheromone is then evaporated before pheromone is deposited. This phenomenon is described in equation 2.

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \sum_{\mathrm{k}=1}^{m} \Delta\tau_{ij}^{k} \quad (2)$$

τ_{ij} is the amount of pheromone on edge (i,j) while ρ is the evaporation rate which must be set between $0 < \rho < 1$ to avoid an unlimited accumulation of pheromone that can lead to stagnation and early convergence. m is the number of ants deployed in the search space while k is associated with ant k. $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid on edge (i,j) by ant k which is given by equation 3.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if ant k uses edge (i, j) in its tour} \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Q is a constant value usually set to 1 while L_k is the tour length of ant k. Equation 2 and 3 suggest that all edges will have its pheromone evaporated or decreased by a small amount before the edges are reinforced based on the solution found by the respective ants. Good solutions (in the case of TSP, the shorter path) will receive higher pheromone thus yielding greater chances of that path being chosen in the next iteration. Edges not in any solution will have their pheromone reduced to a very low level (possibly zero) significantly reducing the probability of being chosen in the future. In terms of performance, AS achieved excellent performance over small TSP instances namely Oliver30.tsp and eil51.tsp but the performance deteriorated when tested on instances larger than 75 cities [39] [40].

The first improvement over AS was made by Dorigo et al [6] with the introduction of elitist AS (EAS). This strategy strongly reinforces the global best solution (S^{gb}) by reinforcing the value of pheromone on the edges of this solution. This is an exploitative measure where the ants are directed toward the global best solution to find better solutions. However, even though the use of a suitable number of elitist ants can enhance the performance of AS, the drawback of this method is that improper number of elitist ants can cause premature convergence to suboptimal tours. In addition to that, determining the suitable number of ants is a tedious parameter tuning task that is computationally extensive and exhaustive. Another well-known improvement over AS is the rank-based AS (RAS) by Bullnheimer et al [44] where the ants are ranked according to the solution quality. Only a certain number of best ants are considered for pheromone update where the amount of pheromone deposited based on their rank. The possible drawback of RAS is the sorting overhead the approach introduced in ranking the ants as the sorting procedure is known to be a computationally exhaustive procedure hence increases the algorithms time complexity.

Gambardella et al proposed the Q-Learning method to ACO called the Ant-Q [42] where AS as the base algorithm. Even though the Ant-Q algorithm produced very good performance when tested on several TSP instances, the algorithm itself is too complex for understanding. Therefore, the authors proposed the Ant Colony System (ACS) [7] with improvement, especially when tested on larger TSP instances. ACS deploys a slightly different approach compared to other conventional ACO algorithms that use AS as the base algorithm. Firstly, the ants used the pseudo-random proportional rule (equation 4) to choose its next city, *j* when at city *i,* by depending on a random variable, q which is randomly distributed over [0,1] and a pre-determined parameter, q_0. This directly balances the exploration of new search areas and the exploitation of prior information. Secondly, the global pheromone update rule (equation 5) only applies to the tour of the best ant. This, coupled with the pseudo-random rule, helps to make the search space more directed. The update process takes place only after all the ants have constructed their solutions. The use of the candidate list was also suggested when ACS applied to large TSP instances as this can reduce the time for the algorithm to locate good solutions. The candidate list consists of a fixed number of preferred destinations for each decision point.

$$\mathrm{j} = \begin{cases} max_{l \in N_i^k}\{[\tau_{il}].[\eta_{il}]^{\beta}\} & \text{if } \mathrm{q} < \mathrm{q}_0 \\ J \text{ using Eq 1} & \text{Otherwise} \end{cases} \tag{4}$$

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \rho\, \Delta\tau_{\mathrm{ij}}^{best} \tag{5}$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{\mathrm{L}_{best}} & \text{if } (i,j) \text{ part of the best tour} \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

0 < α < 1 is the pheromone decay parameter and L_{best} is the length of the global best tour. Thirdly, ACS applies local pheromone update rule (equation 7) to the edges visited by the ants during the solution construction phase. This is to simulate exploration among the ants in the search space.

$$\tau_{ij} \leftarrow (1-\xi).\tau_{ij} + \xi\tau_0 \tag{7}$$

where 0< ξ <1 is the pheromone decay coefficient in local pheromone update.

Stützle and Hoos [43][5] introduced another improved variant of AS known as the Max-Min Ant System (MMAS) which still uses the probabilistic rule as in AS for constructing the fitness solution. However, the first contribution of MMAS is limiting the pheromone trail value to a maximum, τ_{max} or minimum, τ_{min} to prevent early convergence to a sub-optimal solution and also improves exploration. Secondly, only the best ant in every iteration is allowed to deposit pheromone based on the tour length using equation 8. Lastly, the trail-smoothing mechanism is introduced to overcome stagnation in the algorithm which is an undesirable condition where all ants quickly converged to a single solution and repeat the same tours in every iteration thus affecting the exploration process of the search landscape.

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \Delta\tau_{ij}^{best} \tag{8}$$

τ_{ij}^{best} is calculated using equation 6 where L_{gb} can either be global best, iteration-best or alternate between these two. Trail limits as in equation 9 are implemented in MMAS to promote exploration by preventing certain paths from accumulating a high amount of pheromone that will cause this path to be chosen always.

$$\tau_{ij} = \begin{cases} \tau_{max} & \text{if } \tau_{ij} > \tau_{max} \\ \tau_{min} & \text{if } \tau_{ij} < \tau_{min} \\ \tau_{ij} & \text{Otherwise} \end{cases} \tag{9}$$

This ensures the pheromone landscape will always be within the trail limits, $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$ while τ_{max} and τ_{min} will be recalculated every time a new global best tour is found using the equations below.

$$\tau_{max} = \frac{1}{\rho \times L_{best}} \tag{10}$$

$$\tau_{min} = \frac{\tau_{max} \times (1-\sqrt[n]{P_{best}})}{(avg-1) \times \sqrt[n]{P_{best}}} \tag{11}$$

where avg is the average number of choices the ant has at every decision point while P_{best} is a parameter used in MMAS. Lastly, the pheromone trail-smoothing is used when convergence is detected which in turn uses the λ-branching factor (explained later).

2.6 Parameter Settings in ACO

The objective of using a metaheuristic approach is to obtain acceptable solutions in a short period. This is because of the difficulty in solving combinatorial optimization problems and it requires a very long time to solve to optimality hence the use of metaheuristics. The ability to tune the parameters of the metaheuristics in part plays an important role in achieving a fast, acceptable result and the results are very dependent on the parameter settings too. However, parameter tuning is a non-trivial task that requires a deep understanding of the algorithm in use as well as the problem being solved. Parameter tuning is a computationally exhaustive and extensive process and it is impossible to tune the parameters to every problem instance being solved. Similarly, the performance of ACO algorithms are parameter dependant and strongly influenced by the parameter settings [45].

Two main parameters that influence the probabilistic rule are the α and β which are the coefficients that control the relative influence of pheromone trail and heuristic information respectively. Determining the optimal α and β values can have a huge effect on the performance of the ACO algorithm while randomly sampling of the α and β parameters allows the algorithm to comprise of several co-existing strategies bundled up in the algorithm to tackle the optimization problem. As an example, if $\alpha = 0$, this renders the algorithm to act in a greedy manner thus the likelihood of choosing the closest cities. Meanwhile, if $\beta = 0$, only the pheromone trail is taken into consideration in the probabilistic rule without any heuristic information. This leads to poor performance while $\alpha > 1$ is said to cause stagnation behaviour in ACO where all ants construct the same, suboptimal tour. Therefore, parameter setting in ACO is important and usually, researchers apply the parameter suggestions by Dorigo et al [37, p. 71] while there are researchers that perform extensive experiment to determine optimal parameter setting based on the problem they try to solve [46][47][48][49]. This is the main reason why both the α and β parameters were chosen as the main candidates for the heterogeneous approach even though other parameters can be considered for future such as the pheromone evaporation coefficient, ρ.

There are several methods to tune the parameters which can be divided into two categories that are the analytical and empirical methods [50]. The analytical approach applies the mathematical formulations for the recommendations of the parameter settings in ACO [51][52] while the empirical method is conducted by trial-and-error or systematic approach [16]. However, empirical testing is computationally exhaustive and expensive as it requires lengthy computational time even on supercomputers [49]. A possible approach to the parameter exploration is the 'offline' method where optimal parameter settings are explored using several test cases or problem instances. The resulting optimal parameter settings are then implemented during the actual test run of the optimization problem [16][53]. The advantage of this approach is that firstly, this ensures the robustness of the algorithm to parameter settings and secondly, one does not need to find the exact optimal parameter settings in ACO in respect to the α and β values as long as the parameters are within the optimal settings will be enough to ensure good algorithmic performance [16]. Referring to the parameter suggestions by Dorigo et al [38][6], it can be seen that the optimal parameter landscape is reasonably large thus easy to find good to optimal parameter settings via the offline method.

Another approach is the 'online' parameter control where the parameters change over time while the algorithm solves the problem [45][54]. One of the advantages of this approach is the ability to alternate between exploration and exploitation phases, unlike conventional ACO. However, this approach requires additional parameters to be introduced into the algorithm. There are several techniques in the 'online' parameter control approach which are known as the pre-scheduled, adaptive, self-adaptive and search-based adaptation [45]. The pre-scheduled or deterministic technique involves changing the parameters as per some pre-defined rules such as function evaluations, iterations or computational time. This technique is known as the simplest of all but it involves introducing new parameter(s) to realize the approach. In contrast, the adaptive technique accepts feedback from the optimization process and adapts to the current state of the process by changing its parameters. As an example, suppose a difference between fitness solution is set a threshold value and if it is met, then the algorithm changes its parameters according to a set of rules [55]. This technique still requires a pre-defined rule in order to activate the parameter variation. The approach described in Chapter 5 falls under the self-adaptation category where the parameters are encoded into genotype-like solutions in the parameter space. The algorithm searches the solution and parameter space simultaneously to locate instance-optimal parameter settings. The self-adaptive technique does not require any predefined rules but usually involves a hybridization with the Genetic Algorithm (GA) where the parameters are mapped into genotypes and may involve selection, recombination and mutation to evolve the population of artificial ants [56]. However, a minor modification in Chapter 5 (explained in detail in Chapter 5) is to perform parameter adaptation every 5 iterations to allow enough information on the pheromone landscape to decide on the best and worst ant for selection and replacement. Readers are directed to [16][54][55][57][58] for a detailed review approaches to the setting of parameters.

2.7 Genetic Algorithm

The Genetic Algorithm (GA) is another approach for optimization that was initially proposed by John Holland in 1975 [41] to model and analyse the organic adaptation and evolution abilities but generated more interest in the optimization field. The concept of GA is that individuals, usually the fittest, are chosen to produce offspring for next generation. Candidate solutions are represented as genomes, which can be encoded in various ways such as binary, integer or floating-point representations. The population in a GA is usually created at random and is followed by the selection of the parents that are required to produce offspring. The selection process is based on the fitness score of each individual according to the fitness function where the higher the fitness function, the higher the fitness score hence the higher the chances of the individual to be selected for reproduction. The selected parents usually undergo crossover in order to create the offspring. Interestingly, crossover, which involves recombination of genetic information of both parents to produce offspring, is comparable to a certain aspect to the sexual reproduction in biology. On the other hand, the offspring undergoes mutation operator that alters the value of one or several components of the genome in order to maintain population diversity. The most common mutation operator used is the uniform and Gaussian mutation. Again, a minor modification on how the mutation operator is applied in Chapter 5 will be explained later. One example of crossover is the single-point crossover that cuts the solutions at a random position before swapping the parents with the two parts to create two offspring. However, various crossover operators can be applied in GA. It should be noted that the most effective crossover approach is the 2-point crossover while it has been reported that there is a reduction in the performance of the GA when the number of crossover points is increased [42].

The alteration of values of genome introduces new information into the population and can be considered as a random variation. After the selection procedure and mutation process, the member in the population to be replaced will be chosen. There are several ways to implement the replacement procedure such as by ranking and selecting the worst members for replacement or in a more complex method that is by a fitness proportionate selection that gives each solution a chance proportional to their fitness.

2.8 Combinatorial Optimization Problem

The proposed approach is mainly implemented on the Traveling Salesman Problem (TSP) [8] which is one of the most fundamental and popular combinatorial optimization problem (COP). The principal objective of the TSP is to find the shortest tour of a number of cities by determining the order of the cities to be visited where each city must be visited once and only once before returning to the start city. The TSP can be translated into various real-world problems such as the delivery service where the postman must find the shortest route in completing his task. Other variants that take inspiration from the TSP are vehicle routing problem (VRP) [59], printed circuit board (PCB) drilling and many more. The standard formulation of TSP is as follows:

$$f(x) = \min \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} \Psi_{ij} \quad (12)$$

$$\Psi_{ij} = \begin{cases} 1 & \text{if the arc (i, j) is in the tour} \\ 0 & \text{Otherwise} \end{cases} \quad (13)$$

where $\Psi_{ij} \epsilon \{0,1\}$, n is the number of cities and d_{ij} is the distance between i and j.

TSP can be divided into two main categories which are symmetric and asymmetric TSP. In the symmetric TSP, a problem instance is represented by a connected, undirected graph, $G = (V, E)$ where V represents the nodes while E represents the edges connecting the nodes. Each edge has a distance value associated, e_{ij} while the objective is to find the Hamiltonian cycle of the shortest tour. In the Euclidean TSP, each node has its coordinates hence e_{ij} is the straight-line distance between i and j. Therefore, e_{ij} can be calculated using the following equation. Asymmetric TSP will not be discussed here as all the instances used in this study are symmetric TSP instances.

$$e_{ij} = \sqrt[2]{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{14}$$

On the other hand, PCB is a more complex variant of TSP and can be considered as more realistic real-world scenario. PCB holes drilling is an important process in order to connect the conductor of one layer to the conductor of another layer. Holes can be of different diameters and each time when the diameter changes, the head of the machine has to move to the toolbox to change the drilling equipment which takes time. Thus drilling process can be considered as a series of TSPs in which 'cities' are the initial position of drill and 'distance' is referred to the time taken by machine in moving from one position to the other [60]. The motivation in using TSP to test the performance of the proposed approach is because of the huge number of resources as well as the availability of the library with known optimal solutions for a subset of TSP instances thus enabling this study to focus on the development of the metaheuristic rather than the formulation of the problem. In addition to this, a comparison can be carried out against state-of-the-art algorithms that tested their approach on TSP and any improvement in finding better tours can be related to the performance of the algorithm when compared using the same problem instance. Therefore, if the algorithm is tested on lesser-known problem instances, there might be very limited studies conducted that can be used for comparison purposes.

2.9 Performance Metrics for Optimization Algorithm

Experimental research in evolutionary optimization is common, especially in the case of nature-inspired algorithms that often conduct empirical assessments. By the stochastic nature of the optimization algorithms, a certain number of trials must be conducted to gain sufficient experimental data then only performance measures can be performed and based on some statistics. However, it can be difficult performing experiments with a stochastic optimization algorithm and presenting the results in a way that will satisfy high scientific standards. It is widely accepted that empirical analyses are an important aspect of stochastic optimization algorithms especially for metaheuristics that are complex. This section will discuss types of performance measurements or metrics that are usually used in an optimization algorithm that relates to the metrics used in subsequent chapters. There are no single performance metrics that are considered ‘the best’ while in some cases, researchers could not agree on this aspect of reporting empirical results [61][62]. We divide the performance metrics into two categories which are the optimum-based and behaviour-based performance metrics [36].

2.9.1 Optimum-based Performance Metrics

It is important to consider the quality of solutions in the empirical analysis of stochastic optimization algorithms since these algorithms usually do not guarantee optimal solutions. In the case of experimental research, time is the resource that is usually considered. This notion is very subjective as the processing capabilities and platforms used may vary from one researcher to another hence rendering it as unsuitable. However, time which is often expressed indirectly by the number of iterations, the number of generated solutions, and the number of function evaluations, can be considered as useful performance metrics. The most common approach is to restrict the computational resources and to observe the solution quality. The computational resources, in this case, including but not limited to the processing elements, memory and elapsed time.

Optimization algorithms in evolutionary computation are stochastic and produce different solutions on different executions. The performance measurement of the stochastic optimization algorithms must follow or apply the common practice by researchers which is conducting multiple trials of an algorithm. The best solution obtained in multiple executions of an algorithm is often used as a measure for performance as recommended by Eiben and Jelasity [61]. The authors stated that this particular performance metric is important especially when the optimum solution is unknown such as in the cases of optimizing real-world problems. This was supported by Birattari and Dorigo [62]. Using the best solutions out of a predefined number of attempts inside statistical tests that can handle individual values might be acceptable through the use of e.g. the Wilcoxon Rank Sum Test used in this study. It is not important if in a particular experiment one algorithm yields a better "best solution" than the other. Instead, it is preferred that this behaviour is consistent in many experiments and confirmed as statistically significant.

In addition, the arithmetic mean is also used to measure the performance of an algorithm [36][63]. It allows for the reproducibility of results, provided that it is possible to measure the arithmetic mean in a reliable way and a suitably large sample is used [63]. The arithmetic mean can be defined as the average of the best solutions of several trials. These metrics then can be used to compare the performance of several algorithms. Another performance metric is the worst solution found in a number of trials although it is seldom used for comparison purposes. However, this value shows how far the solution is from the known optimal solution for problems such as TSP. Most of the research involving conventional ACO algorithms used best, average and worst solutions across multiple trials as their performance metrics hence also used in this study. These values can also be used to create a ranking-based performance comparison as in Chapter 5 which can also be extended to conduct the Friedman statistical test.

For problems where the optimal solution are known such as TSP, one can easily define a success criterion and calculate the success rate (SR) metric which can be defined as how close the solution is from the known optimal solution in percentage. The higher the success rate, the closer the solution is to the optimal solution as introduced in Chapter 3 and 4. This value acts as an indicator of the probability of the algorithm achieving the optimal performance over a certain number of trials. Lastly, the quantiles representation in boxplots are very useful in illustrating and comparing the performance of several algorithms.

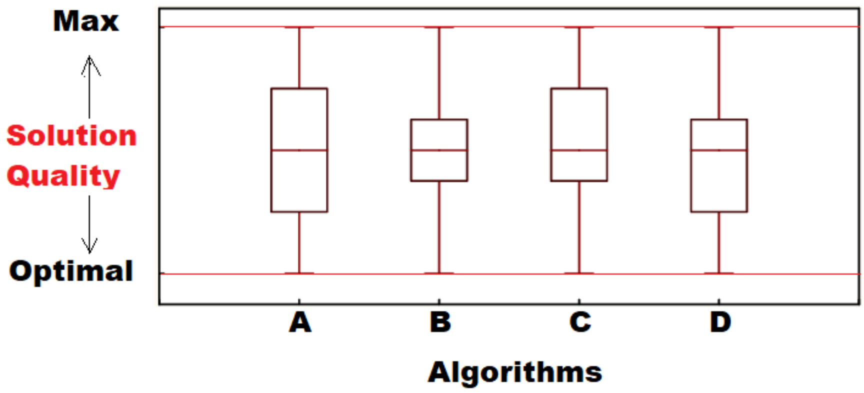


Figure 2.4: Example of boxplots with the same median but different quartile values [63].

As an example, Figure 2.4 illustrates boxplots with the same median but different quartile values where all algorithms managed to find the optimal solution and at the same time have the same maximal value over several trials. Therefore, in this case, the interquartile values of the algorithms can determine, which of these algorithms has a more desirable performance characteristic.

2.9.2 Behaviour-based Performance Metrics

The behaviour-based performance metrics allow the researchers to measure the behaviour of the algorithm during the search process and to estimate when the solutions have sufficiently converged in fitness value or construction. A well-known metric is the standard deviation of the solution set in every iteration [64].

This can be considered as one of the important performance metrics especially in population-based algorithms and has the capability to illustrate to some extent, the exploration and exploitation phases of the algorithm as well as the occurrence of stagnation behaviour in the population. A high standard deviation represents that solutions are more spread out thus indicating the exploration phase while a low standard deviation indicates the spread is smaller hence the algorithm is locating solutions much closer to the optimal thus illustrating the exploitation phase. Otherwise, a standard deviation of zero indicates stagnation behaviour in the ants' colony that means all ants repeat the same tour in every iteration i.e. the algorithm has converged, which can be to a good or bad solution.

Next, Gambardella and Dorigo [65] introduced the average lambda branching factor (λ-branching factor) which uses information from the pheromone trail values thus allowing the researcher to track the algorithm's behaviour as the search progresses. This technique measures the exploration and exploitation of the ants. Assume *ph_max(i,j)* and *ph_min(i, j)* are the maximum and minimum pheromone of all the edges that exit from node i. The branching factor can be represented by the number of edges from node i that is greater than $\lambda.d$ + *ph_min(i, j)* where $0 < \lambda < 1$ and d is the difference between maximum and minimum pheromone amount that exit node i. The branching factor is also able to indicate that if the algorithm has achieved convergence as well as illustrating stagnation behaviour.

Finally, this research work (in subsequent chapters) is the first to propose a metric of the distribution of the ants in the parameter space as an indicator of the heterogeneity effects. In addition to this, 3d graphs and convex hulls are used to illustrate the ants' distribution over time thus indicating the algorithm's robustness toward parameter setting. This also shows the capability of the algorithm to adapt to the changes as the search progress and how the algorithm explores and exploits both the solution space and parameter space concurrently.

2.10 Heterogeneity

Homogeneity or homogeneous population consists of individuals with the same traits or little variation physically or behaviourally. On the contrary, heterogeneity or heterogeneous population comprises of individuals with variation among them. The most obvious is the variation in human beings where generally, we differ in height, weight, skin colour and other traits. It is well known that heterogeneity is ubiquitous in natural systems. Behavioural variation has been observed in social insects by animal behaviour researchers who have been studying the relationship between heterogeneity and population diversity in social insects and how this behavioural variation, especially in social insects, is beneficial to the colony.

Recent studies have shown that intra-colony variations do exist in ant colonies where the ants differ in 'behavioural traits' within the colony [33][28][27][66][67][68][69]. This can be divided into two categories which are variation due to the age and size of the ants [70] and secondly, the behavioural variations such as aggressiveness or choosiness of the ants concerning nest maintenance [69]. Behavioural variation also has been attributed to an increase in colony efficiency and higher colony fitness compared to homogeneous swarm or colony with less behavioural variations [28]. One example of behavioural variation found in ants is the variation in the exploratory behaviour of the workers where some ants might exhibit a higher preference toward exploration compared to others. Both the aggressiveness and exploratory behaviour of the colony are important traits in determining the evolution and efficiency of the colony [71]. The diversity in the population introduced by the intra-colony behavioural variation allows a more efficient task allocation in the division of labour thus indirectly increasing the productivity of the colony.

2.10.1 Heterogeneity in PSO

Typically, swarm intelligence consists of homogeneous agents mainly due to simplicity in implementation [72]. The agents exhibit identical 'behavioural traits' because they are initialized in such a way that they have the same parameter settings. Particle Swarm Optimisation (PSO) [73], a successful and popular swarm intelligence algorithm, is inspired by the social behaviour of animals such as a school of fish, a flock of birds or ant colonies. PSO consists of particles that represent a swarm of candidate solutions that the algorithm will try to search for the optimal solution. Generally, PSO algorithms, as with other swarm intelligence algorithms, are initialized as homogeneous swarms with the particles exhibit identical parameter values. However, heterogeneity has been implemented with success in Particle Swarm Optimisation (PSO) with numerous researches that have been explored and conducted with significant findings. A detailed review of previous work in heterogeneous PSO can be found in [74] although most of the studies are loosely connected to heterogeneity. A more general framework of heterogeneity was firstly modelled by De Oca et al [72] where particles in the PSO algorithm differ either in the neighbourhood size, the model of influence, update rule or update rule parameters. However, the authors were more interested in determining ways to implement heterogeneity in PSO rather than identifying the behaviours that can produce improved performance statistically. Olorunda et al [75] assert that different search behaviours is exhibited by the heterogeneous coevolutionary algorithm which consist of a combination of several evolutionary algorithms that are assigned to each sub-population. However, the overall efficiency of this approach depends on determining optimal sub-population sizes as well as assessing the performance of each sub-algorithm. In another approach, particles were modelled to exhibit distinctive search behaviours sampled from a behavioural pool in the research conducted by Engelbrecht [74]. This allows the swarm to consist of multiple behaviours that include explorative and exploitative particles, among others, to tackle the optimization problem. An improved version of heterogeneous particle swarm optimization (IHPSO) is discussed in [76] and the results showcase that the proposed solution allows the particles to have a different position and velocity update rule that incorporates the particles' best local position and best global position. In addition to this, the decision-making also includes a parameter that determines current fitness proportions of the entire population. The algorithm suggest that heterogeneity introduced in the study is able to converge quickly to good solutions and performs better when compared against other approaches.

Nepomuceno et al [19] implemented an adaptive heterogeneous PSO approach where the heterogeneous particles were allowed to adapt to the changes during the search process. The algorithm keeps track of the success rate of certain behaviours and uses this information when the particle is required to change its behaviour. There are many more heterogeneous PSO algorithms with varying degrees of success and the results indicate that heterogeneity can improve the efficiency and the performance of the algorithm. Therefore, the researches on heterogeneous PSO act as an inspiration for the proposed approach in this study.

2.10.2 Heterogeneity in ACO

Heterogeneity is well studied in PSO as discussed above but there is comparatively less research conducted on heterogeneous ACO. The term ‘behaviour’ in artificial ants is used to refer to the combination of pheromone trail and local heuristics intensity. Heterogeneity in a colony of ants can be set at an individual or colony level where artificial ants with different traits between each other are known as individual-level heterogeneity while sub-colonies of ants with different behaviour for each sub-colony is said to be the latter.

2.10.2.1 Individual-level heterogeneity

Various mechanisms can be used to introduce the heterogeneous approach in ACO. Tsutsui [77] implemented a colony of ants that consist of both donor ants and cunning ants. A partial solution from the donor ant is used by the cunning ant in the next iteration to build its solution. The main reason for deploying this is to speed up convergence and escape from premature convergence. However, determining the right amount of information shared in the solution construction can prove decisive in the performance of the algorithm. In addition to that, over-exploration or exploitation may still exist in the algorithm especially if too little or too much of the partial solution is shared between the solutions. More importantly, the idea of varying the population’s parameters was suggested in [78] where the α and

β values for the whole colony change at every iteration. The values were randomly sampled from a uniform distribution at every iteration. Furthermore, the authors also modified the pheromone deposition and evaporation mechanism to further improve the algorithm in order to escape from local optima. This method introduces heterogeneity into ACO, but the concept lacks an explanation on why and how the parameters change every iteration and how this can improve the performance of the algorithm as well as relation towards the real ant colony.

Lee et al [79] introduced heterogeneous individual ants with different sight, speed and function behaviours for obstacle avoidance in a robotic environment. Although the authors stated that the performance of the proposed approach is better when compared to conventional ACO, they also stressed that there is room for improvement in the proposed approach especially when the main ACO parameters are varied during execution rather than being kept constant. Nugulescu et al [80] reviewed the idea of synthetic genes for artificial ants similar to that of a Genetic Algorithm (GA) approach. The authors suggested several parameters that can be converted from global to local parameters to incorporate the idea. However, the authors did not follow up on their initial idea as there are no published results of a working concept. Chira et al [81][82] discussed the effects of deploying artificial ants with different sensitivity levels to the pheromone trail. The parameter that influences the relative weight of the pheromone trail, α is randomly sampled from a normal distribution with a pre-defined range of 0 to 1. Ants with a low level of pheromone sensitivity (closer to 0) will act as an explorer thus will conduct a random search on the solution landscape while ants with high sensitivity level (closer to 1) will exploit solutions found in order to strongly follow the pheromone trail. This Sensitive Ant Model (SAM) improvised and extends the ACS approach by optimizing the properties which are responsible for inducing heterogeneity in each agent of model which leads to the sustainable search intensification. A similar approach was conducted by Yoshikawa et al [83] who used a *cranky ants* approach that explores paths with a low level of pheromone as opposed to the normal behaviour of standard ACO. This involves modifying the probability rule to include the reciprocal of the pheromone level rather than the pheromone level itself. Nevertheless, Stützle et al [45] suggested that both α and β, should be considered while implementing the parameter variation or adaptation mechanism as they are responsible for controlling the influence of heuristics.

Another heterogeneous ACO was introduced by Hara et al [84] where initially α value is set to constant and *give-up ants* were introduced that construct partial solutions consisting of nodes where the distance from the current node to the next node does not exceed a pre-defined distance, *d*. When the *give-up ants* encounter a situation where the distance of all possible nodes exceeds *d*, then the tour construction will be terminated immediately yielding partial solutions. Then, all partial solutions from the *give-up ants* will be merged to produce one complete tour. As the performance was not satisfactory, the authors then varied the α parameter of the give-up ants from 0 to 1 with a step size of 0.005 for every iteration. Although improvement in performance was noted, the authors indicated that important parameters are problem-dependent hence better performance can be achieved if parameters are varied as the search progresses. Abdelbar et al [85] also implemented a slightly similar approach by introducing *stubborn ants* where these ants have the ability to implement its solution from the previous iteration on the next iteration. The authors introduced a stubbornness parameter to determine the biases of each ant in using the previous solution. This approach enhanced the exploitation of previous tours where a single ant in every iteration will have a higher probability of choosing its previous solution rather than exploring a new path. This somehow reduces the diversity of the colony by limiting the exploration of new search areas. In the meantime, Zufferey et al [86] implemented pre-determined a colony of ants which is categorized into the *normal ants*, *follower ants*, *moody ants* and i*nnovative ants*. *Follower ants* have a higher probability of choosing the previous tour with the highest pheromone trail while *moody ants* can interchange its decision-making by choosing a tour with high in pheromone or inverse to the pheromone value and lastly, *innovative ants* can alternate between the exploration or exploitation phase. The main contribution of this paper was to categorize the ants with different personalities and then vehicle routing problem is being implemented. Although result reported was not according to the state-of-the-art, however, the performance of the better metaheuristic approach of ant personalities is encouraging.

The most recent study of heterogeneous ACO was conducted by Sueoka et al [87] in which both *hard-working* and *lazy ants* were introduced and allowed to interchange between each other in the colony. The *hard-working ants* prefer the path with a high concentration of pheromone level while *lazy ants* perform a random walk on the search landscape. The authors concluded that *lazy ants* play an important role in exploring the search landscape in order to locate the global optimum. Again, this study only focused on exploration and exploitation in context of the parameter that influences the pheromone trail, α while did not take into consideration of the parameter that influences the heuristics, β. This is a key parameter in ACO that can improve the performance as suggested in [45] that should be considered when introducing parameter adaptation method.

Table 2.2 Summary of individual-level heterogeneity.

Title	Author	Year	Drawback
Ant Colony Optimization with cunning ants. [77]	Tsutsui	2006	Exploration vs exploitation imbalance.
Heterogeneous sensitive ant model for combinatorial optimization. [81]	Chira, Dumitrescu, and Pintea	2008	Only single parameter varied (α).
Adaptive Ant Colony Optimization with Cranky Ants. [83]	Yoshikawa	2009	Only involves pheromone coefficient, (α).
Improved Robustness through Population Variance in Ant Colony Optimization. [78]	Matthews,Sutton, Hains, and Whitley.	2009	Lacks explanation and relation to the real world.

Novel ant colony optimization algorithm with path crossover and heterogeneous ants for path planning. [79]	Lee and Lee	2010	Parameters kept constant.
Synthetic Genes for Artificial Ants Diversity in Ant Colony Optimization Algorithms. [80]	Nugulescu and Lascu	2010	The idea not supported by results.
Ant colony optimization using exploratory ants for constructing partial solutions. [84]	Hara, Matsushima, Ichimura, and Takahama.	2010	Parameters were kept constant.
Promoting search diversity in ant colony optimization with stubborn ants. [85]	Abdelbar and Wunsch	2012	Focused on exploitation that can cause stagnation.
Ant Metaheuristics with Adapted Personalities for the Vehicle Routing Problem [86]	Zufferey Farres and Glardon R.	2015	Pre-determined the number of ants for each personality.
On Heterogeneity in Foraging by Ant-Like Colony: How Local Affects Global and Vice Versa. [87]	Sueoka,Nakayama, Ishikawa, Sugimoto.	2016	Only a single parameter varied (α).

Table 2.2 summarizes all individual-level heterogeneous ACO algorithms where each of these algorithms approaches the principle of heterogeneity from a different standpoint, either using different ant roles or through the implementation of problem-specific heterogeneity. Firstly, it can be seen that there is a modest amount of research conducted in the heterogeneous ACO field unlike that of PSO although the

concept has been proven to improve the performance of optimization algorithms. Secondly, the algorithms reviewed mostly adopt static or constant parameter settings or vary only a single ACO parameter even though it is known that both α and β should be taken into account for parameter adaptation. The algorithms discussed above, approach the principle of heterogeneity from a different standpoint, either using different ant roles or through the implementation of problem-specific heterogeneity. The approach taken in this paper is one of biological plausibility for ants with similar roles, but differing behavioural traits, which are being drawn from a mathematical distribution. Therefore, this study analyses the heterogeneity in the ACO approach by randomly sampling the α and β parameters from two different distributions (explained later) within a pre-defined range. This allows each ant to have distinctive 'behavioural traits' in relation to a pair of α and β values that remain constant throughout the search process. In order to measure the effectiveness of the proposed approach in this study, both static (parameters do not change over time) and dynamic approach (the parameter changes over time via adaptive approach) is implemented as most of the heterogeneous ACO approaches presented above are static which put restriction while analysing the efficiency of an approach.

2.10.2.2 Multi-colony heterogeneity

Multi-colony heterogeneity indicates a colony of ants that consists of more than one sub-colony where each sub-colony differs in terms of its parameter settings or 'behavioural traits' while agents in each sub-colony have the same settings. An example of multi-colony heterogeneous ACO was proposed by Zhang et al. [88] where each sub-colony has its own pheromone updating rule that is distinguished from the other in order to achieve a better balance between exploration and exploitation of the search process. Therefore, the authors proposed two sub-colonies with one was implemented with Elitist Ant System (EAS) and the other implemented with Ant Colony System (ACS) characteristics respectively. However, the authors indicated that the algorithm still converges to sub-optimal solutions even though it is capable of overcoming the stagnation problem. Melo et al [89] proposed a multi-caste Ant Colony System (ACS) where the whole colony was divided into several castes and each caste has its own preference towards q0, the parameter that controls the degree of exploration or exploitation in ACS. Two multi-caste variants of ACS were presented that allows the usage of different values of q0 in a single run of algorithm which encompasses const-multi- caste (Multiple castes with a fixed number of ants and q0 value) and Jump-multi-caste (Different castes but ants are allowed to migrate from one group to the other for search). Results reported reveal that multi-caste configurations are subjected to more robustness than ACS which are effective in avoiding poor performance caused due to the suboptimal parameters being selected. Due to this, the proposed solution was effective in finding good solution without adopting the exact configuration. In another study, L. Isabel et al [12] introduced two variants of multi-colony ACO firstly by introducing sub-colonies that exchange communication or good solutions periodically among the sub-colonies and secondly, sub-colonies with different q0 and have the ability to migrate between sub-colonies. The author claimed that an ideal number of colonies is dependent on the number of iterations available and thus continues to improve at a higher rate as the search progress. The results illustrate that multi-colony configurations were able to avoid premature convergence which provides the main reason for the improvement at a later stage of research. Results support the advantage of self-adapting the parameters but multi-colony has also a drawback of computational cost as it was devised to run on a single processor. Mavrovouniotis et al [90] had experimented with both homogeneous and heterogeneous multi-colony ACO. The homogeneous sub-colonies have the same evaporation rate while the heterogeneous sub-colonies were set with different evaporation rate each. The authors had reported an overall improvement in diversity when heterogeneous sub-colonies were applied but were outperformed by the homogeneous approach in most of the test cases. The authors suggested that this can be due to the choice of parameters used to create a heterogeneous approach.

In summary, a real ant colony does not have sub-colonies except castes determined by the division of labour i.e queen, soldier or forager ants [91]. In addition to this, the algorithmic complexity caused by the multi-colony approach as well as the computational cost is not desirable. Therefore, the main reason for the consideration of individual-level heterogeneity in this study is that the population of the colony is not divided into sub-groups but is a whole group of ants with different responsibilities according to their role hence the direction taken in this research that is to analyse individual-level heterogeneity.

2.10.3 Hybrid ACO

Hybrid versions of ACO have been recently proposed in which ACO is combined with other methods to solve combinatorial optimization problems. Blum [92] had suggested that hybridization is recognized to be an essential aspect of high performing algorithms. Pure algorithms are inferior to hybridizations in terms of performance specifically when it comes to solving a complex problem. In fact, most hybridization of the current state-of-the-art ACO algorithms includes approaches and methods from other optimization techniques. Local search-based approaches such as tabu search, local search, iterated local search or hill climbing are the earliest types of hybridization that were incorporated into ACO. However, these hybridizations performed poorly when applied on large problem instances with an increase in the search spaces or highly complicated problems where finding feasible solutions are difficult. Due to this, some researchers created a hybrid ACO by incorporating artificial intelligence (AI) approaches such as constraint programming (CP) into ACO algorithms [92]. One reason why ACO algorithms are especially suited for this type of hybridization is because of the nature of the algorithm that builds its solution constructively. Due to this, the study in chapter 5 takes this as an inspiration and proposes a hybrid algorithm between self-adaptive Heterogeneous ACO that incorporates a GA-like approach and 3-Opt local search.

One of the first studies to incorporate ACO with GA is proposed by Botee and Bonabeau [56] where a GA was used to evolve the parameters while ACO was used to explore the search landscape. ACS was used as the base algorithm where the ant colony was divided into three castes according to their respective q_0 values. The approach was tested on two small scale TSP instances and the results indicate that the approach is time-consuming. Pilat and White [93] tried to improve the performance of ACS by integrating the algorithm with GA thus creating a hybrid approach. Empirical results suggest that the algorithm converges quickly to good solutions initially but then unable to improve thereafter. Therefore, the authors suggested that the approach can be used to find initial solutions before allowing the standard ACS to exploit these initial solutions. The authors in [80] reviewed that a GA approach can be implemented in ACO especially for dynamic problems but they failed to support the claim with any valuable results. Lee et al [94] deploy a hybrid of ACO and GA to solve the path planning of mobile robots. However, it is to be noted that firstly, the algorithm was not tested on a common problem instances that makes comparison fair comparison impossible and secondly, the algorithm is quite complex in nature making it difficult to replicate or compare. A hybrid approach between ACO and GA was proposed by Wang et al [95] that incorporates crossover and inversion to enhance the global search capability of the algorithm while the colony size is set as adaptive. α, β and ρ were drawn from a pre-defined range of values. However, the author did not mention the encoding scheme used as well as the mutation operator. If binary encoding is used in this study, then crossover and inversion can cause algorithm complexity and time-consuming. Another similar approach is by Deng et al [96] who proposed a GA-ACO hybrid algorithm where ACO represents the population in GA while each ant represents the individual component in GA. Even though the comparison results are promising, the authors did not elaborate on the parent selection, replacement and mutation operator thus lacking important information. The authors in [97] implemented the ACO-GA cooperation in order to balance the exploration and exploitation of the algorithms as well as to adapt its parameters. Comparison results indicate better performance in the proposed approach.

Two popular swarm inspired methods in computational intelligence are ACO and PSO where PSO is simple and promising, and requires less computation time, though it faces difficulties for solving discrete optimization problems while ACO has demonstrated to be an efficient and effective tool for combinatorial optimization problems. The latest trend in hybridization is to combine ACO with PSO in which the ACO is subjected to explore the search landscape while the PSO is being used to search optimal parameters in the parameter landscape. Ouyang and Zhou [98] proposed a hybridization of ACO and PSO for solving large scale TSP where the authors included a simple 2-Opt [99] local search procedure to improve the solutions found. The algorithm was compared against the conventional ACO algorithms and improvement is reported over other approaches however comparison was not made against any other state-of-the-art approaches. Elloumi et al [100][101] proposed a hybridization between PSO and ACO where the pheromone trail is used as the particle's weight for better global search. However, a thorough and clear explanation of the cooperation between these two algorithms was lacking. Mahi et al [47] explore in detail the hybridization of PSO-ACO with 3-Opt algorithm. Interestingly, the authors suggested an approach where the PSO algorithm optimizes the α and β values which then act as inputs to the ACO for exploring the search space. The 3-Opt local search procedure is used to improve on the solution found by ACO. Extensive comparison results against other algorithms indicate better performance demonstrating that this is one of the state-of-the-art methods in hybrid ACO therefore used as a comparator in experiments in Chapter 5. A further approach in Gulcu et al [48] proposed a parallel ACO with 3-Opt algorithm where the ant colony is divided into several sub-populations known as a multi-colony approach. The colonies run in parallel on different computers and the global tour is shared among the colonies occasionally to guide the colonies toward the global best tour. In addition to that, the sub-colonies also perform the 3-Opt procedure to their best tour in order to improve the solution found by the ACO algorithm. An interesting element in the proposed approach is that if a sub-colony is stuck in local optima, other sub-colonies are able to provide help by sharing its best tour hence enabling the sub-colony to escape from the phenomenon. However, several computers to perform distributed computing are required to implement this approach and the algorithm is highly complex as it requires occasional communication between the sub-colonies. As a state-of-the-art approach, this is also used as a comparator in the adaptive approach methods described in Chapter 5. However, it should be noted that the real ant colonies only consist of single colonies and are separated by the division of labour.

Most of the studies discussed lack a credible explanation of the proposed approach especially the underlying mechanism and the cooperation of the hybrid algorithms. Secondly, most of the algorithms involve complex procedures that are time-consuming in solving a problem. Having said that, a thorough explanation of the research work was explained in [11] and [12] while the results are competitive when compared against other approaches. Therefore, these two state-of-the-art algorithms will be used to compare against the proposed approach in Chapter 5.

2.11 Concluding remarks

This chapter presents an introduction to swarm intelligence in general and more specifically ACO. Interestingly, individually simple agent, ants as a whole are capable of complex behaviours such as nest building and maintenance, nest defence, foraging for food and many more. These are only possible due to what is known as the 'emergent behaviour' where the colony does not require any centralized control in solving complex problems. This section also primarily focused on the biological explanation of the foraging behaviour of real ants that acts as the main inspiration for the ACO algorithms. More importantly, biological researchers suggest the existence of heterogeneity in real ants and how an individual ant has its own preferences or behaviour especially in solving problems such as choosing a nest. Hence, this acts as an inspiration for the proposed approach.

Moreover, the ACO field has seen tremendous growth since the introduction of AS both in terms of new variants and applications that have been solved by ACO. However, it has been reported in several studies that the performance of heuristics, in general, is highly dependent on the parameter settings hence can easily deteriorate if not well-tune. However, tuning the optimal parameter of ACO for every problem or problem instance is tedious and almost impossible. Therefore, this chapter discussed and proposed both 'offline' and 'online' parameter control techniques through the proposed heterogeneous ACO approach to overcome the aforementioned problem. Apart from that, the static approach is generally deployed especially when it comes to heterogeneous studies that may limit the effectiveness of the algorithm. Therefore, both passive and active parameter exploration approaches are studied and deployed to improve the efficiency of the algorithm effectively.

Lastly, there are certain metrics over which the performance of a certain algorithm is measured which is either fitness solution based or behaviour-based that provides the fundamentals in analysing the efficiency of optimization algorithm when implemented to a certain optimization problem. The inspiration from the heterogeneity in real ants, description of conventional ACO in particular AS and MMAS, and the discussion on parameter adaptation technique provide a strong basis for the next chapter.

Chapter 3

Heterogeneous Ant Colony Optimization via Uniform Distribution.

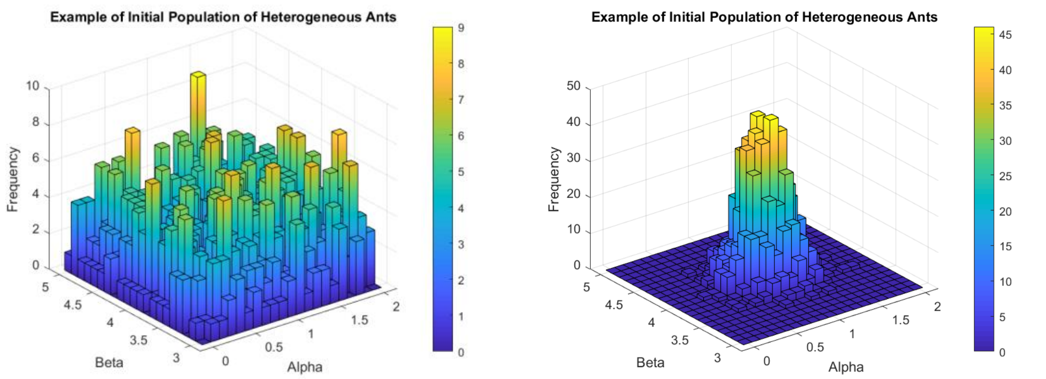
Researchers have proven that real ants differ from one another in terms of their morphological size and behavioural traits even though they belong to the same ant colony. As shown in Chapter 2, these differences allow the colony to achieve greater efficiency in surviving the harsh conditions of the real world. Hence, this chapter presents and analyses the effects of diverse behavioural traits sampled via different distributions in heterogeneous ACO to solve NP-hard combinatorial optimization problems. The heterogeneous ant colony consists of a collection of diverse but bounded agents where each agent has a distinctive bias towards the pheromone trail and local heuristics. The well-known problem of the traveling salesman problem TSP also described in Chapter 2, is used to demonstrate the capability of the proposed method.

3.1 Motivation

Even though ACO has been able to solve various optimization problems and can broadly be regarded as successful, there are still drawbacks affecting the performance of the algorithm. A significant issue with ACO, as in most other metaheuristic approaches is to find a balance between exploitation and exploration. Exploration of the search space denotes action by the search agent in moving towards unexplored areas while exploitation is a process of concentration of the algorithm in the areas of the search space where good quality solutions have been previously been found. Several studies reviewed in [9] show that a proper balance between exploitation and exploration is required in order for a metaheuristic algorithm to achieve good to optimal results. Furthermore, the balance allows the algorithm to find good solutions quickly but at the same time increases its capability of finding new solutions or promising regions. Another drawback of ACO is premature convergence towards local optima or suboptimal solutions. The occurrence of this phenomenon is partially due to non-optimal parameter settings in addition to a lack of diversity in the population [102] of the colony thus causing the ACO algorithm to get stuck in local optima. As a result of this, a number of ACO algorithms display stagnation behaviour where all ants generate the same tour which may not even be a local optimum. Most ACO algorithms are not able to escape from the stagnation behaviour and improve on the fitness solution found because of the absence of diversity in the homogeneous population that is deployed in the algorithm. One potential contributing reason for this is that the population is homogeneous and consists of agents that are set to identical parameters especially in relation to pheromone (α) and heuristics coefficients (β) in ACO. Both α and β are parameters that control the relative importance of the pheromone trail and local heuristics used in the transition probability [6] where a larger α will cause the fitness solution to have a higher influence of the pheromone trail while a larger β will bias the solution toward shorter path. Researchers deploy the homogeneous population concept in most optimization algorithms because it is easy to implement and less complex but varying these parameters according to a pre-defined rule or some adaptive measures can increase the effectiveness of the algorithm [45]. The rule of thumb in any optimization algorithm is that too much emphasis on pheromone trail or local heuristics may hinder the performance of the algorithm through over exploration or exploitation. Finally, on a lesser note of importance, biological researchers have proven that heterogeneity does exist in social insects and plays an important role in determining the efficiency of the colony [103][33][104][28][27]. Therefore, this investigative approach was conducted to study the influence of each ant having different behavioural characteristics or 'traits' in contrast to standard ACO where all ants have the same behavioural traits.

3.2 Formulations of the Heterogeneous Approach

The main aim of this chapter is to study and analyse the ant colonies as heterogeneous, multi-behaviours agents that can further improve the performance of the algorithm. The hypothesis is that with heterogeneity, a mixture of ants that are more inclined towards the exploration of the search space with others that exploit the best path found creates a balance in the search process. This creates a co-existence of search strategies as different strategies are required at different stages of the search process. The conventional ACO with static homogeneous parameter settings will not be able to interchange between exploration and exploitation strategies due to the fixed search behaviour where normally the algorithm starts with exploration before exploiting the solutions found. However, this search strategy of conventional ACO might hinder the performance of the algorithm especially when the algorithm is stuck in local optima. In addition to that, the performance of these algorithms is dependent on the parameter settings. The proposed heterogeneous approach is capable of overcoming this drawback due to the behaviours of the ants of which are randomly initialized either to be more inclined towards exploration or exploitation. The algorithm proposed in this study is a heterogeneous method that works by pre-assigning a random behavioural trait for each of the ants in the population size during initialization that will not change during the iterations, as would be the case with genetic variation in real ants. Each behaviour has a pair of continuous traits that can be related to pheromone trail intensity and visibility or also known as the local heuristic information drawn from a distribution. This research work has studied the effects of initializing the ant's population with two different distributions, the uniform distribution (this chapter) and the normal distribution (chapter 4). In a continuous uniform distribution with the interval of [*a,b*], each ant has an equal probability of being assigned to a value within the range of the interval while in a continuous normal distribution, each ant has a higher probability of being assigned to a value close to the central value known as the mean value, μ. The normal distribution or also known as the Gaussian distribution is the most widely used because most of the phenomenon in nature such as height and weight of the human population are normal distributions hence the tendency to be used in a study. Figure 3.1a illustrates the example of the initial population of the colony drawn from a uniform distribution while Figure 3.1b depicts the initial population of the colony drawn from a Gaussian distribution. The range of values used in both uniform and normal distribution heterogeneous approach is centered around parameter values close to those suggested by Dorigo [6] at the inception of the ACO technique (explained later).



(a) (b)

Figure 3.1: Example of the initial population of a single trial drawn from (a) uniform distribution and (b) normal distribution.

3.2.1 Design of the Proposed Algorithm

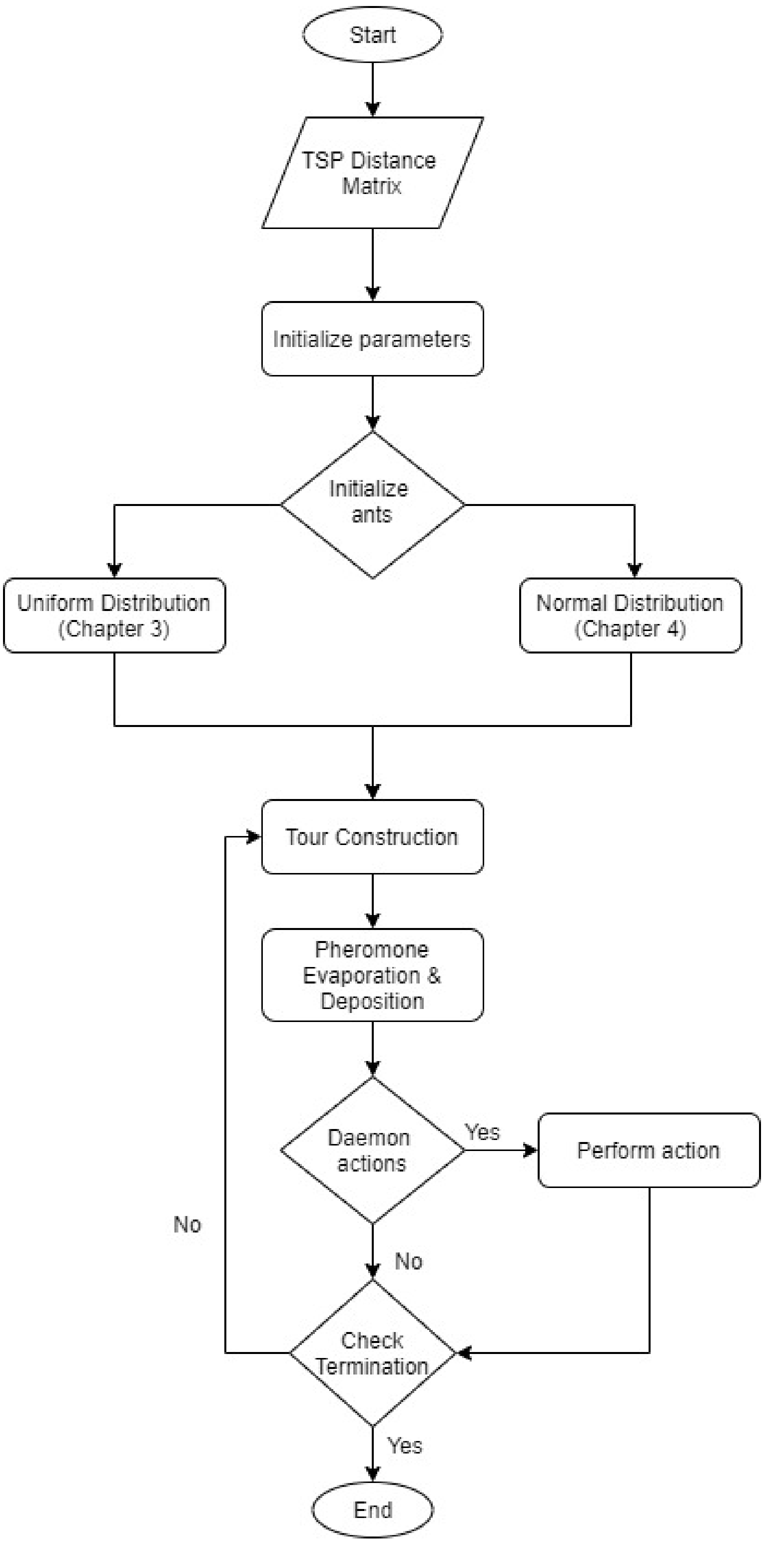


Figure 3.2: Flowchart of the proposed heterogeneous approach.

Figure 3.2 depicts the flowchart of the proposed heterogeneous approach where the algorithm starts by accepting input in the form of distance between nodes of the TSP graph followed by initialization of the parameters such as the stopping criterion, number of ants, constant value Q, initial pheromone and pheromone evaporation rate. The major difference between this approach and the conventional ACO is at the population initialization stage where agents will be assigned with α and β values that are randomly drawn from either uniform or normal distribution rather than identical parameters as in the traditional ACO. In the uniform distribution approach, the ants will have equal probability of being assigned α and β within the range while in the normal distribution, majority of ants will have α and β values around the pre-determined mean α and β. The range (uniform distribution) and selection of mean (normal distribution) for α and β values were based on suggestions of Dorigo et al in [3] as well as additional extensive experiments that have been conducted to evaluate this matter (discussed briefly in the following section). The proposed algorithm begins by randomly placing ants in the nodes of the graph in which every ant moves to a new node according to the probability rule. Each ant has its own preference in deciding the next node due to the individual behavioural trait incorporated in this study. TSP used in this study is a fully connected graph with each edge labelled by trail intensity $\tau_{ij}(t)$ and $\eta_{ij}(t)$ at time t. An ant at node *i* decides the next node with a probability rule that is based on the distance to that node and the amount of trail intensity on the connecting edge. This tour construction phase is repeated until all the ants have completed their tours. The pheromone updating and pheromone evaporation are two main components in the next phase where the pheromone trails are updated once all ants have constructed its tour while pheromone evaporation stops pheromone trails from an unlimited accumulation of pheromone that may cause stagnation in the search process. Depending on the base algorithm used in the study, all ants (AS) or only the ant that generates the iteration-best solution $T_{best}(t)$ (MMAS) is allowed to globally update the pheromone. Pheromone evaporation takes place on all path with the amount of evaporation is dependent on the pheromone evaporation rate, ρ. When base algorithm MMAS was used, both global-best $G_{best}(t)$ and iteration-best $T_{best}(t)$ solutions were used interchangeably to deposit pheromone. Daemon action is an optional stage which is usually used to perform additional steps to the solution found such as local search procedure or additional pheromone deposition. The algorithm terminates when the required number of function evaluations is reached.

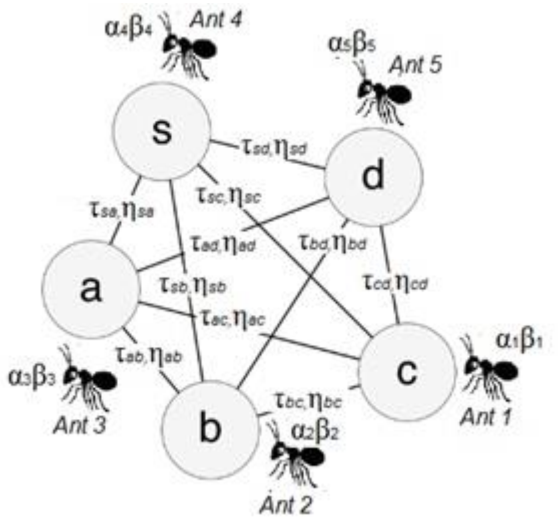


Figure 3.3: The principle of heterogeneity in ACO.

Figure 3.3 illustrates the heterogeneous approach in ACO where each ant will be assigned with individual α and β values compared to identical values for conventional ACO algorithms. The concept of heterogeneity is proposed by modifying the probability rule (Equation 1 in Chapter 2) to incorporate the behavioural traits thus producing equation 15. Therefore, the ants are governed by the probabilistic rule but their preference can be controlled by the heterogeneous elements that allow each ant to have a different perspective while exploring or exploiting the search space by introducing α^k and β^k which represents the individual behavioural traits.

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha^k}[\eta_{ij}]^{\beta^k}}{\sum_{(l \in N_i^k)}[\tau_{il}]^{\alpha^k}[\eta_{il}]^{\beta^k}} \; if\; j \in N_i^k \quad \text{or} \quad P_{ij}^{k} = 0 \; if\; j \notin N_i^k \tag{15}$$

3.3 Experimental Setup, Results & Discussion for Heterogeneous Ant System (HAS)

3.3.1 Experimental Setup

The experiments were conducted on an Intel Core i7 CPU-based computer running Windows 7 equipped with 4GB RAM. The traditional Ant System (AS) and Max-Min Ant System (MMAS) approach, which are used as the base algorithm to implement the heterogeneous approach, are developed using the MATLAB version R2015a. AS was chosen as part of this study due to being the 1st ACO algorithm that proved the artificial ant's concept and the algorithm allows all ants to deposit pheromone thus the effects of heterogeneity is much bigger in AS. It is important to note that MMAS was chosen because it is the best conventional ACO algorithm [5]. Each algorithm is tested using several TSP instances taken from TSPLIB [2] where the value in the problem name indicates the number of cities of the instances i.e Oliver30 is a 30-city TSP instance. All results are of 25 trials in experiments hereafter except stated otherwise. This conforms with the general number of trials in any swarm intelligence-based optimization algorithm study which are usually between 15 to 30 trials. The function evaluations for all the experiments were set as k.n.10000 where k=1 for symmetrical TSPs used, n=number of cities of the TSP instance and 10 000 is the maximum number of iterations unless stated otherwise.

3.3.2 Verifying the Base Algorithm

Firstly, the developed AS and MMAS (the traditional algorithm that was written from scratch on were compared with that of [3] and [5] to show a level of confidence that the developed algorithm is similar to the original version. All the parameters were set according to the authors' recommendations where for AS the parameters were set as follows: α=1, β=5, ρ=0.5 and $m = n$ where m is the number of ants and n is the number of cities related to the TSP. Meanwhile, the parameters for MMAS were set as follows: α = 1, β = 2, $m = n$, ρ = 0.98 and P_{best} is 0.05. Table

3.1 shows the comparison between the developed algorithms against its original versions where the results for the developed algorithms are the average of 15 trials. Several TSP instances were used namely 30-city, 51-city, 100-city and 198-city instances where known optimum tours exist. Equation 4 [105] represents the percentage difference that was used to indicate the similarity of the base algorithm's performance to the original algorithm. As can be seen in the equation below, the average is used as the referral value as not to create bias towards any values used in the equation. If the percentage difference is small, then the performance of the algorithm is said to be similar and vice versa.

$$\frac{|Value\ 1 - Value\ 2|}{(Value\ 1 + Value\ 2)/2}\ x\ 100 \tag{16}$$

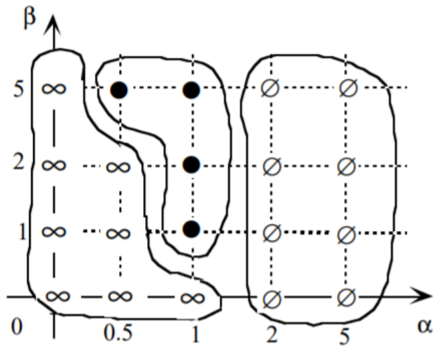
Table 3.1: Developed AS, MMAS vs Original AS, MMAS. Results show the average of the best cost. (Note: Average of 15 trials)

TSP	Optimum [106]	Developed AS	AS	Percentage difference (%)	Developed MMAS	MMAS [5]	Percentage difference (%)
Oliver 30	423.74	423.74	423.74 [3]	0	N.A	N.A	N.A
Eil51	428.87	437.560	437.3 [5]	0.06	427.5	427.1	0.09
kroA 100	21285.4	22451.98	22471.4 [5]	0.09	21299.6	21291.6	0.04
d198	15808.65	16692.24	16702.1 [5]	0.03	15960.2	15956.8	0.02

As can be seen from the table, the developed AS has almost identical result compared to the original AS for Oliver30 while MMAS was never tested using this particular instance by the author. In addition to that, eil51.tsp, kroA100.tsp and d198.tsp were tested on both variants of the developed base algorithm and comparison results suggest that the average best cost of the developed algorithms and that of the original developers' are very similar demonstrating that the base algorithm formulations are working appropriately. The percentage differences are very small thus this supports this claim.

3.3.3 Parameter Exploration for HAS

An ACO algorithm's performance and convergence speed are very much dependant on the parameter selection [107]. However, it is also well known that optimal parameters vary according to problems or even different instances of the same problem [20]. In addition to that, most of ACO algorithms deploy a homogeneous approach where the parameter values were set intuitively and remain static throughout the run. However, varying parameters before or during the run can improve the performance of the algorithm [45]. Therefore, an extensive experiment based on AS (and later MMAS) was conducted to find the best range of α and β for our heterogeneous approach where lower and upper bounds of α and β were based on the recommendation of [3]. We have implemented an approach from [45] that suggests one way of parameter exploration is to find out the good to optimal parameters before the actual run of the algorithm. Instead of creating a homogeneous group with the same α and β values, the proposed approach varies these values to create a heterogeneous colony of ants randomly drawn from a uniform distribution of pre-defined range. Both α and β play an important role in exploration and exploitation of the search space hence varying both parameters will introduce more variance in the agents. In addition, Stützle et al [45] suggest that both α and β are good candidates for parameter adaptation in ACO. Figure 3.4 illustrates the recommendation of very good parameter settings (noted by ● symbol) for AS by Dorigo et al [6]. As can be seen from Figure 3.4, the authors indicate that suggested values for α are between 0.25 and 1.5 while β has a range of 1 to 5. Therefore, extensive experiments were conducted to explore the suitable range to create the heterogeneous approach based on AS where the ants were drawn from a uniform distribution of α between 0 and 1 and 0 to 2 while a uniformly distributed β was varied between 0 and 5, narrowed down to 4 to 5. The other parameters were set according to [3]: 10 000 iterations, $m = n$, $\rho = 0.5$, $Q = 100$, initial pheromone trail = m/L_{nn} where L_{nn} is the tour length of the tsp instance using the nearest neighbour heuristic.



● Parameters that find optimal solution without entering stagnation.

∞ Parameters that do not find good solutions but did not enter stagnation.

∅ Parameters that do not find good solutions and causes the algorithm to go into stagnation.

Figure 3.4: Ant System (AS) behaviour for different α-β combinations. Reproduced from [6]

The following experiment samples the parameters to find the suitable range for the heterogeneous approach by exploring the ranges over several TSP instances that are called training set before implementing the optimal settings on the actual test instances [16, p. 5]. This method of finding the optimal settings is well-known in machine learning [61][108, p. 12]. 3 tsp instances were used to test the algorithm namely oliver30.tsp (integer length optimum = 420, real length optimum = 423.7406), eil51.tsp (integer length optimum = 426, real length optimum = 428. 8716) and eil101.tsp (integer length optimum = 629). Table 3.2 summarizes the outcome of our extensive experiment and the results are average of the best cost found in 15 trials. The table indicates that the Heterogeneous Ant System (HAS) approach with α between 0 and 2, β between 3 and 5 has the best performance by having the lowest average best cost in all three TSP instances compared to other parameter settings. The results also suggest that the performance of the algorithm deteriorates if the colony is too diverse (range of α or β too high) or achieve sub-optimal performance when the colony's diversity is low (range of α or β too small). This parameter exploration experiment allows us to find the right amount of diversity to be introduced in the colony for the heterogeneous approach.

Figure 3.5 illustrates the boxplot from the best cost of 15 trials of HAS with different α and β range as in Table 3.2 which is shown in x-axis when tested on Oliver 30 TSP instance. Four parameter settings of HAS were able to find the optimum tour in all the trials and this can be due to the right amount of diversity in the parameter combinations and secondly due to the fact that Oliver30 is rather a small TSP instance hence HAS with afore-mentioned parameter settings were all able to locate the optimum tour.

Table 3.2: Results from parameter exploration of Heterogeneous Ant System (HAS) where α is uniformly distributed between 0 and 1 and 0 and 2 while β distribution varies. Algorithm tested on oliver30.tsp, eil51.tsp and eil101.tsp. Results represent the average best cost of 15 trials while values in bold represent the best in each category.

α	β	oliver30	eil51	eil101
0 -1	0 -5	427.0934	445.301	699.1238
0 -1	1 -5	425.3379	441.6734	685.7444
0 -1	2 -5	426.0892	439.5271	678.2238
0 -1	3 -5	**423.7406**	436.2947	661.9443
0- 1	4 - 5	**423.7406**	436.3278	659.4744
0 -2	0 -5	427.2749	437.1203	688.2972
0 -2	1 -5	424.6639	442.3749	672.3319
0 -2	2 -5	423.9117	438.0173	665.7093
0 -2	3 -5	**423.7406**	**436.0904**	**645.5318**
0- 2	4 - 5	**423.7406**	436.6167	651.2821

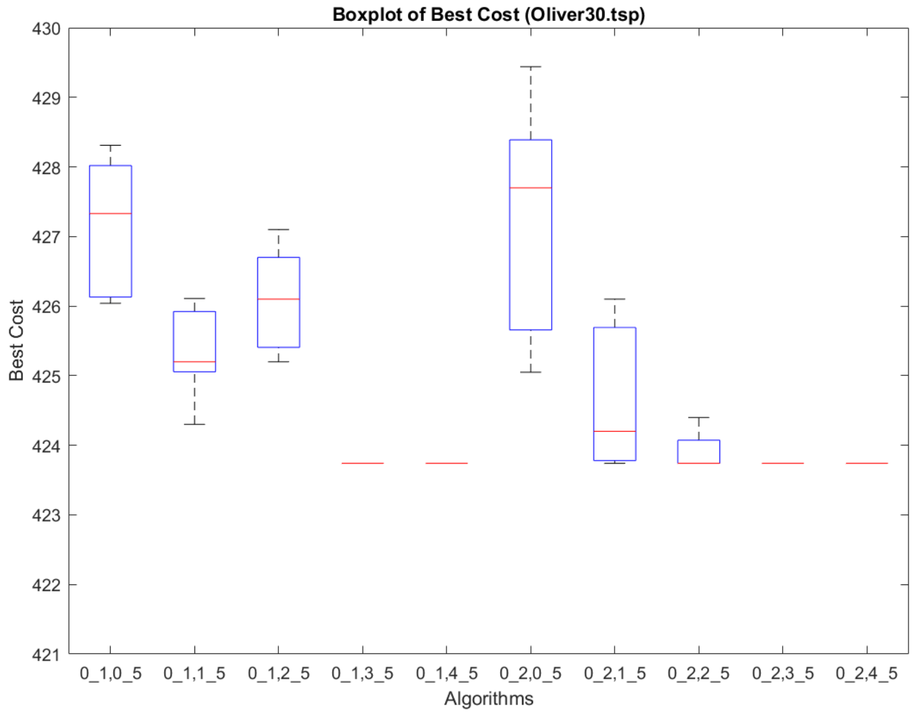
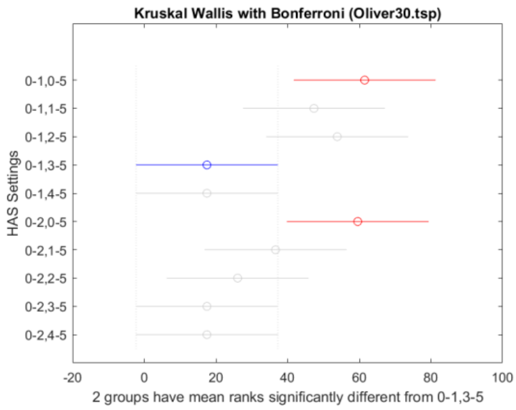


Figure 3.5: Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against oliver30.tsp. *(Note: 0_1 or 0_2 indicates the range of α values for heterogeneous colony while 0_5 to 4_5 indicates the range of β values).*

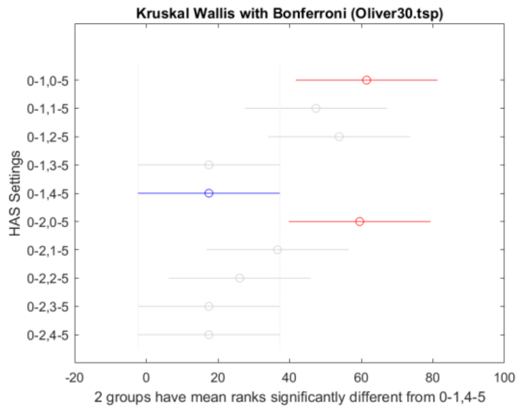
The Kruskal-Wallis statistical test is used to test the performance of HAS with different range of parameter settings in terms of significant difference in performance between each of the parameter settings. In addition to this, multiple comparison test or the post-hoc testing using Bonferroni correction method is performed to identify the best HAS parameter settings. This test is necessary in order to prevent Type 1 error in rejecting the null hypothesis when multiple comparisons are made. Kruskal-Wallis test with a significance level of $\alpha = 0.05$ produced a p-value of 1.6378e-09 hence indicating there is a significant difference between the parameter combinations. With Bonferroni correction, the significance level of $\alpha = 0.05$ is divided by the number of hypotheses in the test i.e, in this case, is 10. Therefore, the post-hoc test with a p-value of less than 0.005 is considered as significantly different.

Table 3.3: p-values of post hoc test using the Bonferroni correction method for HAS tested on oliver30.tsp (Opt=423.74). p-values in bold indicate the significant difference between the algorithms in comparison.

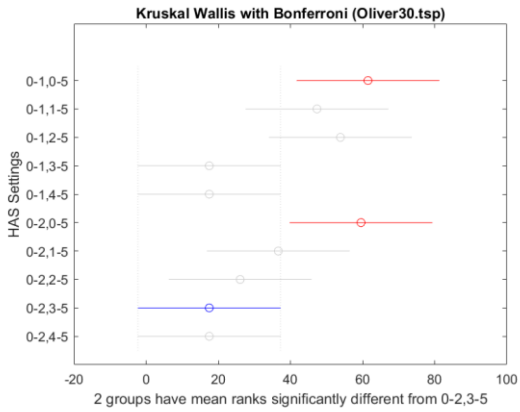
Oliver 30		HAS									
		0_1,0_5	0_1,1_5	0_1,2_5	0_1,3_5	0_1,4_5	0_2,0_5	0_2,1_5	0_2,2_5	0_2,3_5	0_2,4_5
HAS	0_1,0_5	X	1	1	**0.0008**	**0.0008**	1	0.6821	0.0242	**0.0008**	**0.0008**
	0_1,1_5	1	X	1	0.159	0.159	1	1	1	0.159	0.159
	0_1,2_5	1	1	X	0.0172	0.0172	1	1	0.2985	0.0172	0.0172
	0_1,3_5	**0.0008**	0.159	0.0172	X	1	**0.0018**	1	1	1	1
	0_1,4_5	**0.0008**	0.159	0.0172	1	X	**0.0018**	1	1	1	1
	0_2,0_5	1	1	1	**0.0018**	**0.0018**	X	1	0.0479	**0.0018**	**0.0018**
	0_2,1_5	0.6821	1	1	1	1	1	X	1	1	1
	0_2,2_5	0.0242	1	0.2985	1	1	0.0479	1	X	1	1
	0_2,3_5	**0.0008**	0.159	0.0172	1	1	**0.0018**	1	1	X	1
	0_2,4_5	**0.0008**	0.159	0.0172	1	1	**0.0018**	1	1	1	X



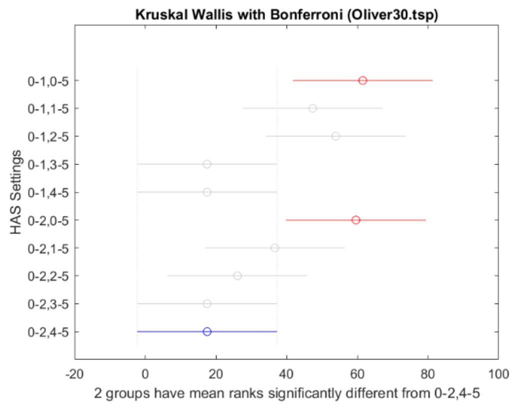
(a)



(b)



(c)



(d)

Figure 3.6: Confidence intervals of each of the HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on oliver30.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap.

Figure 3.6 (and other similar figures hereafter) illustrate the confidence interval comparison between HAS with different α and β combinations as shown by the *y-axis*. The *x-axis* represents the difference in the mean fitness solution of each parameter combination. The blue line indicates the subject (HAS approach) selected for analysis/to be compared with while the selected subject has performance that is statisticaly different against HAS approach represented by the red lines and this is indicated by the non-overlapping lines. Lines that overlap (the grey lines) indicate the performance of the selected HAS approach (blue-line) is not statistically different when compared with the HAS approach with grey lines. Overall, Figure 3.6 supports the results from Table 3.3 where the performance of HAS with α between 0 and 1, β between 3 and 5 (blue line in Figure 3.6a) is significantly different only against that of HAS with α between 0 and 1, β between 0 and 5 and that of HAS with α between 0 and 2, β between 0 and 5. This is indicated by the interval lines which did not overlap (highlighted in red). Figure 3.6b, Figure 3.6c and Figure 3.6d show the same scenario in regard to the significant difference between the parameter combinations.

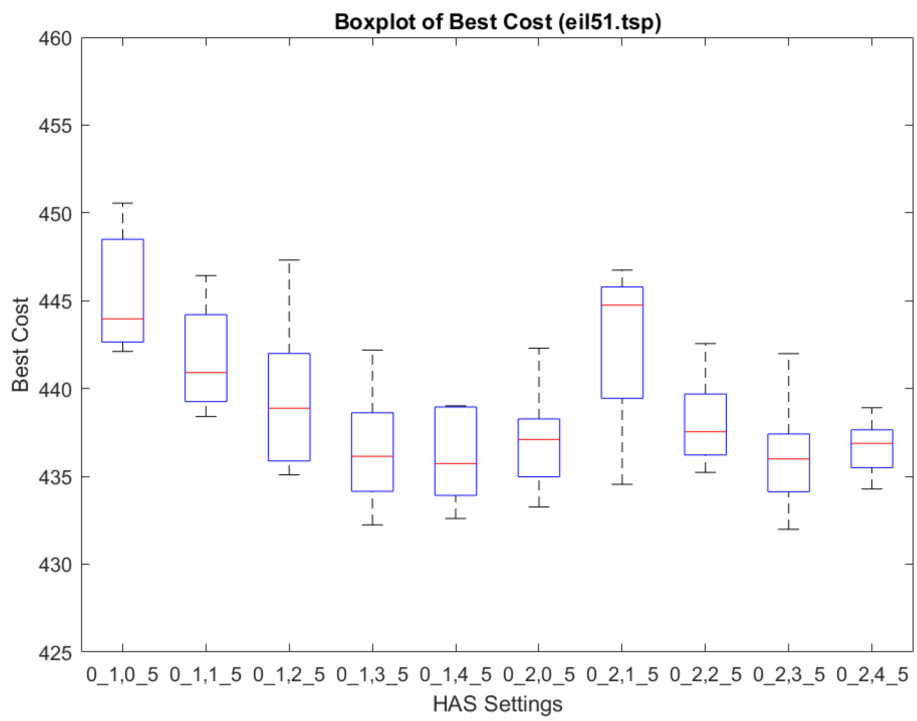


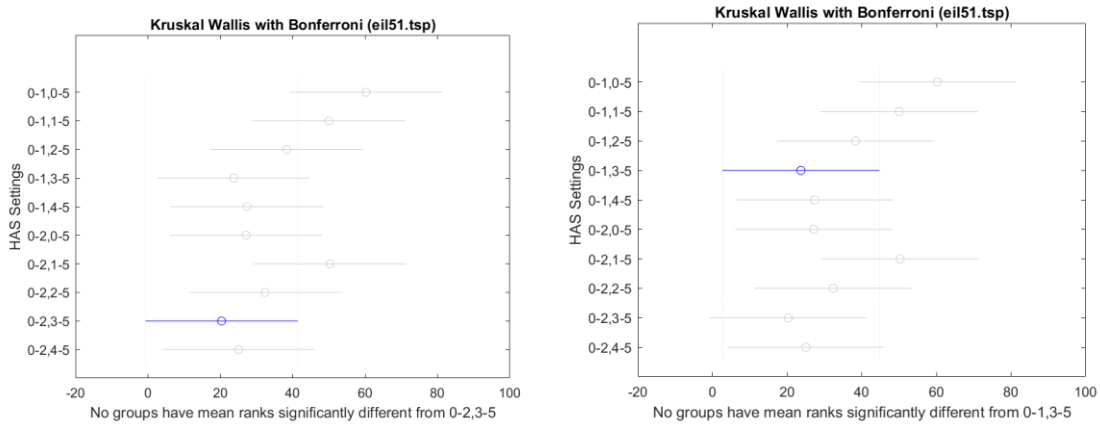
Figure 3.7: Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-5 to 4-5 indicates the range of β values).*

Figure 3.7 represents the best cost of HAS with different α and β range to create the heterogeneous approach in 15 trials. It can be noticed that HAS initialized with α: 0-1, β: 3-5 and α: 0-2, β: 3-5 have the best performance compared against other parameter combinations. HAS initialized with α: 0-2, β: 3-5 managed to find the best cost of 431.99 out of 15 trials with an average of 436.09. In the meantime, the best cost found by HAS initialized with α: 0-1, β: 3-5 is 432.24 with an average of 15 trials of 436.29. Figure 3.9 illustrates the result of multiple comparisons using the

Bonferroni correction method. As indicated by the results shown in Figure 3.7 and supported by Table 3.4 and Figure 3.9, it can clearly be seen that none of the HAS settings is significantly different when compared against other combinations. This is shown by overlapping confidence interval in Figure 3.8 with the closest of being statistically significant for HAS with α: 0-2, β: 3-5 is HAS initialized by α: 0-1, β: 0-5. Table 3.4 also suggest this where the p-value for HAS initialized with α: 0-2, β: 3-5 against HAS initialized with α: 0-1, β: 0-5 is 0.008 which is very close to being statistically significant.

Table 3.4: p-values of post hoc test using the Bonferroni correction method for HAS with different range of α and β values tested on eil51.tsp (Opt=426). p-values in bold indicate significant difference between the algorithms in comparison.

eil51		HAS									
		0_1,0_5	0_1,1_5	0_1,2_5	0_1,3_5	0_1,4_5	0_2,0_5	0_2,1_5	0_2,2_5	0_2,3_5	0_2,4_5
HAS	0_1,0_5	X	1	1	0.0261	0.0261	0.0798	1	0.3807	0.008	0.0392
	0_1,1_5	1	X	1	0.6097	0.6097	1	1	1	0.2464	0.8283
	0_1,2_5	1	1	X	1	1	1	1	1	1	1
	0_1,3_5	0.0261	0.6097	1	X	1	1	0.5664	1	1	1
	0_1,4_5	0.0261	0.6097	1	1	X	1	0.5664	1	1	1
	0_2,0_5	0.0798	1	1	1	1	X	1	1	1	1
	0_2,1_5	1	1	1	0.5664	0.5664	1	X	1	0.2272	0.7715
	0_2,2_5	0.3807	1	1	1	1	1	1	X	1	1
	0_2,3_5	0.008	0.2464	1	1	1	1	0.2272	1	X	1
	0_2,4_5	0.0392	0.8283	1	1	1	1	0.7715	1	1	X



(a) (b)

Figure 3.8: Confidence intervals of each of the HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combinations when tested on eil51.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap.

The lack of statistical significance in both oliver30.tsp and eil51.tsp is due to the small size of the TSP instance where even the base algorithm (AS) performs well when tested on these particular instances with the stopping criteria described in the methodology section. However, it is also not practical to use large TSP instances in this parameter exploration process as it will take a very long time to complete because of the increase in computational time when the problem size increases. Hence, 101 city TSP was used in this experiment to further validate the findings as this problem instance is neither too small to risk finding the optimum and skewing the results nor too large to carry out the required number of repeats. As indicated by Figure 3.9, some clear performance differences can be seen between the parameter combinations. Firstly, it can be noticed that a heterogeneous ant colony randomly initialized with α in the range of 0 and 2 has better performance when compared one to one against its corresponding HAS with α in the range of 0 and 1 (HAS with α: 0-1, β: 0-5 vs HAS α: 0-2, β: 0-5, HAS with α: 0-1, β: 1-5 vs HAS α: 0-2, β: 1-5 and so on). This suggests that the higher amount of diversity created by HAS with α: 0-2 compared to HAS with α: 0-1 thus allowing the colony to have more freedom in choosing a path with a high concentration of pheromone which can improve the performance of the algorithm. Secondly, a heterogeneous colony with β randomly initialized between 0 and 5 is said to be too diverse while range of β from 4 to 5 is said to be too small or have similar behaviours thus the colony might not have enough behavioural diversity to be able to explore and exploit both the parameter and search landscape to achieve optimum parameter settings that can directly lead the algorithm to locate global optima.

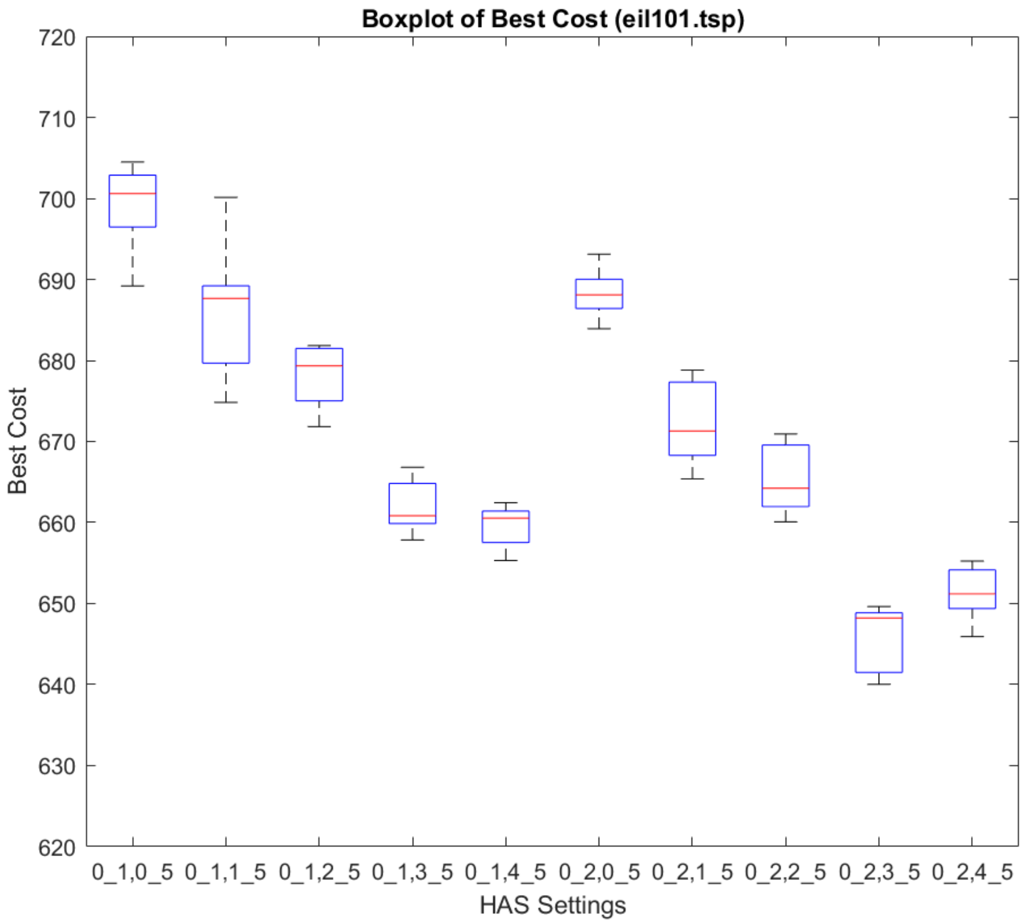
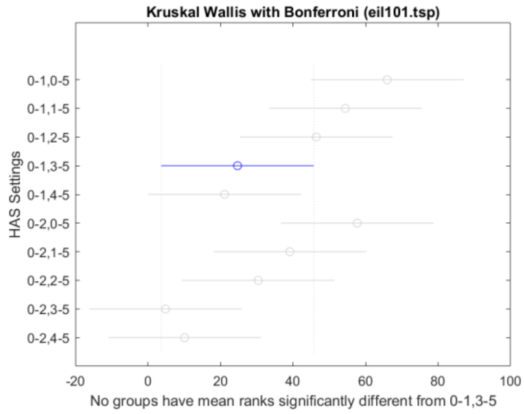


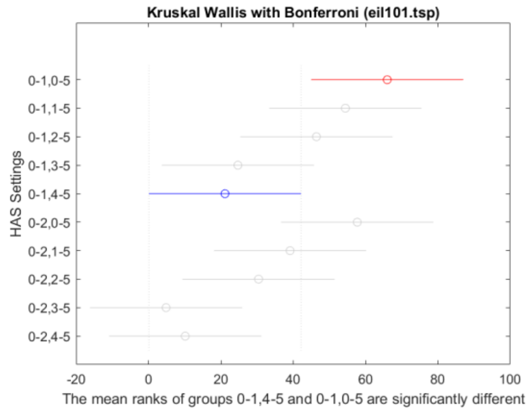
Figure 3.9: Boxplots representing 15 trials of different HAS parameter settings as shown in X-axis tested against eil101.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-5 to 4-5 indicates the range of β values).*

Table 3.5: p-values of post hoc test using the Bonferroni correction method for HAS with different range of α and β values tested on eil101.tsp (Opt=629). p-values in bold indicate significant difference between the algorithms in comparison.

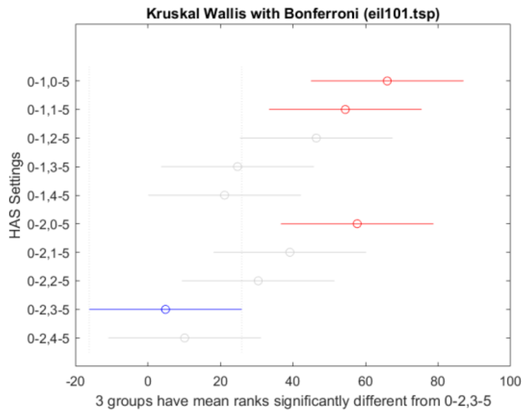
eil101		HAS									
		0_1,0_5	0_1,1_5	0_1,2_5	0_1,3_5	0_1,4_5	0_2,0_5	0_2,1_5	0_2,2_5	0_2,3_5	0_2,4_5
HAS	0_1,0_5	X	1	1	0.0066	**0.0017**	1	0.6099	0.0484	**0.0002**	**0.0002**
	0_1,1_5	1	X	1	0.2837	0.0996	1	1	1	**0.0002**	**0.0021**
	0_1,2_5	1	1	X	1	0.9045	1	1	1	0.006	0.0383
	0_1,3_5	0.0066	0.2837	1	X	1	0.1087	1	1	1	1
	0_1,4_5	**0.0017**	0.0996	0.9045	1	X	0.0348	1	1	1	1
	0_2,0_5	1	1	1	0.1087	0.0348	X	1	0.5459	**0.0001**	**0.0006**
	0_2,1_5	0.6099	1	1	1	1	1	X	1	0.073	0.3455
	0_2,2_5	0.0484	1	1	1	1	0.5459	1	X	0.8432	1
	0_2,3_5	**0.0002**	**0.0002**	0.006	1	1	**0.0001**	0.073	0.8432	X	1
	0_2,4_5	**0.0002**	**0.0021**	0.0383	1	1	**0.0006**	0.3455	1	1	X



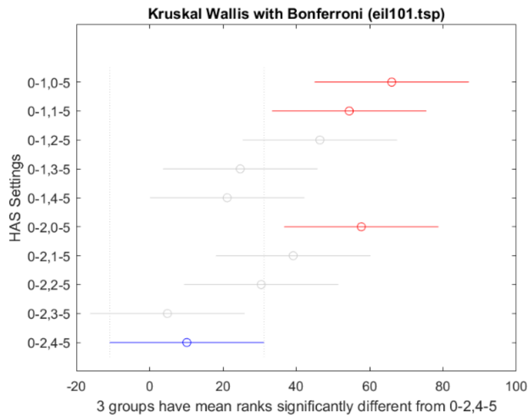
(a)



(b)



(c)



(d)

Figure 3.10: Confidence intervals of each of the HAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combinations when tested on eil101.tsp. Two groups are significantly different if their intervals are disjoint and vice versa if their intervals overlap.

The Kruskal Wallis statistical test indicates p-value of 1.4533e-10 while Table 3.5 and Figure 3.10 illustrate the multiple statistical tests with Bonferroni correction results. As mentioned earlier, p-values less than 0.005 is considered significantly different for post-hoc test with Bonferroni correction. It can be seen from Table 3.5 that HAS initialized with α: 0-2, β: 3-5 and α: 0-2, β: 4-5 have the best performance due to being statistically significant against three other setups of HAS. In addition,

Figure 3.10a shows that HAS initialized with α: 0-1, β: 3-5 has no statistical difference against other parameter combinations of HAS while Figure 3.10b shows that the performance of HAS with α: 0-1, β: 4-5 is statistically different against HAS with α: 0-1, β: 0-5 only. Conversely, both Figure 3.10c and 3.10d show that the performance of HAS randomly initialized with α:0-2, β: 3-5 and α:0-2, β: 4-5 are statistically different against 3 out of 9 other parameter settings of HAS respectively. Therefore, the results from Table 3.2 supported by the statistical tests suggest that HAS initialized randomly from a uniform distribution within the range of α from 0 to 2 and β from 3 to 5 has the best performance when tested on three different problem sizes of TSP. The results also give an early indication that the heterogeneous approach is able to locate good to optimal solution independent of the size of the problem. Hereafter HAS will be initialized with these settings unless stated otherwise.

3.3.4 Comparison against the Base Algorithm (HAS)

Next, the Heterogeneous Ant System (HAS) was compared against its base algorithm, AS [6] based on 4 symmetrical tsp instances. As stated by the authors in [41], the AS (and later MMAS [5]) systems have gone through rigorous and extensive experiments to determine the optimal alpha and beta settings for these problems. The resulting comparisons are therefore made between the randomly initialized heterogeneous system against well-tuned examples of the base ACO algorithms. HAS has the same parameter settings as AS with the only difference between the algorithms is that α is drawn from a uniform distribution between 0 and 2 while β is uniformly varied from 3 and 5. In the meantime, AS used in this experiment is developed and verified in the previous section to be similar to that developed by [6]. Table 3.6 summarizes the comparison between AS and HAS on several TSP instances for 25 trials. It can be seen that HAS was able to locate the best cost of 428 while AS has a best cost of 433 out of 25 trials. In general, HAS improves on the best cost found by AS with an average of 436.00 compared to that of AS which is 437.56. Although HAS was not able to locate the optimum of 426 [109] in the time allotted, it has a better overall performance compared to AS in terms percent of deviation from the optimum. A value is said to be 1% deviation of optimum when it is within the range of 1% to the optimum. In eil51.tsp case, 1% deviation is 1/100 x 426 (optimum value from TSPLIB [109]) =4.26+426 =430.26.

426 < X< 430.26 = 1% deviation of optimum.

430.26 < X < 434.52 = 2% deviation of optimum.

Table 3.6 also shows the comparison between AS and HAS for 100-city tsp, kroA100.tsp. HAS managed to improve on the fitness solution compared to AS where the average best cost for HAS is 22347.6 and that of AS is 22469.4. Although both AS and HAS did not manage to find the optimum for 100-city TSP problem, HAS found the a best cost that is within 5% of the optimum 22 times compared to none by AS. In addition, HAS found a best cost of 22215 compared to 22384 of AS out of 25 trials. The table also summarizes the outcome of 25 trials of d198.tsp using both AS and HAS. AS found a best cost of 16356 throughout the 25 trials while HAS found a best cost of 16186. In addition, HAS has a lower average compared to AS. Although the optimum is not found by any of the algorithms, HAS managed to find fitness solutions that are 3% within the optimum range 6 times and 19 times within 4% of the optimum compared to 0 and 3 times respectively by AS. Lastly, a comparison was carried out between the proposed heterogeneous approach and AS tested on lin318.tsp and it is shown that HAS has better performance in terms of best cost, average and worst best cost compared to AS. Overall, HAS has better performance in all 4 TSP instances when compared against AS. Wilcoxon rank-sum test with a confidence interval of 95% was conducted to determine the statistical significance of the proposed approach against AS. Therefore, a p-value of less than 0.05 indicates a statistically significant performance of the compared algorithm. A post hoc test with Bonferroni correction was not required here because the comparison was done based on 4 different TSP instances. As shown in Table 3.6, the p-values which are less than 0.05 indicate that the performance of HAS is statistically significant against that of AS for all four TSP instances. Even though HAS was not able to locate optimum in all four TSP instances, it managed to have better overall performance as shown in Figure 3.12. As can be seen, the heterogeneous approach has lower median when compared to the well-tuned AS algorithm in all 4 TSP instances. In addition to that, even the third quartile of HAS is lower than the median of AS in all the boxplots. This shows that almost 75% out of 25 trials of the best cost of HAS is better than AS for each of the TSP instance tested. The larger interquartile also indicates the diversity in the fitness solutions found by HAS unlike AS where the interquartile range is small which further support the claim that the heterogeneous approach is able to locate better solutions compared to AS.

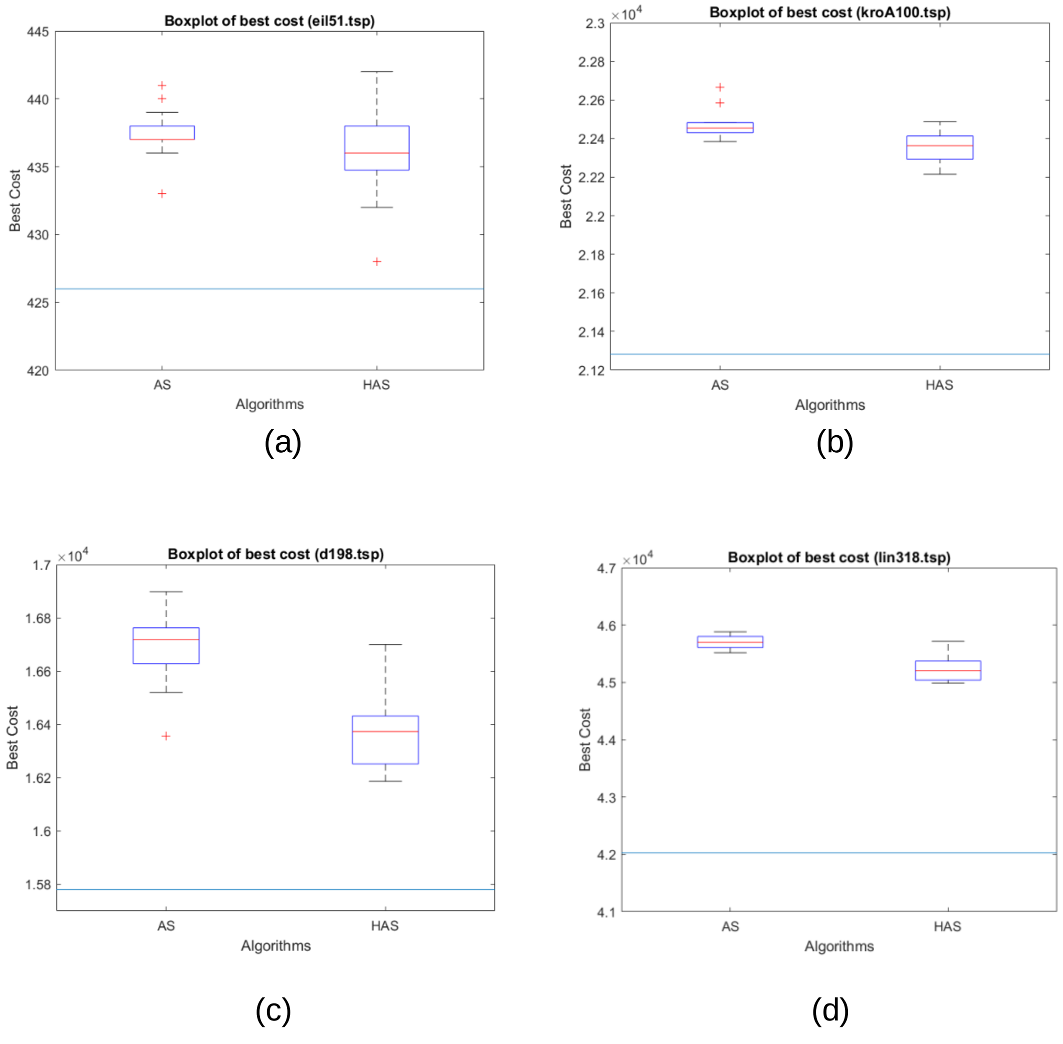
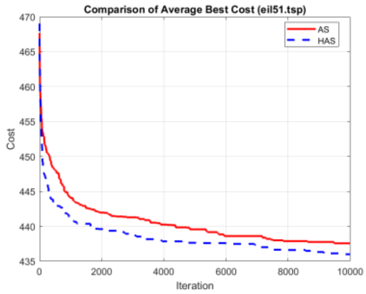


Figure 3.11: Boxplots illustrating the performance comparison of the heterogeneous approach against that of AS in terms of the best cost of 25 trials tested on several TSP instances. The blue line in each figure represents the optimum of that particular instance.

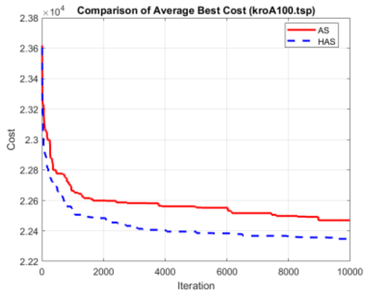
Table 3.6: Best, average and worst cost comparison between AS & HAS for several TSP instances for 25 trials with stopping criterion was set at 10 000 iterations. The optimum for the TSP instances are 426 (eil51.tsp), 21282 (kroA100.tsp), 15780(d198.tsp) and 42029(lin318.tsp). Results in bold represent best in each category.

TSP	Method	Best	Average	Worst	# Optimum Found	≥ 1% dev of opt	≥ 2% dev of opt	≥ 3% dev of opt	≥ 4% dev of opt	≥ 5% dev of opt	≥ 6% dev of opt	p-val
eil51	AS	433	437.56	**441**	0	0	1	21	3	0	0	**0.0143**
	HAS	**428**	**436.00**	442	0	**1**	5	15	4	0	0	
kroA100	AS	22384	22469.4	22666	0	0	0	0	0	0	5	**3e-05**
	HAS	**22215**	**22347.6**	**22487**	0	0	0	0	0	**22**	25	
d198	AS	16356	16572.48	16724	0	0	0	0	3	3	14	**2.8e-08**
	HAS	**16186**	**16359.04**	**16700**	0	0	0	**6**	19	0	0	
lin318	AS	45517	45698.2	45882	0	0	0	0	0	0	0	**6.1e-08**
	HAS	**44986**	**45237.28**	**45716**	0	0	0	0	0	0	0	

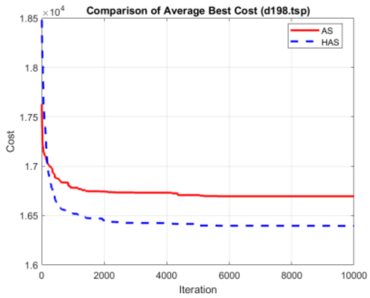
Additionally, it can also be noticed both from Table 3.3 and Figure 3.11 that the heterogeneous approach has a greater effect as the problem size increases thus illustrating the effect of diversity in exploring and exploiting the search space to locate better solutions. Furthermore, the results also indicate that the heterogeneous approach can overcome the parameter tuning problem by having good to excellent performance even when the problem size varies. The various behavioural traits introduced by the heterogeneous approach in the colony allow the algorithm to exhibit exploration and exploitation mode simultaneously unlike conventional ACO algorithms that tend to explore the search space in the beginning before trying to exploit the initial solutions found to locate better solutions. Figure 3.12 illustrates the convergence plot of both HAS and AS for 10 000 iterations for all 4 TSP instances where the figures on the left show the full version of the convergence plot while figures on the right are the truncated version that has been zoomed in for better illustration on the speed of convergence. The heterogeneous approach starts the search process with a higher exploration rate as can be seen in the figures on the left. This is because of the lack of information initially on the pheromone landscape for the colony to exploit. Nonetheless, the colony was able to locate good solutions quickly and exploited those solutions to locate better solutions as shown in the figures on the right which show the convergence curve of HAS improved on the performance of AS very early on. Figures 3.12g and 3.12h show that the convergence plot of the heterogeneous algorithm only improves on the AS after about 2500 iterations. This shows that the algorithm's behaviour changes according to the problem size by further exploring and exploiting the search landscape to locate good solutions. A larger TSP instance or a more complicated problem might need a longer exploration phase which can be achieved by the heterogeneous approach.



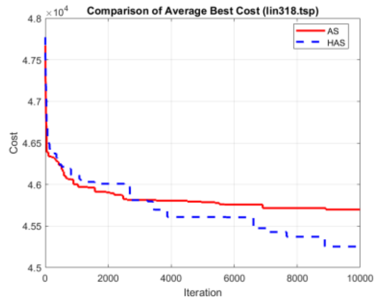
(a)



(b)



(c)

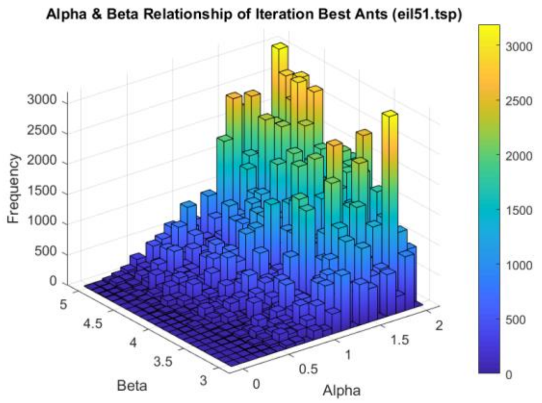


(d)

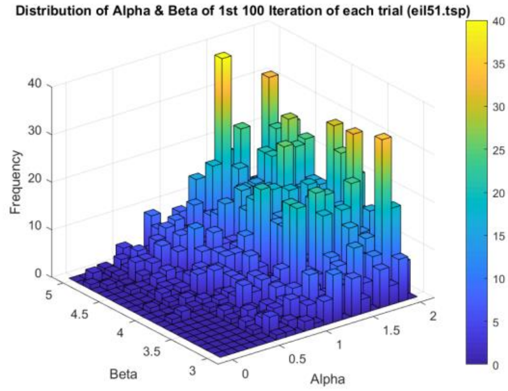
Figure 3.12: Convergence plot of HAS and AS represented by the average best cost of 25 trials over 10 000 iterations.

3.3.5 Heterogeneous Ants' Distribution Analysis (HAS)

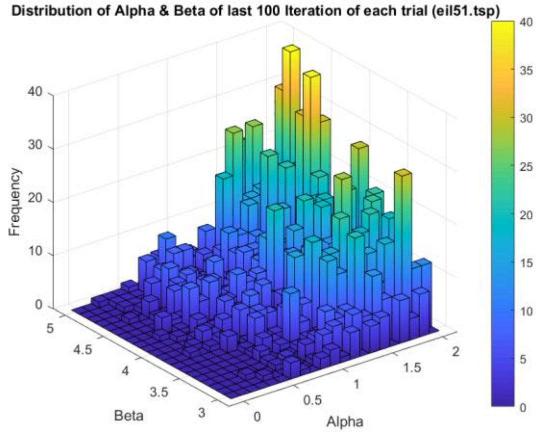
The concept of heterogeneity allows for the exploration and visualization of the parameter space during optimization. The visualization of best-performing ants and their parameters in each iteration provides an understanding of the parameter setting best suited to different phases of the search. It should be noted that these experiments explore this passively, ants are not selected for their performance at this stage. An adaptive method that simultaneously optimizes parameter settings is discussed in Chapter 5.



(a)



(b)

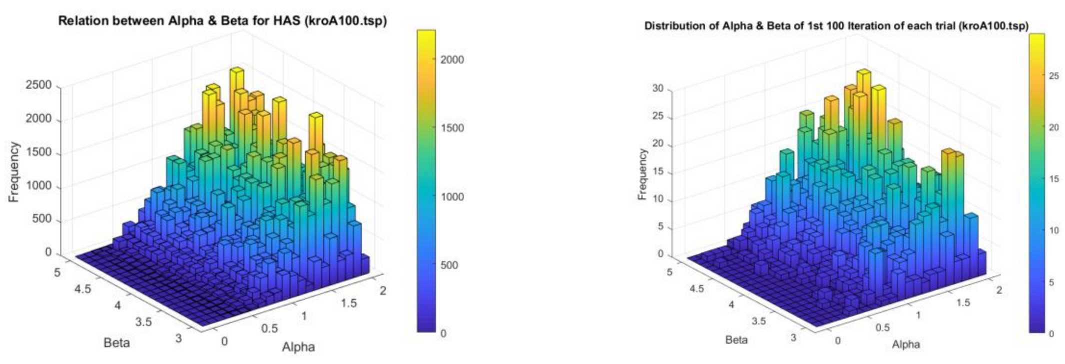


(c)

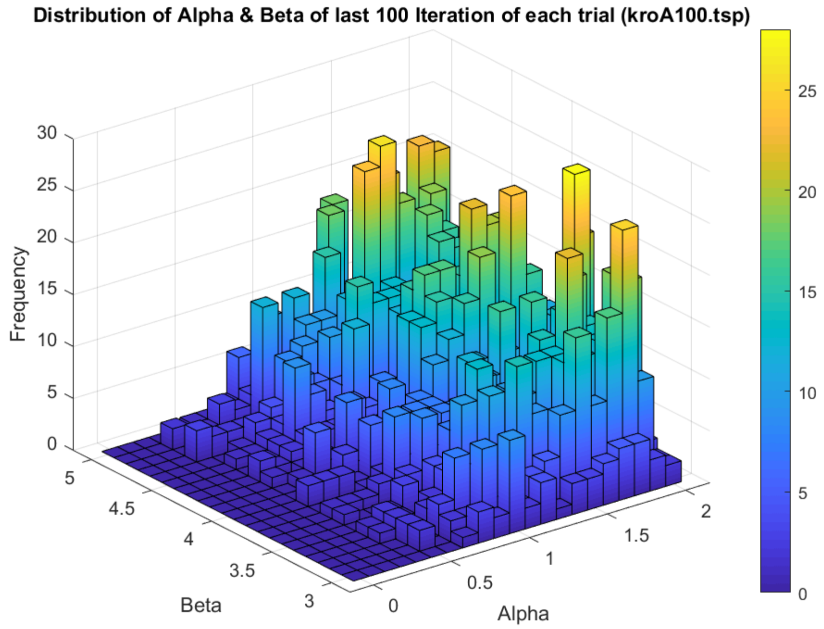
Figure 3.13: Histogram representing Alpha & Beta of iteration-best ants for HAS (eil51.tsp).

Figure 3.13a illustrates the distribution of α and β of iteration-best ants for all trials of HAS when tested on eil51.tsp. As each trial is stopped at 10 000 iterations meaning each trial consists of 10 000 iteration-best ants and each ant has a pair of α and β values assigned randomly from the pre-assigned range. Therefore, the figure shows the distribution of α and β of every iteration of all 25 trials. As can be seen in the figure, It can be seen the alpha values that mostly contribute are between 1.9 and 2, with a strong skew towards these values whereas the beta distribution is much more uniform with a small skew towards beta values of 4.6 and 4.75. This shows that the heterogeneous approach introduces diversity in the algorithm and suggests the mechanism behind the improved performance over the algorithm with a single 'behavioural trait'. On top of that, Figure 3.13b and c represent the α and β values of the first 100 and last 100 iterations of each trial respectively. It can be noticed in Figure 3.14b that during the initial stages of the exploration, ants with low α and high β values tend to perform better in locating good solutions while during the later stages, ants with higher α values perform better as shown in Figure 3.13c. This is because initially, all the edges have the same amount of initial pheromone hence very little information on the pheromone landscape for ants that are more inclined towards higher concentration of pheromone to exploit on. Therefore, ants with a higher preference towards heuristics coefficient (shorter edges) will perform better during the early stages of the search process. As the pheromone build-up on the edges, there will be enough information for ants with a higher preference towards the pheromone coefficient to exploit this information to locate better solutions.

Figure 3.14a illustrates the overall performance of the heterogeneous ant colony in terms of α and β distribution in regard to the iteration-best ants. Figure 3.14b and c in the meantime show the distribution of α and β for 1st 100 and last 100 iterations of each trial for 25 trials. Overall, it is noticeable from Figure 3.14a that ants in the range of 1.5< α<1.8 registered better performance by being able to locate good solutions while β values have a uniform distribution. The distributions are similar for the case of both 1st 100 and last 100 iterations of each trial where ants with higher β values perform best initially while as the search progresses and pheromone deposition increases on the edges hence ants with a higher preference towards pheromone perform better by exploiting this information to locate improved solutions.

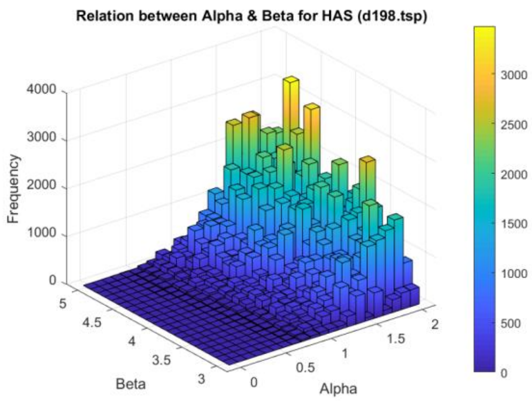


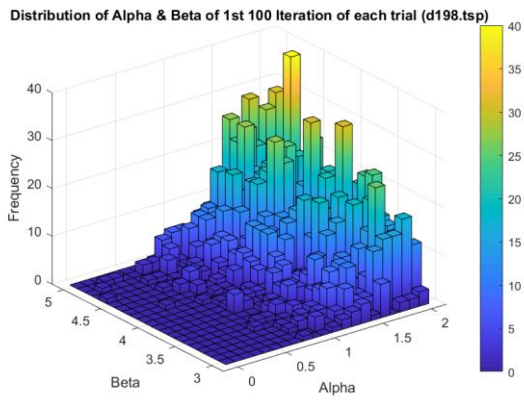
(a) (b)



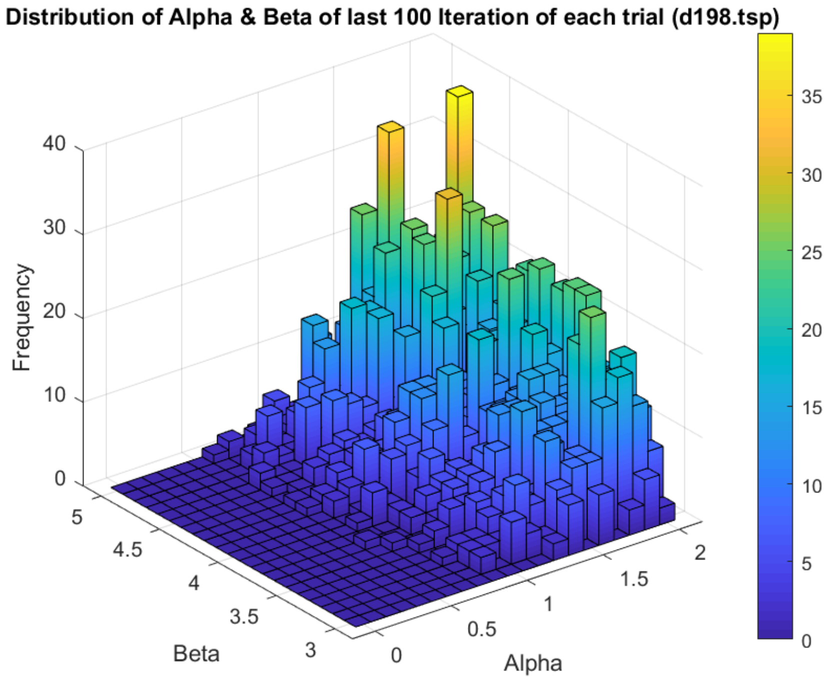
(c)

Figure 3.14: Histogram representing Alpha & Beta of iteration-best ants for HAS (kroA100.tsp).



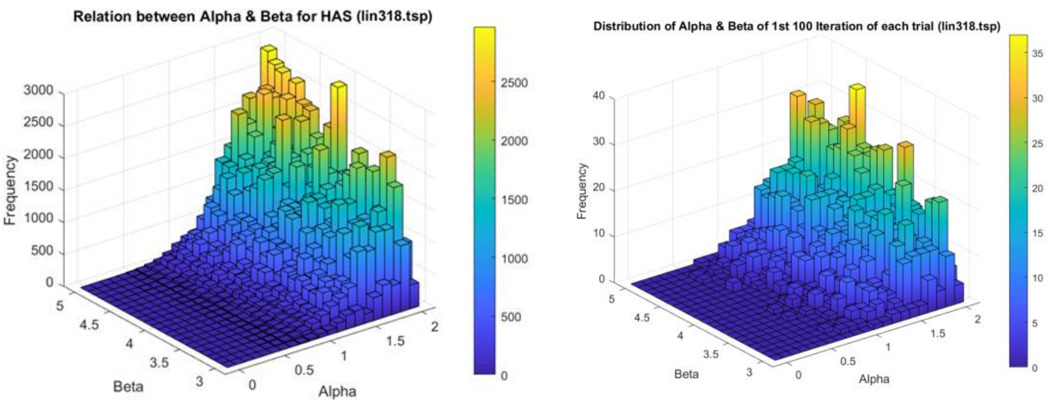


(a) (b)

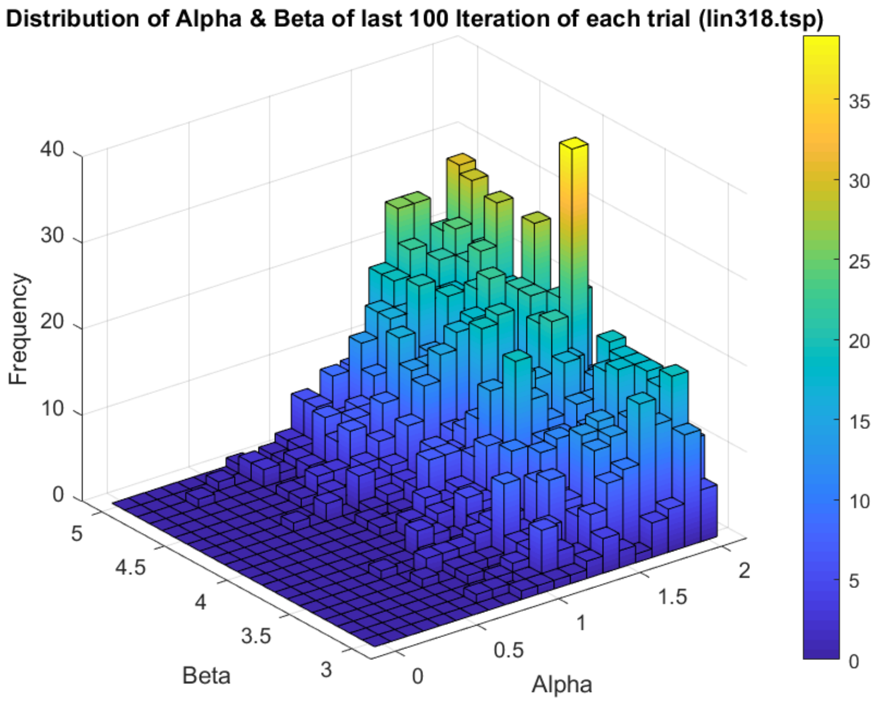


(c)

Figure 3.15: Histogram representing Alpha & Beta of iteration-best ants for HAS (d198.tsp).



(a) (b)



(c)

Figure 3.16: Histogram representing Alpha & Beta of iteration-best ants for HAS (lin318.tsp).

Figure 3.15 and Figure 3.16 both suggest similar trends where strong skewness can be noticed towards higher α and β for all the iterations. The figures also support earlier claims in terms of ants that perform best during the 1st 100 and last 100 iterations. It can be seen clearly that even though both α=1 and β=5 as per suggested in [3] are covered in the pre-determined range for heterogeneous approach, both α and β values that managed to find best cost in every iteration increases rapidly from 0.5 to 2 and a steady increase from 3 to 5 respectively with α values having a peak between 1.5 and 1.9 while β values have a peak between 4.25 and 4.9. The algorithm shows robustness towards parameter settings as different ants perform better at different stages of the search process. In addition to that, the heterogeneous approach managed to achieve good to optimal performance even when tested on different problem instances. Even though the results corroborate the parameter suggestion of [6], but they also suggest that these parameters might not be optimal for all problems or problem sizes.

Although Figure 3.13 to 3.16 indicate that larger α values seem to perform better, this can be due to the synergistic effects of the diverse population especially for HAS which allows all ants to affect the pheromone landscape by depositing pheromone. This enable ants with larger α values to make use the knowledge gathered on the pheromone trail to locate better solutions. Having said that, however, it is worth to analyse the performance of the heterogeneous population by expanding the range of α values to more than 2 in future studies and to investigate whether the result still corroborates to Figure 3.4 or new information can be found.

3.4 Experimental Setup, Results and Discussion for Heterogeneous Max-Min Ant System (HMMAS)

3.4.1 Parameter Exploration of HMMAS

Even though HAS was not able to locate optimal tours of all four TSP instances tested, the proposed approach managed to improve on the performance of the base algorithm significantly. The encouraging results from the initial experiments led to the implementation of the heterogeneous approach on Max-Min Ant System (MMAS) which is one of the best performing conventional ACO algorithms [43]. The α and β values of MMAS were both set as 1 and 2 respectively thus the colony was implemented as homogeneous where all ants have the same behaviour in tackling the optimization problem. One of the disadvantages of homogeneous ants is that the colony is not able to escape from stagnation behaviour where all ants repeat the same tour. Implementing a colony of heterogeneous ants will allow the ants to have a different perspective in their decision making that will indirectly help the colony to escape any stagnation. First of all, the possible range of α and β values were explored in order to create the heterogeneous approach on MMAS where the range of values was explored based on the suggestions by [6] and [43] inclusive of the suggested parameter values. The idea of parameter exploration is to explore the suitable range, unlike parameter tuning where optimal parameters for each problem instance are explored. The experiment was conducted on both eil51.tsp and eil101.tsp while results of parameter exploration using Oliver30.tsp were not reported because almost all parameter combinations of MMAS were able to achieve good to optimal solutions due to the small problem size, hence no strong conclusion can be made of the parameter exploration of the proposed algorithm's based on this TSP instance. Table 3.7 shows the results of parameter exploration experiment for Heterogeneous Max-Min Ant System (HMMAS) where the algorithm is initialized with α values in the range of 0 and 1 as well as 0 and 2 while the range of β values varies. The results indicate the average best cost of 15 trials of the heterogeneous algorithm when randomly initialized from the uniform distribution of α β values within the pre-defined range. Other parameters of the algorithm were set according to that suggested in [43] while the stopping criterion of each trial was set to 10 000 iterations.

The results indicate that HMMAS with α and β randomly initialized from 0 and 2 and 1-3 respectively has the best performance with an average best cost of 428.0 when tested on eil51.tsp. However, the Kruskal Wallis statistical test conducted on the performance of the algorithms when tested on eil51.tsp returned a p-value of 0.282 thus indicating that none of the performance is statistically significant when compared against each other. Figure 3.18 illustrates the boxplots representing the best cost of 15 trials of HMMAS tested on eil51.tsp and it can be seen that HMMAS with parameter settings of α from 0 to 2, β from 1 to 3 as well as α from 0 to 2, β from 2 to 3 have slightly better performance as shown by the lower median compared to other parameter settings of HMMAS. Even though Table 3.7 and Figure 3.18 indicate that HMMAS with parameter settings of α from 0 to 2, β from 1 to 3 have the best performance, the post-hoc confidence intervals show that there is no statistical significance in the performance of the algorithm with the afore-mentioned setting when compared against other settings when tested on eil51.tsp. Furthermore, Figure 3.19 illustrates the boxplots that represent the best cost of HMMAS with several parameter combinations tested on eil101.tsp. It can be noticed that HMMAS with α from 0 to 2, β from 1 to 3 managed to locate the optimum of 629 while also have the lowest median and lowest worst cost.

Table 3.7: Results from parameter exploration of Heterogeneous Max-Min Ant System (HMMAS) where α is uniformly distributed between 0 and 1 as well as 0 and 2 while β distribution varies. The heterogeneous algorithm tested on eil51.tsp and eil101.tsp. Results represent the average best cost out of 15 trials while values in bold represent the best in each category.

α	β	Average	
		eil51	eil101
0-1	0-2	430.07	639.4
0-1	1-2	429.00	638.87
0-1	0-3	429.53	639.33
0-1	1-3	429.20	638.4
0-1	2-3	429.33	638.47
0-2	0-2	428.93	637.67
0-2	1-2	428.73	637.13
0-2	0-3	429.13	637.8
0-2	1-3	**428.00**	**635.13**
0-2	2-3	428.26	636.00

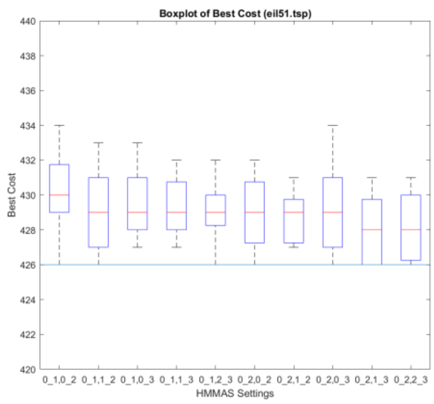


Figure 3.17: Boxplots representing 15 trials of different HMMAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-2 to 2-3 indicates the range of β values).*

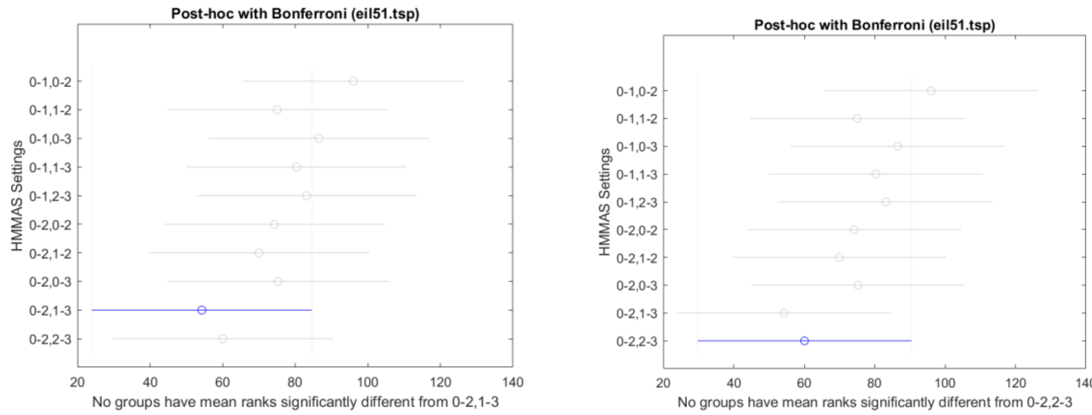


Figure 3.18: Confidence intervals of each of the HMMAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on eil51.tsp.

The Kruskal-Wallis statistical test produced a p-value of 0.4482 which indicates the performance of the algorithm is not statistically significant. The post-hoc confidence intervals shown in Figure 3.20 also suggest similar performance of HMMAS with different parameter settings as all the confidence interval lines overlap each other thus indicating high similarity. Although statistical significance in performance is not achieved by any parameter combinations of HMMAS for both TSP instances, this gives an idea of a suitable range for both α and β values to create the heterogeneous approach in HMMAS. The lack of statistical significance can be attributed to the relatively good performance of the base algorithm when tested on small-sized TSP instances. However, the minor improvements also suggest that the heterogeneous approach on MMAS can still produce better results while eliminating the need for parameter tuning. Therefore, hereafter, HMMAS will be randomly initialized from a uniform distribution of α from 0 to 2 and β from 1 to 3 unless stated otherwise.

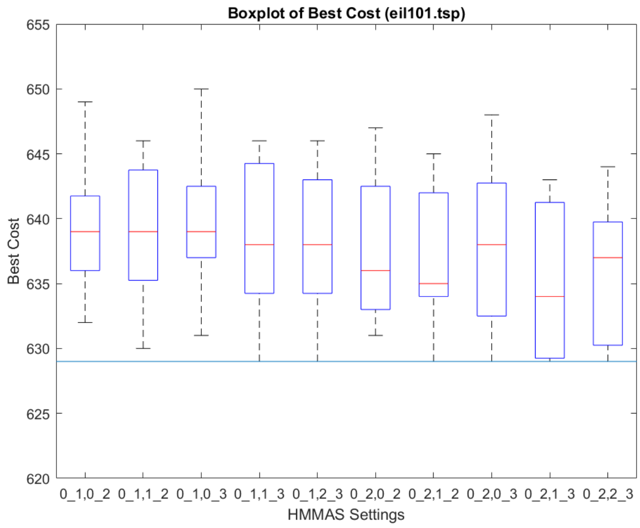


Figure 3.19: Boxplots representing 15 trials of different HMMAS parameter settings as shown in X-axis tested against eil101.tsp. *(Note: 0-1 or 0-2 indicates the range of α values for heterogeneous colony while 0-2 to 2-3 indicates the range of β values).*

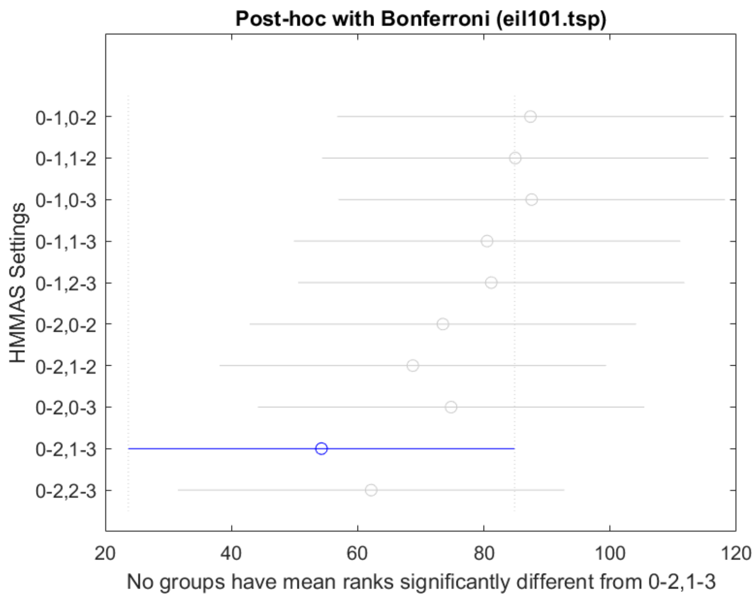


Figure 3.20: Confidence intervals of each of the HMMAS parameter settings with post hoc Bonferroni correction method and its corresponding statistically significant combination when tested on eil101.tsp.

3.4.2 Comparison against the Base Algorithm (HMMAS)

MMAS is one of the best conventional ACO algorithms however the algorithm too underwent rigorous parameter tuning to determine optimal parameter combinations. It has been suggested that each problem instance or each problem may have its optimal parameter values [53][110][45][21]. Even then the parameter tuning process is usually based on prior experience or trial-and-error method hence it is a time-consuming and computationally expensive process to determine the optimal parameter for each and every problem thus almost impossible. Therefore, the heterogeneous approach will be able to overcome this problem by including potential optimal parameter settings within the range of values randomly drawn from the distribution. Equally important, MMAS was designed in such a way that the artificial agents exhibit a ‘slow learning’ behaviour where the colony performs high exploration initially then followed by exploitation [43]. However, different processes (exploration or exploitation) may be required at different stages of the search process thus the conventional MMAS setup may not be optimal. The introduction of the heterogeneous approach will allow the colony to adapt to the process required at the specific stage thus enabling the algorithm to overcome this conundrum.

The heterogeneous approach is compared against MMAS on 4 TSP instances namely eil51, kroA100, d198 and lin318 where all parameters were set according to the authors’ recommendation except that for heterogeneous MMAS where the α and β were randomly initialized from a uniform distribution of the pre-defined range as reported above. The stopping criterion was set at 10 000 iterations in order to have a fair comparison against the original study. Table 3.8 summarizes the comparison results of the performance of HMMAS against MMAS and it obvious HMMAS has a slightly poor performance for eil51 and kroA100 compared to MMAS in terms of average best cost even though both algorithms managed to find the optimum. Having said that, HMMAS managed to locate the optimal solution 10 and 11 times respectively compared to 4 times by MMAS in both TSP instances. This indicates HMMAS has a higher chance of solving a problem to its optimality as opposed to its base algorithm. The reason for slightly poor performance by HMMAS for both eil51 and kroA100 can be attributed to the improved performance of MMAS especially when tested on smaller TSP instances.

Table 3.8: Best, average and worst cost comparison between HMMAS & MMAS tested on several TSP instances for 25 trials with stopping criterion of 10 000 iterations. The optimum for the TSP instances are 426 (eil51.tsp), 21282 (kroA100.tsp), 15780(d198.tsp) and 42029(lin318.tsp). Results in bold represent best in each category.

TSP	Method	Best	Average	Worst	# Optimum Found	≥ 1% dev of opt	≥ 2% dev of opt	p-val
eil51	MMAS	**426**	**427.4**	**430**	4	**25**	**25**	0.8796
	HMMAS	**426**	427.6	431	**10**	23	**25**	
kroA100	MMAS	**21282**	**21299.6**	21390	4	**25**	**25**	0.3078
	HMMAS	**21282**	21316.6	**21379**	**11**	**25**	**25**	
d198	MMAS	15846	15961.12	16137	0	10	22	**1.27e-04**
	HMMAS	**15795**	**15871.68**	**16006**	0	**21**	**25**	
lin318	MMAS	42220	42438.72	42862	0	15	**25**	**7.06e-04**
	HMMAS	**42186**	**42324.44**	**42684**	0	**22**	**25**	

Furthermore, it can be seen from the table and Figure 3.21 that even though HMMAS has a 40% success rate of finding the optimal tour compared to 16% of MMAS, the proposed approach also found higher worst cost in 2 out of 25 trials for eil51.tsp and has a slightly higher median as well as larger interquartile range for both eil51.tsp and kroA100.tsp. It is also important to note that MMAS managed to locate tours much closer to the optimum in both test cases as indicated by the respective boxplots. The performance of HMMAS improved significantly compared to MMAS when tested on larger TSP instances such as d198.tsp and lin318.tsp as can be seen in the table below. HMMAS was able to locate the best cost of 25 trials as well as have lower average best cost and lower worst best cost. The algorithm has a success rate of 90% in locating solutions within 1% of the optimum for both d198.tsp and lin318.tsp even though was not able to locate the optimal tour. Even though the base algorithm indicates drop in performance when tested on larger TSP instances, the improved performance of HMMAS in these larger TSP instances can be attributed to the ability of the algorithm to explore and exploit the search space simultaneously as the search progresses. This is because of the individual 'behavioural traits' introduced by the heterogeneity in the colony that consist of ants that are either explorative or exploitative thus enabling the algorithm to diversify and intensify its search capability in both solution and parameter space albeit in a static approach in the later. Finally, the statistical tests using the Wilcoxon rank-sum test with a 95 % confidence interval indicate that the proposed approach has a statistically significant performance over MMAS when tested on d198 and lin318 respectively while both the algorithms have similar performance for eil51 and kroA100 tsp instances.

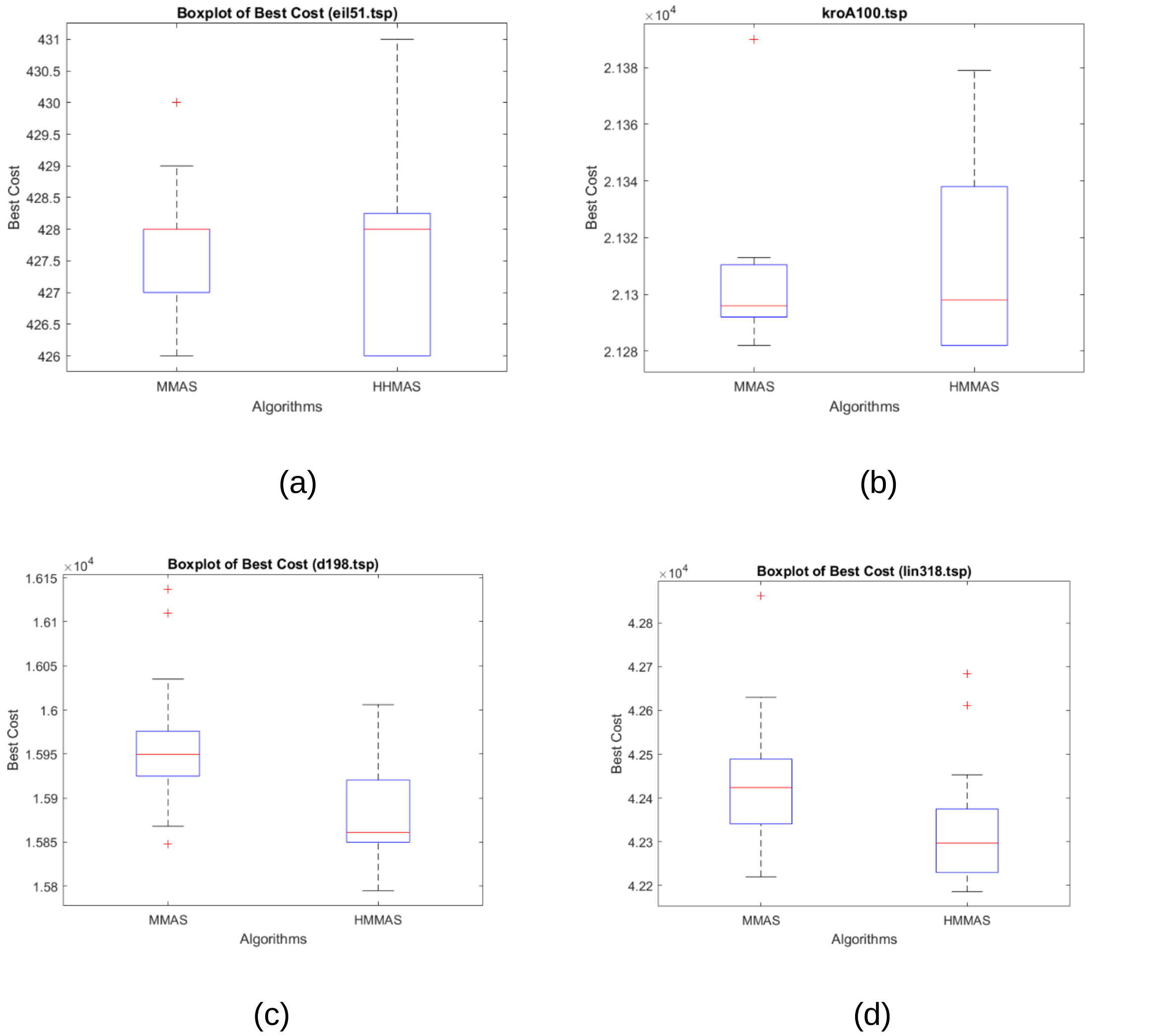
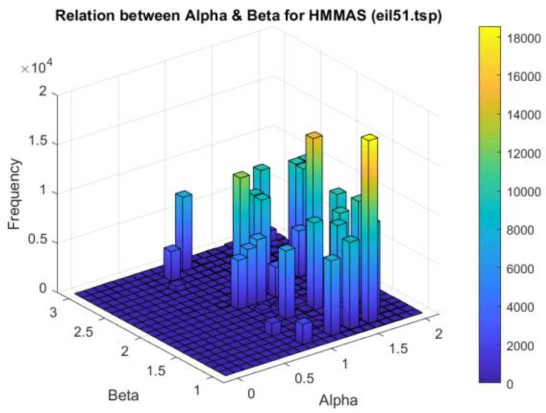


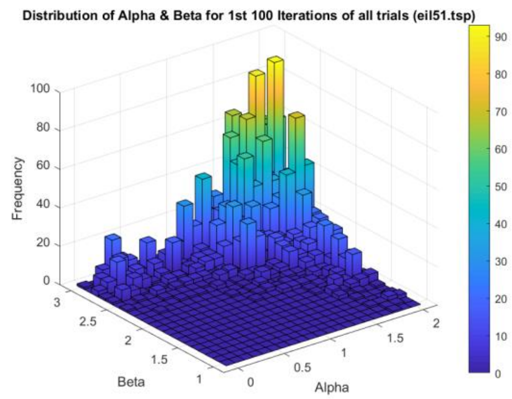
Figure 3.21: Boxplots representing the best cost of 25 trials of both MMAS and HMMAS tested on several TSP instances.

3.4.3 Heterogeneous Ants' Distribution Analysis (HMMAS)

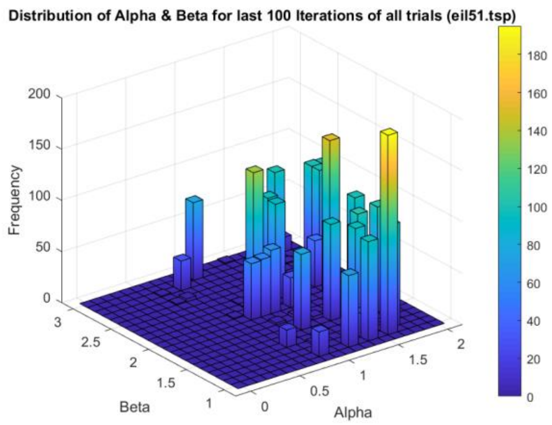
The major difference in MMAS is that the algorithm only allows the iteration-best ant to deposit pheromone on the edges it has traversed. Therefore, it can be noticed that the diversity effect of the whole heterogeneous colony reduces as only one ant may alter the pheromone landscape after finding the optimal or sub-optimal tour unlike HAS where all ants were able to deposit pheromone hence greater effect of diversity can be noticed. Figure 3.22 illustrates the distribution of α and β values that represents the iteration-best ants of all iterations in 25 independent trials as well as the distribution of ants in 1st 100 and last 100 iterations of all the trials when tested on eil51.tsp. It can be seen in Figure 3.22a that the algorithm converges to several sets of iteration-best ant that was able to locate the iteration-best tour in all the trials. Importantly, ants with high beta values tend to locate iteration-best fitness solution in the initial stages of the search process. This indicates the advantage of ants with a higher preference towards heuristics when less information on pheromone landscape. Figure 3.22c suggests that when there is enough pheromone information, then ants with higher preference towards pheromone may exploit this information to locate better solutions.



(a)

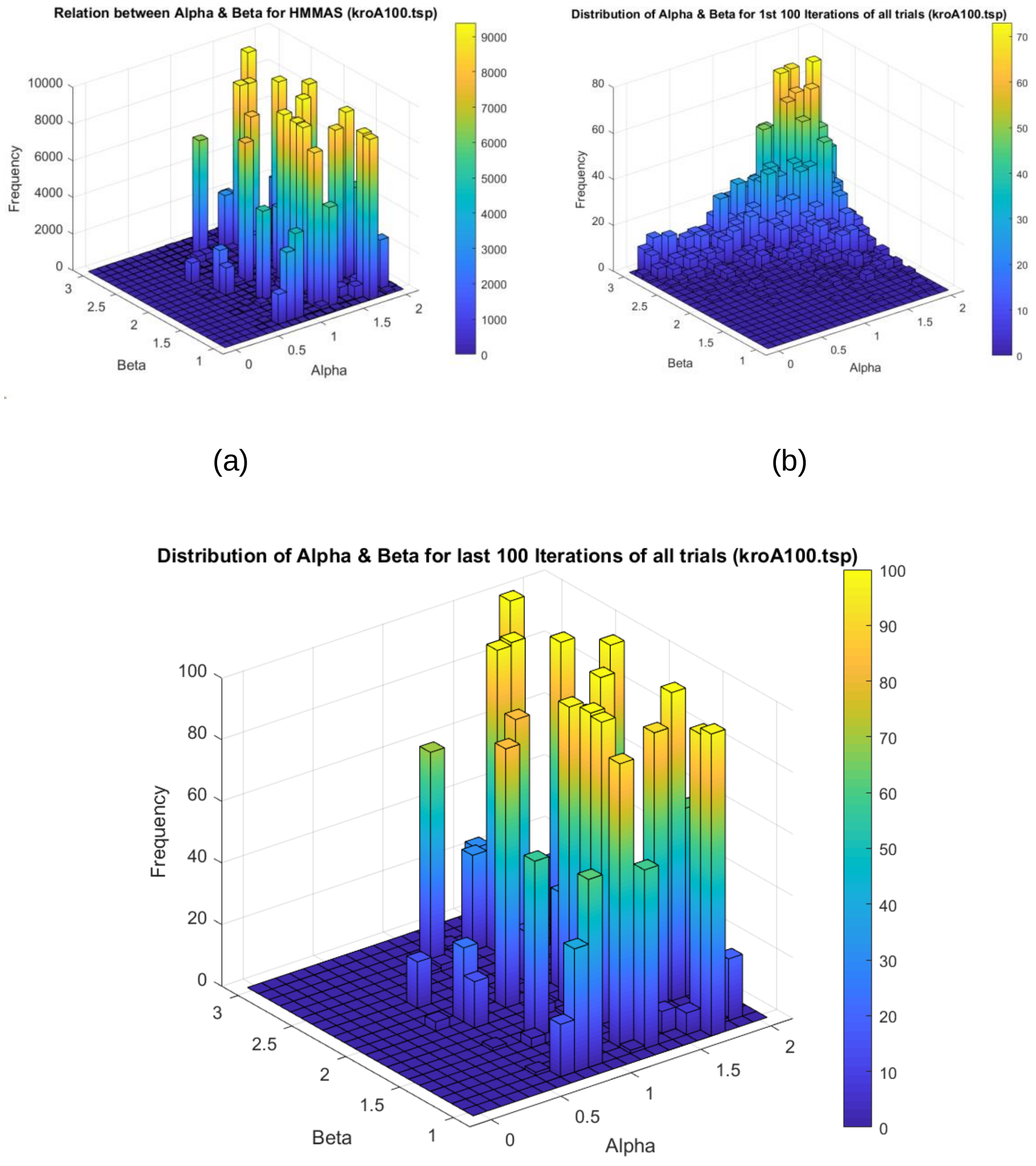


(b)



(c)

Figure 3.22: 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on eil51.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS.



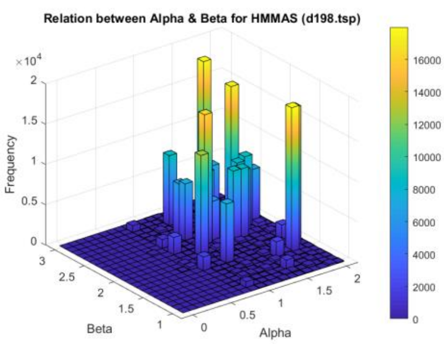
(a) (b)

(c)

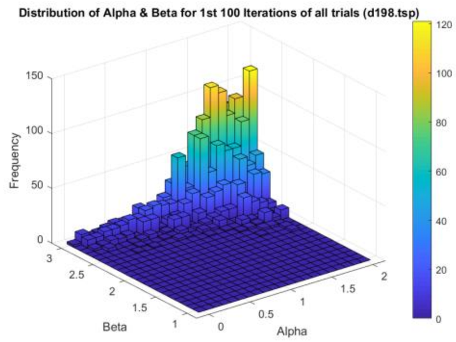
Figure 3.23: 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on kroA100.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS.

Figure 3.23a, b and c illustrate a similar scenario of the distribution of ants for HMMAS where the algorithm again converged to several sets of iteration-best ants.

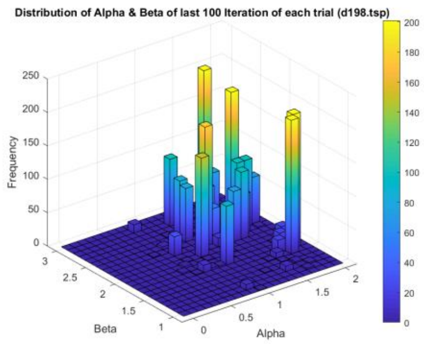
Interestingly, the diversity in the colony allows ants in the higher end of both alpha and beta spectrum to locate better solutions initially before exploitative ants managed to find improved solutions. Meanwhile, Figure 3.24 and 3.25 show the effects of heterogeneity in HMMAS when tested on d198.tsp and lin318.tsp respectively. It can be seen clearly from the figures that the ants that perform well initially are not necessarily the best ants in the later stages of the search process. This is the advantage of the proposed approach where it is capable of passively adapt as the search progresses. The results support the claim that different search strategy or parameter configurations may be required at different stages of the search process while suggesting the heterogeneous approach is able to overcome this problem as the algorithm is able to explore and exploit the parameter space in conjunction with optimising the problem.



(a)

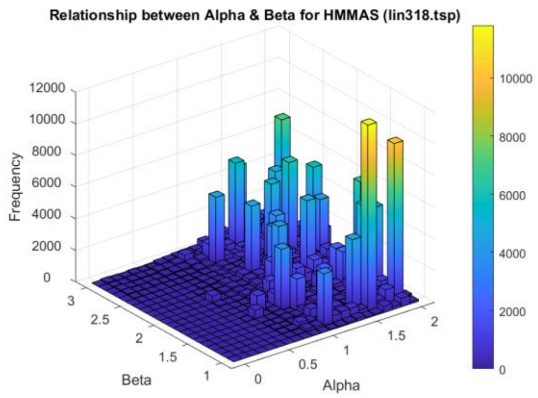


(b)

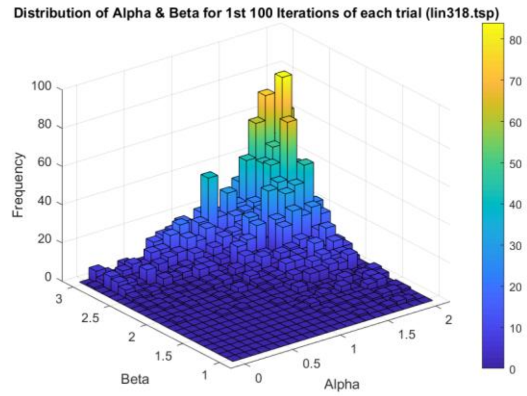


(c)

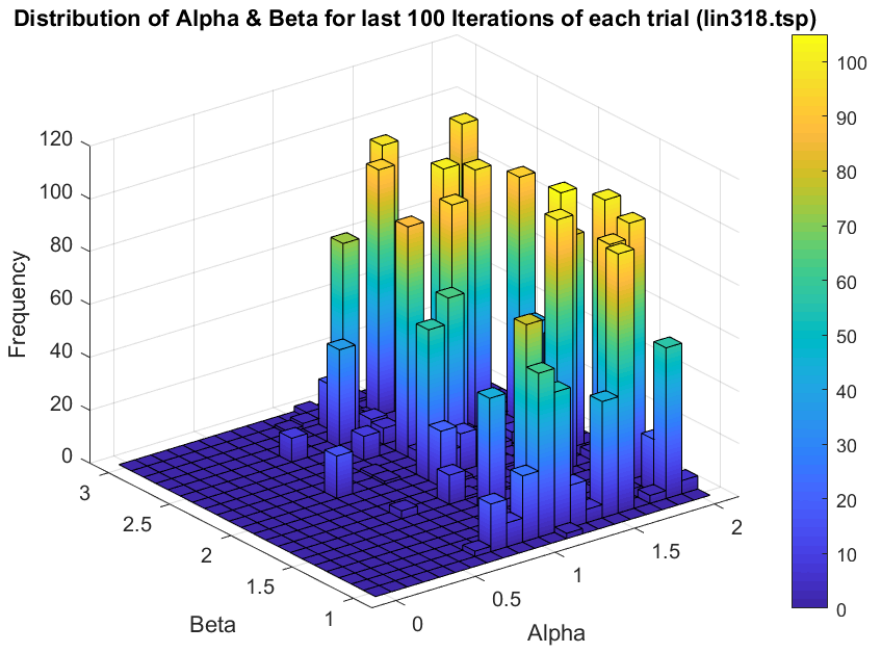
Figure 3.24: 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on d198.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS.



(a)



(b)



(c)

Figure 3.25: 3d histograms representing the alpha and beta values of iteration-best ants for HMMAS on lin318.tsp, (a) Alpha and beta of 1st 100 iterations (c) Alpha and beta of last 100 iterations of HMMAS.

3.5 Concluding Remarks

In summary, a heterogeneous ACO has been introduced which implements artificial ants that have different ‘behavioural traits’ compared to the traditional homogeneous approach. This computational work in ACO is in relation to the biological aspect of real ants where ants are known to have diversity in their population. This chapter presents a heterogeneous approach initialized from a random distribution of a pre-defined range of values of α and β which represents the coefficients related to pheromone trail and heuristics respectively. Most importantly, these two parameters are chosen as heterogeneous elements because these two parameters are responsible for determining the performance of the algorithm. However, most of the previous work focused on either one of these parameters while some studies include too many parameters thus difficult to determine the effects of each parameter that contribute to the improvement in the algorithm.

The overall results indicate that the heterogeneous approach outperforms the conventional ACO algorithms, especially for larger TSP instances. Even though the base algorithms have gone through rigorous parameter tuning, the heterogeneous Ant System (HAS) and heterogeneous Max-Min Ant System (HMMAS) both managed to improve on the performance tremendously. It can be seen in the section above that the percentage of improvement made by HAS is larger compared to that of HMMAS. This is due to the nature of the base algorithm AS itself that allows all ants (heterogeneous ants in HAS) to deposit pheromone on the pheromone landscape that indirectly enables greater contribution of each ant in the decision-making process. Generally, only single ant (global best, iteration-best or alternate between these two) is allowed to deposit pheromone in MMAS. This is the same for HMMAS, therefore, reducing the effects of colony diversity in the performance of the algorithm. However, having said that, MMAS is chosen as the base algorithm hereafter mainly due to being proven in this chapter that HMMAS managed to locate solutions much closer to the optimum. In addition to that, it is also well known that MMAS has an improved performance over AS by being able to locate solutions closer to the optimum hence another reason for better performance over HMMAS in smaller TSP instances.

The proposed approach which is implemented by varying both the α and β parameters within a pre-defined range corroborate the parameter suggestions by Dorigo and Stützle [37, p. 71] to a certain extent but at the same time indicates that determining optimal parameter settings even for a single parameter is considered as a non-trivial task. This can clearly be seen where even though the ants were initialized from the same predefined range of values, the optimal parameters for each TSP instance were different. In other words, the optimal parameter settings are problem-dependent and it is impossible to fine-tune the algorithm for every problem or problem instances. This shows that the heterogeneous algorithm is robust to parameter settings by effectively exploring and sampling the parameter space while concurrently optimizing the problem. Another key advantage of this heterogeneous approach is that the algorithm is not complicated or has minimal complexity unlike some parameter-tuning algorithms such as F-Race [111]. F-Race is an algorithm that intends to optimize the parameters before being applied to another optimization algorithm. This approach is time-consuming and tedious while the algorithm is complex.

This chapter implements a static heterogeneous approach where the parameters assigned to the ants do not change over time and the algorithm explores the parameter space passively. Therefore, the discovery of distinct distributions of parameter settings for α and β is interesting and demonstrates the algorithms' sensitivity to these parameters. Moreover, it is worthy to note the ability of the heterogeneous algorithm to passively adapt to different search strategy over time. One possibility that arises from this chapter is the adaptive approach in Chapter 5 that allows ants to automatically self-adapt over time. In the meantime, the heterogeneous approach continues with a colony of ants initialized from the Gaussian distribution in the next chapter. As the Gaussian distribution is ubiquitous in nature, it provides an interesting proportion for the implementation of the heterogeneous approach.

Chapter 4

Heterogeneous Ant Colony Optimization using a Gaussian Distribution

The Gaussian distribution is widely used in research especially in Artificial Intelligence related studies as it fits the distribution of many natural phenomena such as human heights, blood pressure and many more. This chapter further extends the concept of heterogeneity by introducing and analysing the effects of initializing the α and β parameters of each individual ant in HACO from a Gaussian distribution compared to the previous chapter that used a uniform distribution. In addition, the spread (standard deviation) of the Gaussian distribution is analysed in order to create optimal diversity in the colony. To assess the performance of this method, comparisons with state-of-the-art heterogeneous ACO algorithms are conducted on several TSP and Printed Circuit Board (PCB) problem instances. Overall, empirical results indicate the diversity introduced by the Gaussian heterogeneous approach improves the algorithm‘s performance over other approaches and alleviates the parameter tuning process required in finding optimal parameter settings as well as tackling the management of exploration-exploitation within ACO.

4.1 Motivation

Premature convergence occurs in most of the population-based optimization algorithms mainly due to the imbalance of search information propagation that leads the population to converge towards sub-optimal solutions and combined with the loss of population diversity will cause the algorithm to not be able to escape from this phenomenon. It is even worse in the case of conventional ACO and even in some of the state-of-the-art ACO algorithms as the population is homogeneous where it is usually set as such that all ants will have identical parameters. Therefore, introducing a diverse population via heterogeneity can help the algorithm to maintain diversity thus indirectly improve the exploration and exploitation capability of the algorithm. Empirical results of the heterogeneous approach introduced via the uniform distribution in the previous chapter indicate that the algorithm is able to improve on the performance of the base algorithms, especially in larger TSP instances. However, there were some indications that the algorithm was stuck in local optima and was unable to perturb the solutions found in order to locate better solutions. This can be due to the inclusion of ants with sub-optimal performance during the initialization period as all ants have an equal probability of being randomly initialized with any values within the pre-defined range. The uniform distribution introduced a wide range of heterogeneity which includes a large number of extremal values in the specified range. Many natural phenomena are distributed according to a normal or gaussian distribution where the majority of the values are located around the mean and comparatively few around the extremal values. This chapter investigates the use of a Gaussian distribution for heterogeneous ACO. Even if the scope of the study is widened to other areas such as physics or psychology, many random variables and data are found to follow a Gaussian distribution trend [112]. The data that does not exactly fit the Gaussian distribution can also be approximated as one because of the central limit theorem that states the normalized sum of randomly independent variables produces Gaussian-like distributions regardless of the source population distribution [113]. Several general attributes mimic the Gaussian distribution such as human heights or exam scores where the attributes have a small probability of occurring at the high and low tails of the distribution while a large probability of occurring around the mean. Therefore, this chapter implements a heterogeneous approach where the ants are randomly initialized from a Gaussian distribution with a pre-defined mean and standard deviation.

4.2 The Framework

This chapter focuses on implementing the heterogeneous approach on Max-Min Ant System (MMAS) only because empirical results in Chapter 3 have shown that HMMAS has better performance compared to HAS. The basic algorithmic framework used in this study is still the same as the previous chapter except that the ants' colony is randomly initialized from Gaussian distribution rather than a uniform distribution. The same probabilistic rule used in the previous chapter (Equation 2) is maintained in this study, modified to accommodate the heterogeneous approach. Figure 4.1 indicates the example of a heterogeneous ant colony initialized from a Gaussian distribution with a mean α=1 and mean β=4 while the standard deviation is set to 0.2.

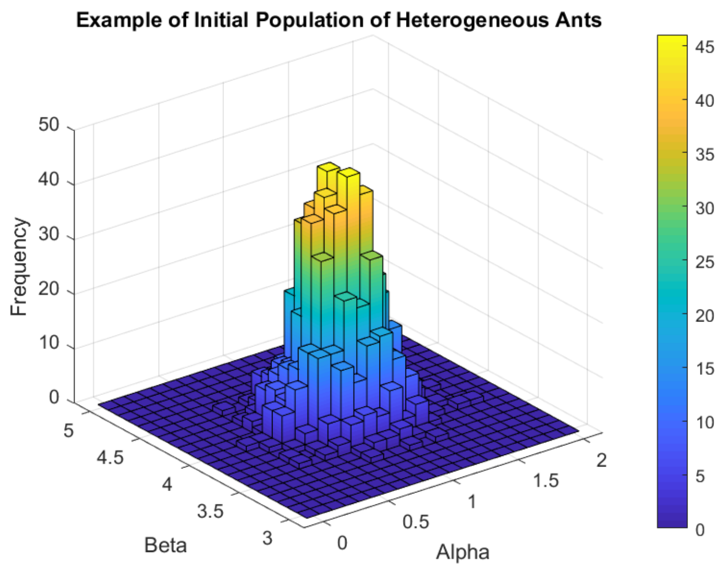


Figure 4.1: Example of the initial population drawn from normal distribution consisting of 25 different populations representing 25 trials.

Algorithm 1: Pseudocode of Gaussian Heterogeneous MMAS

1. Input: Distance Matrix of TSP / PCB Drilling;
2. Initialize parameters;
3. Initialize ants:
4. **for** *i* = 1 : number of ants **do**
5. Alpha(i) = normrnd (mean α, s.d α);
6. Beta(i) = normrnd (mean β, s.d β);
7. **end for**
8. Start Iteration:
9. **for** *it* = 1 : Max Iteration **do**
10. **for** *k* = 1 : number of ants **do**
11. Position each ant on starting node;
12. **while** *TourSize* < *n* + 1 **do**
13. Tour Construction;
14. **end while**
15. **end for**
16. **end for**
17. Update Solution;
18. Update Pheromone;
19. Pheromone Evaporation;
20. Check if the termination condition is met;
21. **if** True **then**
22. Go to 26;
23. **else**
24. Go to 8;
25. **end if**
26. End

The Gaussian distribution, also known as normal or bell-shaped distribution is a continuous probability distribution that is commonly associated with real-valued variables in nature. The two most important parameters in a Gaussian distribution are the mean, μ, and standard deviation, σ parameters. Therefore, the settings of these parameters for both α and β are key to the performance of the Gaussian heterogeneous approach.

The experiments shown later will initialize the colony randomly from a Gaussian distribution with mean centred around α and β coefficients suggested in [5] while the standard deviation is pre-determined via parameter exploration experiment. Algorithm 1 represents the pseudocode of Gaussian Heterogeneous MMAS (known as GHMMAS hereafter) and it can be seen from lines 4 to 7 where the 'normrnd' function in MATLAB was used to mimic the random behavioural traits of agents in a colony. The α and β coefficients were set to 1 and 2 respectively in the homogeneous approach in [5] while this study implements a heterogeneous approach from a Gaussian distribution with mean $\alpha = 1$ and mean $\beta = 2$ respectively. This is to create a colony of ants where each ant will have a pair of parameters close to the values suggested in [5]. In addition to that, exploratory experimentation was conducted in order to determine the right amount of spread (standard deviation) to create optimal diversity in the heterogeneous colony. A narrow Gaussian distribution may eliminate ants with α and β values within the suggested values of [6] and will limit the benefits of heterogeneity while too wide a Gaussian spread may include ants with poor performing values thus rendering the performance of the algorithm.

4.2.1 Exploring Gaussian Properties

Extensive experiments were conducted in order to determine the optimal standard deviation for the implementation of GHMMAS. MMAS was used as the base algorithm while the parameters were set according to suggestion in [5] except that the ants were initialized randomly from a Gaussian distribution with mean α and β were set to 1 and 2 respectively and the standard deviations for both α and β are varied from 0.1 to 0.4 only because larger standard deviation may include poor parameter values that may cause stagnation behaviour in the algorithm as suggested in [6]. Therefore, smaller standard deviation allows the α and β values to be centered around and much closer the mean values which were suggested to be optimal values. A change in the mean values will cause a shift in the symmetrical axis of the distribution while a variation of the standard deviation will increase or decrease the sharpness and the spread of the curve. The stopping criterion for this exploratory experiment was set at 1 000 iterations as to reduce the computational cost.

Table 4.1 compares the performance of GHMMAS with various standard deviations for both α and β tested on TSP instances of eil51 and eil101. It can be seen that GHMMAS with parameter settings in row 5 (standard deviations of 0.2) has the best performance compared to other parameter combinations. The algorithm with these settings was able to locate the optimal solutions of eil51 and eil101 4 times and once. In addition to that, the algorithm has an average best cost of 427.9 and 641.9 as a result of 15 trials when tested on eil51 and eil101.tsp respectively. However, the Kruskal Wallis non-parametric statistical test produced p-values of 0.1732 and 0.1993 for both eil51 and eil101.tsp thus indicating there are no significant differences in terms of the performances of the algorithm with different settings. Figures 4.2 and 4.3 illustrate the boxplots of the best cost of 15 trials of GHMMAS with different standard deviation combinations for eil51.tsp and eil101.tsp respectively. It can be seen in Figure 4.2 that only 2 out of 12 parameter combinations were not able to locate the optimal solution and 4 parameter combinations have a median of 428. Additionally, the figure also illustrates the interquartile range for GHMMAS with mean $\alpha = 1$, $\sigma = 0.2$ and mean $\beta = 2$, $\sigma = 0.2$ is from 426.5 to 429.5 thus supporting the results in Table 4.1 by indicating that this parameter combination has the best performance overall.

Table 4.1: Results of GHMMAS with variation in distribution spread (standard deviation (s.d)) of α and β tested on eil51.tsp (Opt = 426) and eil101.tsp (Opt = 629). Values in bold are the best in each category. Results are of 15 trials, each trial = 1 000 iterations.

Num	Mean α	S.d α	Mean β	S.d β	Best Cost		Average		Worst Cost		No of Opt	
					eil51	eil101	eil51	eil101	eil51	eil101	eil51	eil101
1	1	0.1	2	0.1	**426**	633	429.1	643.3	435	**652**	3	0
2	1	0.2	2	0.1	**426**	**629**	428.7	647.3	433	659	2	**1**
3	1	0.3	2	0.1	**426**	631	430.6	648.9	436	665	1	0
4	1	0.1	2	0.2	**426**	638	428.9	648.9	432	660	1	0
5	1	0.2	2	0.2	**426**	**629**	**427.9**	**641.9**	**431**	657	**4**	**1**
6	1	0.3	2	0.2	**426**	631	429.8	643.1	435	658	2	0
7	1	0.1	2	0.3	**426**	630	429.6	642.9	439	655	2	0
8	1	0.2	2	0.3	**426**	630	428.6	645.6	**431**	658	1	0
9	1	0.3	2	0.3	427	635	429.3	644.6	435	658	0	0
10	1	0.1	2	0.4	**426**	634	429.4	642.0	432	654	1	0
11	1	0.2	2	0.4	428	639	429.9	646.9	434	657	0	0
12	1	0.3	2	0.4	**426**	632	428.9	645.1	437	666	2	0

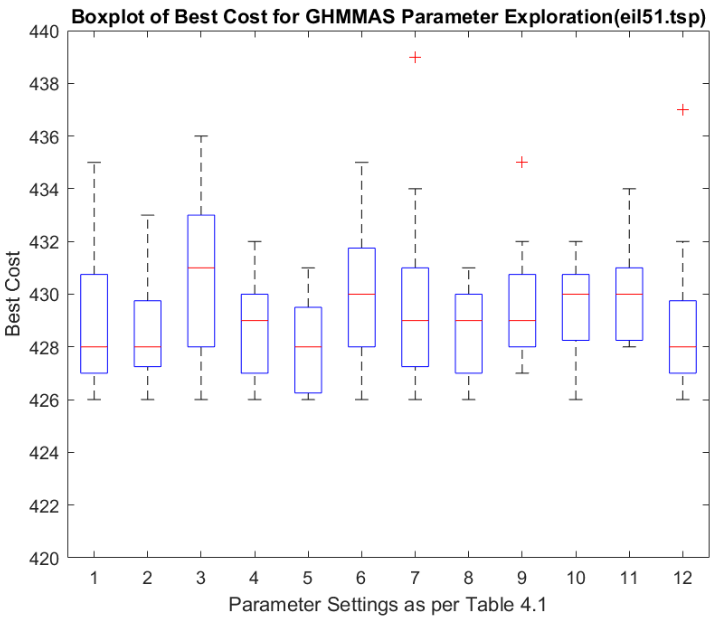


Figure 4.2: Boxplots representing 15 trials of different GHMMAS parameter settings as shown in X-axis tested against eil51.tsp. *(Note: Ex: 0.1, 0.1 indicate the standard deviation for mean α and β). X-axis represents 'Num' column in Table 4.1.*

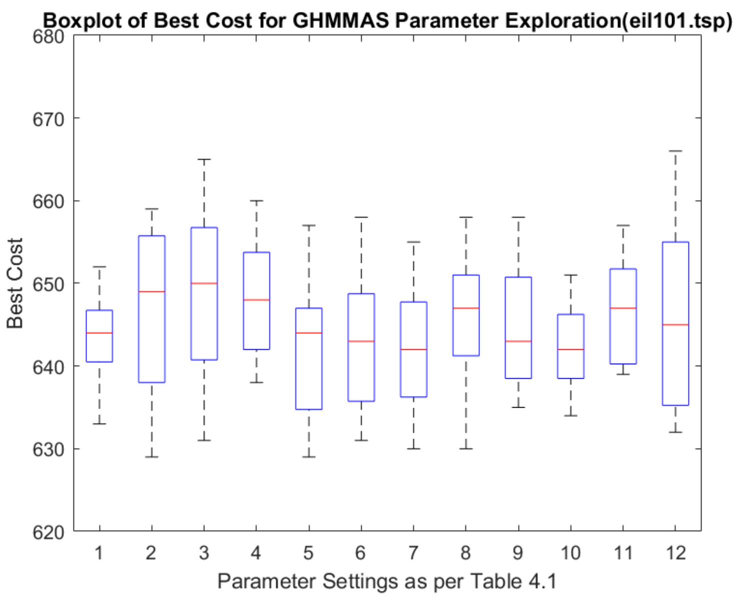


Figure 4.3: Boxplots representing 15 trials of different GHMMAS parameter settings as shown in the X-axis tested against eil101.tsp. *(Note: Ex: 0.1, 0.1 indicate the standard deviation for mean α and β). X-axis represents 'Num' column in Table 4.1.*

Figure 4.3 shows that only two parameter combinations for GHMMAS managed to locate the optimal solution. Although the median of GHMMAS with mean $\alpha = 1$, $\sigma = 0.2$ and mean $\beta = 2$, $\sigma = 0.2$ is slightly higher, this parameter combination has the lowest interquartile range which is from 634 to 647 compared to others. Therefore, experiments hereafter involving GHMMAS will use this parameter combination.

4.3 Experimental Setup, Results & Discussion

4.3.1 Experimental Setup

The experiments were conducted on an Intel Core i7 CPU-based computer running Windows 7 equipped with 4GB RAM. The proposed approach used Max-Min Ant System (MMAS) as the base algorithm which is developed using the MATLAB version R2015a mainly because the previous chapter has shown that the heterogeneous approach on MMAS has a marked performance compared to that implemented on AS. MMAS was chosen clearly because it is the best conventional ACO algorithm [5]. Various studies have acknowledged both empirically and theoretically that different optimal parameter settings are required for different problem instances of the same problem and also for different problems. Each algorithm is tested using several medium-sized TSP instances taken from TSPLIB namely kroA100, d198, pr226, and lin318 TSP instances while the optimal solution of each instance can be referred at [8]. The approach is also compared against the Max-Min Paraconsistent Ant Algorithm (MMPAS) [114] tested on the Printed Circuit Board (PCB) drilling problem instances [106]. This is to show that the approach is independent and robust to different problems and also problem sizes. The PCB drilling problem is an example of a real-world problem where the time taken by the CNC machine reduces the efficiency of the drilling process thus indirectly affects the output of the manufacturing line and the company’s profit. The movement of the CNC machine has to be optimized in order to improve efficiency by minimizing the machine's drilling path. Therefore, the algorithm explores the PCB layout to identify the optimal drilling path. All experimental results are from 20 trials hereafter unless stated otherwise. This conforms with the general number of trials in any swarm intelligence-based optimization algorithm study which are usually between 15 to 30 trials and also to achieve a fair and stable comparison against other algorithms. The function evaluations for all the experiments were set as k.n.10000 where k=1 for symmetrical TSPs used, n=number of cities of the TSP instance and 10 000 is the maximum number of iterations unless stated otherwise. The computation time for the algorithm to locate the optimal solution is not used as a performance indicator in this study because this measure is not accurate as the algorithm's performance is dependent on various elements such as the hardware specifications, processor speed, how the program is coded et cetera.

4.3.2 Comparison against the Base algorithm & HMMAS

The proposed approach is compared against MMAS and HMMAS on four TSP instances namely kroA100, d198, pr226, and lin318.tsp. All three algorithms managed to find the optimal solution of 21282 when tested on kroA100.tsp and the GHMMAS has the lowest mean out of 20 trials followed by MMAS and HMMAS respectively while GHMMAS could not improve on the worst cost found by HMMAS which was 21379. Next, the algorithms were tested on d198.tsp with an optimum of 15780 where GHMMAS has the lowest average of 15866.95 and found the lowest worst cost of 15944. The best fitness solution found by GHMMAS is 15801 which is not close to the best solution found by HMMAS even though neither algorithm could locate the optimal solution. Lastly, it can be seen that the proposed approach has marked improvement over other algorithms when tested on pr226.tsp and lin318.tsp respectively. The Gaussian heterogeneous approach managed to locate the best solution, an average that is closer to the optimum, and lowest worst best cost when tested on both TSP instances mentioned above. The improvement noticed on GHMMAS over HMMAS and MMAS can be attributed to the reduced number of ants in the extremal and initializing the colony of ants through the Gaussian distribution centred around the optimal parameter whereas HMMAS, which uses uniform distribution, may include ants in the extremities that can be detrimental to the performance of the algorithm. Interestingly, this approach also conforms to an extent to the parameter combination suggestion in [5] and [6] because the Gaussian heterogeneous approach initializes the ants with α and β values centred around the 'optimal' values suggested by the original authors. However, the diversity created by the Gaussian distribution allowed the heterogeneous ants to have improved exploration-exploitation balance thus achieving improved performance over the uniform as well as the homogeneous approach. Furthermore, it is also important to note that the performance of the homogeneous approach deteriorates significantly with the increase in problem size. The absence of diversity in the colony prevents the algorithm from efficiently alternate between exploring the search landscape and exploiting the solution found because it has been acknowledged that different search strategies are required at a different stage of the search process in order to achieve optimal performance in an optimization algorithm [21] [12].

Table 4.2: Comparison between GHMMAS, HMMAS & MMAS tested on several TSP instances for 20 trials with stopping criterion of 10 000 iterations. The optimum for the TSP instances are 21282 (kroA100.tsp), 15780 (d198.tsp), 80369 (pr226.tsp) and 42029 (lin318.tsp). Results in bold represent best in each category.

TSP	ALGORITHM	BEST COST	AVERAGE	WORST COST
Kroa100	MMAS	**21282**	21302.3	21390
	HMMAS	**21282**	21310.55	**21379**
	GHMMAS	**21282**	**21301.9**	**21379**
d198	MMAS	15848	15958.2	16137
	HMMAS	**15795**	15879.3	16006
	GHMMAS	15801	**15866.95**	**15944**
pr226	MMAS	80691	80983.8	81330
	HMMAS	80464	80682.3	81011
	GHMMAS	**80401**	**80552.6**	**80851**
lin318	MMAS	42220	42425.5	42862
	HMMAS	42186	42301.95	42612
	GHMMAS	**42110**	**42245**	**42453**

Table 4.3: Results of the Wilcoxon rank-sum test conducted on GHMMAS, HMMAS & MMAS based on the solutions found in each of the 20 trials. Results in bold represent statistical significance.

TSP	GHMMAS vs HMMAS	GHMMAS vs MMAS	HMMAS vs MMAS
kroA100	0.4331	**0.0659**	0.671
d198	0.5425	**3.47E-05**	**1.30E-03**
pr226	**0.064**	**5.83E-04**	**3.60E-03**
lin318	**0.0514**	**1.80E-05**	**3.92E-04**

Table 4.3 represents the p-values of the Wilcoxon rank-sum test conducted on the performance of all three algorithms with a confidence interval of 90% where a p-value of 0.1 or lower indicates that the performance in-test is statistically significant. It can be seen that the performance of GHMMAS is statistically significant over HMMAS on pr226.tsp and lin318.tsp but not on the other two TSP instances. This is likely to be due to the fact that the instances are relatively small and so the differences between the algorithms are harder to determine. Initialization of the colony from the Gaussian distribution creates better diversity by reducing the number of poor-performing ants hence the improved performance in GHMMAS. In the meantime, the performance of GHMMAS is statistically significant over MMAS in all four TSP instances while the performance of HMMAS is significantly different in 3 out of 4 TSP instances thus indicating that both heterogeneous algorithms significantly outperform MMAS, in terms of quality of solutions found, robustness and scalability of the problem size. Moreover, Figure 4.4 illustrates the boxplots representing the best cost of 15 trials of each algorithm when tested on all four TSP instances. It can be seen in all 4 TSP instances that GHMMAS has the lowest median and the lowest worst best cost while also being able to locate the best fitness solution in 3 out of 4 TSP instances. The empirical results suggest that by enabling ants to be randomly initialized with different behavioural traits, a colony may consist of a mix of explorative ants as well as exploitative ants thus allowing the algorithm the ability to explore the search landscape and exploit the solution found throughout the search process.

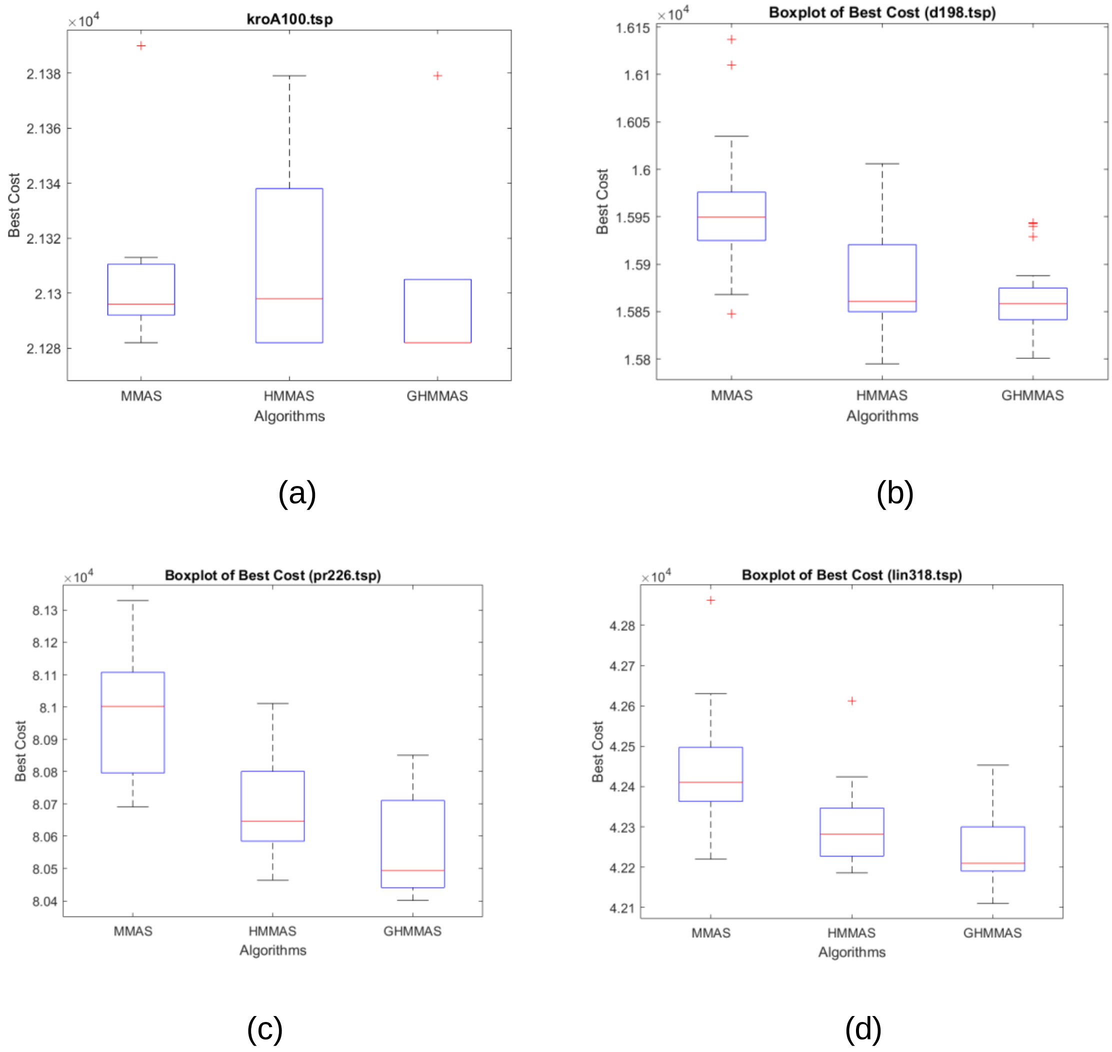


Figure 4.4: Boxplots representing the best cost of 20 trials of MMAS, HMMAS, and GHMMAS tested on several TSP instances.

4.3.3 Printed Circuit Board (PCB) Holes Drilling Problem

The PCB drilling problem is similar to the TSP where the objective function is to locate the minimum distance from one hole to another for the CNC machine to travel for the drilling process. The location of holes drilled on a PCB is a crucial process especially in companies producing electronics. The optimization of the drilling of these holes produces a notable reduction in drilling time thus able to hugely improve productivity, reduce time and cost as well as increase revenue indirectly. A detailed review on PCB hole-drilling optimization using swarm intelligence can be found at [115]. Unlike TSP instances that have abundance of literature, limited work has been conducted in optimizing the PCB drilling problem especially by using ACO approach. However, some of the studies that uses ACO to solve this particular problem have repeatability problem due to important information not reported such as the stopping criterion [116][117], no general problem instances such as TSP used to test the algorithm [118] or computational time was used as stopping criterion (not repeatable due to different processing power contributes to the performance of the algorithm)[119], [120]. Due to this, GHMMAS was tested on several PCB instances taken from TSPLIB [109] and compared against Max-Min Paraconsistent Ant Algorithm (MMPAS) [114] which implements a hybrid approach of MMAS and Paraconsistent Logic (PL) by deploying the concept of recruitment learning. As the search progresses, the ants become more knowledgeable in regard to the search landscape hence able to explore efficiently in order to locate the optimal solution. For a fair comparison, the parameter settings of GHMMAS were set similar to [114] except that the ants were initialized randomly from a Gaussian distribution with mean $\alpha = 1$, mean $\beta = 2$ and standard deviations for both were set to 0.2. The stopping criterion was set at 2000 iterations and no local search procedure was applied just as in [114]. Table 4.4 shows the comparison between GHMMAS and MMPAS in terms of the best fitness solution found and the mean of the fitness solution over 20 trials. It can be noticed that GHMMAS was not able to improve on the best fitness solution found by MMPAS for a280.tsp and tsp225.tsp respectively although the local optimum found by GHMMAS was close However, GHMMAS found the best fitness solution when tested on pcb442.tsp as compared to MMPAS. In addition to that, the mean performance of GHMMAS is closer to the optimum in comparison to MMPAS for all three PCB drilling instances. The improvement in performance especially in larger PCB instances can be attributed to the high number of ants that introduce a huge diversity in the colony thus enabling the algorithm to fully explore the search space as well as further perturb the solution found to locate optimal solutions. The individual behavioural traits in the ants enabled the algorithm to escape from stagnation behaviour by allowing the ants to have individual perspective in tackling the search landscape rather than following the same tour. It is also important to note that the proposed approach was able to achieve close to optimal performance even when the algorithm is stopped at 2 000 iterations rather than 10 000 iterations as in previous experiments. This gives an early indication that the heterogeneous approach is able to achieve excellent performance even within budget (stopping criterion) limitations.

Table 4.4: Best cost and average best cost comparison between GHMMAS and MMPAS tested on several PCB instances for 20 trials with stopping criterion of 2 000 iterations. Results in bold represent best in each category.

PCB	OPTIMUM	ALGORITHM	BEST COST	AVERAGE
a280	2579	MMPAS	**2593**	2642.77
		GHMMAS	2597	**2635.62**
tsp225	3919	MMPAS	**3937**	4012.07
		GHMMAS	3943	**3982.7**
pcb442	50778	MMPAS	52129	53290.1
		GHMMAS	**51966**	**52671**

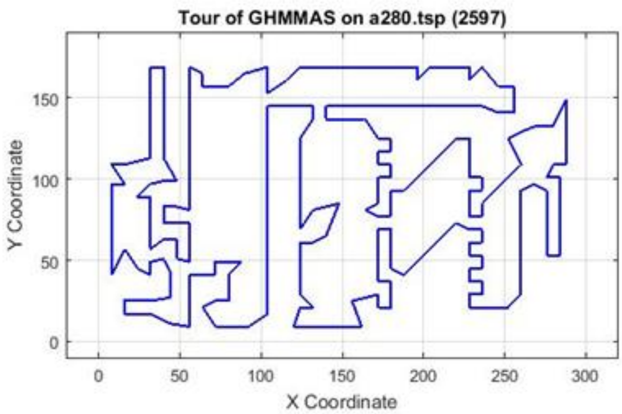


Figure 4.5: The optimized tour (2597) found by GHMMAS when tested on a280.tsp.

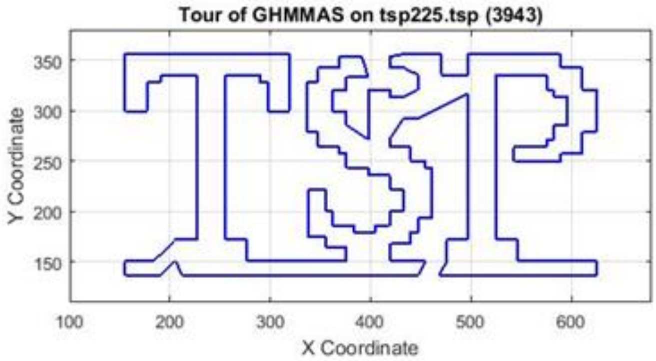


Figure 4.6: The optimized tour (3943) found by GHMMAS when tested on tsp225.tsp.

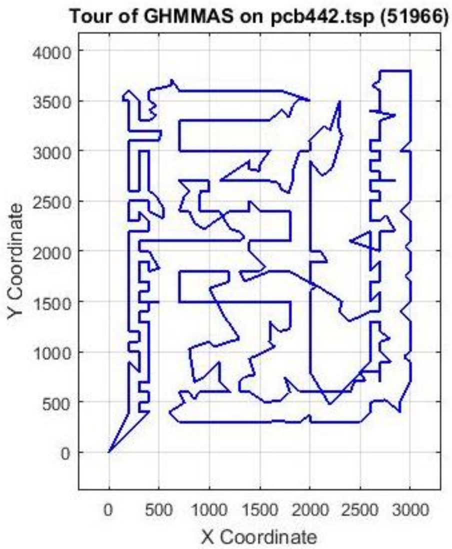
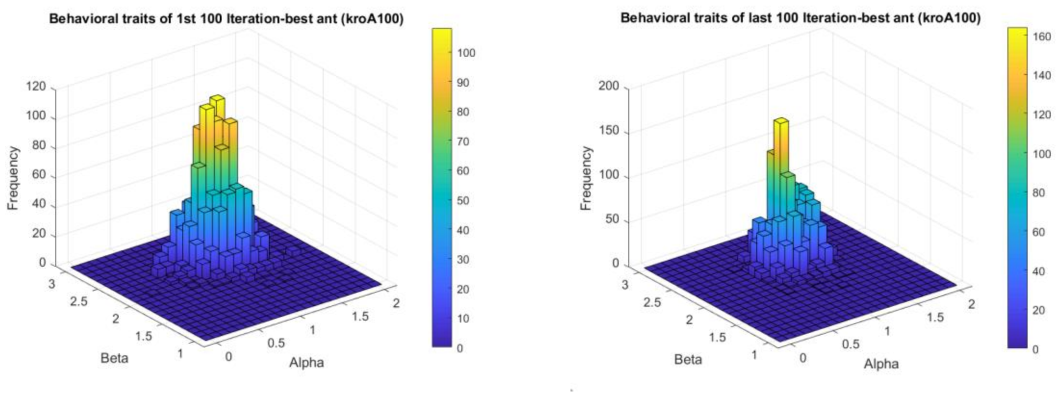


Figure 4.7: The optimized tour (51966) found by GHMMAS when tested on pcb442.tsp.

Figure 4.5 to 4.7 illustrate the best tour found by GHMMAS for a280.tsp, tsp225.tsp, and pcb442.tsp respectively where the tours found are close to the optimal except for pcb442.tsp. It is clearly noticeable the overlapping path in pcb442.tsp that can be further optimized possibly by using local search procedures such as 2-opt or 3-opt. This will create a hybrid approach that can further improve on the fitness solution found as well as reduce the computational time i.e number of iterations required to find good to optimal solutions. However, the local search procedure is not implemented in this chapter (but will be explored in Chapter 5) in order to highlight the ability of the heterogeneous approach in improving the performance of the base algorithm as well as achieve better or comparable results against some of the state-of-the-art algorithms.

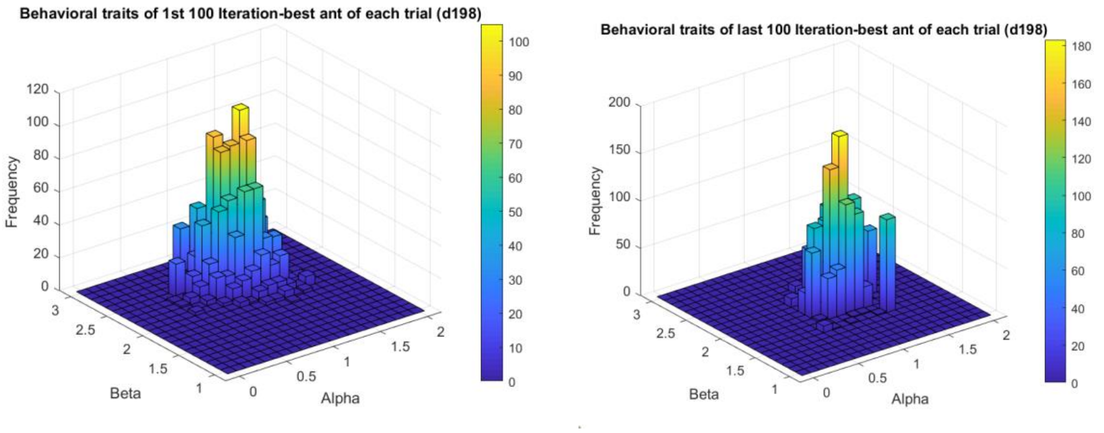
4.3.4 GHMMAS Ants' Distribution Analysis

Figure 4.8 illustrates the alpha and beta values of the iteration-best ant for GHMMAS when tested on kroA100.tsp where Figure 4.8a represents alpha and beta values of the first 100 iteration-best ants of each trial while Figure 4.8b shows the values for the last 100 iteration-best ants. It is noticeable that ants with $\beta > 2$ were able to locate the iteration-best fitness solutions during the early stages of the search process. This is likely to be due to a lack of information on the pheromone landscape at this early stage for the ants to exploit hence the ants with a higher preference on heuristics were able to locate good solutions that will then be used for exploitation. Interestingly, ants with a slightly higher preference towards the pheromone were able to locate iteration-best fitness solution with the increase in pheromone deposited on the landscape in later stages of the search process. This shows that the algorithm is able to alternate between exploration and exploitation whenever necessary. In addition to that, the algorithm also is able to explore the parameter space making it robust to parameter selection.



(a) (b)

Figure 4.8: 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on kroA100.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS.

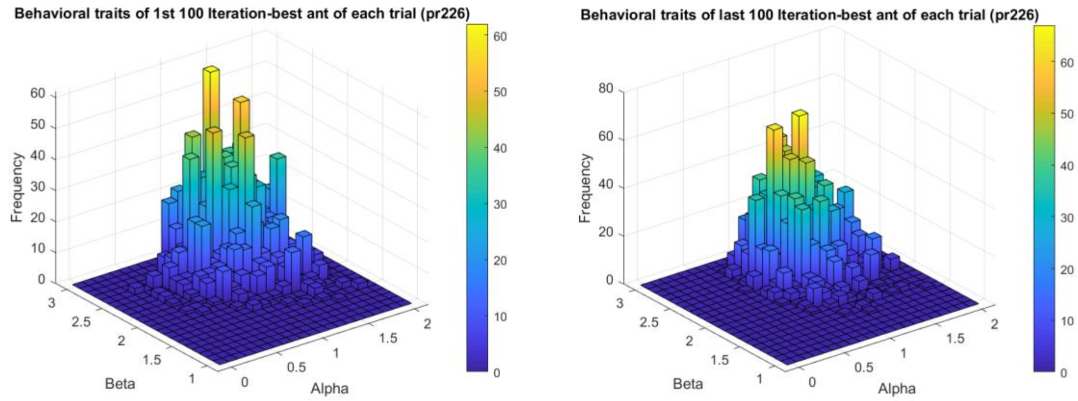


(a) (b)

Figure 4.9: 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on d198.tsp, (a) Alpha and beta of 1^{st} 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS.

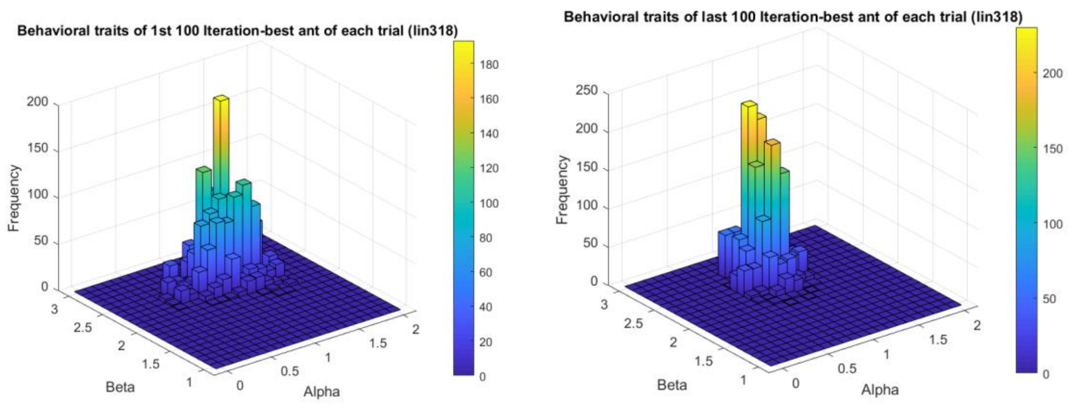
Figure 4.9a and 4.9b represent the 3d histograms for GHMMAS when tested on d198.tsp. In this problem instance, similar to the previous figures where it can be seen clearly that ants with a higher preference towards heuristic information perform well during the early stages while ants with slightly lower dependency on heuristics perform well during later stages. Importantly, the colony was able to explore both the solution and parameter landscape when tested on d198.tsp. This highlights the advantage of the proposed approach over conventional, homogeneous approach by exploring the parameter space to find the optimal parameter as per the situation need throughout the search process. It is also important to highlight the alpha values were between 1 and 1.5 which supports the suggested values in but at the same time confirms the importance of the proposed approach that allows continued exploration of both search space and parameter space.

The relationship between alpha and beta for GHMMAS on pr226 is more diverse when compared to the relationship in the previous experiments above. The algorithm explores a huge area of the parameter space as shown in Figure 4.10a and 4.10b respectively. It is also important to note that an increase in the number of ants with a higher preference towards pheromone in later stages of the search process as can be seen in Figure 4.9b.



(a) (b)

Figure 4.10: 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on pr226.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS.



(a) (b)

Figure 4.11: 3d histograms representing the alpha and beta values of iteration-best ants for GHMMAS on lin318.tsp, (a) Alpha and beta of 1st 100 iterations (b) Alpha and beta of last 100 iterations of GHMMAS.

In the meantime, Figures 4.11a and 4.11b represent the distribution of alpha and beta values for GHMMAS when tested on lin318.tsp. Both figures suggest that the algorithm is capable of implementing different search strategies at various stages of the search process. This will indirectly enable the algorithm to escape from local optima and stagnation behaviour. The results also suggest that the tedious job of parameter tuning can be overcome by implementing the heterogeneous approach that automatically explores the parameter landscape to locate the optimal parameters for different problem instances. One thing to note is that the distributions look quite different for some of the problem instances. This reinforces the idea that some sort of adaptive technique is required, either passive as in this chapter or active as in the following chapter.

Lastly, Figure 4.12 a, b, c, and d illustrate the standard deviations of the fitness solutions found by the ants in every iteration for GHMMAS when tested on several TSP instances for a single trial. The standard deviations represent diversity of the colony during the search while higher standard deviations indicate the algorithm in explorative mode and lower standard deviations suggest the algorithm is in exploitative mode. It can clearly be seen in the figures that the standard deviations decrease gradually over time thus indicating a clear switch of the algorithm from explorative behaviour to exploitative. Hence, these demonstrate that the proposed approach are able to maintain colony diversity throughout the search process, able to balance between exploration and exploitation of the search space. The figures suggest convergence to an acceptable fitness solution while also able to prevent stagnation behaviour from occurring.

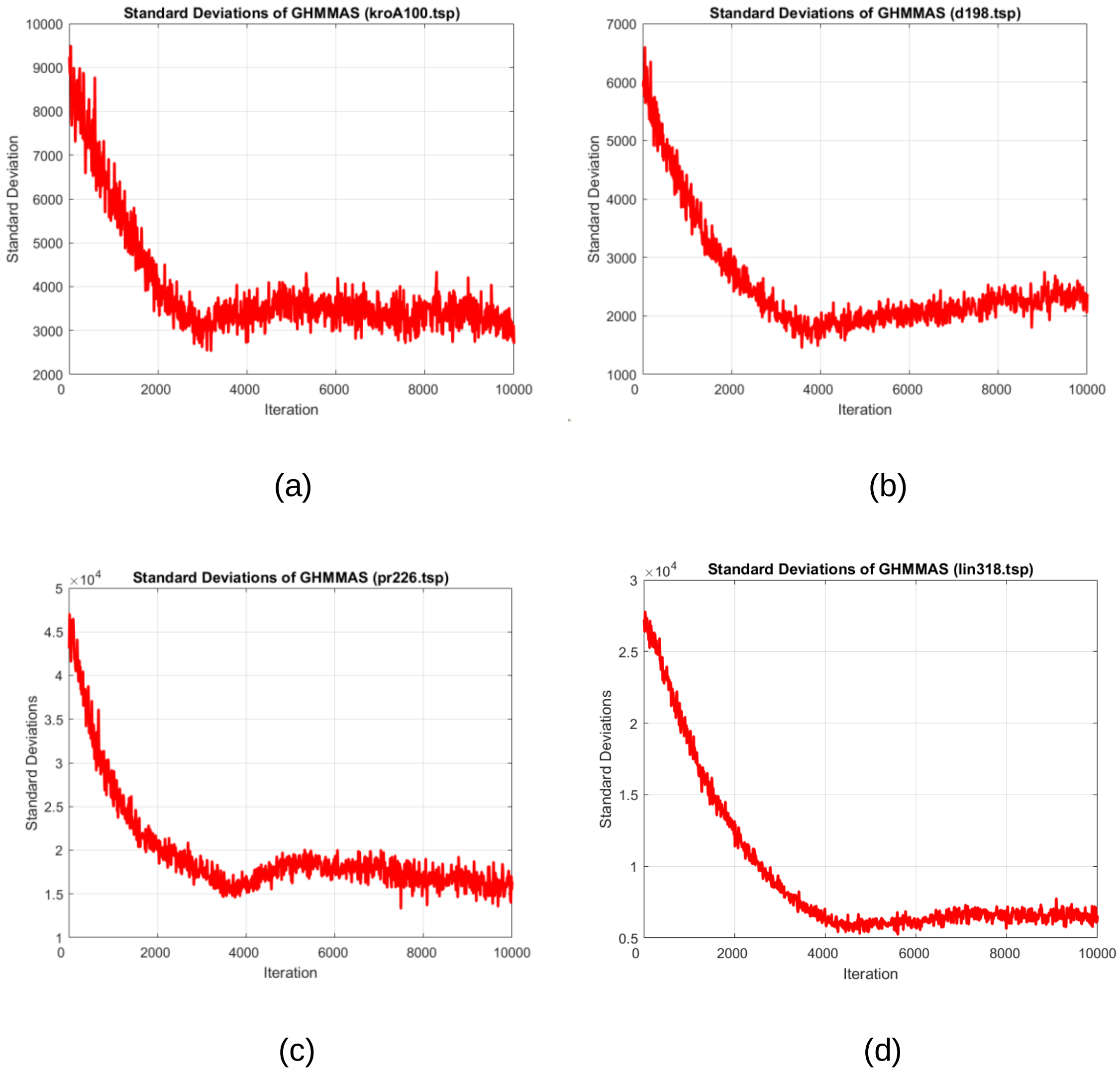


Figure 4.12: Standard deviations of fitness solutions found by the ants for every iteration (a) kroA100.tsp, (b) d198.tsp, (c) pr226.tsp and (d) lin318.tsp.

4.4 Concluding remarks

This chapter presents a heterogeneous approach randomly drawn from a Gaussian distribution with a predefined mean and standard deviation. The results suggest that the GHMMAS consistently outperforms both the uniform heterogeneous approach and the homogeneous approach on medium-sized TSP instances. GHMMAS has the best performance in terms of the best fitness solution found over a number of trials, the best average fitness solution and the best worst fitness solution (worst cost indicates how poor the performance is with lower worst cost indicates better performance). Initializing the ant colony with a higher number of ants centred around these values while also reducing the number of ants in the far extremities of the range using the Gaussian distribution contributed to the increased performance in the proposed approach. The algorithm also was able to overcome stagnation behaviour as illustrated by the standard deviations where the algorithm was able to passively switch between the exploration and exploitation strategies while optimizing the search space.

In addition to the TSP instances, the static Gaussian heterogeneous approach was tested on PCB holes drilling problem, which is a real-world problem, and compared against MMPAS that implements heterogeneity via recruitment learning. Although GHMMAS was not able to improve on the best fitness solution found by MMPAS on two out of three PCB instances, the best fitness solution found is not far off though. The results suggest that GHMMAS can be improved further with the introduction of 2-opt or 3-opt hybrid approaches. Another possibility is to create an adaptive approach where the ants will be able to actively search the parameter space and adapt to the best performing ant. This will be discussed in detail in the following chapter.

Chapter 5

A Heterogeneous Adaptive Max-Min Ant System for Traveling Salesman Problem

The previous chapter suggested optimal parameter settings of ACO are dependent on the problem instances. This chapter introduces an adaptive approach to a heterogeneous ant colony population that evolves the alpha and beta controlling parameters for ACO to locate near-optimal solutions. This is a development from the previous two chapters where the individual alpha and beta values of ants do not change over time thus remain constant. The adaptive approach is able to modify the exploitation and exploration characteristics of the algorithm during the search process to reflect the dynamic nature of the proposed approach. In addition, the 3-opt local search heuristic is integrated into the proposed approach to further improve the fitness of the solution. An empirical analysis of the proposed algorithm tested on a range of Travelling Salesman Problem (TSP) instances shows that the approach has better algorithmic performance when compared against state-of-the-art algorithms from the literature.

5.1 Motivation

One of the drawbacks of most ACO algorithms is its failure to continue to explore new solutions once the algorithm converges thus getting stuck in local optima (i.e the best solution within a subset of neighbouring solutions) or unable to escape stagnation behaviour (e.g. where all ants construct the same tour, i.e in the TSP). To counter this, several studies have suggested that different optimization strategies are required at different stages of the search process [13][12][121][46]. Another limitation of the conventional ACO is that the performance of the algorithm is significantly dependant on the parameter settings which are often set before the run by trial and error method and these parameters remain constant throughout the optimization process. These claims are supported by the empirical results from the previous chapter which show that the optimal ACO parameter settings may change over time as well as for different problems or even different instances of the same problem. In addition, parameter tuning for each and every problem is almost impossible for real-world problems as it is a time-consuming and computationally expensive process. In most cases, researchers spend a significant amount of time to fine-tune the ACO parameters based on experience while others implement settings suggested from the literature. Whilst the tuning of parameters is possible for standard, general problem instances, on larger real-world problems it is often not possible to conduct a thorough exploration of the parameter settings due to the computational complexity involved in the calculation of the objective function. In addition, the algorithm must maintain its exploratory nature even after converging to a set of solutions in order to improve the overall performance by being able to continuously explore and exploit the search space efficiently especially when applied to large problem instances. However, it is also important to note that over-exploration where the algorithm continuously explores new search space without really perturbing the solutions found or converging to an optimal solution may occur and this undesirable scenario causes a waste of valuable computational resources and function evaluations. Self-adaptive approaches have been shown to work well in other metaheuristics such as in EAs [5][8][9][123] and PSO [124][125][126][127][128], however, little research has been conducted in regards to the analysis of self-adaptation methods in ACO.

Therefore, enabling the algorithm with the ability to learn the optimal parameter settings via a self-adaptation method [55][57][54], where parameters are encoded as genomes of individuals, can alleviate the costly parameter tuning procedure as well as creating an algorithm that is robust to parameter settings. This approach explores the synergistic effects of the adaptive evolutionary process and heterogeneity to allow convergence towards colony-level parameter setting through the self-adaptive approach indirectly enabling the algorithm to locate better solutions. The population diversity is preserved by implementing a Gaussian mutation to the selected ants to prevent the algorithm from stagnation. A detailed description of the algorithm is described in the following section.

5.2 Heterogeneous Adaptive MMAS – The Framework

The framework described in the following section is based on the idea of optimal heterogeneity that can lead the algorithm to colony-level parameter convergence and is able to adapt during the search process in order to escape from local optima while continuing exploring the search landscape. This is achievable by introducing a set of rules for parameter adaptation to occur in order for the parameter values to be close to the optimal values. The heterogeneous nature of the population of ants introduced in Chapter 4 and 5 allow the initial population to explore the most promising areas of the search space initially but there is no additional mechanism to modify these as the search progresses. Therefore, the proposed approach in this chapter considers the initial population of ants as the initial population for an evolutionary algorithm that will adapt the parameters throughout the optimization. Algorithm 5.1 represents the pseudocode of the algorithm which includes the introduction of both the greedy homogeneous and the Gaussian heterogeneous ants as well as the adaptive mechanism.

```
Input: Distance matrix of TSP;
Initialize ACO parameters;
Initialize ants:
        for l=1: number of ants, do
                AlphaHet(i) = mean α, s.d α;
                BetaHet(i) = mean β, s.d β;
        end for
AlphaHo=0; BetaHo=10;
Start Iteration:
for it = 1 : Max Iteration do
        if it < 6 then
                Alpha=AlphaHo;
                Beta=BetaHo;
        end if
if it >= 6 then
        Alpha=AlphaHet;
        Beta=BetaHet;
end if
for k = 1 : number of ants do
        Position each ant on starting node;
        while T ourSize < n + 1 do
                Tour Construction;
                3-opt local search;
                Adaptation mechanism;
        end while
end for
end for
```

```
Update Solution;
Update Pheromone;
Pheromone Evaporation;
Check if termination condition is met;
if True then
        Go to End;
else
        Repeat Iteration ;
end if
End
```

Algorithm 5.1: Pseudocode of Heterogeneous Adaptive MMAS

Figure 5.1 illustrates the principle of the proposed approach where the parameters evolved in this study are the α and β values that control the relative importance of pheromone and heuristics respectively. These parameters were chosen because the performance of ACO is very much dependant on them [40] [20]. In addition to that, several studies have shown that these parameters are the best candidate for parameter adaptation analysis [45][53][129][130]. As shown in Figure 5.2, each ant has its own 'behavioural traits' represented by the α and β values. The ants will have fitness solutions associated with them based on the tours they built in every iteration and given a rank according to the mean fitness value over a certain number of iterations (discussed below as the 'adaptive interval'). Next, the highest and lowest-ranked ants are selected as the mean best and mean worst ants. Once an offspring is produced from the process of mutation of the mean best ant, the offspring then replaces the mean worst ant and is included in the population for exploration and exploitation in the next iteration. Additionally, the use of elitism ensures that the population retains the fittest individual in the population and where the mean worst ant over several iterations is replaced by the child of the mean best ant that had undergone mutation. Through the use of selection, elitism and mutation, the EA can generate new promising parameter settings during the ACO run as well as maintaining diversity in the population. The first step in solving a problem via the Evolutionary Algorithm (EA) is to represent a solution to a problem. Choosing the right encoding scheme is crucial in determining how the solution space is defined. As this study encodes the parameters of the ants, an individual ant can be represented by a pair of the parameters (α,β) which is later evolved. To achieve this, the algorithm requires representation and this is achieved here through the use of a floating-point representation. It is worth noting that other encodings are possible, in particular, binary encodings although the floating-point encoding here is preferred due to the ability to test fine-grained changes to the range from the mutation operation. By representing the parameters as the genotype, the heterogeneous adaptive approach explores and exploits both the solution and parameter space simultaneously to locate the optimal fitness solution as well as colony-level convergence to instance-optimized parameters. The proposed adaptive approach uses GHMMAS, which implements heterogeneity from a normal distribution, as the base algorithm as empirical results in Chapter 4 show that it has a better performance compared to that of the heterogeneous approach drawn from a uniform distribution.

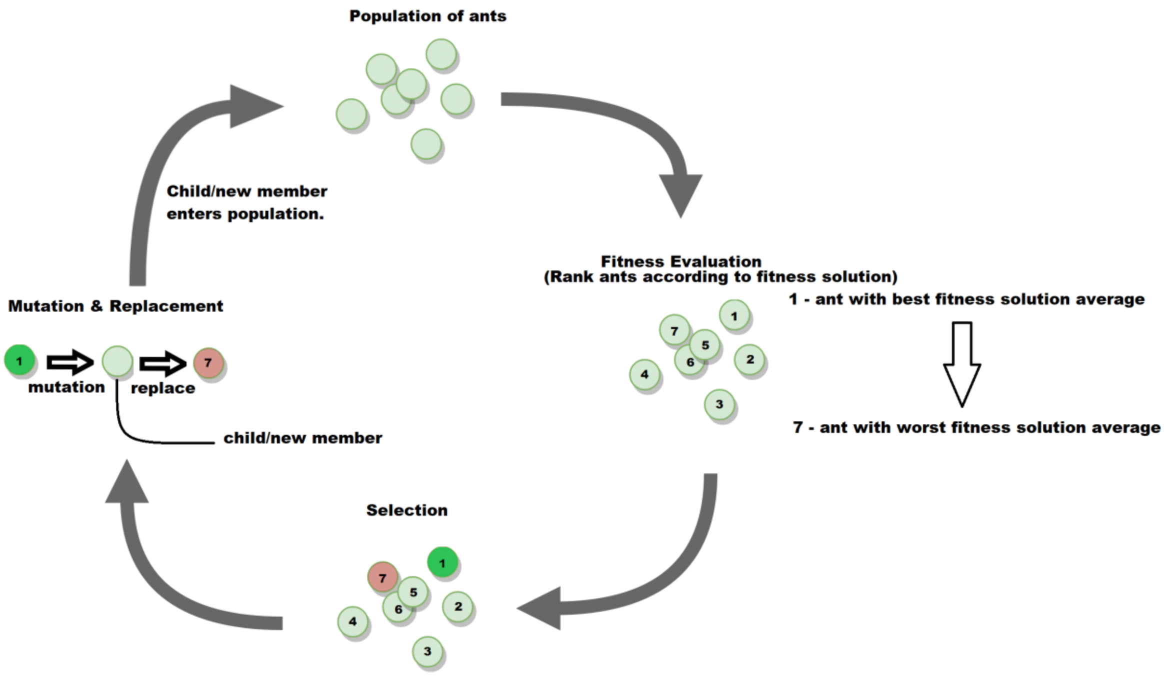


Figure 5.1: The iterative procedure of the proposed approach.

	Ant_1	Ant_2	Ant_n
Iteration 1	$\alpha_1\beta_1 = F_{11}$	$\alpha_2\beta_2 = F_{21}$	$\alpha_n\beta_n = F_{n1}$
Iteration 2	$\alpha_1\beta_1 = F_{12}$	$\alpha_2\beta_2 = F_{22}$	$\alpha_n\beta_n = F_{n2}$
.			
.			
Iteration 5	$\alpha_1\beta_1 = F_{15}$	$\alpha_2\beta_2 = F_{25}$	$\alpha_n\beta_n = F_{n5}$

Figure 5.2: Ant’s fitness solution representation in every iteration.

5.2.1 Greedy Ants Initialization

It has been reported that the choice of the initial tour plays an important role in the final solution [131][132]. Therefore, in this approach, a group of greedy, foraging ants was deployed for several iterations to act as a guide for the Gaussian heterogeneous ants to explore and exploit the solution found rather than starting by locating random tours. This method to speeds up and improve the initial tours found by the colony of ants. Initially, the pheromone landscape lacks useful information for the ants to utilize in their exploration hence the use of these extremely greedy ants to quickly locate sub-optimal solutions for the heterogeneous ants to further perturb in order to locate better solutions. The possibility of getting stuck in local optima during the greedy ants' stage, if any, is overcome by the ability of the algorithm to escape from local optima via continuous exploration due to the mechanism of the heterogeneous approach. The greedy, ants consist of ants with a relative importance of 0 towards pheromone (α=0) and a very high preference towards the heuristic (next-hop distance) (β=10 [133]) that indicates high β during early stages of the search process is desirable. This will allow the greedy, homogeneous colony to locate good solutions for the heterogeneous ants to exploit very early on rather than starting with random tours as per conventional ACO algorithms. An experiment was conducted to determine an optimal setting for the number of iterations required for the aforementioned approach. Two different variants of the proposed algorithm were created where Heterogeneous Adaptive ACO–5 (HAACO-5) is the adaptive approach with greedy, homogeneous ants that were deployed for 5 iterations while HAACO-10 is the same algorithm except that the greedy, homogeneous ants were deployed for 10 iterations. The algorithms were tested on medium-scaled TSP instances of ch150.tsp and kroA200.tsp with a known optimum of 6528 and 29368 respectively [109] while the stopping criterion was set at 10 000 function evaluations without any local search procedure. The experiment is designed test the settings on selected test cases of the same problem before applying the resulting settings for all the instances of the same problem [16]. The reason behind this experiment is that there might be a waste of valuable function evaluations if too many iterations are used for the greedy ants phase and the colony of ants may exhibit stagnation behaviour where all ants perform the same tour. On the contrary, potential sub-optimal solutions may not be found if too few iterations used for this phase thus defeating the main purpose of this initialization stage. It should be noted that this number of greedy tours approximates those seen in the other approaches [47][48] (compared against in section below), so as to confer no advantage to either algorithm.

Table 5.1 compares the result of the experiment and it can be seen clearly that the algorithm that limits the greedy ants' exploration stage at 5 iterations has better overall performance compared to that of the other approach in terms of the best cost found, average best cost and worst best cost. A Wilcoxon rank-sum statistical test with a 90% confidence interval gives a p-value of 0.03 and 0.1 respectively when tested on the performance of HAACO-5 over HAACO-10 for both TSP instances thus indicating the results are statistically significant. Figures 5.3 and 5.4 illustrate the performance of the HAACO with two different deployment approach as stated above. The boxplots represent the best cost of 20 trials of the proposed approach. The result indicates that allowing the greedy, homogeneous ants to explore the search space for the first 5 iterations based on a greedy approach instead of random initial tours produces good initial solutions. The analysis also suggests that enabling the homogeneous ants to explore the search landscape for 5 iterations produced the best performance. From iteration 6 onwards, the algorithm introduces a heterogeneous population of ants randomly drawn from a Gaussian distribution that will exploit the good regions found by the greedy homogeneous ants to further locate better solutions. Therefore, this approach will be implemented in the studies hereafter. The mean and standard deviations for both α and β parameters of the heterogeneous ants remain as in the previous chapter. The population then evolves as the search progresses where the worst ant in every adaptive interval (AI) was replaced with the child of the best ant in that AI that had undergone a Gaussian mutation (explained in the following section).

Table 5.1: Performance comparison of the heterogeneous adaptive approach with a different number of iterations for greedy ants to explore the search space (-5 indicates 5 iterations and -10 indicates 10 iterations for greedy ants to explore).

Algorithm	Best		Average		Worst	
	ch150	kroA200	ch150	kroA200	ch150	kroA200
Greedy-5	**6548**	**29452**	**6572.1**	**29638**	**6601**	**29813**
Greedy-10	6554	29478	6582.6	29700	6622	30115

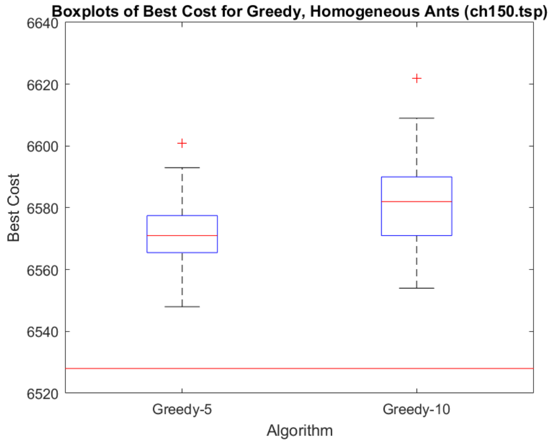


Figure 5.3: Boxplots representing the best cost of heterogeneous adaptive approach with different initial greedy ants run tested on ch150.tsp (20 trials).

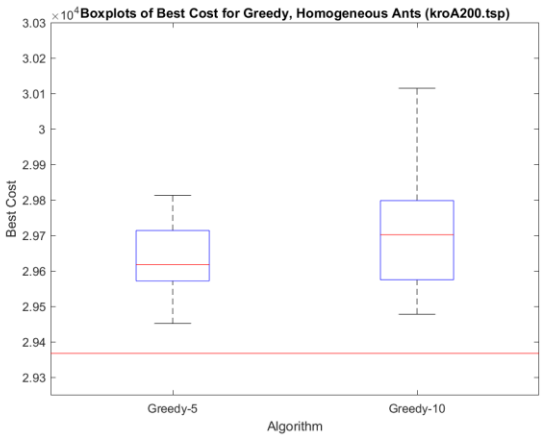
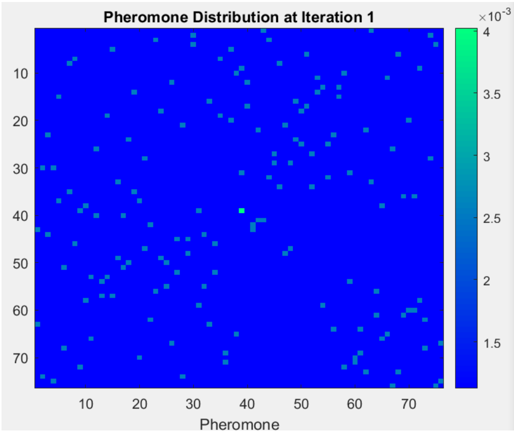
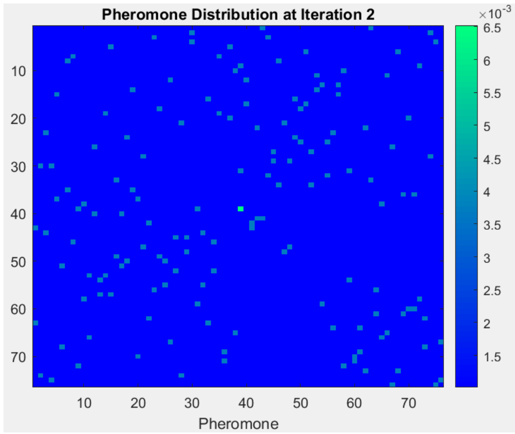


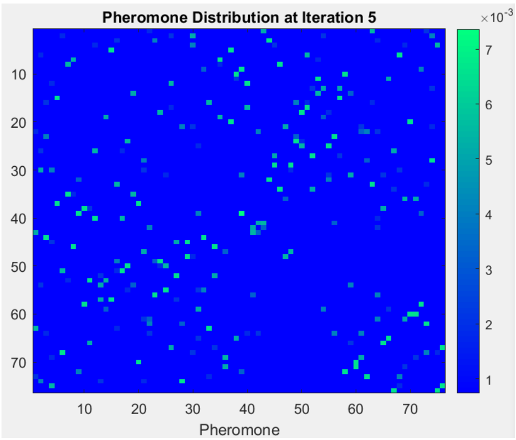
Figure 5.4: Boxplots representing the best cost of heterogeneous adaptive approach with different initial greedy ants run tested on kroA200.tsp (20 trials).



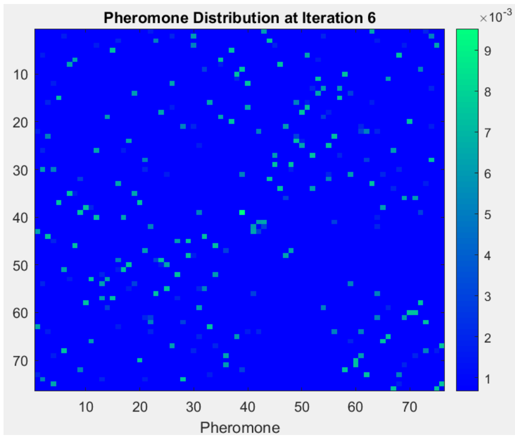
(a)



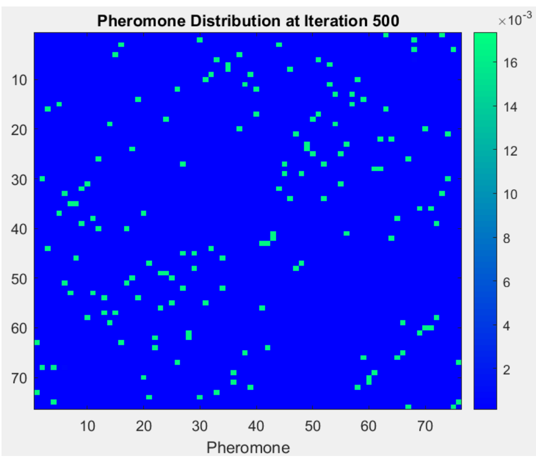
(b)



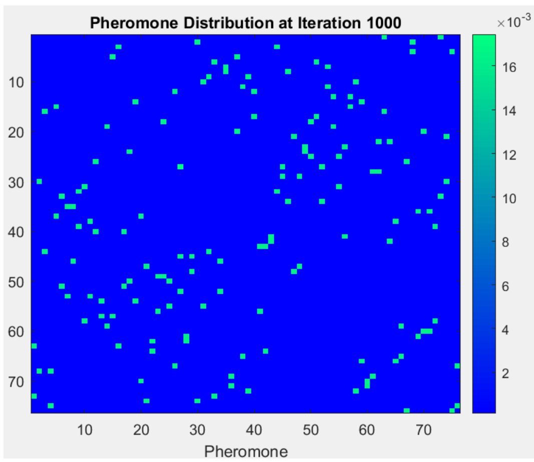
(c)



(d)



(e)



(f)

Figure 5.5: Pheromone distributions of the heterogeneous adaptive ants (eil76.tsp).

Figures 5.5 (a) to (f) illustrate an example of the pheromone distribution for a single run of heterogeneous adaptive approach on eil76.tsp (Opt=538) throughout the search process. Pheromone distributions from Iteration 1 to Iteration 5 represent the tours made by the greedy homogeneous ants while the pheromone distributions from Iteration 6 onwards represent the tours made by the heterogeneous ants. It is noticeable that the pheromone intensity diminishes a little in iteration 6 when the heterogeneous ants were introduced but the heterogeneous ants do not totally forget those paths found by the greedy ants but exploit those paths while also explore new search areas. Figure 5.5c and f can be compared and it can be seen that the greedy homogeneous ants were able to locate the sub-optimal solution as early as 5th iteration itself thus giving the heterogeneous ants a head start rather than starting from random tours. Thus this approach of the greedy ants acts as a guide for the heterogeneous ants towards good regions.

5.2.2 Adaptive Interval

As the heterogeneous adaptive approach does not employ additional function evaluations, the assessment of the quality of α and β parameter settings is based on the mean performance of ants with those parameter settings and a frequency of sampling of this information must be specified. Furthermore, a parameter must be defined to determine how often the evolution of the parameters takes place. Both of these factors are considered in the *adaptive interval (AI)* which is the number of ACO iterations that are completed between evolutionary steps. Setting this value is important because it determines how many evolutionary steps are possible within a given ACO run of fixed length and because it determines the robustness of the sampling that underpins the objective function calculation. For example, low values of *AI* (e.g. 1) indicate that parameter adaptation occurs in each ACO iteration. This provides many EA iterations but each one will be based on only one tour from each ant, leading to potentially volatile changes to the best performing ants through time. A more moderate setting such as 5 will yield 1/5 of the evolutionary steps but based on a more robust sample of 5 tours generated by the ants along with the amount of pheromone deposition is taken into account as well. This parameter was analysed across several TSPs: eil101.tsp, ch150.tsp, and d198.tsp respectively and 5 was selected as the best performing of these. Figure 5.6,5.7 and 5.8 illustrate the boxplots of the proposed approach with different adaptive intervals while the yellow lines indicate the average best cost. The figures indicate that *AI*=5 produces the best performance when compared against other adaptive intervals that we have analysed in this section. The results show that fast adaptation is preferable over a slow adaptation mechanism where information over 5 iterations are used to determine or select the best and worst ant for replacement.

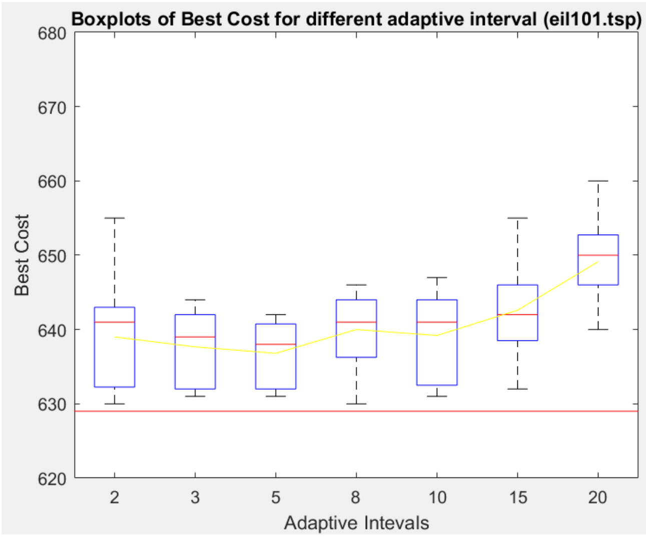


Figure 5.6: Boxplots representing the best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on eil101.tsp.

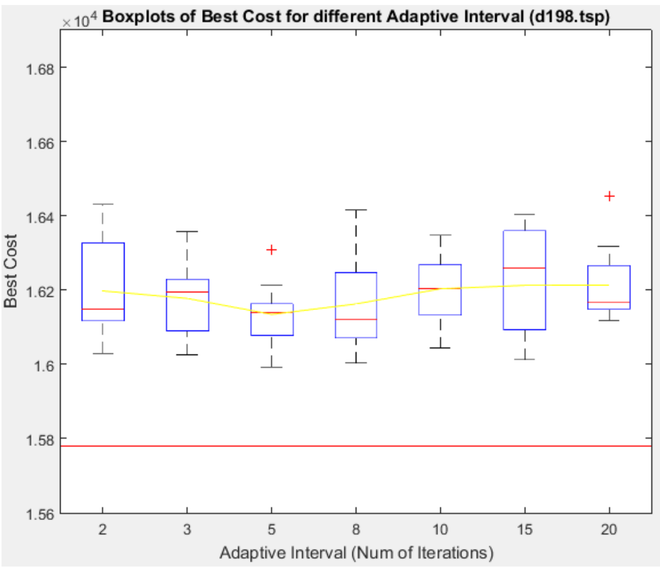


Figure 5.7: Boxplots representing the best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on ch150.tsp.

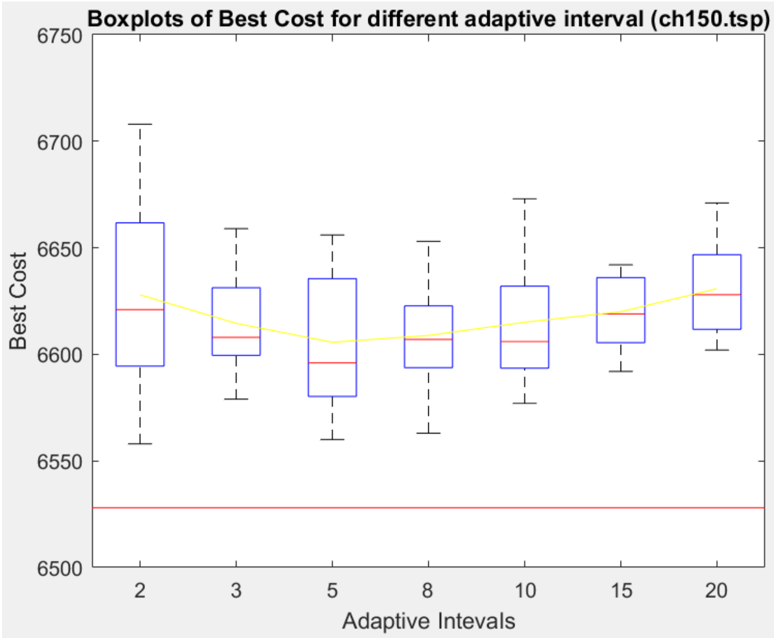


Figure 5.8: Boxplots representing the best cost of 20 trials of heterogeneous adaptive ants with different *Adaptive Intervals (AI)* tested on d198.tsp.

5.2.3 Mutation

One of the key aspects of the algorithm is that it requires a mechanism to explore the parameter space and to generate new parameter values for evaluation. However, in contrast the algorithm also has to preserve the information that it has already gathered thus drastic alteration is not preferable. This suggests that a very low amount of mutation is desirable to prevent total loss of the fittest individuals over time especially when a small population size is used. In addition, the mutation operator is also capable of preventing premature convergence to non-optimal solutions by enabling the algorithm to escape from local optima. On top of that, the mutation operator maintains the diversity in the evolving populations. Therefore, this study implements an approach where the mean best ant over 5 iterations (AI = 5 as explained above) will undergo mutation to cause a small, random change to the genotype before replacing the mean worst ant. The offspring will then join the heterogeneous population and through this repeated action will explore the parameter space and locate the instance-optimal parameter setting. The crossover operator, usually paired with the mutation operator in EA, is not used in this study because the genetic algorithm with random mutation alone results in better performance as compared against that of mutation with crossover and inversion operator [134].

The degree of random change that each application of the mutation operator can cause is related to the width of the distribution used. There are various mutation operators however two of the most popular mutation operators are the uniform and Gaussian mutation. Usually, the uniform mutation operator in GA replaces the value of the chosen gene or allele with that randomly drawn from the uniform distribution within the upper and lower bound. Likewise, the Gaussian mutation operator adds a random value to the allele drawn from the Gaussian distribution with mean = 0 and standard deviation determine empirically by the researcher. The Gaussian mutation operator which is a commonly used operator is used in this study due to the high probability of creating an offspring that is much closer to the genes of the parents especially when elitism selection method is used. The Gaussian mutation is advantageous as it supports fine-tuning of the subject in study and flexible to implement. An empirical study was conducted to compare the performance of both uniform and Gaussian mutation operator in order to be implemented in the final setup of the heterogeneous adaptive approach. Initial experiments were conducted to determine the suitable range for the uniform mutation operator, $M = U[-a; a]$ and standard deviation, σ for Gaussian mutation with mean 0, $M = G(0; \sigma)$. Table 5.2 and 5.3 show the results of the initial experiments conducted where the range of $[-0{:}05; 0{:}05]$ produces the best performance for uniform mutation while heterogeneous adaptive with a Gaussian mutation and standard deviation of $0{:}05$ has an overall best performance. In addition to the results, the extent to which the uniform mutation and the standard deviation, σ of the Gaussian mutation operator was explored, it was expected that a large mutation width (large range of values for uniform mutation or high σ for Gaussian mutation) will cause the algorithm to engage in excessive exploration thus mimicking a random walk and too small a width (small range of values for uniform mutation or low σ for Gaussian mutation) will cause the algorithm to converge to a local optimum too quickly and very little exploration of the parameter space.

Table 5.2: Comparison of the average best cost of heterogeneous adaptive approach with different uniform mutation range tested on several TSP instances where the results in bold indicate best in each category. Results are of 20 trials, each trial = 1000 iterations.

Range		Best			Average			Worst		
LB	UB	eil51	kroA100	kroA200	eil51	kroA100	kroA200	eil51	kroA100	kroA200
-0.05	0.05	**426**	21416	**29499**	428.0	**21490.2**	**30023.4**	431	**21609**	**30664**
-0.075	0.075	**426**	**21355**	29601	428.7	21617.1	30058	434	21860	30833
-0.1	0.1	**426**	21416	29540	**427.7**	21531.4	30059.4	**430**	21877	30857
-0.2	0.2	**426**	21443	29524	428.7	21518	30107.9	433	21719	30967
-0.3	0.3	**426**	21379	29847	429.3	21587.5	30138.2	434	21768	30727
-0.4	0.4	**426**	21443	29603	430.4	21573.7	30150.2	434	21883	30991
-0.5	0.5	428	21393	29672	431.9	21604	30440.2	440	21899	31140

Table 5.3: Comparison of the average best cost of heterogeneous adaptive approach with different standard deviation for the Gaussian mutation operator tested on several TSP instances where the results in bold indicate best in each category. Results are of 20 trials, each trial = 1000 iterations.

TSP	Opt	Standard deviation							
		0.5	0.4	0.3	0.2	0.1	0.075	0.05	0.025
st70	675	690.6	689.4	684	681.7	680.4	682.8	**676.5**	679.4
ch150	6528	6649.8	6638.6	6608.6	6598.2	6596.2	6586.6	**6578.8**	6584.6
kroA200	29368	30708.1	30397.7	30525.4	30333.7	30084	29964.9	**29633.2**	29720.4

Hence, a suitable mutation width is required in order to achieve good performance. The mean best ant which has individual α and β values will undergo mutation using the equations below where *M*1 and *M*2 are two random values based on the mutation distribution used in the experiments. These values are then added to the α and β of the mean best ant thus creating a child of the mean best ant which replaces the mean worst ant. Table 5.4 compares the performance of the proposed approach with both uniform and Gaussian mutation with optimal settings as discussed above. It can be seen clearly that the heterogeneous adaptive with Gaussian mutation has an overall best performance in terms of best, average and worst best cost in almost all TSP instances when compared against the heterogeneous adaptive with uniform mutation algorithm. Therefore, Gaussian mutation with *mean* = 0 and standard deviation, *σ* = 0:05 will be applied in all experiments hereafter unless stated otherwise.

$$\alpha_{meanworst} = \alpha_{meanbest} + M_1[-a, a]/M_1(0, \sigma) \tag{16}$$

$$\beta_{meanworst} = \beta_{meanbest} + M_1[-a, a]/M_2(0, \sigma) \tag{17}$$

Table 5.4: Performance comparison of heterogeneous adaptive approach with uniform and Gaussian mutation operator respectively tested on several TSP instances. Results are of 20 trials, each trial = 1 000 iterations and those in bold indicates the best of each category.

TSP	Opt	Best		Average		Worst	
		Uni	Gau	Uni	Gau	Uni	Gau
st70	675	**675**	**675**	679.9	**676.5**	684	**678**
eil101	629	631	**630**	638.5	**632.5**	645	**635**
lin105	14379	**14379**	**14379**	14453.6	**14411.8**	14525	**14483**
ch150	6528	**6554**	6566	6581.7	**6578.8**	6599	**6595**
kroA200	29368	29604	**29483**	30150.4	**29633.2**	30973	**29755**

5.2.4 Local Search

The heterogeneous adaptive algorithm is further improved by the 3-opt local search heuristic where 3 edges are deleted from the tour and reconnected in some other possible ways in order to find the optimal solution (in this case, lowest best cost of all possible tours made up from the reconnection). 2-opt and 3-opt are the most common local search heuristics used to improve the tours while $k > 3$ (*k-opt)* produces better tours but with a significant increase in computational time which causes the algorithm to be ineffective. The heterogeneous adaptive approach implements the basic 3-opt local search approach to improve the solution found. Even though other approaches such as the “fixed-radius search” and “don’t look bits” are known to be better than the 3-Opt, this study did not implement any of it in order to confer no advantage to the heterogeneous adaptive approach over the algorithms in comparison. Only the tour that belongs to the iteration-best ant is perturbed by the 3-opt local search heuristic. This is because of the additional computational time as well as the extra function evaluation count if all tours in each iteration undergo this procedure.

5.3 Experimental Setup

An Intel Core i7 CPU-based computer with Windows 7 equipped with 4GB RAM was used to conduct the experiments and analysis. Matlab version R2015a was used to implement the base algorithm Max-Min Ant System (MMAS) [5]. The results of the developed base algorithm were shown to closely replicate the performance to that of the original authors which can be referred to Chapter 4. The performance of the proposed approach was measured on several symmetric Euclidean TSP instances with known optimum indicated in Table 5.5 [8]. Various sizes of TSP instances were used because different sizes of the search landscape impact the behaviour of the algorithm thus can determine how the algorithm adapts to the change in the search space. In addition, these cities were chosen in order to enable the proposed approach to be compared against state-of-the-art adaptive and hybrid algorithms from the literature. Table 5.6 summarizes the parameter settings of HAACO where the settings for the base algorithm of MMAS are obtained from [5] as well as the suggestion by Dorigo and Stützle [107, p. 71] that suggest some parameters that produces good performances. In addition to this, specific parameters in relation to the HAACO are from our preliminary experiments.

Table 5.5: TSP instances, size of the problem, and its known optimum.

TSP	Size	Opt
eil51	51	426
Berlin 52	52	7542
st70	70	675
eil76	76	538
rat99	99	1211
kroA100	100	21282
eil101	101	629
lin105	105	14379
ch150	150	6528
kroA200	200	29368

Table 5.6: Parameter settings as in the proposed approach.

Parameter	Symbol	Value
Number of ants	m	10
Pheromone importance (Heterogeneous)	α	Gau dist: mean = 1, s.d = 0.2
Heuristics importance (Heterogeneous)	β	Gau dist: mean = 5, s.d = 0.2
Pheromone importance (Greedy ants)	α	0
Heuristics importance (Greedy ants)	β	10
Initial pheromone	τ0	1/(n*Lnn)
Pheromone Evaporation	ρ	0.02
Stopping criterion	-	10 000 function evaluations
Constant value	Q	1
Greedy ants initialization	-	5 iterations
Adaptive Interval	AI	5 iterations
Gaussian mutation operator	-	mean = 0, s.d = 0.05
Local search	-	3-opt
Number of trials	-	20 trials

5.4 Results & Discussion

5.4.1 Overall Results

The aim of this section is to present the results and analysis of the HAACO approach. Moreover, the ability of the approach to explore and exploit the parameter space to locate instance-optimal parameter settings whilst generating competitive TSP tours is described. The proposed approach is compared against two state-of-the-art hybrid ACO algorithms for TSP which are the parallel co-operative ACO algorithm with 3-opt [48] and hybrid PSO-ACO-3opt algorithm [47], as well as two variants of MMAS, where the MMAS algorithm was developed and its similarity is proven in Chapter 4. MMAS1 is a variant of MMAS with 3-opt and greedy, homogeneous ants for the first 5 iterations while MMAS2 represents a standard MMAS implementation also augmented with 3-opt. [48] and [47] have some similarities to HAACO where the algorithms also deploy co-operative approaches, migration strategies (similar to adaptive approach), uses 3-opt local search and have been compared against a number of algorithms and proven to perform well on several TSP instances. For comparison purposes, all parameters were set according to [47] and [48] except specific parameters such as mean α and mean β of heterogeneous ants which were set to 1 and 5 respectively (as explained above).

Tables 5.7 to 5.12 represent the performance comparison of HAACO against [47], [48] and MMAS variants in terms of best cost, average best cost and worst best cost respectively. A ranking mechanism was used to rank the performance of each algorithm with a ranking of 1 to 5 where 1 is the best algorithm in each TSP instance with 5 being the poorest. Referring to Table 5.7, HAACO has the overall best performance as it is capable of locating the optimum or lowest fitness solution in 7 over 10 instances and it has the second-best fitness solutions in the other two TSP instances while the algorithm is third in the ranking for ch150.tsp. Table 5.8 supports this claim by indicating the average ranking of each algorithm and HAACO has an average ranking of 1.4 followed by [47] in second with an average of 1.7. The results indicate that the proposed algorithm has a better performance when compared to the state-of-the-art algorithms.

Table 5.7: Fitness solution comparison of HAACO against other approaches tested on several TSP instances. The result in bold indicates the best of each category.

TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt [48]	MMAS 1	MMAS 2
eil51	426	**426**	**426**	**426**	427	**426**
Berlin52	7542	**7542**	**7542**	**7542**	**7542**	**7542**
st70	675	**675**	676	676	**675**	682
eil76	538	**538**	**538**	**538**	**538**	**538**
rat99	1211	**1211**	1224	1213	1212	1212
kroA100	21282	**21282**	21301	**21282**	21315	21379
eil101	629	630	631	**629**	631	631
lin105	14379	**14379**	**14379**	**14379**	**14379**	**14379**
ch150	6528	6566	**6538**	6570	6554	6566
kroA200	29368	29483	**29468**	29533	29485	29488

Table 5.8: Ranking comparison of HAACO against other approaches tested on several TSP instances based on the fitness solution in Table 5.7. The result in bold indicates the best of each category.

TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt[48]	MMAS 1	MMAS 2
eil51	426	**1**	**1**	**1**	2	**1**
Berlin52	7542	**1**	**1**	**1**	**1**	**1**
st70	675	**1**	2	2	**1**	3
eil76	538	**1**	**1**	**1**	**1**	**1**
rat99	1211	**1**	4	3	2	2
kroA100	21282	**1**	2	**1**	3	4
eil101	629	2	3	**1**	3	3
lin105	14379	**1**	**1**	**1**	**1**	**1**
ch150	6528	3	**1**	4	2	3
kroA200	29368	2	**1**	5	3	4
average		**1.4**	1.7	2	1.9	2.3

Table 5.9 shows the average fitness solution over 20 trials, corresponding to Table 5.10 which illustrates the results via the ranking system. Both HAACO and PACO-3Opt [48] have similar performance where both have lowest average best cost in 4 over 10 instances with an average ranking of 1:9. Lastly, another performance measure that was used in this comparison is the lowest of 20 worst fitness solutions of each algorithm for each TSP instance as shown in Table 5.11 while Table 5.12 translates the results into a ranking system. Both tables indicate that HAACO has a better performance in terms of worst best cost with an average ranking of 1.8 compared to 1.9 by [48].

Table 5.9: Comparison of HAACO against other approaches based on the average fitness solution. The result in bold indicates the best of each category.

TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt [48]	MMAS 1	MMAS 2
eil51	426	427.5	426.45	**426.35**	429.4	428.5
Berlin52	7542	**7542**	7543.2	**7542**	**7542**	**7542**
st70	675	**676.5**	678.2	677.85	683.8	685.2
eil76	538	542	**538.3**	539.85	542.8	543.5
rat99	1211	**1214.1**	1227.4	1217.1	1216.9	1219.4
kroA100	21282	21364.2	21445.1	**21326.8**	21528.3	21513.7
eil101	629	632.5	632.7	**630.55**	640.4	640.9
lin105	14379	14411.8	**14379.15**	14393.0	14429.2	14433
ch150	6528	6578.8	**6563.95**	6601.4	6603.9	6581
kroA200	29368	**29633.2**	29646.05	29644.5	29799.4	29760.3

Table 5.10: Ranking comparison of HAACO based on the average fitness solution in Table 5.9. The result in bold indicates the best of each category.

TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt [48]	MMAS 1	MMAS 2
eil51	426	3	2	**1**	5	4
Berlin52	7542	**1**	2	**1**	**1**	**1**
st70	675	**1**	3	2	4	5
eil76	538	3	**1**	2	4	5
rat99	1211	**1**	5	3	2	4
kroA100	21282	2	3	**1**	5	4
eil101	629	2	2	**1**	3	4
lin105	14379	3	**1**	2	4	5
ch150	6528	2	**1**	4	5	3
kroA200	29368	**1**	3	2	5	4
average		**1.900**	2.300	**1.900**	3.800	3.900

Table 5.11 Fitness solution comparison of HAACO against other approaches tested on several TSP instances based on the lowest worst fitness solution over 20 trials. The result in bold indicates the best of each category.

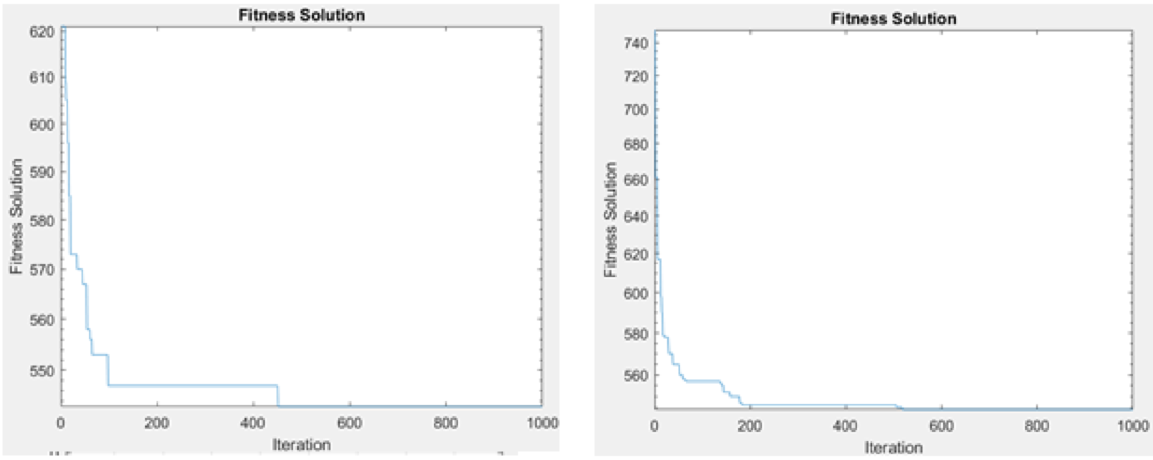
TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt [48]	MMAS 1	MMAS 2
eil51	426	430	428	**427**	433	432
Berlin52	7542	**7542**	7548	**7542**	**7542**	**7542**
st70	675	**678**	681	679	691	692
eil76	538	545	**539**	542	547	551
rat99	1211	**1218**	1230	1225	1226	1229
kroA100	21282	21445	21554	**21382**	21917	21810
eil101	629	**635**	638	639	651	650
lin105	14379	14483	**14381**	14422	14542	14594
ch150	6528	**6595**	6622	6627	6675	6617
kroA200	29368	29755	29957	**29721**	30307	30033

Table 5.12: Ranking comparison of HAACO based on the average fitness solution in Table 5.11. The result in bold indicates the best of each category.

TSP	Opt	HAACO	PSO-ACO-3Opt [47]	PACO-3Opt [48]	MMAS 1	MMAS 2
eil51	426	3	2	**1**	4	5
Berlin52	7542	**1**	2	**1**	**1**	**1**
st70	675	**1**	3	2	4	5
eil76	538	3	**1**	2	4	5
rat99	1211	**1**	5	2	3	4
kroA100	21282	2	3	**1**	5	4
eil101	629	**1**	2	3	5	4
lin105	14379	3	**1**	2	4	5
ch150	6528	**1**	3	4	5	2
kroA200	29368	2	3	**1**	5	4
average		**1.8**	2.5	1.9	4	3.9

5.4.2 Parameter Adaptation

The improved performance in HAACO can be attributed to the capability of the algorithm to search the parameter space and quickly converge towards optimal parameter settings as well as exploiting the neighbouring regions via the Gaussian mutation. The greedy, homogeneous sub-colony was able to locate sub-optimal solutions very early in the search process and this acts as a guide for the Gaussian heterogeneous ants to exploit to locate better solutions. When considering a limited budget of function evaluations i.e 1000 iterations, this mechanism provides better starting solutions compared to a random start approach as used in most of ACO algorithms. Figures 5.9a and 5.9b show the comparison of the best cost of HAACO with and without the use of the greedy homogeneous ants for a single trial when tested on eil76.tsp (optimum: 538) . Figure 5.9a shows that the greedy ants enabled the algorithm to start with better, shorter tours very early on while Figure 5.9b illustrates that the algorithm starts off with random tours due to lack of pheromone information during the early stages thus starting with poor fitness solution. Several function evaluations were required before the algorithm managed to locate good solutions hence a waste of function evaluations.

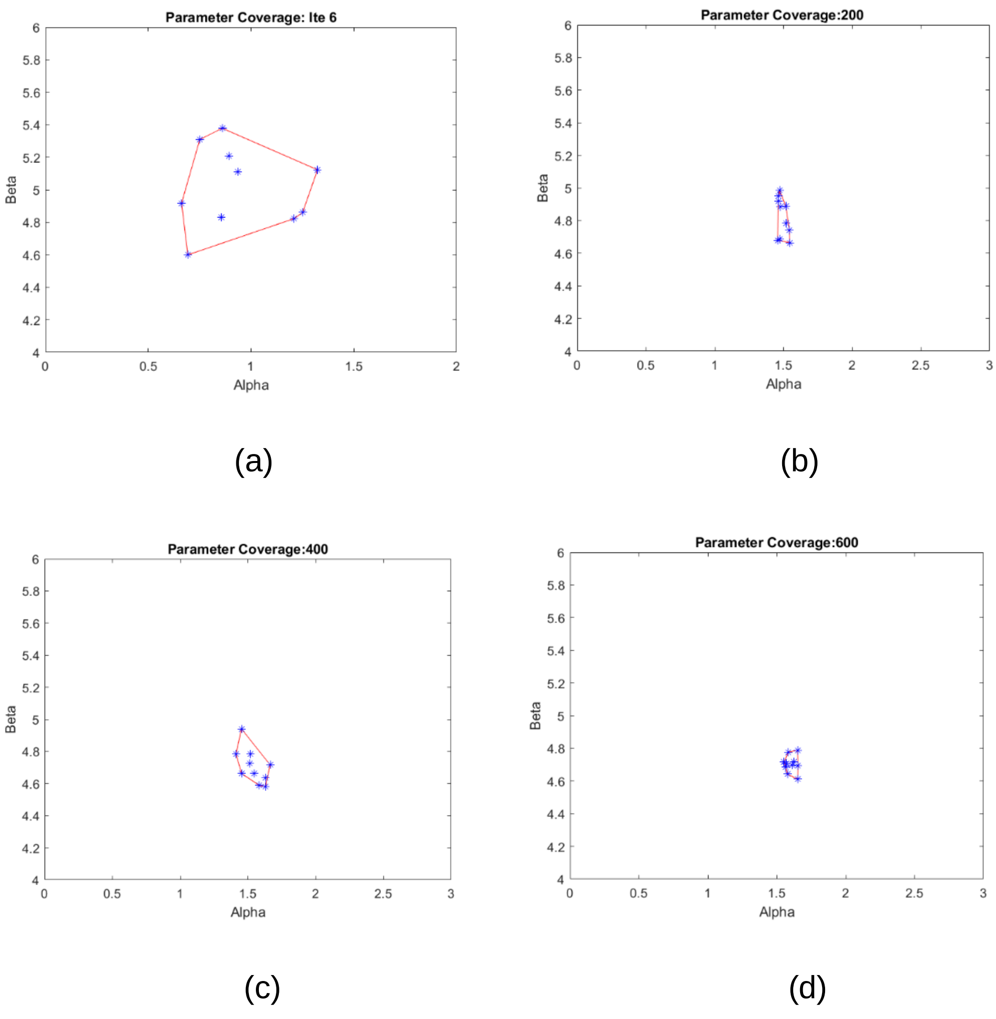


(a) HAACO with greedy (b) HAACO without greedy

Figure 5.9: Best cost plot of a single trial of HAACO tested on eil76.tsp.

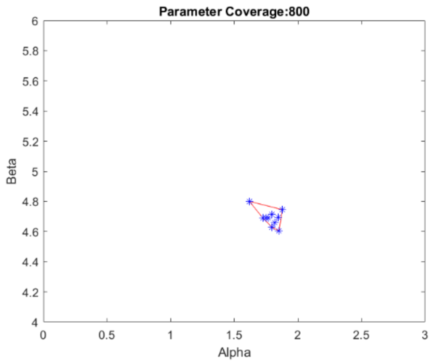
It can also be seen that the HAACO with greedy ants initialization was able to locate a tour of 540 by iteration 100 while HAACO without greedy ants initialization only managed to reach 550 by iteration 500 while possibly running out of function evaluations to further explore or exploit the search space especially in situations of limited budget evaluations.

Figures 5.10, 5.11 and 5.12 illustrate the parameter convergence using convex hulls to show the convergence of parameter values during different stages of the optimization process when tested on st70.tsp (optimum:675), lin105.tsp (optimum:14379) and ch150.tsp (optimum:6528). The convex hull forms the perimeter of the outermost points in a Euclidean space while the histograms represent the number of ants within a particular range of values. From these figures, it can be seen that the initial distribution of heterogeneous ants introduced from iteration 6 onward is followed by exploration and exploitation of the parameter space by these ants in order to locate the instance-optimal parameter settings. It can also be noticed that in general, ants with high beta and low alpha values perform best. This depicts the exploration phase of the algorithm where the algorithm explores the fitness landscape to locate good to sub-optimal solutions. In addition to that, there is less information on the pheromone landscape for the ants to exploit hence the colony adapts to ants with high beta and low alpha values. The strategy changes as the search progresses when there is an accumulation of pheromone on the edges hence ants become less reliant on the heuristics and utilizes the information on the pheromone landscape. Therefore, ants with higher alpha perform better in later stages and the colony adapts to this. This shows that the proposed approach is capable of adapting its strategy between exploration and exploitation whenever necessary and at the same time exploring the parameter space to locate instance-optimal parameter settings.

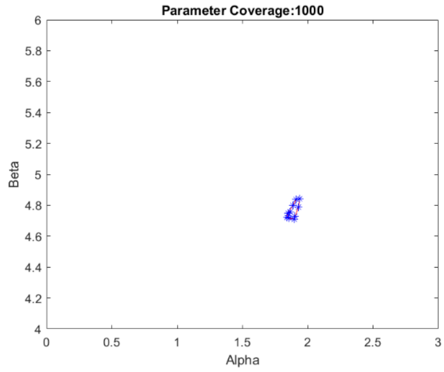


(a) (b)

(c) (d)

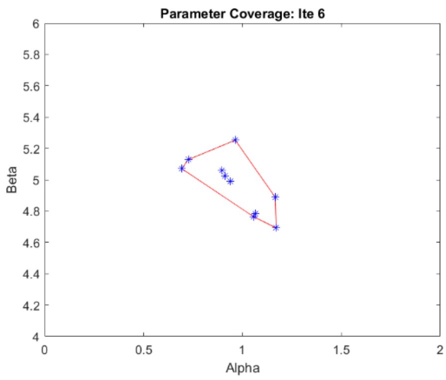


(e)

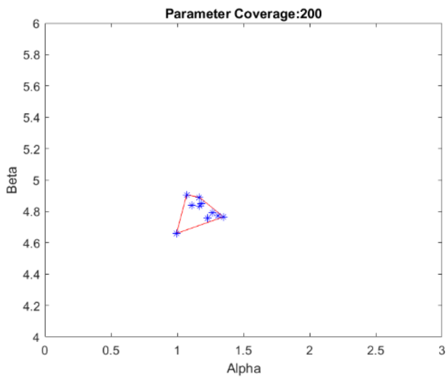


(f)

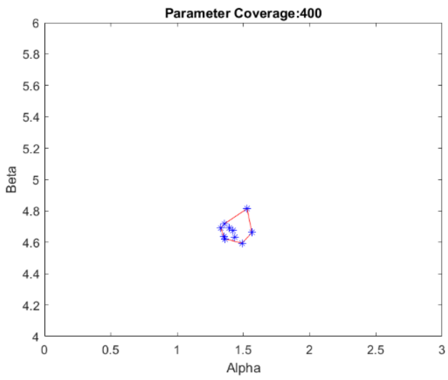
Figure 5.10: Parameter convergence of HAACO tested on st70.tsp.



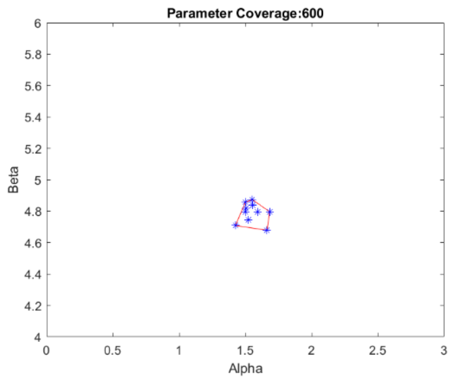
(a)



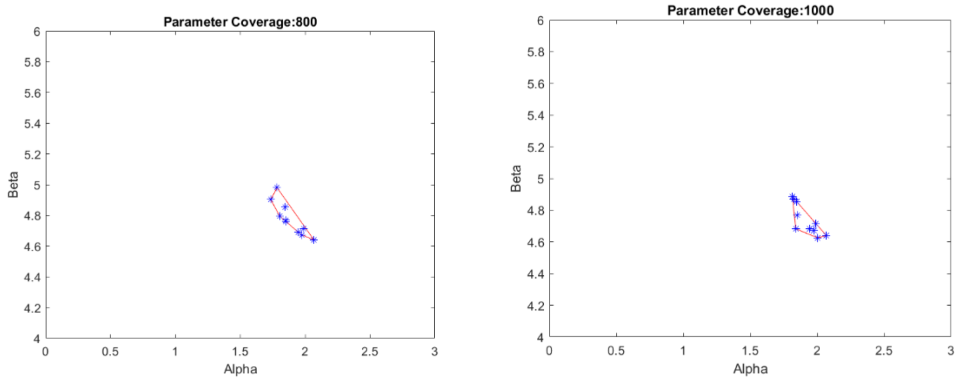
(b)



(c)

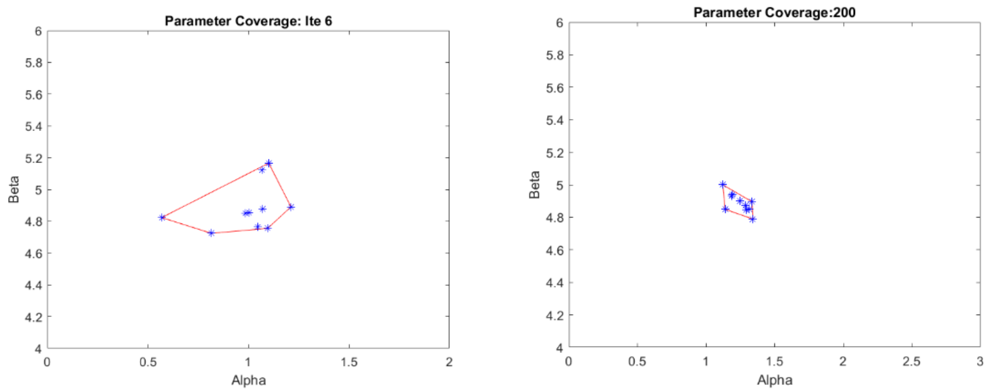


(d)

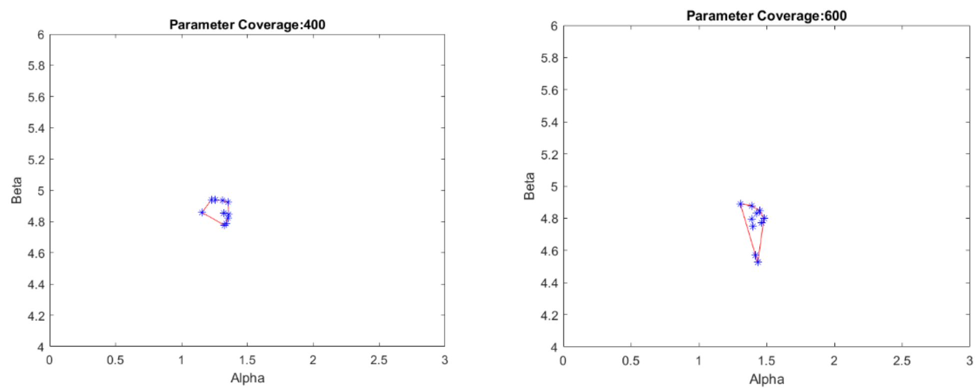


(e) (f)

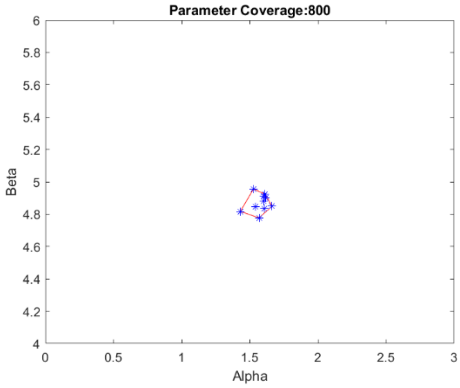
Figure 5.11: Parameter convergence of HAACO tested on lin105.tsp.



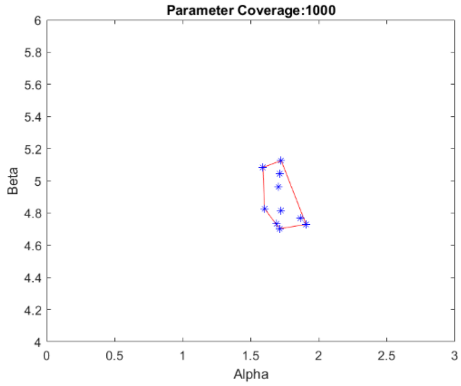
(a) (b)



(c) (d)



(e)

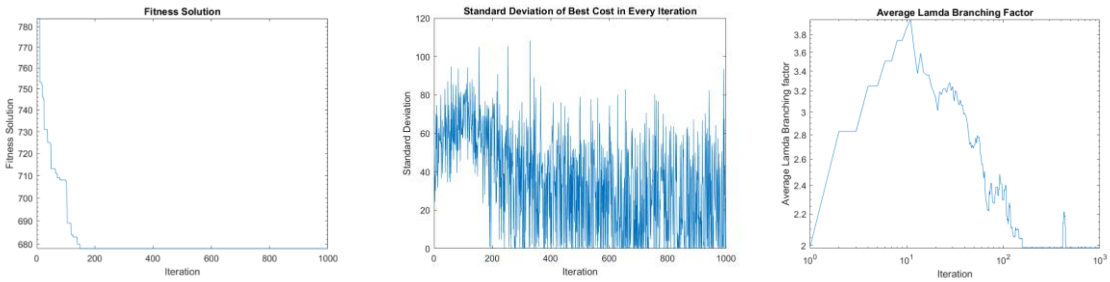


(f)

Figure 5.12: Parameter convergence of HAACO tested on ch150.tsp.

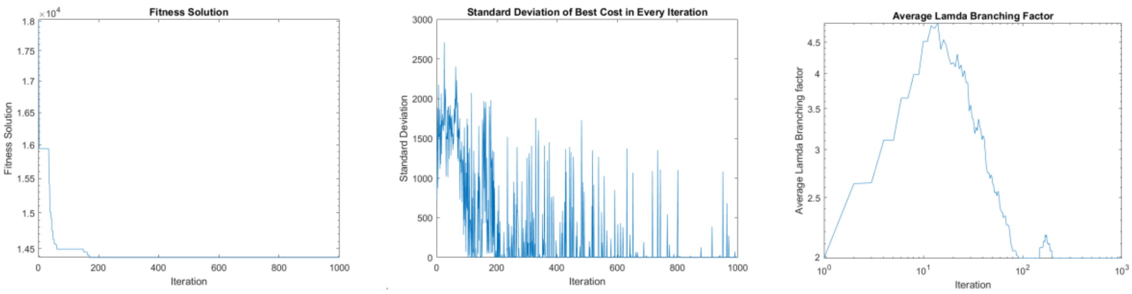
5.4.3 Algorithm's Behaviour

Figures 5.13a, 5.14a and 5.15a indicate that the algorithm quickly converges to good solutions before exploiting these parameter settings to achieve better solutions. The change from the exploration phase to exploitation can be noticed in Figure 5.13b, 5.14b and 5.15b respectively where the algorithm has a high standard deviation early on before it converges to sub-optimal solutions and further perturbation is necessary to locate better solutions. The standard deviations also indicate that the proposed approach did not enter stagnation behaviour in all three TSP instances. However, the standard deviation of HAACO is almost 0 when tested on lin105.tsp as indicated in Figure 5.15b. The reason could be that the algorithm converges to the optimal solution that was found very early on during the search process. Lastly, the fast convergence claim is supported by the average lambda branching factor of HAACO in all three TSP instances where the number of branches being explored starts to decrease from iteration 200 onwards as can be seen in Figure 5.13c, 5.14c and 5.15c respectively. This also indicates the nature of the algorithm which is capable of escaping from local optima as observed in Figures 5.13c and 5.14c where the algorithm still explores new edges even after a period of convergence and this can be due to the Gaussian mutation operator that leads the algorithm to new neighbouring regions in the parameter space. The advantage of the algorithm is in its ability to converge to good parameter solutions quickly and exploit those values to obtain optimal settings. In addition, the small mutation width that was used in this study allows the ants to move to neighbouring regions rather than execute a large jump to non-optimal search space. Finally, it is important to note that this algorithm does not incur any additional function evaluation calls to implement the adaptive approach unlike other hybrid approaches that utilizes PSO to optimize the parameters [47]. In addition to that, this approach also is not computationally extensive as it doesn't demand heavy computation or computational resources compared to Parallel ACO approach [48].



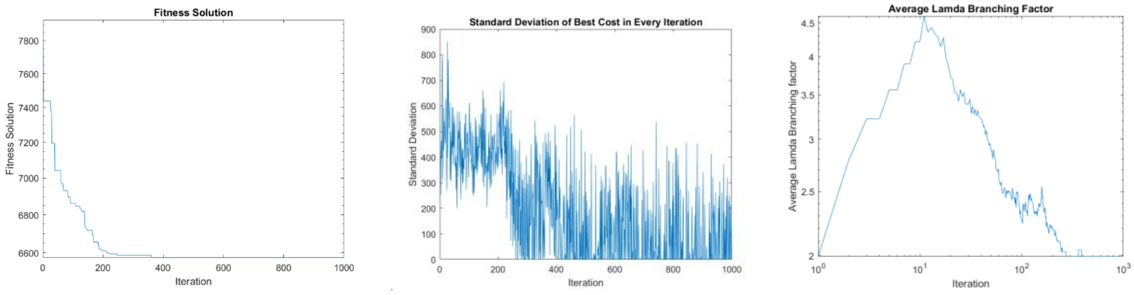
(a) (b) (c)

Figure 5.13: Analysis of HAACO tested on st70.tsp.



(a) (b) (c)

Figure 5.14: Analysis of HAACO tested on lin105.tsp.



(a) (b) (c)

Figure 5.15: Analysis of HAACO tested on ch150.tsp.

5.5 Conclusion

A heterogeneous adaptive approach with Gaussian mutation was presented in this chapter. The relevance of the adaptive approach including the introduction of the greedy ants, adaptive interval and amount of mutation were studied using several symmetrical TSP instances. Empirical studies have shown that adapting the parameters of an algorithm has an advantage over fine-tuned algorithms in their ability to react to the changing features of a search landscape as it approaches the optimum. The proposed approach is more feasible than the time-consuming task of fine-tuning the parameters that usually require prior knowledge of the algorithm as well as the problem in hand. The proposed approach allows the algorithm to be able to adapt or interchange between different strategies i.e exploration and exploitation throughout the search process. Comparison against the two state-of-the-art algorithms also suggests that the proposed approach is able to explore the parameter space to locate the instance-optimal settings that would then allow the ants to explore the search space to find better fitness solutions. This enhances the robustness of the algorithm towards parameter settings. In addition to this, the algorithm is able to escape from local optima with the introduction of the mutation operator and prevents the algorithm from going into stagnation behaviour by allowing the ants to exhibit individual behaviour. The small amount of mutation allows the algorithm to explore neighbouring regions rather than performing a random jump. It is also noticeable that this approach suits one with a low or small budget of function evaluations as it is capable of converging to good or sub-optimal parameters quickly before the exploitation of those areas to locate optimal settings. Future work requires the investigation of the approach on larger TSP instances. In conclusion, this study has explored the possibilities of enabling the algorithm to self-adapt its α and β parameters thus allowing the algorithm to explore both parameter space and search space simultaneously to locate better solutions.

Chapter 6 Conclusion & Future Work

The last decade has seen a rise in the interest of solving combinatorial hard optimization algorithms using algorithms inspired by swarm intelligence. These algorithms use a decentralized approach and collective intelligence that are adopted from the behaviour of social insects such as flock of birds or swarm of bees. ACO is a highly successful branch of swarm intelligence that is inspired by the foraging behaviour of real ants. It has been widely reported that the performance of ACO is critically dependant on the parameter settings. However, parameter tuning is a non-trivial task mainly because each problem or even each problem instance has its own optimal parameter settings but it is impossible to tune the parameters for each and every problem and problem instances. Even if one tries to, a deep understanding of both the algorithm and the problem being solved is required. In addition, the optimal parameter settings may change as the search landscape changes.

This thesis addresses and presents new insights into the implementation of individual 'behavioural traits' via the heterogeneous ACO approach for static TSP and PCB drilling problems, although the approach is generic and not limited to these problems. This research work takes inspiration from recent studies that have shown that real ants in a colony do differ in terms of their behaviours and preferences. The animal behaviour scientists suggest that the different personalities could arise due to the ants' genetics, its morphology, the changes in the environment or the availability of food resources. Therefore, the idea is to develop a heterogeneous ACO that closely resembles the newfound nature of real ant colonies that is able to improve on the performance of the conventional ACO algorithms as well as overcome the problems that have been discussing in previous chapters.

6.1 Significance of the Research & Contributions

6.1.1 Heterogeneous Ant System (HAS) & Heterogeneous Max-Min Ant System (HMMAS)

Chapter 3 implements and analyses a heterogeneous approach with ants randomly initialized from a uniform distribution within a pre-defined range of values. The results clearly show that the heterogeneous approach in ACO is able to produce improved performance over the standard, parameter-tuned algorithms on which they are based with increased robustness to parameter settings. The performance difference was particularly marked when heterogeneity was implemented on the Ant System (AS). This is likely to be due to the greater contribution of each ant to the pheromone trail, highlighting the effect of diversity where all ants were allowed to modify the pheromone landscape unlike in the Max-Min Ant System (MMAS). Even though HMMAS still produced sup performance over the base algorithm. the smaller gains made by HMMAS can be explained by the increased performance of the base algorithm, locating solutions closer to the optimum and the effect of individual variance or heterogeneity is limited in HMMAS due to algorithm's limitation of only a single agent to modify the pheromone limiting the overall heterogeneity advantage. Having a variety of 'behavioural traits' rather than a single behaviour shows the advantage in the performance of the algorithm. Recording the best performing alpha and beta values provided some support for the parameter values suggested by both Dorigo et al [6] and Stützle et al [5], but also highlighted instances where these parameter settings were not optimal. The discovery of distinct distributions of parameter settings for alpha and beta is interesting and demonstrates the algorithms' sensitivity and robustness to these parameters. These distributions remained stable despite being tested on multiple problem sizes. This study shares a new understanding and knowledge of the advantages of the heterogeneity approach in ACO especially and swarm intelligence as a whole thus acting as a guide in developing effective algorithms. Based on the results discussed in Chapter 3, it can be concluded that the proposed algorithm is highly competitive when compared against conventional ACO algorithms hence it provides a base for further research work.

6.1.2 Gaussian Heterogeneous Max-Min Ant System (GHMMAS)

GHMMAS is introduced in Chapter 4 as another heterogeneous approach to ACO that implements different ’behavioural traits’ within a population drawn from a Gaussian distribution thus enabling each ant pre-assigned with a pair of traits in relation to α, the relative importance of pheromone, and β, the relative importance of heuristics. The diversity in the population introduced by Gaussian distribution enables GHMMAS to display a better performance compared to HMMAS by virtue of initializing the population centered around values suggested in [40]. It was also found that performance can be improved through the introduction of the right amount of spread introduced via the standard deviation of the Gaussian distribution reduces and eliminates ants in the extreme-low and extreme-high values of the range which might cause the algorithm to perform poorly. The results also highlight ants with a higher preference towards heuristics (higher β) are useful during the early stages of the search process while ants with a slightly lower preference towards heuristics perform better during later stages. The shift in pattern is because the pheromone landscape has less information initially thus requiring ants with a higher preference of heuristics information to build good solutions and as the search progress, more pheromone is deposited hence ants with higher α and lower β perform better during later stages. In regard to the PCB drilling problem, the huge amount of diversity introduced by GHMMAS enables the algorithm to fully explore the search space to locate the optimal solution.

6.1.3 Heterogeneous Adaptive Ant Colony Optimization (HAACO)

Chapter 5 extended the study to implement and analyse a self-adaptive approach of the heterogeneous ACO by adapting the parameters as the search progresses. The approach is based on the collective intelligence of the heterogeneous individuals and their interactions in not only exploring the fitness landscape but also the parameter landscape simultaneously. This allows the algorithm to automatically adjust its search strategy by alternating between exploration and exploitation whenever required. Empirical results tested on several TSP instances have shown that the adaptation of the algorithm's parameters have an advantage over fine-tuned algorithms in their ability to react to the changing features of a search landscape as it approaches the optimum. The proposed approach is more feasible than the time-consuming task of fine-tuning the parameters that usually requires prior knowledge of the algorithm as well as the problem that is being solved. The algorithm is also able to autonomously locate instance-optimal parameter settings as demonstrated by the convex hull diagrams that showcased the algorithm's ability to explore the parameter space thus rendering the algorithm to be scalable to various problem sizes. It is also noticeable that this approach suits applications with a low or small budget of function evaluations as it is capable of converging to good or sub-optimal parameters quickly before the exploitation of those areas to locate optimal settings.The experimental evaluations such as the best cost found over a certain number of trials, average best costs and rank indicators indicate that the proposed approach outperforms various state-of-the-art hybrid approaches as illustrated in Chapter 5. Furthermore, ACO algorithms are known to suffer from stagnation behaviour where the artificial ants construct the same tours repeatedly from early stages. This condition is more likely to happen if the parameters are not optimal as the algorithm is unable to improve on the solution found hence higher possibility of getting trapped in local optima. Therefore, the average branching factor illustrations suggest that the proposed approach is able to overcome the stagnation behaviour issue as the individual 'behavioural traits' allows each ant to have a different view, preference or perspective in tackling the search landscape.

6.2 Limitations of The Proposed Work

There are some limitations in the contributions of this thesis as follows:

1. The main framework of the proposed approach revolves around both α and β, which are the two main parameters in ACO that affect or determine the performance of the algorithm. In addition to this, the implementation of the proposed approaches, especially the range of values, are based on the parameter suggestion by Dorigo et al [6].
2. The proposed approaches were tested mainly on static optimization problems such as TSP and PCB drilling. Even though the later is a more realistic real-world problem, the proposed approaches ought to be tested on dynamic problems where the global optima changes over time. Preliminary results in this thesis suggest that the heterogeneous approach may produce good performance in dynamic problems due to its ability to explore the search and the parameter landscape simultaneously.

6.3 Conclusion

Overall, empirical results presented and discussed in this thesis have shown significant improvement in the proposed approach over the base algorithms while the self-adaptive heterogeneous ACO is shown to perform better than state-of-the-art hybrid ACO algorithms. Both of these are the result of replacing the static parameters with that of the heterogeneous approach where each ant is incorporated with its own distinguished parameters known as ‘behavioural traits’. This allows the ants to explore and exploit throughout the search process. Furthermore, the self-adaptive heterogeneous approach has the ability to adapt these parameters as the search progresses and thus is able to explore and exploit the search regions autonomously.

In conclusion, the proposed approach is able to maintain diversity in the population of ants that allows the algorithm to be able to effectively achieve a balance between exploration and exploitation of the search space throughout the search process. It is also important to note that the algorithm is able to achieve excellent performance when tested on various problem sizes. This shows that the algorithm is robust to parameter settings and scalable. Lastly, the heterogeneous approach can be applied to real-world problems due to its improved performance coupled with the simplicity in implementation. Some likely problems would include the delivery truck route optimization, collision avoidance, telecommunication routing network and many more.

6.4 Future Work

The effectiveness of the heterogeneous ACO introduced in this study can be further explored in several ways. One of the areas for further analysis is to explore initializing the ants from different distributions such as binomial, Poisson or exponential distributions. This will then give a much clearer picture of the best distribution to implement heterogeneity in ACO.

The self-adaptive heterogeneous ACO approach has been developed in a way that is well suited for dynamic environments where the global optimum changes over time. Therefore, the algorithm should be able to adapt to the changes in order to locate the global optimum. In addition to that, many of the real-world problems are dynamic in nature such as route planning or scheduling problems. Therefore, the effectiveness and suitability of the heterogeneous approach in solving the real-world problem can be implemented and analysed. The heterogeneous approach can incorporate some other behaviours of real ants such as recruitment learning. Rather than random initialization, ants can be defined to have specific roles such as explorative ants, lazy ants that prefers random walk and many more. In addition, ants can have the ability to recruit other ants when the need arises such as promising region in a solution space. This can either be at individual-level or multi-colony approach where ants with same behaviour can be grouped into a sub-colony and exchange of information between the sub-colonies. Finally, the heterogeneous approach in this study as well as previous studies in PSO indicate the ability of this approach to improve on the performance of the algorithm, mainly algorithms in the field of swarm intelligence., The heterogeneous approach is wide in nature and has wide applicability thus can possibly be applied to improve other swarm algorithms such as artificial bee colony or the more recent intelligent water drop algorithm.

References

[1] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms Second Edition*. Luniver Press, Frome, UK, 2008.

[2] L. M. Gambardella, “Coupling Ant Colony System with Local Search,” UNIVERSITE LIBRE DE BRUXELLES, 2005.

[3] M. Dorigo and T. Stützle, *Ant colony optimization*. MIT Press, 2004.

[4] M. Dorigo, M. Birattari, and T. Stützle, “Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique,” *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.

[5] T. Stutzle and H. Hoos, “MAX MIN Ant System and Local Search for the Traveling Salesman Problem,” *Ieee Int. Conf. Evol. Comput.*, pp. 309–314, 1997.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996.

[7] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[8] G. (Gerhard) Reinelt, *The traveling salesman : computational solutions for TSP applications*. Springer-Verlag, 1994.

[9] Christian Blum, “ACO Applied to Group Shop Scheduling: A Case Study on Intensification and Diversification,” in *ANTS ’02 Proceedings of the Third International Workshop on Ant Algorithms*, 2002, pp. 14–27.

[10] M. Dorigo and T. Stützle, “Ant colony optimization: Overview and recent advances,” *Int. Ser. Oper. Res. Manag. Sci.*, vol. 272, no. May, pp. 311–351, 2019.

[11] R. Sagban, K. Ruhana Ku-Mahamud, M. Shahbani, and A. Bakar, “Nature-inspired Parameter Controllers for ACO-based Reactive Search,” *Res. J. Appl. Sci. Eng. Technol.*, vol. 11, no. 1, pp. 109–117, 2015.

[12] L. Isabel and D. A. Melo, “Self Adaptation in Ant Colony Optimisation,” University of Coimbra, Portugal, 2018.

[13] N. R. Sabar and A. Aleti, “An Adaptive Memetic Algorithm for the Architecture

Optimisation Problem," in *Artificial Life and Computational Intelligence. ACALCI 2017*, 2017.

[14] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1. Elsevier B.V., pp. 19–31, 01-Mar-2011.

[15] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "An adaptive particle swarm optimization algorithm based on optimal parameter regions," *2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc.*, vol. 2018-Janua, pp. 1–8, 2018.

[16] F. J. L. Cláudio F. Lima, Zbigniew Michalewicz, Ed., *Parameter Setting in Evolutionary Algorithms*, Volume 54. Springer Science & Business Media, 2007, 2007.

[17] B. Doerr and C. Doerr, "Theory of Parameter Control for Discrete Black-Box Optimization: Provable Performance Gains Through Dynamic Parameter Choices," 2020.

[18] C. R. Stephens, I. García Olmedo, J. M. Vargas, and H. Waelbroeck, "Self-Adaptation in Evolving Systems," 1997.

[19] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO inspired by ants," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7461 LNCS, pp. 188–195.

[20] D. Gaertner and K. Clark, "On Optimal Parameters for Ant Colony Optimization algorithms TSP classifications," pp. 83–89, 2005.

[21] A. Aleti, "An Adaptive Approach to Controlling Parameters of Evolutionary Algorithms," Swinburne University of Technology, 2012.

[22] K. Y. Wong and Komarudin, "Parameter tuning for ant colony optimization: A review," *Proc. Int. Conf. Comput. Commun. Eng. 2008, ICCCE08 Glob. Links Hum. Dev.*, pp. 542–545, 2008.

[23] P. A. Kulkarni, "Explore-exploit-explore in ant colony optimization," in *Advances in Intelligent Systems and Computing*, 2019, vol. 828, pp. 183–189.

[24] A. E. Negulescu, S. C. Negulescu, and I. Dzitac, "Balancing Between Exploration and Exploitation in ACO," *Int. J. Comput. Commun. Control*, vol. 12, no. April, pp. 265–275, 2017.

[25] Y. Liu, J. Ma, S. Zang, and Y. Min, “Dynamic Path Planning of Mobile Robot Based on Improved Ant Colony Optimization Algorithm,” in *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, 2019, pp. 248–252.

[26] M. Mavrovouniotis, “Ant Colony Optimization in Stationary and Dynamic Environments,” 2013.

[27] O. Blight, G. Albet Díaz-Mariblanca, X. Cerdá, and R. Boulay, “A proactive–reactive syndrome affects group success in an ant species,” *Behav. Ecol.*, vol. 27, no. 1, pp. 118–125, Jan. 2016.

[28] A. P. Modlmeier and S. Foitzik, “Productivity increases with variation in aggression among group members in Temnothorax ants,” *Behav. Ecol.*, vol. 22, no. 5, pp. 1026–1032, Sep. 2011.

[29] T. A. O’shea-Wheller, N. Masuda, A. B. Sendova-Franks, and N. R. Franks, “Variability in individual assessment behaviour and its implications for collective decision-making.”

[30] E. J. H. Robinson, O. Feinerman, and N. R. Franks, “How collective comparisons emerge without individual comparisons of the options.”

[31] “They’ve Got Personality: Ant ‘Superorganisms’ Have Unique Temperaments | Live Science.” [Online]. Available: https://www.livescience.com/61114-ant-superorganism-personality.html. [Accessed: 13-Feb-2020].

[32] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence : from natural to artificial isystems*. Oxford University Press, 1999.

[33] T. A. O’Shea-Wheller, N. Masuda, A. B. Sendova-Franks, and N. R. Franks, “Variability in individual assessment behaviour and its implications for collective decision-making,” *Proc. R. Soc. London B Biol. Sci.*, vol. 284, no. 1848, 2017.

[34] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, “Error, communication and learning in ant societies,” *Eur. J. Oper. Res.*, vol. 30, no. 2, pp. 168–172, 1987.

[35] S. Goss, J. L. Deneuborg, and J. M. Pasteels, “Self-organized shortcutes in the Argentine ant,” *Naturwissenschaften*, vol. 76, no. 1959, pp. 579–581, 1989.

[36] M. Mavrovouniotis, “Ant Colony Optimization in Stationary and Dynamic

Environments," 2013.

[37] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

[38] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: An autocatalytic Optimizing Process," 1991.

[39] M. Dorigo, V. Maniezzo, A. Colorni, M. Dorigo, V. Maniezzo, and A. Colorni, "Title: Positive feedback as a search strategy Positive Feedback as a Search Strategy," 1991.

[40] A. Colorni, M. Dorigo, and V. Maniezzo, "The Ant System: An autocatalytic optimizing process." Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.

[41] A. Colorni, M. Dorigo, and V. Maniezzo, "An investigation of some properties of an 'Ant algorithm,'" in *PPSN 92*, 1992, pp. 509–520.

[42] L. M. Gambardella and M. Dorigo, "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem," pp. 252–260, 1995.

[43] T. Stützle, H. H. Hoos, and H. Hoos, "Improvements on the Ant-System: Introducing the MAX-MIN Ant System Improving the Ant System: A Detailed Report on the MAX-MIN Ant System," 1996.

[44] B. Bullnheimer, R. F. Hartl, and C. Strauu, "A New Rank Based Version of the Ant System -A Computational Study," *Cent. Eur. J. Oper. Res. Econ.*, vol. 7 (1), pp. 25–38, 1999.

[45] T. Stutzle *et al.*, "Parameter Adaptation in Ant Colony Optimization," in *Autonomous Search*, 2010, pp. 191–215.

[46] D. Gaertner, "Natural Algorithms for Optimisation Problems Final Year Project Report," Imperial College London, 2004.

[47] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem," *Appl. Soft Comput.*, vol. 30, no. C, pp. 484–490, May 2015.

[48] Ş. Gülcü, M. Mahi, Ö. K. Baykan, and H. Kodaz, "A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem," *Soft Comput.*, vol. 22, no. 5, pp. 1669–1685, Mar. 2018.

[49] R. Laptik, “Ant system initial parameters distribution,” *Elektron. ir Elektrotechnika*, vol. 4, no. 4, pp. 85–88, 2011.

[50] E. Ridge and P. Thesis, “Design of Experiments for the Tuning of Optimisation Algorithms,” 2007.

[51] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” *Theor. Comput. Sci.*, vol. 344, no. 344, pp. 243–278, 2005.

[52] P. Pellegrini and D. Favaretto, “Quantifying the exploration performed by metaheuristics,” *J. Exp. Theor. Artif. Intell.*, vol. 24, no. 2, pp. 247–266, 2012.

[53] P. Pellegrini, T. Stützle, and M. Birattari, “Off-line vs. On-line Tuning: A Study on MAX-MIN Ant System for the TSP.”

[54] P. Pellegrini, T. Stützle, and M. Birattari, “A critical analysis of parameter adaptation in ant colony optimization,” *Swarm Intell*, vol. 6, pp. 23–48, 2012.

[55] A. E. Eiben, Z. Michalewicz, M. Schoenauer, and J. E. Smith, “Parameter control in evolutionary algorithms,” *Stud. Comput. Intell.*, vol. 54, no. 2, pp. 19–46, 2007.

[56] H. M. Botee and E. Bonabeau, “Evolving Ant Colony Optimization,” *Adv. Complex Syst.*, vol. 01, no. 02n03, pp. 149–159, 1998.

[57] T. Stützle *et al.*, *Parameter Adaptation in Ant Colony Optimization*. Springer-Verlag Berlin Heidelberg, 2011.

[58] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, “Parameter Control in Evolutionary Algorithms: Trends and Challenges,” 2012.

[59] N. Z. Naqvi, H. K. Matheru, and K. Chadha, “Review Of Ant Colony Optimization Algorithms OnVehicle Routing Problems And Introduction ToEstimation-Based ACO,” *Proc. Int. Conf. …*, vol. 8, pp. 161–166, 2011.

[60] W. C. E. Lim, G. Kanagaraj, and S. G. Ponnambalam, “PCB drill path optimization by combinatorial cuckoo search algorithm.,” *ScientificWorldJournal.*, vol. 2014, p. 264518, Feb. 2014.

[61] A. E. Eiben and M. Jelasity, “A critical note on experimental research methodology in EC,” *Proc. 2002 Congr. Evol. Comput. CEC 2002*, vol. 1, no. May, pp. 582–587, 2002.

[62] M. Birattari and M. Dorigo, “How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs?,” *Optim. Lett.*, vol. 1, no. 3, pp. 309–311, 2007.

[63] N. Ivkovic, D. Jakobovic, and M. Golub, “Measuring Performance of Optimization Algorithms in Evolutionary Computation,” *Int. J. Mach. Learn. Comput.*, vol. 6, no. 3, pp. 167–171, 2016.

[64] R. Sagban, K. R. Ku-Mahamud, and M. S. Abu Bakar, “ACOustic: A Nature-Inspired Exploration Indicator for Ant Colony Optimization.,” *ScientificWorldJournal.*, vol. 2015, p. 392345, 2015.

[65] L. M. Gambardella and M. Dorigo, “Ant-Q: A Reinforcement Learning approach to the traveling salesman problem,” *Mach. Learn. Proc. 1995*, pp. 252–260, Jan. 1995.

[66] Y. Ulrich *et al.*, *Emergent behavioral organization in heterogeneous groups of a social insect*, no. March. 2020.

[67] E. A. Langridge, A. B. Sendova-Franks, and N. R. Franks, “How experienced individuals contribute to an improvement in collective performance in ants,” *Behav. Ecol. Sociobiol.*, vol. 62, no. 3, pp. 447–456, Jan. 2008.

[68] J. N. Pruitt and S. E. Riechert, “How within-group behavioural variation and task efficiency enhance fitness in a social group,” *Proc. R. Soc. B Biol. Sci.*, vol. 278, no. 1709, pp. 1209–1215, 2011.

[69] C. Doran, M. C. Stumpe, A. Sendova-Franks, and N. R. Franks, “Exploration adjustment by ant colonies.”

[70] A. P. Modlmeier, J. E. Liebmann, and S. Foitzik, “Diverse societies are more productive: a lesson from ants.”

[71] A. Hui and N. Pinter-Wollman, “Individual variation in exploratory behaviour improves speed and accuracy of collective nest selection by Argentine ants,” *Anim Behav*, vol. 93, pp. 261–266, 2014.

[72] M. A. Montes De Oca, J. Peña, T. Stützle, C. Pinciroli, and M. Dorigo, “Heterogeneous particle swarm optimizers,” *2009 IEEE Congr. Evol. Comput. CEC 2009*, no. January, pp. 698–705, 2009.

[73] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” pp. 1942–1948, 1995.

[74] A. Engelbrecht, “Heterogeneous particle swarm optimization,” *Swarm Intell.*, pp. 191–202, 2010.

[75] O. Olorunda and A. P. Engelbrecht, “An analysis of heterogeneous cooperative algorithms,” *2009 IEEE Congr. Evol. Comput. CEC 2009*, pp.

1562–1569, 2009.

[76] D. Boudjehem and B. Boudjehem, "Improved heterogeneous particle swarm optimization," *J. Inf. Optim. Sci.*, vol. 38, no. 3–4, pp. 481–499, 2017.

[77] S. Tsutsui, "cAS: Ant Colony Optimization with Cunning Ants," in *Parallel Problem Solving from Nature - PPSN IX*, 2006.

[78] D. C. Matthews, A. M. Sutton, D. Hains, and L. D. Whitley, "Improved Robustness through Population Variance in Ant Colony Optimization," Springer, Berlin, Heidelberg, 2009, pp. 145–149.

[79] J. W. Lee and J. J. Lee, "Novel ant colony optimization algorithm with path crossover and heterogeneous ants for path planning," in *Proceedings of the IEEE International Conference on Industrial Technology*, 2010.

[80] S. C. Negulescu and A. E. Lascu, "Synthetic Genes for Artificial Ants . Diversity in Ant Colony Optimization Algorithms," vol. V, no. 2, pp. 216–223, 2010.

[81] C. Chira, D. Dumitrescu, and C. M. Pintea, "Heterogeneous sensitive ant model for combinatorial optimization," *Genet. Evol. Comput.*, p. 163, 2008.

[82] C. Chira, D. Dumitrescu, and C.-M. Pintea, "LEARNING SENSITIVE STIGMERGIC AGENTS FOR SOLVING COMPLEX PROBLEMS," *Comput. Informatics*, vol. 29, pp. 337–356, 2010.

[83] M. Yoshikawa, "Adaptive Ant Colony Optimization with Cranky Ants," in *Intelligent Automation and Computer Engineering*, X. Huang, S.-I. and Ao, and O. and Castillo, Eds. Springer Netherlands, 2010, pp. 41–52.

[84] A. Hara, S. Matsushima, T. Ichimura, and T. Takahama, "Ant colony optimization using exploratory ants for constructing partial solutions," in *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 2010.

[85] A. M. Abdelbar and D. C. Wunsch, "Promoting search diversity in ant colony optimization with stubborn ants," vol. 12, pp. 456–462, 2012.

[86] N. Zufferey, J. Farres, and R. Glardon, "Ant metaheuristic with adapted personalities for the vehicle routing problem," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9335, pp. 3–15, 2015.

[87] K. O. Yuichiro Sueoka, Kazuki Nakayama, Masato Ishikawa, Yasuhiro

Sugimoto, “On Heterogeneity in Foraging by Ant-Like Colony: How Local Affects Global and Vice Versa,” in *ANTS 2016*, 2016.

[88] P. Zhang and J. Lin, “An adaptive heterogeneous multiple ant colonies system,” in *Proceedings - 2010 International Conference of Information Science and Management Engineering, ISME 2010*, 2010.

[89] L. Melo, F. Pereira, and E. Costa, “Extended experiments with ant colony optimization with heterogeneous ants for large dynamic traveling salesperson problems,” *Proc. - 14th Int. Conf. Comput. Sci. Its Appl. ICCSA 2014*, pp. 171–175, 2014.

[90] M. Mavrovouniotis, S. Yang, and X. Yao, “Multi-Colony Ant Algorithms for the Dynamic Travelling Salesman Problem.”

[91] C. R. Smith, K. E. Anderson, C. V. Tillberg, J. Gadau, and A. V. Suarez, “Caste Determination in a Polymorphic Social Insect: Nutritional, Social, and Genetic Factors,” *Am. Nat.*, vol. 172, no. 4, pp. 497–507, Oct. 2008.

[92] C. Blum, “Ant colony optimization: Introduction and recent trends,” *Phys. Life Rev.*, vol. 2, pp. 353–373, 2005.

[93] M. L. Pilat and T. White, “Using Genetic Algorithms to optimize ACS-TSP.”

[94] J.-W. Lee, B.-S. Choi, K.-T. Park, and J.-J. Lee, “Comparison between heterogeneous ant colony optimization algorithm and Genetic Algorithm for global path planning of mobile robot,” *2011 IEEE Int. Symp. Ind. Electron.*, pp. 881–886, 2011.

[95] P. Wang, Y. Zhang, and D. Yan, “An improved self-adaptive ant colony algorithm based on genetic strategy for the traveling salesman problem Optimal solution for travelling salesman problem using heuristic shortest path algorithm with imprecise arc length AIP Conference An Improved Self-,” *Proceedings*, vol. 1820, p. 40061, 2018.

[96] W. Deng, H. Chen, and H. Li, “A Novel Hybrid Intelligence Algorithm for Solving Combinatorial Optimization Problems ALGORITHMS,” vol. 8, no. 4, pp. 199–206, 2014.

[97] T. A. Ei-Mihoub, A. Hopgood, and I. A. Aref, “Self-adaptive Hybrid Genetic Algorithm using an Ant-based Algorithm,” in *2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 2014.

[98] A. Ouyang and Y. Zhou, “An improved PSO-ACO algorithm for solving large-

scale TSP," *Adv. Mater. Res.*, vol. 143–144, no. October, pp. 1154–1158, 2011.

[99] O. Mersmann, B. Bischl, J. Bossek, H. Trautmann, M. Wagner, and F. Neumann, "Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness."

[100] W. Elloumi, H. El Abed, A. Abraham, and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP," *Appl. Soft Comput. J.*, vol. 25, pp. 234–241, 2014.

[101] W. Elloumi, N. Baklouti, A. Abraham, and A. M. Alimi, "Hybridization of Fuzzy PSO and Fuzzy ACO applied to TSP," *13th Int. Conf. Hybrid Intell. Syst. HIS 2013*, pp. 105–110, 2014.

[102] M. Bhattacharya, "Diversity Handling In Evolutionary Landscape."

[103] T. J. Czaczkes, C. Grü, S. M. Jones, and F. L. W. Ratnieks, "Synergy between social and private information increases foraging efficiency in ants," 2011.

[104] Deborah M Gordon and D. M. Gordon, "Ant encounters: interaction networks and colony behavior," *Prim. complex Syst.*, p. 167, 2010.

[105] T. J. Cole and D. G. Altman, "Statistics Notes: What is a percentage difference?," *BMJ*, vol. 358, p. j3663, Aug. 2017.

[106] G. Reinelt, "TSPLIB." [Online]. Available: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html. [Accessed: 31-May-2018].

[107] M. Dorigo and T. Stützle, *Ant colony optimization*. MIT Press, 2004.

[108] J. Brownlee, *Clever Algorithms, Nature-Inspired Programming Recipes*. Australia: Creative Commons, 2011.

[109] Reinelt and Gerhard, *The traveling salesman : computational solutions for TSP applications*. Springer-Verlag, 1994.

[110] M. Maur, M. López-Ibáñez, and T. Stützle, "Pre-scheduled and adaptive parameter variation in MAX-MIN ant system," in *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 2010.

[111] M. Birattari, Y. Zhi, B. Prasanna, and T. Stũtzle, "F-Race and iterated F-Race: An overview," in *Experimental Methods for the Analysis of Optimization Algorithms*, 2010, no. October.

[112] L. Zhang, F. Yang, and A. Z. Elsherbeni, "ON THE USE OF RANDOM

VARIABLES IN PARTICLE SWARM OPTIMIZATIONS: A COMPARATIVE STUDY OF GAUSSIAN AND UNIFORM DISTRIBUTIONS," 2009.

[113] S. G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean J. Anesthesiol.*, vol. 70, no. 2, pp. 144–156, Apr. 2017.

[114] L. Eduardo da Silva, G. Lambert-Torres, L. Eduardo Borges da Silva, T. Silveira, R. Menezes Salgado, and H. César Brandão de Oliveira, "Max-Min Paraconsistent Ant Algorithm," *Int. J. Res. Surv.*, vol. 8, no. 2, pp. 599–604, 2014.

[115] Z. A. N. Wahida, M. F. F. Ab Rashid, and N. M. Z. Nik Mohamed, "A Review of Multi-holes Drilling Path Optimization Using Soft Computing Approaches," *Arch Comput. Methods Eng*, vol. 26, pp. 107–118, 2017.

[116] H. Abdullah, M. S. Zakaria, T. M. L. T. Zahari, N. Talib, W. K. Lee, and A. Saleh, "Simulation of ant colony optimization on hole making performance," no. July, pp. 3–4, 2018.

[117] O. Montiel-Ross, N. Medina-Rodríguez, R. Sepúlveda, and P. Melin, "Methodology to optimize manufacturing time for a CNC using a high performance implementation of ACO," *Int. J. Adv. Robot. Syst.*, vol. 9, pp. 1–10, 2012.

[118] A. T. Abbas, K. Hamza, and M. F. Aly, "CNC Machining Path Planning Optimization for Circular Hole Patterns via a Hybrid Ant Colony Optimization Approach," *Mech. Eng. Res.*, vol. 4, no. 2, 2014.

[119] T. Eldos, A. Kanan, and A. Aljumah, "Solving the printed circuit board drilling problem by ant colony optimization algorithm," *Lect. Notes Eng. Comput. Sci.*, vol. 1, pp. 584–588, 2013.

[120] T. Eldos, "Adapting The Ant Colony Optimization Algorithm To The Printed Circuit Board Drilling Problem," *Recent Res. Telecommun. Informatics, Electron. Signal Process.*, pp. 58–63.

[121] M. Maur, "Adaptive Ant Colony Optimization for the Traveling Salesman Problem."

[122] R. Hinterding, Z. Michalewicz, and A. E. Eiben, "Adaptation in evolutionary computation: a survey," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 65–69.

[123] P. J. Angeline, "Adaptive and Self-adaptive Evolutionary Computations," in

Computational Intelligence: A Dynamic Systems Perspective, 1995, pp. 152--163.

[124] J. Kennedy, "Particle swarm: Social adaptation of knowledge," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1997, pp. 303–308.

[125] F. V. Nepomuceno and A. P. Engelbrecht, "Behavior changing schedules for heterogeneous particle swarms," *Proc. - 1st BRICS Ctries. Congr. Comput. Intell. BRICS-CCI 2013*, pp. 112–118, 2013.

[126] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Inf. Sci. (Ny).*, vol. 181, no. 20, pp. 4515–4538, Oct. 2011.

[127] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO for real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013, pp. 361–368.

[128] X.-F. F. Xie, W.-J. J. Zhang, and Z.-L. L. Yang, "Adaptive particle swarm optimization on individual level," in *International Conference on Signal Processing Proceedings, ICSP*, 2002, vol. 2, pp. 1215–1218.

[129] M. L. Pilat and T. White, "Using Genetic Algorithms to Optimize ACS-TSP," in *Ant Algorithms. ANTS 2002*, 2002, pp. 282–287.

[130] G. Di Caro and P. Marco Dorigo, "Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks," 2004.

[131] M. M. Sysło, N. Deo, and J. S. Kowalik, *Discrete optimization algorithms : with Pascal programs*. Prentice-Hall, 1983.

[132] A. Adrabiński and M. M. Sysło, "Computational experiments with some approximation algorithms for the travelling salesman problem," *Applicationes Mathematicae*, vol. 18, no. 1. pp. 91–95, 1983.

[133] D. Gaertner and K. Clark, "On Optimal Parameters for Ant Colony Optimization algorithms."

[134] D. B. Fogel and J. W. Atmar, "Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Processes Using Linear Systems," 1990.