HYBRIDIZATION OF ENHANCED ANT COLONY SYSTEM AND TABU SEARCH ALGORITHM FOR PACKET ROUTING IN WIRELESS SENSOR NETWORK

HUSNA BINTI JAMAL ABDUL NASIR

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2020

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

HUSNA BINTI JAMAL ABDUL NASIR

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"HYBRIDIZATION OF ENHANCED ANT COLONY SYSTEM AND TABU SEARCH ALGORITHM FOR PACKET ROUTING IN WIRELESS SENSOR NETWORK"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***07 Februari 2019.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
February 07, 2019.

Pengerusi Viva: *(Chairman for VIVA)*	**Prof. Dr. Zulkhairi Md Dahalin**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Mohd Aizaini Maarof**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Dr. Yuhanis Yusof**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Ku Ruhana Ku Mahamud**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Eiji Kamioka**	Tandatangan *(Signature)*

Tarikh:
(Date) ***February 07, 2019***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Di dalam rangkaian sensor tanpa wayar (WSN), peningkatan masa penghantaran berlaku apabila agen pencarian memfokus pada nod sensor yang sama, manakala masalah optima setempat berlaku apabila agen terperangkap di dalam gelintaran buta-tuli ketika pencarian. Algoritma kawanan pintar telah digunakan untuk menyelesaikan masalah ini termasuklah sistem koloni semut (ACS) iaitu salah satu variasi pengoptimuman koloni semut. Walau bagaimanapun, ACS menghadapi masalah optima setempat dan stagnasi di dalam persekitaran sederhana dan besar disebabkan mekanisme penjelajahan yang tidak berkesan. Kajian ini mencadangkan penghibridan antara algoritma ACS Tertingkat dan Gelintaran Tabu ((EACS(TS)) untuk penghalaan paket di dalam WSN. EACS(TS) memilih nod sensor yang mempunyai nilai feromon yang tinggi dikira berdasarkan nilai feromon semasa dan tenaga yang tersisa di setiap nod sensor. Masalah optima setempat dapat dielakkan dengan menanda nod yang tidak mempunyai nod jiran yang berpotensi sebagai nod Tabu dan menyimpannya di dalam senarai Tabu. Pengemaskinian feromon setempat dilakukan untuk menggalakan penjelajahan ke nod sensor lain yang berpotensi manakala pengemaskinian feromon global dilaksanakan untuk menggalakan pengeksploitasian nod sensor yang optimum. Eksperimen telah dijalankan di dalam persekitaran simulasi WSN yang disokong oleh rangka kerja RMASE untuk menilai prestasi EACS(TS). Sejumlah 6 set data telah dijalankan untuk menilai keberkesanan algoritma yang di cadangkan. Keputusan menunjukkan EACS(TS) mengatasi dari segi kadar kejayaan, kadar kehilangan paket, latensi, dan kecekapan tenaga apabila dibandingkan dengan algoritma penghalaan kecerdasan kawanan tunggal seperti EEABR, BeeSensor, dan Termite-hill. Pencapaian yang baik juga telah dicapai untuk kadar kejayaan, kadar penghantaran, dan latensi apabila dibandingkan dengan algoritma hibrid yang lain seperti FSACO, ICSCA, dan BeeSensor-C. Hasil daripada kajian ini menyumbang kepada algoritma penghalaan yang optimum di dalam WSN. Ini boleh menghasilkan kualiti servis yang baik dan meminimumkan penggunaan tenaga.

Kata Kunci: Sistem koloni semut, Gelintaran tabu, Rangkaian sensor tanpa wayar, Penghalaan paket

Abstract

In Wireless Sensor Network (WSN), high transmission time occurs when search agent focuses on the same sensor nodes, while local optima problem happens when agent gets trapped in a blind alley during searching. Swarm intelligence algorithms have been applied in solving these problems including the Ant Colony System (ACS) which is one of the ant colony optimization variants. However, ACS suffers from local optima and stagnation problems in medium and large sized environments due to an ineffective exploration mechanism. This research proposes a hybridization of Enhanced ACS and Tabu Search (EACS(TS)) algorithm for packet routing in WSN. The EACS(TS) selects sensor nodes with high pheromone values which are calculated based on the residual energy and current pheromone value of each sensor node. Local optima is prevented by marking the node that has no potential neighbour node as a Tabu node and storing it in the Tabu list. Local pheromone update is performed to encourage exploration to other potential sensor nodes while global pheromone update is applied to encourage the exploitation of optimal sensor nodes. Experiments were performed in a simulated WSN environment supported by a Routing Modelling Application Simulation Environment (RMASE) framework to evaluate the performance of EACS(TS). A total of 6 datasets were deployed to evaluate the effectiveness of the proposed algorithm. Results showed that EACS(TS) outperformed in terms of success rate, packet loss, latency, and energy efficiency when compared with single swarm intelligence routing algorithms which are Energy-Efficient Ant-Based Routing (EEABR), BeeSensor and Termite-hill. Better performances were also achieved for success rate, throughput, and latency when compared to other hybrid routing algorithms such as Fish Swarm Ant Colony Optimization (FSACO), Cuckoo Search-based Clustering Algorithm (ICSCA), and BeeSensor-C. The outcome of this research contributes an optimized routing algorithm for WSN. This will lead to a better quality of service and minimum energy utilization.

Keywords: Ant colony system, Tabu search, Wireless sensor network, Packet routing

Acknowledgement

Alhamdulillah, it is impossible for me to accomplish this thesis without the bless from Allah and contribution from so many gracious people who have given me all sorts of assistance. To these people, I am indeed grateful.

Firstly, I would like to record my sincere gratitude to my first supervisor, Prof. Dr. Ku Ruhana Ku Mahamud, for her supervision, advice, and guidance since the beginning of my research. Working under her supervision will become of my unforgettable experiences in my life. I would like to also thank my second supervisor, Prof. Dr. Eiji Kamioka, for his generosity and support to enrich my research work through useful advice and research exchange program opportunity. Their continuous encouragement and support in various ways was able to release me from my hopeless feeling that occasionally happened during my stressful day.

Secondly, I would like to express my sincere appreciation to my best friend, my best work mate, my best advisor, my best supporter, my love, Saufi Bin Bukhari, for his never ending support to realize my dream despite all the hurdles and challenges to juggle between work, study and family. Not forgetting to my kids, Danish Adni, Daniel Adib and Daris Adel, thanks for all the sacrifices and patient. Also, thanks to my lovely parents Jamal Abdul Nasir and Siti Mariam and also to my siblings, family in law, and relatives. What I know is that I am indebted to them for being patient, inspiring and supportive.

Finally, I would like to thank all the people who have been very kind in offering their help and assistance in one way or another in the course of completing my research. To those who I couldn't name them specifically here because there are simply so many of them, I offer my apology. But I am deeply indebted to them for their kindness and generosity.

May Allah bless them all.

Table of Contents

Permission to Use ... i

Abstrak ... ii

Abstract ... iii

Acknowledgement ... iv

Table of Contents ... v

List of Tables ... vii

List of Figures ... viii

List of Appendices ... x

List of Abbreviations ... xi

CHAPTER 1: INTRODUCTION ... 1

1.1 Problem Statement ... 11

1.2 Research Objectives ... 15

1.3 Significance of the Research ... 16

1.4 Scope and Limitation of the Research ... 17

1.5 Structure of the Thesis ... 17

CHAPTER 2: LITERATURE REVIEW ... 19

2.1 Overview of Wireless Sensor Network ... 19

2.1.1 Design Challenge and Routing Challenge in Wireless Sensor Network ... 20

2.1.2 Issues and Limitations of the Routing Algorithms in Wireless Sensor Network ... 22

2.2 Overview of Metaheuristic Algorithms ... 22

2.2.1 Swarm Intelligence Algorithm ... 24

2.2.2 Local Search Algorithm ... 29

2.3 Ant Colony Optimization ... 30

2.4 Tabu Search ... 34

2.5 Routing Algorithms in Wireless Sensor Networks ... 36

2.5.1 Performance Evaluation Criteria of Routing Algorithms in Wireless Sensor Network ... 37

2.5.2 Single Swarm Intelligence Approaches in WSN Packet Routing ... 39

2.5.3 Hybrid Swarm Intelligence Approaches in WSN Packet Routing ... 49

2.6 Summary ... 61

CHAPTER 3: RESEARCH FRAMEWORK AND METHODOLOGY 63
3.1 Research Framework 64
3.1.1 Node Selection Phase 65
3.1.2 Local Search Phase 66
3.1.3 Exploration Control Phase 67
3.1.4 Exploitation Control Phase 68
3.1.5 Validation Phase 69
3.2 Simulation Design and Implementation 69
3.3 Comparative Measures 72
3.4 Summary 75
CHAPTER 4: ENHANCED ANT COLONY SYSTEM AND TABU SEARCH ALGORITHM 77
4.1 Ant Colony System Implementation 77
4.2 Tabu Search Implementation 82
4.3 Enhanced Ant Colony System and Tabu Search Algorithm 86
4.4 EACS(TS) Evaluation Design and Implementation 91
4.5 Summary 97
CHAPTER 5: EXPERIMENTAL RESULTS 99
5.1 Performance Evaluation Criteria 99
5.2 Experimental Design 102
5.3 EACS(TS) Parameter Tuning 104
5.3.1 Effect of β value 104
5.3.2 Effect of q_o value 105
5.3.3 Effect of ρ value 106
5.3.4 Effect of α value 107
5.4 Experimental Results of EACS(TS) 108
5.4.1 Experimental Results on Single Swarm Intelligence Algorithms 109
5.4.2 Experimental Results of Hybrid Algorithms 120
5.5 Summary 134
CHAPTER 6: CONCLUSION AND FUTURE WORK 138
6.1 Contribution of the research 140
6.2 Future Work 142
REFERENCES 144
APPENDICES 175

List of Tables

Table 2.1: Summary of single swarm intelligence routing algorithms in WSN........49
Table 2.2: Summary of hybrid swarm intelligence routing algorithms in WSN.......60
Table 5.1: Simulation parameters 102
Table 5.2: Source node characteristics..103
Table 5.3: Destination node characteristics 103
Table 5.4: TS parameter values 103
Table 5.5: Effect of ß value to the EACS(TS) algorithm 105
Table 5.6: Effect of q_o value to the EACS(TS) algorithm 106
Table 5.7: Effect of ξ value to the EACS(TS) algorithm..107
Table 5.8: Effect of ρ value to the EACS(TS) algorithm 108
Table 5.9: Scheduling parameters to investigate the performance of the algorithms for different numbers of sensor nodes 109
Table 5.10: Scheduling parameters to investigate the performance of the algorithms for different simulation time 116
Table 5.11: Scheduling parameters to investigate the performance of ACO variant algorithms for different simulation time 121
Table 5.12: Statistical test for the throughput values of the third experiment 123
Table 5.13: Scheduling parameters to investigate the performance of hybrid algorithms for different numbers of sensor nodes 126
Table 5.14: Scheduling parameters to investigate the performance of hybrid algorithms for large numbers of sensor nodes 130
Table 5.15: Parameters to investigate the performance of the algorithms in submitting large size of packets 133

List of Figures

Figure 1.1: Examples of NP-complete problem 6
Figure 1.2: Ant behaviour in foraging process (Perretto & Lopes, 2005) 7
Figure 2.1: Metaheuristics algorithms 24
Figure 2.2: Process of TS algorithm (Zapfel et al., 2010) 36
Figure 3.1: Research framework 65
Figure 3.2: Events between application layer and MAC layer 71
Figure 4.1: Ants in a WSN system...79
Figure 4.2: Situation where forward ant is trapped in the blind alley 84
Figure 4.3: Situation where node F is excluded by forward ant from moving destination candidate 84
Figure 4.4: Situation where forward ant returns to the previous node 85
Figure 4.5: Flowchart of EACS(TS) algorithm 89
Figure 4.6: Pseudocode of EACS(TS) algorithm 90
Figure 4.7: High-level workflow of EACS(TS) algorithm in constructing routing path..93
Figure 4.8: High-level workflow of EACS(TS) algorithm in performing backward movement 94
Figure 4.9: High-level workflow of EACS(TS) algorithm in submitting data packet...95
Figure 5.1: Success rate of different algorithms by using different numbers of sensor nodes in 300 seconds 111
Figure 5.2: Packet loss rate of different algorithms by using different numbers of sensor nodes in 300 seconds 112
Figure 5.3: Latency of different algorithms by using different numbers of sensor nodes in 300 seconds 113
Figure 5.4: Energy efficiency of different algorithms by using different numbers of sensor nodes in 300 seconds 115
Figure 5.5: Success rate of different algorithms by using nine sensor nodes in 100 seconds 117
Figure 5.6: Energy efficiency of different algorithms by using nine sensor nodes in 100 seconds 118

Figure 5.7: Throughput of different algorithms by using nine sensor nodes in 100 seconds....119
Figure 5.8: Latency of different algorithms by using nine sensor nodes in 100 seconds....120
Figure 5.9: Throughput of different ACO variant algorithms by using 100 sensor nodes in 300 seconds....122
Figure 5.10: Energy consumption of different ACO variant algorithms by using 100 sensor nodes in 300 seconds....124
Figure 5.11: Energy efficiency of different ACO variant algorithms by using 100 sensor nodes in 300 seconds....125
Figure 5.12: Success rate of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy sensor nodes in 5000 seconds....127
Figure 5.13: Number of packet received of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy sensor nodes in 5000 seconds....128
Figure 5.14: Energy consumption of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy of sensor nodes in 5000 seconds....129
Figure 5.15: Latency of EACS(TS), IEEABR, BeeSensor-C and BeeSensor by using large numbers of sensor nodes in 200 seconds....131
Figure 5.16: Energy efficiency of EACS(TS), IEEABR, BeeSensor-C and BeeSensor by using large numbers of sensor nodes in 200 seconds....132
Figure 5.17: Energy consumption of EACS(TS), ICSCA, PSO-ECHs, and LEACH by using different numbers of sensor nodes in submitting large size of packets....134

List of Appendices

Appendix I: Effect of β value to the success rate of EACS(TS) algorithm............175

Appendix II: Effect of β value to the throughput of EACS(TS) algorithm...……..176

Appendix III: Effect of β value to the latency of EACS(TS) algorithm…………..176

Appendix IV: Effect of β value to the energy consumption of EACS(TS) algorithm……………………………………………………………………..177

Appendix V: Effect of β value to the energy efficiency of EACS(TS) algorithm...177

Appendix VI: Effect of q_o value to the success rate of EACS(TS) algorithm…….178

Appendix VII: Effect of q_o value to the throughput of EACS(TS) algorithm…….178

Appendix VIII: Effect of q_o value to the latency of EACS(TS) algorithm………..179

Appendix IX: Effect of q_o value to the energy consumption of EACS(TS) algorithm..…………………………………………………………………..…..179

Appendix X: Effect of q_o value to the energy efficiency of EACS(TS) algorithm.180

Appendix XI: Effect of ρ value to the success rate of EACS(TS) algorithm……..180

Appendix XII: Effect of ρ value to the throughput of EACS(TS) algorithm ……..181

Appendix XIII: Effect of ρ value to the latency of EACS(TS) algorithm………...181

Appendix XIV: Effect of ρ value to the energy consumption of EACS(TS) algorithm……………………………………………………………………..182

Appendix XV: Effect of ρ value to the energy efficiency of EACS(TS) algorithm……………………………………………………………………..182

Appendix XVI: Effect of α value to the success rate of EACS(TS) algorithm …..183

Appendix XVII: Effect of α value to the throughput of EACS(TS) algorithm…...183

Appendix XVIII: Effect of α value to the latency of EACS(TS) algorithm…..…..184

Appendix XIX: Effect of α value to the energy consumption of EACS(TS) algorithm……………………………………………………………………..184

Appendix XX: Effect of α value to the energy efficiency of EACS(TS) algorithm……………………………………………………………………..185

List of Abbreviations

ABC	Artificial Bee Colony
ACA	Ant Colony Algorithm
ACLR	Ant Colony based Location-aware Routing algorithm
ACO	Ant Colony Optimization
ACS	Ant Colony System
AFSA	Artificial Fish Swarm Algorithm
AS	Ant System
BABR	Basic Ant-Based Routing Algorithm
CH	Cluster Head
CS	Cuckoo Search
DGA	Distributed Genetic Algorithm
EACS(TS)	Enhanced Ant Colony System and Tabu Search
EAS	Elitist Ant System
EEABR	Energy-Efficient Ant-Based Routing
EPWSN	Energy and Path aware ACO algorithm for routing of Wireless Sensor Networks
FSACO	Fish Swarm Ant Colony Optimization
GA	Genetic Algorithm
HCS	Hierarchical Cuckoo Search
IABR	Improved Ant-Based Routing
ICSCA	Improved Cuckoo Search-based Clustering Algorithm
IEEABR	Improved Energy Efficient Ant Based Routing
LBTAS	Local Best Tour Ant System
MMAS	Max-Min Ant System
NP	Nondeterministic Polynomial
Prowler	Probabilistic Wireless Network Simulator
PSO	Particle Swarm Optimization
QAP	Quadratic Assignment Problems
QoS	Quality of Service
RAS	Rank-based Ant System
RMASE	Routing Modelling Application Simulation Environment
SA	Simulated Annealing

SNGF	Stateless Non-deterministic Geographic Forwarding
SRA	Smart Routing Algorithm
Termite-hill	Termite Hill
TS	Tabu Search
TSP	Travelling Salesman Problem
TSRA	Tabu Search-based Routing Algorithm
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

The Wireless Sensor Network (WSN) has become an important information revolution and research area in computer networking. A computer network allows users to communicate and share data with high connection speed and bandwidth. Several types of hardware and software in wired networks are combined by a series of cables to establish a computer network. In wired networks, applications are inflexible in variation because they are limited to a fixed area that leads to high installation cost and limited connections. Due to these disadvantages, a wireless network has been proposed as a solution. A wireless network is developed through combinations of various resources that connect and communicate through the internet from different geographic locations. Wireless networks can also support internet and mobility services and, at the same time, can reduce installation cost compared to wired networks. In order to further improve wireless networks, a WSN is introduced to reduce the deployment and maintenance costs while, at the same time, improving the security and network lifetime. The WSN is based on a large-scale networking area that consists of sensor nodes with limited power, to gather useful information from the surrounding network (Okdem & Karaboga, 2009; Mittal & Kumar, 2015; Parenreng & Kitagawa, 2017). However, the WSN has different constraints and requirements compared to traditional wireless networks. This includes using broadcast communication method that is prone to packet loss, constant changes of topology caused by dead nodes, and reliance on non-rechargeable or irreplaceable battery that has limited lifetime.

Wireless sensor network technology was first applied in the military and heavy industrial area. In the 1950s, the first WSN application, a sound surveillance system, was introduced by the United States Military, based on acoustic sensors on the ocean bottom to detect and track Soviet submarines (Desai, Jain, & Merchant, 2007). This application is still being used nowadays by National Oceanographic and Atmospheric Administration to monitor conditions in the ocean. At the same time, the network of air defence radar was also developed by the United States in an effort to defend its territory. In 1980, a distributed sensor network program was initiated by the Defence Advanced Research Projects Agency to study the implementation of distributed/wireless sensor networks (Wang & Balasingham, 2010). With the collaboration of Carnegie Mellon University and Massachusetts Institute of Technology Lincoln Labs, a large amount of research and applications on WSN were produced by academia and scientific researchers (Chong & Kumar, 2003). Some commercial applications developed using the architecture of the WSN include natural disaster prevention, forest fire detection, air quality monitoring and weather stations (Ali, Ming, Chakraborty, & Iram, 2017). WSNs were also implemented to support heavy industrial applications such as waste-water treatment (Derbew & Libsie, 2014; Zakaria & Michael, 2017), power distribution (Suryadevara, Mukhopadhyay, Kelly, & Gill, 2015; Katyara, Izykowsk, Chowdhry, Musavi, & Hussain, 2018) and specialized factory automation (Shin, Chin, Yoon, & Kwon, 2011; Frotzscher, Wetzker, Bauer, Rentschler, Beyer, Elspass, & Klessig, 2014; Aijaz, 2018).

Management of sensor nodes in a WSN is the main component that needs to be considered in controlling the network's lifetime. There are various issues in WSN such as packet routing, energy efficiency, node localization, time synchronization, load balancing, and security (Hu & Cao, 2010; Derr & Manic, 2015; Yildiz, Bicakci, Tavli, Gultekin, & Incebacak, 2016; Lu et al., 2018). Other issues in WSNs that can be considered as main issues and are often discussed by many researchers are routing packets to destination nodes (Luo & Li, 2012; Zeng & Dong, 2016; Wang, Zhang, Gao, Wang, & Li, 2017) and energy efficiency of all available sensor nodes (Okafor & Fagbohunmi, 2013; Wang, Chen, Wu, & Shu, 2016; Biswas, Das, & Chatterjee, 2018).

Robustness and scalability are two main aspects that have been considered in implementing any routing algorithm in the WSN system (Aliouat & Aliouat, 2013; Loganathan, Sabapathy, Ghazali, Ahmad, & Osman, 2017). Due to their unlimited transmission range, sensor nodes act as an intermediate medium in WSNs to forward packets from source to destination (Frey, Rührup, & Stojmenović, 2009; Arafath, Khan, & Sunitha, 2018). Sensor nodes will communicate with each other through radio signal broadcast to send or receive information. Routing packets in WSNs aim to maximize throughput, minimize latency, avoid overload and minimize energy consumption of each sensor node in order to increase the network's lifetime. At the same time, a good routing algorithm influences the balancing of forwarding packets on each sensor node.

Sensor nodes are geographically distributed in large scale areas in the WSN. The main functions of sensor nodes are to sense any changes in the WSN and communicate between available sensor nodes to forward packets from the source node to destination node (Sutar & Bodhe, 2010; Kumar, Jain, & Barwal, 2014;

Mittal, Gupta, & Choudhury, 2018). However, available sensor nodes in the WSN have very limited capabilities in terms of energy, memory, computational power and communication capacity (Khan, Gansterer, & Haring, 2013; Tubiello, Poehls, Webber, Marcon, & Vargas, 2018). The routing algorithm in the WSN should consider this limitation in order to select optimal sensor nodes to forward packets to destination nodes to ensure that all packets arrive in the minimum time. In addition, packet loss problems may occur due to depletion of energy of sensor nodes and it could affect the energy consumption of each sensor node which will eventually reduce network lifetime.

Load balancing is also critical in the WSN system because an effective load balancing algorithm can reduce the energy consumption of each sensor node and, at the same time, extend the network lifetime of the WSN system (Wajgi & Thakur, 2012; Javaid et al., 2015; Qiu, Shen, & Chen, 2017). In order to solve this problem, all forwarding packets need to be equally spread among sensor nodes in the WSN system. A good load balancing algorithm must be capable of balancing entire sensor nodes through fair distribution of entire packets across available sensor nodes by considering packet characteristics and sensor node capacity in order to obtain optimal node utilization.

Routing and load balancing are categorized as a Nondeterministic Polynomial (NP)-complete problem (Liu, Xu, & Sun, 2012; Karthikeyan & Subramani, 2014). The NP-Complete problem is a problem that cannot be solved by an exact algorithm in a polynomial time (Blum & Roli, 2003). Figure 1.1 shows the example of NP-complete problems which are grouped by the type of problem such as routing, scheduling, assignment, subset problems and others. One of the most effective ways to solve these problems is to use metaheuristics algorithms such as Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS), and Ant Colony Optimization (ACO) that combined higher level strategies and local improvement procedure in performing a robust search of a solution space and at the same time escaping from the local optima (Glover & Kochenberger, 2006).

Simulated annealing (Azami, Ranjbar, Rostami, & Amiri, 2013; Xenakis, Foukalas, & Stamoulis, 2016), GA (Chakraborty, Mitra, & Naskar, 2011; Elhoseny, Yuan, Yu, Mao, El-Minir, & Riad, 2015), TS (El Rhazi & Pierre, 2009; Varsha, Singh, & Bala, 2017) and ACO (Fathima & Sindhanaiselvan, 2013; Bouarafa, Saadane, & Rahmani, 2018) are several of the metaheuristic algorithms that move from one solution to another in the process to construct the best solution to solve routing and energy efficiency problems in the WSN system. By using these methods, a feasible solution can be produced even though it will not be close to the optimal solution.

Ant colony optimization is one of the applications of swarm intelligence that is inspired by the foraging behaviour of ants that work together to find the shortest path between nest and food source (Blum, 2005; Singh, Singh, & Kumar, 2010). Swarm intelligence is a sub-category of artificial intelligence that is motivated by the intelligent behaviour of groups such as natural systems of social insects like bees, ants, wasps, and termite (Jangra, Awasthi, & Bhatia, 2013). Other examples of swarm intelligence include artificial bee colony algorithms that study the foraging behaviour of honey bees and particle swarm intelligence that studies the behaviour of bird flocking and fish schooling (Zhao et al, 2010).

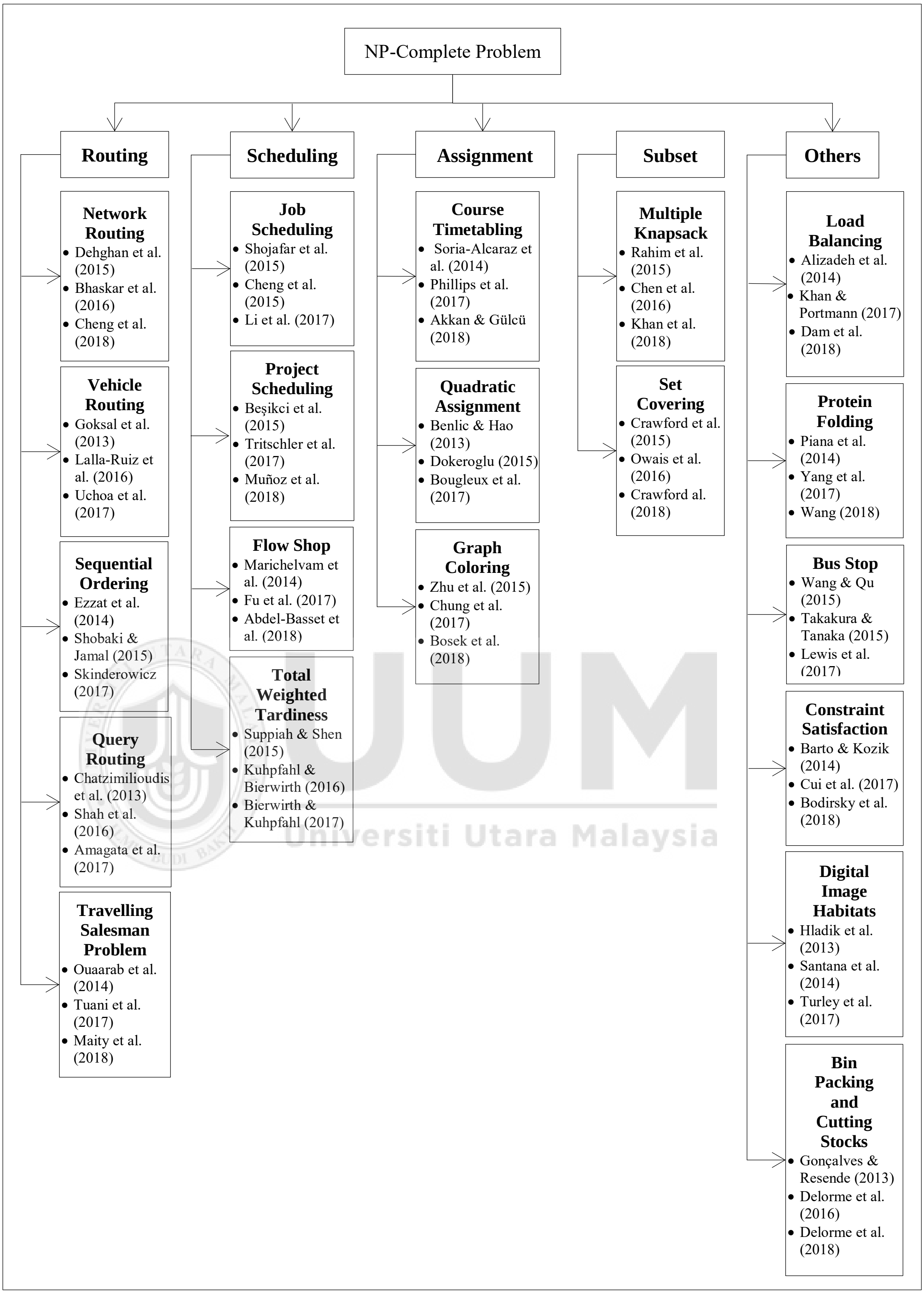


Figure 1.1. Examples of NP-complete problem

Experiments to investigate the behaviour of real ants like foraging and nest construction have been undertaken by many researchers. A double bridge experiment was conducted by Goss, Aron, Deneuborg and Pasteels (1989) to investigate the foraging behaviour of ants. Figure 1.2 (a) shows that ants move in a continuous path from nest to food source. However, ants will randomly choose whether to turn left or right when an obstacle appears in the way because they have no idea which is the best path or the shortest path to move to the destination, as shown in Figure 1.2 (b). At this point, there is no pheromone on either path, so half the ants will choose the short path and the other half will choose the long path. Then, ants will deposit a certain amount of pheromone while moving from nest to food source on both paths. By assuming that all ants move at the same speed, ants that choose the short path will reach the food source and return to the nest faster. This will increase the amount of pheromone in the short path and influence more ants to travel on the short path rather than a long path, as shown in Figure 1.2 (c). Figure 1.2 (d) shows that all ants will choose the short path after a transitory phase due to the large amount of pheromone accumulated on that path.

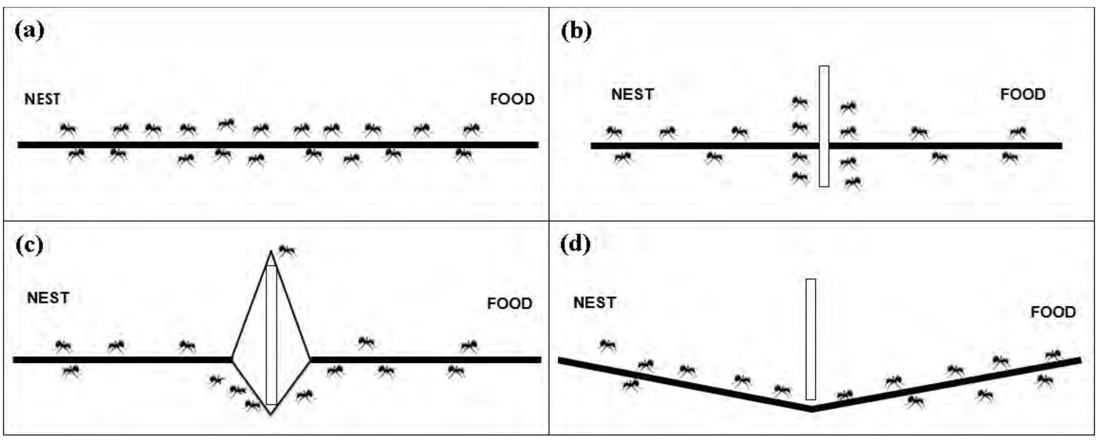


Figure 1.2. Ant behaviour in foraging process (Perretto & Lopes, 2005)

There are many variations of the ACO algorithm such as Ant System (AS), Ant Colony System (ACS), Max-Min Ant System (MMAS), Rank-based Ant System (RAS) and Elitist Ant System (EAS) (Dorigo & Stützle, 2004). Ant colony optimization has been successfully applied to solve many routing problems such as the network routing problem (Ye & Mohamadian, 2014; Yang, Ping, Aijaz, & Aghvami, 2018), travelling salesman problem (Holzinger et al.; 2016; Gülcü, Mahi, Baykan, & Kodaz, 2018), vehicle routing problem (Tan, Lee, Majid, & Seow, 2012; Kuo & Zulvia, 2017), sequential ordering routing problem (Gambardella, Montemanni, & Weyland, 2012; Ezzat, 2013; Skinderowicz, 2017) and query routing problem (Gómez Santillán et al., 2010; Hanane & Fouzia, 2014). Additionally, ACO has been used to solve routing problems between forwarding packets and available sensor nodes in WSN systems (Anjali & Kaur, 2013; Sundaran, Ganapathy, & Sudhakara, 2017; Zou & Qian, 2018). On the other hand, load balancing problems in the WSN have also been successfully solved by using the ACO algorithm by equally distributing all forwarding packets to available sensor nodes (Yang, Xu, Zhao, & Xu, 2010; Laouid et al., 2017).

Ant colony system is considered as one of the best ACO variants for solving NP-complete problems (Schyns, 2015; Bukhari, Ku-Mahamud, & Morino, 2017; Liu et al., 2018). In ACS, exploitation and exploration mechanisms are introduced during the path construction process to balance the probability between random selection and fixed selection based on certain parameters in state transition rule. Ant colony system applies local pheromone updates to evaporate the pheromone intensity at each traversed path and global pheromone update to increase the attractiveness of the best solution so far to be used for the next iteration. Therefore, a systematic and deterministic exploration and pheromone update mechanism can enhance the performance of the ACS algorithm.

The ACO algorithm is applied in WSNs because it is easily adapted to solve both static (Singh & Behal, 2013; Mavrovouniotis, 2013) and dynamic (Lissovoi & Witt, 2015; Mavrovouniotis & Yang, 2018) combinatorial optimization problems. However, the performance of ACO algorithms to solve routing and load balancing problems in WSNs can be further extended in order to gain maximum throughput, minimum latency, minimum energy consumption of sensor nodes, minimum stagnation problems, to balance entire sensor nodes and, at the same time, to extend the network lifetime of the WSN network. Stagnation in the WSN may also occur when all packets are assigned to the same sensor nodes which lead to the nodes having high workload. The stagnation problem in a WSN network can potentially be solved when all sensor nodes are well utilized. The local optima problem will also occur when the path searching process by the agent is trapped in a blind alley where further movement will result in a loop (Czubak, 2013). To solve this problem, it is crucial to detect the potential occurrence of local optima and mitigate it using effective methods such as backward movement and known bad path marking.

Hybridizing an ACS with local search algorithms such as TS, GA, and SA will improve the solutions produced during the path construction phase (Gambardella, 2015). Tabu search is a good candidate to be hybridized with ACS as both algorithms complement each other. Ant colony system works based on a constructive approach (Angelo, Bernardino, & Barbosa, 2015) while TS works based on local search (Paquete & Stützle, 2018). Tabu search is also based on a systematic search that can prevent the algorithm from random solutions. Research by Yoshikawa and Otani (2010) proposed a hybrid routing algorithm for Travelling Salesman Problem (TSP) that combined ACS and TS in preventing the ant agent from getting trapped in the blind alley during routing process. On the other hand, Alobaedy (2015) proposed hybrid ACS and TS in improving the scheduling process in grid computing in terms of makespan. Hybrid ACS and TS is also suitable to be applied in solving packet routing in WSN which is one of the NP-complete problems.

Glover (1986) proposes the TS algorithm which is one of the metaheuristics algorithms based on guided local search to solve mathematical optimization. Like ACO, TS has also successfully solved many optimization problems such as the Travelling Salesman Problem, job scheduling, network routing and vehicle routing problem (Gendreau & Potvin, 2014). Tabu search that is based on local search can avoid local minimum by using many mechanisms such as diversification and memory (Rothlauf, 2011). The TS algorithm is flexible when implemented in the WSN because it uses the concept of responsive exploration and adaptive memory (Orojloo & Haghighat, 2016). There are four types of memory: frequency (long-term); recency (short-term); influence; and, quality (Glover & Laguna, 1997). Many research works that are based on hybrid TS use only one or two of these memories (Dhivya & Sundarambal, 2012; Sahni, Bala, & Sharma, 2016). However, the performance of hybrid TS algorithms to solve the routing problem in WSNs can be further extended to gain minimum latency and energy consumption.

This research aims to solve the routing problems in WSNs including packet loss, energy efficiency, latency and local optima. The proposed routing algorithm using

ACS and TS and focuses on balancing the load to all sensor nodes by encouraging exploitation and exploration during the sensor node selection mechanism while preventing becoming trapped in a blind alley. Even though the hybridization concept of ACS and TS has never been applied in WSN, but it has been successfully applied in solving problems in other research domains (Yoshikawa & Otani, 2010; Alobaedy, 2015). However, the proposed algorithm differs from these hybrid algorithms where it considers the energy efficiency of each sensor nodes and the whole system during searching process in preventing the dead node problem that will affect the network lifetime.

1.1 Problem Statement

In WSN, most algorithms including several ACO variants have been designed to efficiently transmit packets to available sensor nodes. However, these available algorithms are far from being ideal. Latency and packet loss still occurs when the number of packets is increased, and available sensor nodes are insufficient to cater for all packets (Yan, Gao, & Yang, 2011; Tall & Chalhoub, 2017) which will eventually lead to stagnation in the WSN environment. Tabu Search-based Routing Algorithm (TSRA) was proposed by Orojloo and Haghighat (2016) in balancing packet transmission taking into considerations of remaining energy of each sensor nodes. Sensor nodes with low energy level will be marked as “taboo” and not been used to forward packets. Sun, Dong, and Chen (2017) tried to solve the energy consumption problem in WSNs by proposing a new ant-based routing algorithm that considers communication transmission distance and heuristics function during the path construction process. However, neither algorithm considered the throughput and latency which may increase the transmission time of packets.

Sensor node selection strategy is an important part of the WSN as it can ensure the selected sensor nodes have high possibility to route packets successfully. Exploration and exploitation of available sensor nodes must be considered in order to control the energy level of sensor nodes. The lack of an effective method to control the energy level may lead to dead nodes where certain sensor nodes are under heavy traffic load which drastically depletes their energy. Energy-Efficient Ant-Based Routing (EEABR) and Improved Energy-Efficient Ant-Based Routing (IEEABR) proposed by Camilo, Carreto, Silva, and Boavida (2006) and Zungeru, Ang, and Seng (2012a) respectively are among ACO variants that focused on energy efficiency of sensor nodes in WSN. However, both algorithms only considered the exploitation of optimal path without taking into consideration exploration to the other potential path. This will lead to the hotspot problem where the energy level at certain sensor nodes will drain drastically. The modifications on state transition rule to select the next nodes may give the opportunity to the searching ants to explore and exploit the potential sensor nodes. Elhabyan and Yagoub (2014) propose a hybrid routing algorithm that combines the PSO and clustering technique called PSO-C that focuses on packet delivery ratio and energy consumption of sensor nodes. The cluster head is responsible for collecting and delivering data from all cluster members to the destination node on each simulation round. However, the hotspot problem may occur on cluster head since it uses a lot of energy during the packet submission process that will affect the energy efficiency of the cluster and the whole system.

Many routing algorithms in the WSN do not always consider the load balancing problem that will affect the energy efficiency of sensor nodes. A good routing algorithm should distribute load to all available sensor nodes efficiently in order to reduce the energy consumption and hotspot problem. Singh and Behal (2013) combine a mobile sink technique with an ACO algorithm to solve routing problems in WSNs. However, the pheromone update technique that reduces the pheromone value of sensor nodes was not applied and the lack of this technique may lead to unbalanced selection of sensor nodes. Termite-hill algorithm (Zungeru, Ang, & Seng, 2012b) limits the pheromone update and pheromone evaporation rate within certain range to control the pheromone value on sensor nodes. This approach only encourages the ant in the next iteration to reselect the optimal sensor nodes without exploring other potential sensor nodes which eventually leads to reduction in nodes utilization during routing process. A Bee Sensor-C, based on an Artificial Bee Colony (ABC), and cluster technique was proposed by Cai, Duan, He, Yang, and Li (2015) to encourage the multipath construction method and improve energy efficiency in the WSN routing environment. Nodes are selected based on the remaining energy and number of hops during the searching process. However, there is no control element in exploitation of the sensor node that will lead to unbalanced distribution of load among sensor nodes.

Searching agent in WSN routing algorithms often get trapped in the local optima where there is no potential routing path to move during routing process (Li et al., 2010). Wang, Zhan, and Zhang (2018) proposed Distributed Genetic Algorithm (DGA) to tackle the local optima problem in WSN by maximizing the number of disjoin set. However, DGA only considers the number of disjoin set and computational time without considering throughput and balancing factor. The stagnation nature of pheromone also occurs in ACS when applied in large size networks due to the high exploitation process (Mathiyalagan, Suriya, & Sivanandam, 2010). This will lead to local optima problems where ants get trapped in blind alleys during the node searching process; ants cannot reach the destination node and all available sensor nodes are previously visited nodes. This problem can lead to the ant becoming stuck where the exploration process cannot progress further within the network. Therefore, the ACS algorithm needs to be improved in terms of its exploration mechanism and ability to correct the construction phase after each cycle. In order to solve the local optima problem in ACS, Yoshikawa and Otani (2010) proposed a hybrid algorithm that combined the ACS and Tabu search algorithm. Even though this research was applied in TSP, the concept of hybridizing these two metaheuristics algorithms can also be used in WSN with some modification. WSN differ from TSP where energy efficiency of each sensor nodes and the whole system must be considered in order to prevent the dead node that will affect the network lifetime of the whole system.

This leads to several research questions that must be answered as follows:

1. How does an extended state transition rule increase the accuracy of the selected optimal path?
2. How to integrate TS technique in solving local optima problem in ACS?
3. Can the improved local pheromone update lead to fair distribution of packets to available sensor nodes?
4. Can the extended global pheromone update technique reduce the latency during the routing process?
5. How efficient is the proposed algorithm in solving the routing problem in the wireless sensor network?

1.2 Research Objectives

The main objective of this research is to develop an enhanced ACS and TS based routing algorithm in the WSN that can route packets to suitable sensor nodes, minimize the forwarding time of packets to the destination node, minimize energy consumption of sensor nodes, balance the workload of entire sensor nodes, prevent the local optima problem during the routing process, and improve network lifetime of the WSN.

Specific objectives of the research are:

i. To formulate a state transition rule that considers the energy efficiency and energy consumption aspects in sensor node selection strategy to find optimal sensor nodes that can prolong the network lifetime.

ii. To design an enhanced ACS and TS algorithm in preventing local optima problems in the WSN while at the same time increasing the throughput value.

iii. To develop an extended local pheromone update technique that can balance the load on each sensor node and encourage exploration in the searching process to prevent the hotspot problem.

iv. To develop an extended global pheromone update to encourage the exploitation of the selected optimal path in reducing latency during the routing process.

v. To develop a simulation model that can be used to evaluate the performance of the proposed algorithm.

1.3 Significance of the Research

The WSN is an efficient computer paradigm that can be used as a solution to several challenging applications in science, engineering and economics such as traffic monitoring, habitat monitoring, healthcare and military surveillance (Camilo et al., 2006; Yan et. al., 2011). Wireless sensor network optimization involves efficient management of available sensor nodes to forward available packets to the destination by considering throughput, latency and energy consumption of sensor nodes (Ennaji & Boulmalf, 2009; Yan et al.; 2011). Therefore, managing sensor nodes is crucial in the WSN environment.

The outcome of this research will contribute to a new routing algorithm that combines ACS and TS techniques that could improve the performance of available ACO algorithms in WSN environments. The new proposed ACS algorithm could also enhance the classical approach of the ACO algorithm by dynamically routing packets to available sensor nodes while preventing the local optima problem in order to minimize the forwarding time of each packet and energy consumption of each node. At the same time, it tries to balance packets allocation in the entire sensor nodes by encouraging exploration and exploitation during the searching process. Thus, this research output is a new member of the ACO family that offers a new alternative to enhance the performance of available ACO algorithms in WSNs concerning the routing aspect. The proposed hybrid algorithm also has great potential in solving the routing problem in other research domains such as TSP, sequential ordering problem, and vehicle routing problem.

1.4 Scope and Limitation of the Research

This research focuses on developing a routing algorithm to solve the packet loss problem, energy efficiency, and load balancing problems in WSNs. The proposed algorithm combines the technique from ACS and TS algorithms where the main focus is on improving the way ants search the best nodes in terms of minimizing the forwarding time of each packet from source node to destination node, minimizing the energy consumption of each sensor node, preventing the local optima problem and, at the same time, trying to balance all loads on available sensor nodes. The ACS and TS algorithm is selected where a new technique is proposed for sensor nodes selection strategies and pheromone update techniques. However, this research does not cater for sensor node localization and fault tolerance in the WSN. Throughout all the improvements, the proposed algorithm can potentially reduce the local optima problem and maximize the network lifetime in WSN environments.

1.5 Structure of the Thesis

The structure of this thesis is as follows. In Chapter 2, an overview of the WSN, ACO and TS algorithm concepts are introduced. Then, ant-based approaches for managing resources in WSNs and current hybrid approaches to optimize routing packets in WSNs is discussed.

Chapter 3 covers the Enhanced Ant Colony System and Tabu Search (EACS(TS)) framework for routing packets in WSNs. Research methods that have been used to fulfil the research objectives are elaborated.

Chapter 4 presents the proposed enhanced ant-based routing algorithm which focuses on energy efficiency and submission time factors. The routing packets scenario and details of the proposed algorithm are discussed in this chapter. The design and implementation of the EACS(TS) are also described.

Experimental results and analysis of applying EACS(TS) in WSNs are presented in Chapter 5. The performance of the proposed algorithm is compared to existing algorithms, i.e., EEABR, Termite Hill, BeeSensor, and AODV algorithms in terms of success rate, energy consumption, energy efficiency, latency, throughput and lifetime of the system.

Chapter 6 discusses the contribution of the research and highlights the future research directions.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the review of previous studies that have been conducted in the area of WSN, metaheuristic algorithms including swarm intelligence and local search approaches, performance evaluation criteria applied by previous studies, and existing single and hybrid routing algorithms in WSN. Challenges, issues, and limitations of previous routing algorithms are also discussed with the aim to determine the gap that has been addressed by this work.

The overview of wireless sensor network is discussed in Section 2.1, design and routing challenges in Section 2.1.1, followed by issues and limitations of the routing algorithms in Section 2.1.2. Metaheuristics algorithms that consist of swarm intelligence and local search are discussed in Section 2.2. Previous works on ACO algorithms are presented in Section 2.3 followed by discussion of TS algorithm in Section 2.4. Section 2.5 discussed the routing algorithms in WSN which covers the performance evaluation criteria in Section 2.5.1, single swarm intelligence approach in Section 2.5.1 and hybrid swarm intelligence approach in Section 2.5.2. Lastly, the summary of the chapter is presented in Section 2.6.

2.1 Overview of Wireless Sensor Network

Wireless sensor network is a large-scale distributed network that consists of many small sensor nodes that are interconnected to perform various network operations. Sensor nodes are typically small, portable, and lightweight (Engmann, Katsriku, Abdulai, Adu-Manu, & Banaseka, 2018) with the capabilities to sense events, real time monitoring, perform light computation and calculation, transmit data, temporary data storage, and communicate between each other during data transmission (Ketshabetswe, Zungeru, Mangwala, Chuma, & Sigweni, 2019). Sensor nodes act as an intermediate medium to sense and transmit data from source node to destination node by using multi hop technique. Sensor nodes can gather and forward raw data to destination node or use their processing abilities to carry out simple computation operation and submit only the partially and required processed data. However, sensor nodes have limited capabilities in terms of memory, storage, computation power, and communication capabilities (Mohindru & Singh, 2018).

There are many types of WSN such as terrestrial WSN, underwater WSN, underground WSN, mobile WSN and multimedia WSN that have successfully been applied in many critical applications such as healthcare, military, industrial, environment, and habitat (Nasir & Ku-Mahamud, 2016). Due to dynamic nature of distributed system, there a many aspects that needs to be considered such as in packet routing, sensor node localization, load balancing, time synchronization, and security issues. Among all, packet routing is one of the main issues that is often discussed by many researchers (Luo & Li, 2012; Zeng & Dong, 2016; Mostafaei, 2018, Sarkar & Murugan, 2019). There are many issues and limitations that need to be overcome in packet routing to prolong the network lifetime of the WSN system.

2.1.1 Design and Routing Challenges in Wireless Sensor Network

One of the challenges in WSN is to design, develop and implement the routing environment using the energy efficient software and hardware with the aim to minimize the energy usage in the system (Gupta & Sikka, 2015). Sensor nodes that have limited battery power are responsible in sensing, collection, data communication, data processing, and data transmission need to save their powers to prevent the occurrence of dead node (Loganathan et al., 2017). Sensor nodes are also able to work without human interventions in managing the network configurations, maintenance, adaptation and repair by itself especially when distributed in the large scale of networks. In the real environments, sensor nodes are also prone to the physical attacks when exposed to adversaries and bad weather (Rathod & Mehta, 2011). A good WSN design must consider the robustness of each sensor nodes and also the whole system. To achieve this objective, the system must be constructed to adapt and tolerate with individual node failure without affecting the whole system.

In WSNs, packet routing is one of the main issues that need extra focus in order to maximize throughput, minimize delay, minimize energy consumption of sensor nodes and avoid overload on certain sensor nodes (Dai, 2009; Li, Lim, & Liu, 2010). In order to achieve these objectives, the optimal routing path that can ensure packets submission to the destination node must be discovered by the routing algorithm to reduce the possibility of packet loss. A good routing algorithm can discover several alternatives of routing path and fairly distribute packets to all potential sensor nodes to balance the load in the system while reducing the congestion and energy consumption of each sensor node.

Local optima is another routing challenge in WSN systems which happens when the path discovery is stuck in a dead loop during packet submission (Li et al., 2010). This problem needs to be considered in order to reduce the submission time of packets, reduce packet loss rate, and improve the global optimal solution.

Metaheuristic algorithms such as swarm intelligence and local search are among the algorithms proven in solving the routing problem in WSNs (Vijayalakshmi & Anandan, 2018; Zou & Qian, 2018).

2.1.2 Issues and Limitations of the Routing Algorithms in Wireless Sensor Network

In real environments such as earthquake early warning system or fire detection, sensed data need to be delivered within time constraints with the purpose of observation or immediate reactive action (Wang & Ni, 2012; Lule & Bulega, 2015). Data needs to be sensed and reached at the destination node as fast as possible. Many routing algorithms in WSN try to transmit data in minimal time and at the same time prevent data loss, congestion, and noise. Priority approach has also been applied by routing algorithms where real time data are put in high priority list and non-real time data in the low priority list (Karim, Nasser, Taleb, & Alqallaf, 2012).

Management of sensor nodes is another issue faced by routing algorithms in WSN. Nature characters of sensor nodes that have limited battery power and storage capacity give huge impacts to the packet routing process. In certain cases, packet loss problem happens when sensor nodes that carry that data suddenly died due to depletion of energy. Packet loss also happens when the storage of sensor nodes are full that leads to corrupted or missing routing records (Ez-Zaidi & Rakrak, 2017).

2.2 Overview of Metaheuristic Algorithms

The word metaheuristic, that comes from the Greek verb, is the combination of the word meta meaning “upper level” and heuristic meaning “to find”. Heuristic is the basic algorithm that searches the solution space to find a good solution and can be categorized as local search algorithms and constructive algorithms (Bianchi, Dorigo, Gambardella, & Gutjahr, 2009). Local search algorithms work by improving the pre-existent solution by modifying its components while constructive algorithms develop the solution by combining the components of the solution one by one until the solution is completely discovered.

As stated by Blum and Roli (2003), metaheuristics is the concept of exploring the search space by implementing different strategies. These strategies are used to balance between exploration of the new search space (diversification) and exploitation of the previously accumulated search experience (intensification). It is important to balance the diversification and intensification in order to identify a high quality solution in a short time within the regions in the search space. Figure 2.1 shows that the metaheuristic consists of two main categories, local search and population-based where evolutionary computing and swarm intelligence are under the population-based category.

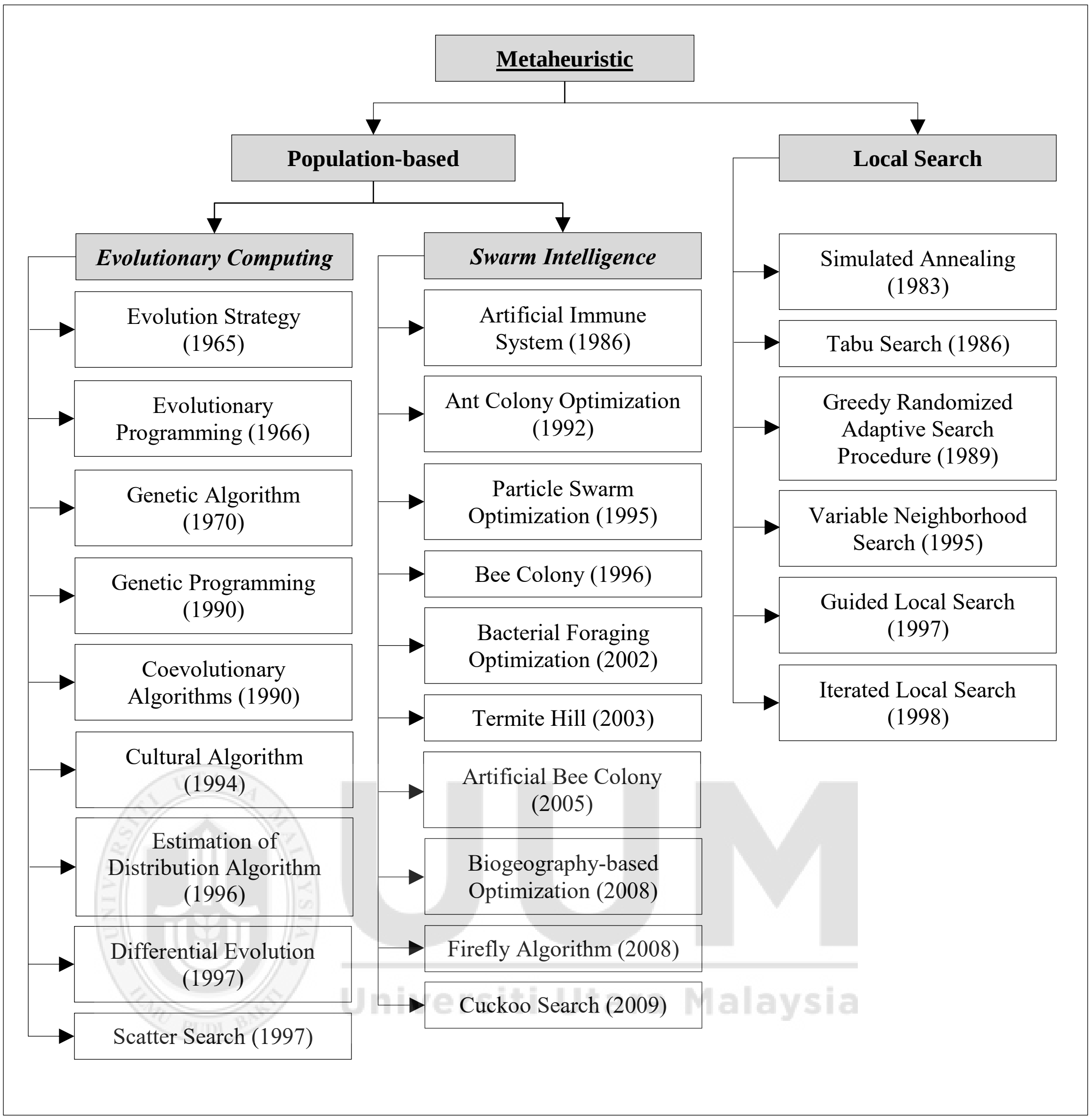


Figure 2.1. Metaheuristics algorithms

2.2.1 Swarm Intelligence Algorithm

Swarm intelligence consists of nature-inspired algorithms such as Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Termite Hill (Termite-hill), firefly algorithm, cuckoo search, and Particle Swarm Optimization (PSO) (Yang, 2014b). These algorithms are inspired by the biological behaviour of natural systems such as colonies of bees or ants, schools of fish, and flocks of birds that can be related to the optimization problem which is part of the computing field (Pintea, 2014). The concepts of swarm intelligence which include quality, proximity, diverse response, adaptability and stability (Lim & Jain, 2009) are suitable for solving distributed optimization problems such as WSN packet routing, grid scheduling and sensor nodes allocation. There are several swarm intelligence algorithms that are commonly used to improve the performance in WSNs such as ACO, PSO, ABC, Termite- Hill and Cuckoo Search.

The ACO algorithm is inspired by the foraging behaviour of real ant colonies that can help users to design metaheuristic algorithms and solve optimization problems (Dorigo & Stützle, 2004). This algorithm simulates the behaviour of real ants in finding the shortest path between the nest and food sources. Several ants work together to construct a good solution where a decision is built step by step by a single ant until a complete solution is found. The ant colony uses stigmergy, which is an indirect communication between them, by depositing a certain amount of a chemical substance called pheromone that can be detected by all ants in the environment (Dorigo & Stützle, 2004). Every ant will deposit pheromone along the trail as they move from the nest to the food source and vice versa. The strength of the pheromone will attract ants to follow a chosen path that is considered as a good or optimal solution. Therefore, they will choose the shortest or optimal path based on the pheromone value. According to the concept, the path with high pheromone value is shorter than the path with low pheromone value. This behaviour is the basis for cooperative communication in ACO. Details about ACO are further explained in Section 2.3.

The ABC algorithm that is inspired from the foraging behaviour of honeybee swarms was proposed by Karaboga (2005). Three categories of bees in the ABC algorithm, employed bees, scouts and onlookers, are responsible for finding the food source around the hive. The number of employed bees is equivalent to the number of food sources in the colony. In the ABC algorithm, the food source corresponds with the possible solution of the optimization problem while nectar amount is represented as the quality or fitness to that solution. On the other hand, the number of employed bees is represented as the number of solutions in the population. Employed bees will determine and move to the food source based on the information in their memory to check the nectar amount of the food source. Employed bees will return to the hive to share the information obtained from the food source with onlookers. The information is presented by using the waggle dance by the employed bees in the dance area. Onlookers in the hive will determine the best food source based on the dance presented by the employed bees. After a few times, the food source will be abandoned and employed bees will be transformed into scouts. The scout is responsible for moving randomly in the colony to find a new food source to be explored. Like the other swarm intelligence algorithms, exploitation and exploration must be carried out together. In ABC algorithms, scouts perform the exploration process while employed bees and onlookers control the exploitation process.

The termite-hill algorithm that is based on the termite behaviour in building the hill, is one of the intelligence entities that can perform self-organization to achieve complex tasks (Zungeru, Ang, and Seng, 2012b). There are five (5) concepts of swarm intelligence in the termite-hill algorithm: stigmergy, multiple interactions, randomness, positive feedback, and negative feedback. The movement of termites in the colony is based on the randomness concept where it can continue the current solution that fits to the environment or encourage a new solution to the system. Termites build a termite hill by dumping collected pebbles in one place. A termite moves randomly by carrying only one pebble at a time with the objective of finding a suitable place to drop the pebble. There is no direct communication between termites during the construction process and the only way to communicate is by using the pheromone value like that used by the ACO. This indirect communication is called stigmergy where termites move toward the largest pile based on the pheromone value. Termites will sense the pheromone value and move toward it to drop pebbles. This positive feedback concept will increase the pheromone value on the pile to attract other termites to drop more pebbles and to arrive faster. In the early stage, several small piles are quickly developed due to random dumps by termites. In order to reduce the number of piles, the negative feedback concept is applied where the pheromone value is decreased by the evaporation process. Negative feedback can encourage a large pile to grow and prevent small piles from continuing to attract termites. This concept is important to remove poor or old solutions in the system. If the number of termites is insufficient, the pheromone will evaporate before more pebbles are dropped onto the pile. In maintaining the pheromone value on the pile, multiple interactions are applied in the development process of the termite hill.

The PSO algorithm, proposed by Eberhart and Kennedy (1995), is inspired by the social behaviour of some animals such as fish, birds, herds and insects. Particle swarm optimization is established by having candidate solutions which are particles that work in a population called a swarm. One of the social behaviours applied by PSO is to cooperate among particles in finding food by changing the search pattern among swarm members. Particles will move randomly in the search space by using the combined information from the best solution based on their learning experience and the best solution found by the other neighbourhood particles. When the new best position is discovered by particles, it will become a guide to the movement of the whole swarm. This process is repeated until the optimal solution is obtained in the system. Particle swarm optimization also considers the exploration of particles to a new best solution and exploitation to the previous best solution. It is important for PSO to balance both criteria in order to prevent the premature convergence problem in local optima.

The Cuckoo Search (CS) algorithm, proposed by Yang and Deb (2009), is based on brood parasitism where some cuckoo species lay their eggs in the nests of other host birds. This approach is utilized by the female parasitic cuckoo, who can mimic the pattern and colours of the other host species egg, in order to increase productivity and reduce the possibility of eggs being thrown away by the host. In certain situations, when the host birds realize the unfamiliar egg in their nest, they will throw away the egg or leave their nest and build a new one. When the cuckoo egg hatches earlier than the host eggs, the cuckoo chick will remove the host eggs from the nest. This action ensures the cuckoo chick gets enough food from the host bird. The cuckoo chick can also mimic the behaviour of the host chick in getting more food from the host. There are three rules in developing the CS algorithm where each cuckoo lays only one egg at a time and puts it in a random nest, the nest with high quality eggs will be carried forward to the next generation, and the host bird has a probability to identify an unfamiliar egg and decide to throw away the egg or move to a new location to build a new nest. The CS algorithm is suitable to be adopted in solving the optimization problem where the host eggs in the nest act as the current solution and the cuckoo egg represents a new solution. This approach aims to replace the old solution with a new, better solution in the system.

2.2.2 Local Search Algorithm

Local search algorithm, which is also known as neighbourhood solution, applies local movement to improve the solution locally (Vob, 2001). Several techniques have been developed to establish the local search such as iteration, greed, random, steepest descent algorithm and variable neighbourhood search (Aarts & Lenstra, 2003; Gendreau & Potvin, 2010; Zapfel, Braune, & Bogl, 2010). Local search that is applied in metaheuristics algorithms such as Tabu search and SA are proved to perform better in WSN systems (Shekofteh, Yaghmaee, Khalkhali, & Deldari, 2010; Kaur & Gangwar, 2015; Keskin, Altınel, & Aras, 2015).

Tabu Search (TS), developed by Glover (1986), is a metaheuristics algorithm based on local search. The TS algorithm has the ability to prevent the local optima problem by applying various mechanisms such as diversification and memory (Rothlauf, 2011). Tabu search is flexible when applying the concept of adaptive memory and responsive exploration. There are four types of memory that operate in TS: frequency (long-term memory), recency (short-term memory), influence, and quality (Glover & Laguna, 1997). However, only one or two types of memory are used at one time by many applications to complete standard operations. Many optimization problems such as network routing, job scheduling, and the TSP have been successfully resolved by the TS algorithm. A detailed explanation about TS is further covered in Section 2.4.

Simulated Annealing (SA), which is another optimization algorithm, was developed by Kirkpatrick, Gelatt, and Vecchi (1983). SA is inspired from the physical process of metal cooling and freezing in a crystalline state in the annealing of materials (Yang, 2014a) and has been implemented to overcome many combinatorial optimization problems (Moschakis & Karatza, 2015; Wei, Zhang, Zhang, & Leung, 2018). Local optima problems can be prevented by using SA where hill-climbing movements are applied in finding the global optimum solution. SA will also control the maximum allowed decrease in solution quality (Zapfel et al., 2010). The SA algorithm has also been successfully applied to improve packet routing in WSNs (Zhang, Zhang, & Bu, 2014; Mohammadi & Noghabi, 2016).

2.3 Ant Colony Optimization

A combination of priori information (heuristics) and posteriori information (pheromone) is a key success of ACO algorithms. Priori information (called greedy strategy) is the quality of candidate solution while posteriori information (called autocatalytic process or positive feedback) is the goodness of the previously obtained solution. The ACO algorithm combines heuristics to create a solution from a list of candidate solutions with the accumulated experience from the previous iterations in getting good solutions.

The Ant System (AS) is the first member of ACO algorithms that was proposed by Colorni, Dorigo, and Maniezzo (1991) and Dorigo (1992). The main objective of AS is to simulate the foraging behaviour of real ants to find an optimal path from nest to food sources. The AS is also the first ACO algorithm introduced to solve the TSP (Dorigo, Maniezzo, & Colorni, 1996) where an ant which is responsible for finding the shortest route visits all the cities exactly once in a given set (Dorigo & Gambardella, 1997a). Three versions of AS were proposed by Dorigo, Maniezzo, and Colorni (1991a, 1991b, 1996): ant-density, ant quantity and ant-cycle. The difference between these three versions is the pheromone updating techniques which affect the quantity and time. In ant-density and ant-quantity, the ants update the pheromone directly after moving from one city to another. However, the pheromone update in the ant-cycle will only be done after all the ants have constructed the tours and each tour quality is based on the amount of pheromone deposited by each ant. Ant solution construction and pheromone update are two main phases in AS. Nevertheless, when the size of the test-instances increases, the performance of AS tends to decrease compared to the later variants of ant algorithms.

Dorigo and Gambardella (1997a, 1997b) propose an Ant Colony System (ACS) which is an upgraded version of AS to improve the performance of TSP and some other problems. Ant colony system differs from AS in three main aspects. First, action choice rule in ACS is more aggressive than AS where it exploits the search experience accumulated by the ants more strongly than AS. Second, pheromone deposit and pheromone evaporation take place only on the global best solution. Third, when an ant moves from city r to city s, some pheromone will be removed from the arc to increase the exploration of new paths. A pseudo-random proportional rule is used by ants in ACS to select the next city that has not yet been visited. This rule is a transaction between exploitation and exploration. Exploitation uses the information from the previous iteration with the maximum combination of heuristics value and pheromone trails while exploration refers to the possibility to add a new edge to the solution. The pseudo-random proportional rule is calculated based on either the a random variable ranging from 0 to 1 and a parameter to control the possibility of exploration or exploitation for the case of exploitation or a random variable based on the probabilistic decision rule for the case of exploration (Dorigo & Stützle, 2004). After each iteration, only the global best solution will be allowed to deposit pheromones in the ACS. In contrast to AS, trail update only applies to the arc of the global best solution. Apart from that, a global update rule with a combination of local update rules is applied to ACS. Evaporation rate and length of global best tour are elements to calculate the global pheromone update. After having crossed an arc, a local update rule will be used immediately during the tour construction. This is to prevent an already chosen arc from being selected by the following ant and the exploration of an unvisited arc being increased. Local pheromone update is calculated based coefficient value and the initial value of pheromone trail (Dorigo & Gambardella, 1997b).

The MMAS is another improvement of the AS-based algorithm, proposed by Stützle and Hoos (2000). The MMAS shows a better performance compared to other ACO algorithms for Quadratic Assignment Problems (QAP) and TSP. The MMAS differs from AS in four main aspects. First, a greedier search mechanism is used by the MMAS to allow better exploitation of the best solution. Second, the pheromone trail is controlled by the MMAS in order to prevent premature stagnation during the search process (ants converge early to one sub-optimal solution) by limiting the pheromone trail to the interval $[T_{min,} T_{max}]$. Third, the MMAS allows higher exploration at the start of the algorithm by initializing the pheromone trail to the upper trails limit. Finally, pheromone trails are reinitialized by the MMAS when no better tour has been generated for a certain number of iterations or when the system reaches stagnation. In the MMAS, only the best ant or the global best solution is allowed to add a certain amount of pheromone. Therefore, the arc that will always be the best solution will get a large amount of pheromone. The upper and lower pheromone trail limits are used in the MMAS to avoid stagnation. The pheromone trail limit has the effect of indirectly limiting the probability T_{ij} of selecting a city j when an ant is in city i to an interval $[T_{min,} T_{max}]$ with $0 < T_{min} \leq T_{ij} \leq T_{max} \leq 1$.

The Local Best Tour Ant System (LBTAS) that uses local information to guide the ants' search process was proposed by Kaegi and White (2003) as a new version of AS. The main modification of the LBTAS is that each ant updates pheromone values according to its own best tour from the start of the algorithm. This prevents the use of global best solution observed by all ants like in the ACS and MMAS. Each ant works individually in the LBTAS and, at the same time, indirectly cooperates with other ants. The LBTAS showed better performance compared to the AS and $AS_{elitist}$ when applied to TSP. This proves that the local search procedure on LBTAS gives a high change to a better version of AS.

Energy-Efficient Ant-Based Routing (EEABR) proposed by Camilo et al. (2006) is the version of AS that is applied in WSN. EEABR used forward ant to explore the potential path and backward ant to update the pheromone value. EEABR used probabilistic decision rule during searching process and global pheromone update to control the pheromone value of selected node. Energy level and travelled distance are two important keys that are used by EEABR in calculating both the probabilistic decision and global pheromone with the aim to optimize the routing process and at the same time to increase the network lifetime of the system.

Hybrid ACO algorithms also have been applied in WSN in improving the performance of single algorithms. Cui, Liu, and Zhao (2015) proposed ACO-GA that combined ACO and GA in solving the routing problem in WSN. ACO that is based on global parallel and distribute search capabilities is an excellent algorithm for route discovery process. ACO accumulates the pheromone to find the optimal path in transmitting packets in WSN. On the other hand, GA that has a global search capability is applied to improve the convergence speed and prevent the local optima problem that ACO does not cater by using crossover and mutation approach. The combination of both algorithms showed good results when compared to the single algorithms such as ACO and GA.

Fish Swarm Ant Colony Optimization (FSACO) proposed by Li, Keegan, and Mtenzi (2018) combined two swarm intelligence algorithms which are ACO and Artificial Fish Swarm Algorithm (AFSA) in improving the performance of packet routing in WSN. FSACO used state transition rule and global pheromone update from ACO for route discovery process where the path length and energy level of sensor nodes are considered for both formulas. At the same time, FSACO also used crowd factor and heuristics information from AFSA to prevent the congestion during routing process. Experimental results showed that FSACO performed better than single ACO variants such as EEABR and Sensor Ant.

2.4 Tabu Search

The TS algorithm is one of the metaheuristic algorithms that explore the solution space beyond local optimality based on the local heuristic search procedure. There are two types of memory in TS algorithms: attributive memory that records the information of some attribute solutions that change when moving from one solution to another; and, explicit memory that records the complete solution (Glover & Laguna, 1997). For example, in the packet routing scenario in a WSN, a new solution vector will be created when moving a packet *p* from sensor node *Na* to sensor node *Nb*. Therefore, the TS memory can record the whole complete solution or record only the attributes that change the solution which is the part when packet *p* assigned to sensor node *Nb*. In this situation, attribute memory will prevent the TS algorithm in using the old solution for *k* number of iterations with the same sensor node. Nevertheless, this Tabu attribute can be overridden if the move will produce a better solution than the best so-far-solution. The duration parameter for the move, which is called Tabu tenures, is effective based on the size of the problem instance.

The TS algorithm is initiated by the initial solution either by random discovery or by using any ad-hoc algorithms such as the ACS, MMAS, and ABC algorithms (Xhafa, Alba, Dorronsoro, Duran, & Abraham, 2008). The searching process will be continued by the TS algorithm to find the local optima. At this stage, the neighbour solution is saved as a current solution if it is not a Tabu. On the other hand, if the neighbour solution is better than the best-so-far solution, it will be marked as a current best-so-far solution. In the scenario that the neighbour solution is Tabu, the inspire level will check the status to override if this solution is better than best-so-far solution. If the termination condition is not met, the TS algorithm will update the memory and start a new iteration after movement to the neighbour solution. This searching process is repeated in the neighbourhood as a guide to explore interesting areas in the search space efficiently (Costa, 1994). There are several issues in implementing TS algorithms such as the size of the Tabu list, the information needing to be saved in the memory, how to perform diversification, and the method to be used to move to the neighbourhood (Thesen, 1998).

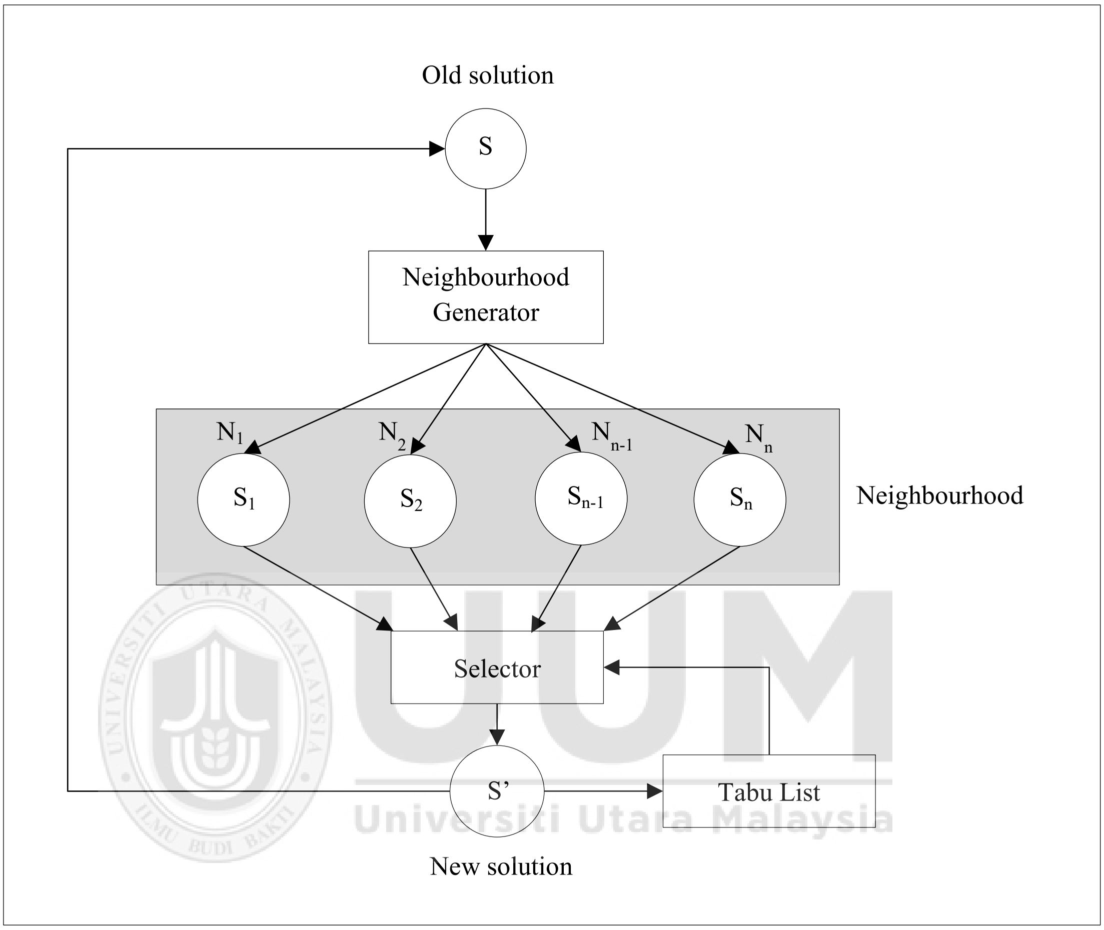


Figure 2.2. Process of TS algorithm (Zapfel et al., 2010)

2.5 Routing Algorithms in Wireless Sensor Networks

Routing packets from the source node to the destination node is crucial due to the limitations of sensor nodes in terms of battery power, storage, and memory to sense, collect, and transmit data from various locations in WSN. Section 2.5.1 discusses the performance evaluation criteria that are used by researchers in evaluating the performance of their proposed algorithms in routing packets in WSN, New or improvement of existing algorithms is commonly achieved using two approaches such as single algorithm or hybrid algorithm. While a single algorithm is commonly proposed, several researchers have proposed a hybrid algorithm by adopting good components and/or functions from more than one algorithm to improve specific criteria or tackle specific problems. Section 2.5.2 discusses in detail about the single swarm intelligence algorithms implemented in WSNs such as EEABR, BeeSensor, and Termite-hill while hybrid swarm intelligence algorithms such as PSOABC and Bee-Sensor-C are elaborated in Section 2.5.3.

2.5.1 Performance Evaluation Criteria of Routing Algorithms in Wireless Sensor Network

Performance evaluation criteria that commonly used by researchers in evaluating the performance of their proposed algorithms are energy consumption, energy efficient, success rate, packet loss rate, latency, throughput, number of alive node, number of dead node, and residual energy (Oldewurtel & Mahonen, 2010; Nasir, Ku-Mahamud, & Kamioka, 2017). Good routing algorithms supposedly can fulfil at least one of these criteria during experiments. In order to achieve this objective, sensor nodes in WSN should be organized to sense and transmit packets optimally without affecting the network lifetime of the system. Packets need to be distributed fairly to all potential sensor nodes in balancing the energy usage with the aim to prevent the sensor nodes from drastically drain their energy that can cause the dead node (Levendovszky, Tornai, Treplan, & Olah, 2011). Energy efficiency is one of the criteria to measure the energy usage during packet routing process where it measures based on the number of successfully arrived packets and the energy consumed (Zungeru et al, 2012a). Most of the routing algorithms used energy consumption and energy efficiency as a performance metrics during experiments (Singh & Behal, 2013; Zungeru et al., 2012a; Li et al., 2018; Gupta, 2018).

Success rate and packet loss rate are two important keys in measuring the ability of routing algorithms in submitting packets from source node to destination node. Success rate is measured by the total number of packets that successfully arrived at the destination node per all submitted packets while the packet loss rate is contradict to it where it measures the number of packets that failed to arrive at the destination node per all submitted packets. IEEABR which is one of the ACO variants uses success rate along with latency, energy consumption and energy efficiency to evaluate its performance. Variants of ABC such as BeeSensor and Bee-Sensor-C that have been applied in WSN also used success rate and packet loss rate to measure their performances.

Many researchers have also used throughput as one of the criteria to evaluate the ability of routing algorithms in transmitting packets. Throughput is measured by the number of successful packets arrived from source node to the destination node per second (Alazzawi & Elkateeb, 2009). Throughput is applied as performance metric during experiments to see the relationship between the number of packets received and time where the large number of packets arrived in the short time indicates the high throughput value and high quality of transmission path. Zungeru et al. (2012b), Singh and Behal (2013), and Li et al. (2018) used throughput along with energy consumption and energy efficiency as a performance metrics in ensuring the quality of packets transmission in terms of time, energy and quantity of successful packets received.

Latency is another important performance evaluation criterion that is commonly used in the performance evaluation where it measures the time taken to submit packets from source node to destination node. Many ACO variants applied in WSN such as Yan et al. (2011), Luo and Li (2012), Zungeru et la. (2012a), and Cui et al. (2015) that aimed to reduce the transmission time of packets used latency as one of the performance metrics. These algorithms minimized the packets transmission time by discovering the optimal path with the combination of state transition rule and pheromone update.

2.5.2 Single Swarm Intelligence Approaches in WSN Packet Routing

This section discusses in details of single swarm intelligence algorithms that are applied in WSN in terms of their contributions, research methodologies, performance evaluation criteria and also the drawbacks and gaps that can be studied. At the end of this section, all these algorithms are summarized based on their performance evaluation criteria.

Camilo et al. (2006) propose Energy-Efficient Ant-Based Routing (EEABR) with the aim of reducing the communication load and energy consumption in WSNs. EEABR is the fundamental ACO algorithm in WSN that is used as a benchmark study by many ACO algorithm in validating their performance. Two types of ants have been used in EEABR, the forward ant that explores the system in finding the optimal sensor nodes, and the backward ant that updates the pheromone value of traversed sensor nodes. A probabilistic decision rule is applied by the forward ant in evaluating the capacity of the neighbour nodes during the node selection phase while the global pheromone update is applied by the backward ant to encourage the optimal sensor nodes to be selected by the following ant in the next iteration. Experiments have been done in evaluating the energy efficiency of EEABR in three different conditions: mesh network, mobile network, and static network. Experimental results show that the energy efficiency of EEABR is better than the Improved Ant-Based Routing (IABR) Algorithm and Basic Ant-Based Routing Algorithm (BABR). However, the EEABR algorithm did not explore alternative paths that may lead to hotspot problems on certain sensor nodes which will affect the load balancing and network lifetime of the system. The performance of EEABR was only compared and evaluated with the other variants of ACO but not with the other swarm intelligence algorithms.

The study by Yan et al. (2011) proposes an improved AS called ASW to solve the routing problems in static wireless sensor networks. The objectives of this research are to minimize delay and energy consumption of sensor nodes during submission of packets from source node to destination node. The proposed algorithm is similar to AS on the node selection process but with a different pheromone update mechanism. In ASW, different amounts of pheromone are assigned to every ant during the pheromone update process depending on the minimum energy consumed by each ant. Comparison has been made between AS, ACS and ASW on the average delay and average energy consumption. Experimental results show that ACS achieves the lowest energy consumption followed by ASW and AS. On the other hand, AS achieves the lowest average delay value followed by ASW and ACS when routing packets using 50, 100, 150, 200, 250, and 300 sensor nodes. As can be concluded from the experimental results, ASW performed average for both performance metrics. However, similar to EEABR, the performance of the proposed ASW algorithm was only compared with the other family of ACO but not with the other swam intelligence algorithms.

Luo and Li (2012) propose an MMAS-based routing algorithm for reducing the packet loss, delay and energy consumption of sensor nodes during the routing process in WSNs. The search angle has been proposed in this algorithm to limit the ant's search area during nodes selection activities. By using the search angle approach, nodes only broadcast their information to their neighbours within the search angle area to reduce the energy consumption of each sensor node and to increase the search speed of ants. The quantities of pheromones are different from each path where the good path discovered by ants gets more pheromone compared to others. However, the quantity of pheromone remains limited to the maximum and minimum range, as in MMAS approach, to control stagnation in the WSN environment. Experimental results show that the proposed algorithm performs better than the MMAS basic algorithm in terms of delay, packet loss, and dead nodes aspects. Regardless of the performance, there was no pheromone evaporation rate that reduces the pheromone value of the optimal sensor nodes in this proposed algorithm. This problem leads to the hotspot problem where the energy at certain sensor nodes is quickly depleted due to the heavy load that will affect the network lifetime of the system.

Almshreqi, Ali, Rasid, Ismail, and Varahram (2012) propose a SensorAnt algorithm in balancing the energy consumption during packets routing by utilizing and optimizing all sensor nodes in WSNs. SensorAnt is based on an ACO algorithm where the quality of paths and hops are measured in selecting the optimal path to forward the data from the source node to destination node. The quality is measured by considering the number of hops, minimum residual battery power of sensor nodes, and average energy of route and network. Two types of ant are used in SensorAnt, ant-forward to find an optimal route to the destination node and ant-backward to put the pheromone value on the visited sensor nodes. The pheromone value is stored in the sensor node's memory and will be updated in order to prevent the hotspot problem on certain sensor nodes. The performance of SensorAnt was compared with EEABR in terms of energy consumption and energy efficiency. Experimental results show that the SensorAnt performs better for both performance metrics. However, the performance of SensorAnt was only compared with the ant-based algorithm and the other performance metrics such as delay, throughput and packet loss were not considered.

Improved Energy Efficient Ant Based Routing (IEEABR) was proposed by Zungeru et al. (2012a) in improving EEABR routing algorithms. IEEABR uses the same concept as EEABR where the forward ant is used to explore the potential path and backward ant is responsible to update the pheromone value of selected sensor nodes. IEEABR also applies the same formula as EEABR for the probabilistic decision rule and pheromone update technique. However, IEEABR differs from EEABR in terms of memory usage where the routing table is intelligently initialized in the early stage to give priority to potential neighbour nodes. A routing table in IEEABR can also intelligently update in case of link or node failure to reduce the congestion problem in WSNs. In order to balance the energy consumption of each node, the number of neighbour nodes is considered by IEEABR in calculating the probability distribution of nodes. Performances of IEEABR were compared with BABR, SC, FF, FP, and

EEABR in terms of latency, success rate, energy consumption, and energy efficiency. In both static and dynamic scenarios, IEEABR shows good performances in energy consumption and energy efficiency aspects. Notwithstanding the results, experiments were only done to compare the performance of IEEABR with the ant-based routing algorithms but not with relevant swarm intelligence algorithms.

Saleem, Ullah, and Farooq (2012) proposed the BeeSensor routing algorithm that is based on the foraging behaviour of honey bees. Three types of agents operate in BeeSensor: scouts, packers and foragers. Packers reside in the hive which is the software module in the sensor node that processes sensed data from other sensor nodes. Packers in the source node broadcast to scouts that are responsible for finding the optimal path to the destination node. Scouts will evaluate the quality of paths and return to the source node once the destination node is found. Foragers are launched once scouts have returned to the source node. Foragers are responsible for evaluating the quality of visited paths and to transmit data packets from source node to destination node. The remaining energy of sensor nodes and path length are the two elements in evaluating the quality of the path and expressed through the number of waggle dances by bees. The performance of the BeeSensor algorithm was compared with EEABR, FF-Ant, FP-Ant, SC-Ant and AODV in terms of packet delivery ratio, latency, energy efficiency, control overhead, lifetime, and energy consumption. BeeSensor achieved the best performance for energy efficiency, control-overhead and lifetime value. On the other hand, FP-Ant was the best algorithm for packet delivery ratio and EEABR was the best for latency. However, BeeSensor did not consider the local optima problem during the searching process by scouts that may affect the latency and packet delivery ratio.

The termite-hill routing algorithm was proposed by Zungeru et al. (2012b) with the aim of solving routing problems in static, dynamic and mobile WSN environments. Autocatalytic behaviour that is used in finding a solution in a reasonable time is applied by termites in the Termite-hill algorithm. Termite-hill uses the same concept as EEABR which has a forward soldier and backward soldier in finding the optimal path between source node and destination node. Termite-hill also applies a pheromone value to communicate between termites in the system. There are 3 types of pheromone: initial, lower and upper. Initial pheromone is calculated in the early stage to find the probability distribution of packets in the system. Pheromone update and pheromone evaporation are guided by the range between lower pheromone and upper pheromone to the selected sensor nodes. This limitation is essential in encouraging termites in the next iteration to reselect the optimal sensor nodes. Experiments were done to compare the performance of Termite-hill with SC, FF, and AODV in static, dynamic, and mobile environments. Termite-hill attained better performance in terms of throughput, energy consumption, and energy efficiency. However, the performance of Termite-hill in a large sized network was not validated because the experiments only covered small numbers of sensor nodes.

A study by Okafor and Fagbohunmi (2013) proposes an ant-based routing algorithm that aims to reduce the energy consumption among sensor nodes in WSNs. In this proposed algorithm, pheromone values are stored in the node's memory instead of the ant's memory as in the traditional ACO algorithm. This approach reduces the energy and size of data that must be carried by the searching ant. The selection of the next nodes depends on the neighbour nodes' energy levels where the nodes with high energy levels will be selected by ants in forwarding packets to destination nodes. A pheromone evaporation technique is applied in order to reduce the attraction to the optimal path and to encourage exploration to the other potential path. The number of visited sensor nodes by the ant during the searching process will be used as an element to calculate the pheromone value. However, experiments were only done to see how the sensor nodes communicate within the range and how the ant moves in the WSN to find the optimal path. Important performance metrics that are always used in evaluating the routing performance such as energy efficiency, latency, and throughput were not considered by the proposed algorithm.

Mobile sink with a combination of ACO was proposed by Singh and Behal (2013) to improve the network lifetime in dynamic WSN environments. The ant is responsible for calculating the energy of sensor nodes and deciding the next best location for the mobile sink node. The location with high energy level of sensor nodes will be selected to move the mobile sink node. Thus, it will save the energy of available sensor nodes and balance the entire WSN environment. The performance of the proposed algorithm was compared with the other routing algorithms such as Termite-hill, FF and AODV in terms of throughput, energy consumption, energy efficiency and network lifetime aspects. In both static and dynamic sink environments, the proposed algorithm performed best in terms of throughput and energy consumption aspects. The proposed algorithm also achieved the highest energy efficiency and lifetime value when routing packets in the dynamic sink environment. Meanwhile, Termite-hill performed better in the static sink environment for energy efficiency and lifetime aspects. Even though the proposed algorithm showed a good performance, there was no pheromone update function that may lead to an unbalanced selection of nodes to forward packets.

Orojloo and Haghighat (2016) propose a packet routing algorithm in WSNs called TSRA that is based on the TS approach. The main objectives of this research are to balance the packet transmission among sensor nodes to reduce energy consumption and prolong the network lifetime of the system. The remaining energy of sensor nodes and required transmission energy are two important factors considered by TSRA as link cost. Tabu tenures and Tabu list, which are beneficial features of TS algorithm, are used in avoiding the selection of low energy sensor nodes. The size of the Tabu list is calculated based on the number of nodes, wireless communication coverage, and network size. The movement in the neighbourhood search space by the TSRA also considers the hop counts and energy consumption value in reducing the average cost of routing. The performance of TSRA was validated against traditional Ant Colony Algorithm (ACA), Ant Colony based Location-aware Routing algorithm (ACLR), and Energy and Path aware ACO algorithm for routing of Wireless Sensor Networks (EPWSN) in terms of energy consumption, network lifetime and routing cost. Experimental results showed that TSRA performed better than the other algorithms in all performance metrics. Unfortunately, TSRA did not consider the local optima problem that may affect delay and throughput of the whole system.

An enhanced version of the AS algorithm, called Smart Routing Algorithm (SRA), was proposed by Bouarafa et al. (2018) to improve the routing performance in WSNs. During the packets submission from the source node to destination node, the predecessor node will broadcast a request message to its successors which are known as neighbour nodes. Neighbour nodes that receive the message will store the predecessor ID in the neighbour list and, at the same time, broadcast their ID, residual energy, and location to the predecessor node. From this point, sender node and receiver node are connected to each other for communication. By using the acknowledgement obtained from the successors nodes, the SRA will calculate the probability of each node by considering the sensor node's remaining energy and distance between the two nodes. After the packet has successfully arrived at the destination node, the SRA will perform a pheromone update to the traversed sensor nodes. Evaporation rate and path length are the two elements considered in updating the pheromone value. The performance of the SRA was evaluated in terms of path length and energy consumption. From the experiments that were executed in 50 iterations, the SRA was proven to preserve the network lifetime of the WSN because there are no dead nodes during experiments due to the energy balance among sensor nodes. Despite the good performance, the other important performance metrics such as delay and throughput were not considered by the SRA and the performance was not compared with other routing algorithms.

Table 2.1 summarizes all single swam intelligence algorithms covered in Section 2.5.1 in terms of performance metrics used in evaluating their performance. Energy consumption is the most selected performance metric followed by energy efficiency and latency. Thus, it can be concluded that energy consumption and energy efficiency are important elements in ensuring the lifetime of a WSN. At the same time, latency that measures the submission time of packets from source node to destination node is also important in increasing the throughput value of routing algorithm. The EEABR proposed by Camilo et al. (2006) has been selected by many researchers as a benchmark to evaluate their algorithm like Almshreqi et al. (2012), Zungeru et al. (2012a), Saleem et al. (2012) and Zungeru et al. (2012b). Zungeru et al. (2012a, 2012b), who proposed IEEABR and Termite-hill, also adopted and adapted the routing concept from EEABR. Despite many new ACO based algorithms that have been proposed in WSN, EEABR is still relevant to be used as a benchmark study due to it concepts that mimic the traditional ACO approach proposed by Dorigo (1992). Method and experiment set up used by EEABR are also suitable and easy to be adopted and adapted by the other routing algorithms. Based on this table, EEABR (Camilo et al., 2006) IEEABR (Zungeru et al., 2012a), Termite-hill (Zungeru et al., 2012b), and BeeSensor (Saleem et al., 2012) are used as benchmark algorithms to be compared with the proposed algorithm in the experiments due to the concept and performance metrics used that are similar and comparable to the proposed algorithm. Even though research done by Bouarafa et al. (2018) is the recent ACO-based algorithm in WSN, it is not used as a benchmark algorithm because it was not validated and compared with the other research work.

Table 2.1

Summary of single swarm intelligence routing algorithms in WSN

Authors	Performance Evaluation Criteria											
	Energy Consumption	**Energy Efficiency**	**Latency**	**Throughput**	**Success Rate**	**Packet Loss Rate**	**Network Lifetime**	**Dead Node**	**Path Length**	**Standard Deviation**	**Control Overhead**	**Routing Cost**
Camilo et al. (2006)		√								√		
Yan et al. (2011)	√		√									
Luo and Li (2012)			√			√		√				
Almshreqi et al. (2012)	√	√										
Zungeru et al. (2012a)	√	√	√		√							
Saleem et al. (2012)		√	√		√		√				√	
Zungeru et al. (2012b)	√	√		√								
Okafor & Fagbohunmi (2013)												
Singh & Behal (2013)	√	√		√			√					
Orojloo & Haghighat (2016)	√						√					√
Bouarafa et al. (2018)	√								√			

2.5.3 Hybrid Swarm Intelligence Approaches in WSN Packet Routing

Hybridization occurs when two or more algorithms are combined with the objective of improving a specific performance metric that is not achievable using a stand-alone algorithm (Masrom, Abidin, & Omar, 2012; Fister & Fister, 2015). Algorithms could be combined partially or fully to be able to obtain the best features of the hybridization algorithm. Hybridization between algorithms can be categorized into low level and high level based on the degree of connection between algorithms.

The degree of inner exchange procedure among algorithms reflects the level of hybridization. Low level hybridization is known as strongly coupled hybridization while high level hybridization is called loosely coupled hybridization (Masrom, Abidin, Omar, & Nasir, 2014). One of the algorithms is the main algorithm in low level hybridization while part of the other algorithm is called during the execution time. Low level hybridization can be represented as Algorithm1(Algorithm2) (Xhafa, Gonzalez, Dahal, & Abraham, 2009) where Algorithm1 is the main algorithm and Algorithm2 is the subordinated algorithm (Jourdan, Basseur, & Talbi, 2009; Xhafa, Kolodziej, Barolli, & Fundo, 2011). On the other hand, each algorithm operates fully in high level hybridization while preserving its own identity. There is a chain of algorithm operations in high level hybridization type where the flow can be illustrated as ($Algorithm1 \rightarrow Algorithm2 \rightarrow Algorithm3 \rightarrow \cdots \rightarrow Algorithmn$). Based on this flow, the output from the Algorithm1 is passed to Algorithm2 and so on. High level hybridization can be represented as Algorithm1 + Algorithm2.

This section discusses in detail the hybrid routing algorithms that are applied in WSN. The contributions, research methodologies, experimental results, gaps and drawbacks of these algorithms are highlighted. Summarization of these algorithms based on the performance evaluation criteria are also provided at the end of this section. The study by Xiu-li, Hong-wei, and Yu (2008) proposes a multipath routing algorithm called MACS which combines the ACS and MMAS in WSN routing packets. The multipath routing method is applied in MACS where more than one ant is used in each iteration with the aim of minimizing the transmission delay of data, to reduce energy consumption of nodes and to balance the energy of each sensor node in the WSN. Two types of ant used in searching process: forward ant and backward ant. The forward ant is responsible for searching optimal nodes and performing local pheromone updates while the global pheromone update is performed by the backward ant. The pheromone value of each node is indicated by the pheromone trail based on the MMAS algorithm in order to prevent premature stagnation during the searching process and, at the same time, to influence the exploration of new nodes. The performance of the MACS algorithm was compared with directed diffusion, ACS and the MMAS algorithm in terms of total energy consumption and average transmission delay. Experimental results show that MACS performs better than the other algorithms in both aspects. However, residual energy of each node was not considered during the searching process that may lead to the hotspot and dead node problems.

The study by Li and Shi (2013) proposes an energy-effective Quality of Service (QoS) routing algorithm based on ACO and Stateless Non-deterministic Geographic Forwarding (SNGF) to solve the routing problem and to balance the energy consumption of nodes in WSNs. The SNGF algorithm is used to speed up the convergence of ants in order to find an optimal routing path. Nodes are divided into two groups by the SNGF algorithm in which the first group contains nodes that are closer to the destination node and another group which contains nodes that are further from the destination node. In the routing process, none of the further nodes are selected by the ant as the next hop because they are stored in a forbidden table. This will speed up the convergence of the ACO algorithm. The load of the sensor node is calculated based on its queue length and residual energy while the pheromone value is calculated by the forward ant based on bandwidth and load of node. Pheromone update value is applied by the backward ant by considering the delay of the path. Experimental results show that the proposed algorithm performs better than a basic ACO algorithm and AODV algorithm in terms of average time end-to-end delay and network lifetime. Nevertheless, the division of sensor nodes based on the location will abandon the sensor nodes with high energy levels that are further from the destination node. This situation will lead unbalanced energy consumption among sensor nodes.

Tewari and Vaisla (2014) propose a hybrid ACO and greedy algorithm with main objectives to influence the energy conservation and balance all loads to available clusters. Based on the energy value, all sensor nodes in the system are divided into small groups called clusters. The greedy algorithm is responsible for evaluating the energy level of each cluster in order to balance the energy of all clusters in the system. The ACO in this hybrid algorithm is responsible for balancing the distribution of all packets to available clusters. Experimental results show that the proposed hybrid algorithm performs better than the traditional cluster method, which is LEACH, in terms of throughput and network lifetime. However, the performance of the proposed algorithm was not compared with the other routing algorithms such as traditional ACO and greedy algorithm. The other important performance metrics such as energy consumption, energy efficiency and latency were also not considered in the proposed algorithm.

Karthikeyan and Subramani (2014) propose a hybrid algorithm called PSOABC, which combines particle swarm optimization (PSO) and artificial bee colony (ABC) to improve the QoS-based routing in WSNs. In the proposed algorithm, the bee colony is applied as an agent to discover the optimal path between the source node to destination node. Multiple forward agents that are sent to the destination node will communicate with available sensor nodes along the path. As soon as the forward agent arrives at the destination node, it will become a reverse agent. The reverse agent will return to the source node and update the routing table which consists of routing information. The PSO agent will be responsible to forward packet from the source node to destination node by referring to the routing table. In order to improve the QoS of WSN routing, several performance metrics are used to evaluate the performance of the proposed algorithm. Experimental results show that PSOABC performs better than traditional PSO algorithm in terms of delay, throughput, and packet loss aspects. However, the proposed algorithm only focuses on optimizing the path in WSN without considering the load balancing and energy efficiency aspects. This could potentially lead to stagnation and may downgrade the performance of the WSN system as the load distribution is not performed effectively.

Bee-Sensor-C proposed, by Cai et al. (2015), is inspired by BeeSensor (Saleem et al., 2012) that combined the bee algorithm and cluster technique to establish the multipath routing protocol while focusing upon the energy-aware aspect. There are three phases in developing Bee-Sensor-C: cluster formation, multipath construction, and data transmission. Sensor nodes are divided into several clusters during the cluster formation phase and the most powerful sensor node in each cluster will be selected as a cluster head. The information of each sensor node such as ID, source node ID, waiting time and remaining energy are carried by HiveHeader which is an agent that is used in Bee-Sensor-C to evaluate the condition of sensor nodes in each cluster. The multipath construction phase is done by using the technique from the BeeSensor algorithm to connect all cluster heads to the destination node by using a multi hop technique and, at the same time, to balance the energy consumption in the system. In the data transmission phase, all data in the cluster are submitted by each cluster member to the cluster head to be forwarded to destination nodes. The performance of Bee-Sensor-C was compared with BeeSensor, IEEABR, and FF-Ant. Bee-Sensor-C performs best in terms of energy efficiency, control overhead, energy standard deviation, latency, packet delivery rate, and routing building time. Despite the best performance, these experiments were done using small packet size and the performance of Bee-Sensor-C was not compared with other hybrid algorithms.

Rao and Rani (2015) have proposed an algorithm that focuses on maximizing network lifetime and increasing energy efficiency of sensor nodes in WSNs by combining the cluster technique and ACO algorithm in solving WSN routing problems. At first, each cluster will select the best sensor node to become the cluster head by considering the residual energy and distance from sensor node to destination node. Ant colony optimization is used in finding optimal paths between each cluster head to neighbour nodes and the pheromone update technique is applied on visited paths to overcome stagnation in the system. The performance of the proposed algorithm was evaluated in terms of energy consumption and number of survival nodes when compared to the LEACH and PARA algorithm. Experimental results show that the proposed algorithm performs better for both performance metrics when compared with the other two (2) algorithms. However, the load balancing aspect was not considered in the proposed algorithm that could diminish the energy of certain sensor nodes at a quicker rate and affect the lifetime of the whole system.

Cui et al. (2015) have proposed a hybrid algorithm called ACO-GA that combines ACO and GA in solving the routing problem in WSNs. This algorithm adopts the behaviour of a traditional ACO where the memory behaviour prevents the ant from visiting the visited path and communication behaviour uses pheromones to communicate with each other. Pheromone value is exploited as an indicator to select the optimal path. Routes with high pheromone value will be selected to forward packets to the destination node. However, ACO always leads to local optima problems and has low convergence speed (Song, Sun & Cao, 2010; Yoshikawa & Otani, 2010) that influences the packet loss problem. ACO-GA applies GA to solve these problems by inserting the mutation that can prevent premature convergence and crossover that replaces the old solution in the previous iteration by the new solution of the current iteration. The performance of ACO-GA was compared with the traditional ACO and GA in terms of fitness value. Experimental results show that ACO-GA can route packets in small simulation time and low energy consumption. However, this hybrid algorithm did not consider the throughput, packet loss and energy efficiency factor and, at the same time, did not compare its performance with other hybrid algorithms.

The Hierarchical Cuckoo Search (HCS) algorithm, proposed by Boucetta, Idoudi, and Saidane (2016), aims to maximize the residual energy of sensor nodes and network lifetime. Sensor nodes are divided into cell-based clusters on geographic location. The sensor node with the highest energy in each cell will be selected as the cluster head that is responsible as an agent to transmit the data from cell members to the destination node. Sensor nodes in the cell will exchange their residual energy through a Hello message. After comparing their energy with the other cell members, sensor nodes with the highest residual energy will send a *Ch_request* and *Ch_electing* message to declare leadership. If there are multiple announcements at any one time, the fastest sensor node will be selected as the cluster head. Data that are received from cell members will be forwarded by the cluster head using the CS algorithm. The aim of the cluster head is to determine the potential neighbour cluster head to forward data by using a multi hop technique. The quality of the neighbour cell is calculated by using a fitness function that considers the residual energy of the cluster head and the number of cell members. The neighbour cell that has the highest fitness will be selected to forward data to the destination node. The performance of the HCS was compared with LEACH and M-GEAR in terms of dead nodes and residual energy. Even though HCS performs better on both performance metrics, it remains insufficient to validate its performance. The other important performance metrics, such as latency and throughput, need to be considered by the HCS during experimentation in measuring the time taken to submit packets and the number of packets received by the destination node.

Mohammadi and Noghabi (2016) propose a hybrid SA and TS algorithm called SAT in reducing the energy consumption and average length distance to submit packets from the source node to destination node. In the initial stage, the primary route is constructed by using a typical algorithm that considers the route with the lowest estimated energy consumption. In this stage, the primary route may or may not be the optimal path but will be used as a benchmark to accept or reject new candidates generated by the TS and SA algorithms. Before the first phase starts, by referencing the primary route, the source node is set as the current node and the next node is set as the Tabu node which will not be considered in the routing optimization process.

Then, in the first phase, the TS algorithm is applied to find the best neighbour node by selecting the node with the highest ratio of initial energy and distance to the destination node. The current node is marked as the source node and the next node is marked as the Tabu node. After moving to the Tabu node, the second phase will start where the SA algorithm is used to optimize the subsequent nodes by selecting the nodes with the highest ratio of energy and distance to the destination node, but the next node will not be marked as a Tabu node after it is selected. This phase will repeat until the destination node is discovered. If the estimated energy for the new routing path is less than the primary route, it will overwrite the primary route and will be the new benchmark. The second phase will continue to run until the termination criteria is satisfied which is either all the possible routes are measured or until the initial energy became zero. The routing optimization is performed to each neighbour route to find the most optimal route from source to destination. The experimental results show that SAT performs better in terms of average length distance and energy consumption when compared with a traditional TS algorithm called a TSRA. However, the function of the SA algorithm remains unclear in terms of avoiding local optima in the second phase. Furthermore, the proposed algorithm only focused on the minimal distance and estimated energy consumption between the source and the destination but not on nodes utilization to avoid hotspot problems.

Li et al. (2018) propose FSACO algorithm which combines AFSA and ACO to enhance the routing process in WSN. This hybrid algorithm consists of pseudorandom proportional route selection model in ACO and crowd factor in AFSA in the initial route discovery to find the global optimum solution. Heuristic information from AFSA is used as the initial pheromone value in the probabilistic route discovery scheme to determine the exploration and exploitation of sensor nodes during neighbours searching process. Due to the drawback of ACO where sensor nodes with high pheromone value are always selected by the ant that will lead to the hotspot problem, the crowd factor from AFSA is integrated to represent the congestion degree within sensor nodes radius. This mechanism is used to prevent stagnation during packet submission and at the same time to reduce the local optima problem. The global pheromone update applied by FSACO considers the path length and residual energy of selected sensor nodes. The performance of FSACO was compared with improved routing algorithm based on ACO (IACO), EEABR, and SensorAnt in terms of route setup time, convergence time, energy consumption, energy standard deviation, network lifetime and throughput. Experimental results show that FSACO outperform the other algorithms in all performance metrics. However, the proposed algorithm did not specify the mitigation process when the ants get trapped in local optima and the comparison was only done with single ACO variants.

Gupta (2018) has proposed an Improved Cuckoo Search-based Clustering Algorithm (ICSCA) with the objective of balancing energy consumption among cluster heads in WSNs. This hybrid algorithm combines the CS algorithm with the clustering technique. The fitness value of all available sensor nodes will be calculated and 20% of the highest fitness value will be selected as a cluster head (CH). Subsequently, the host nest and egg which is the cluster head will be initialized. The cost function of each host nest will be evaluated based on the total energy and distance among host nest members. The best nest with the highest cost function will be selected as the best host nest (F_{best}). After this stage, the iterative process will be done in order to select the high-quality nest with the best set of CHs. The new host nest (F_{new}) will be initialized and evaluated in the iterative process where the F_{best} will be replaced by F_{new} if the value of F_{new} is higher than F_{best}. The iterative process will be continued until the stopping condition is met which is *Max_Generation*. The performance of ICSCA was compared with LEACH, E_OEERP, and PSO-ECHS in terms of energy consumption and residual energy. From the experimental results, ICSCA performs best in both performance metrics. However, ICSCA only takes into consideration the energy aspect without considering the other important performance metrics such as throughput and latency that may affect the submission time of packets from source node to destination node.

Table 2.2 summarizes all performance evaluation criteria used by hybrid swarm intelligence routing algorithms discussed in Section 2.5.2. Energy consumption and latency are the most chosen performance metrics by researchers in evaluating the performance of their routing algorithms. Thus, it can be concluded that latency, which measures the submission time of packets from source node to destination node, is important in increasing the throughput value during the routing process. At the same time, energy consumption that measures the energy used by sensor nodes during the routing process is also important in preventing the dead node problem that will affect the network lifetime of WSN systems. Energy consumption and latency also are used as benchmark to evaluate the load balancing factor of routing algorithm (Zhou, Trajcevski, Tamassia, Avci, Khokhar, & Scheuermann, 2017; Yousif, Badlishah, Yaakob & Amir, 2018) where a good load balancing algorithm leads to a lower latency and energy consumption.

Table 2.2

Summary of hybrid swarm intelligence routing algorithms in WSN

Authors	**Performance Evaluation Criteria**													
	Energy Consumption	**Energy Efficiency**	**Residual Energy**	**Latency**	**Throughput**	**Success Rate**	**Packet Loss Rate**	**Network Lifetime**	**Alive Node**	**Dead Node**	**Path Length**	**Standard Deviation**	**Control Overhead**	**Building Time**
Xiu-li et al. (2008)	√			√										
Li & Shi (2013)				√				√						
Tewari & Vaisla (2014)					√			√						
Karthikeyan & Subramani (2014)				√	√		√							
Cai et al. (2015)		√		√		√						√	√	√
Rao & Rani (2015)	√								√					
Kamaei et al. (2015)			√					√						
Cui et al. (2015)	√			√										
Boutekkouk et al. (2015)									√	√				
Boucetta et al. (2016)			√							√				
Mohammadi & Noghabi (2016)	√										√			
Li et al. (2018)	√				√			√				√		√
Gupta (2018)	√		√											

Most of the research works discussed above include a combination of metaheuristics algorithms which consist of evolutionary computing, swarm intelligence, and local search categories. Research by Li et al. (2018), Karthikeyan and Subramani (2014), and Xiu-li et al. (2008) combine two swarm intelligence approaches while research by Cui et al. (2015) combines swarm intelligence and evolutionary computing approaches. There are also algorithms that combine swarm intelligence with the clustering technique such as BeeSensor-C as proposed by Cai et al. (2015), HCS as proposed by Boucetta et al. (2016), and ICSCA as proposed by Gupta (2018). Swarm intelligence has also been combined with the local search as proposed by Tewari and Vaisla (2014) while Mohammadi and Noghabi (2016) proposed a hybrid routing algorithm that combines two local search algorithms.

2.6 Summary

Based on all the research works covered in this chapter, it can be summarized that packet routing is an NP-complete problem because there are no exact algorithms to completely solve the routing problem, either by using a standalone algorithm or hybrid/combined algorithms. Improvements to the routing algorithm are essential to handle all submitted packets from source node to destination node by maximizing the packet delivery rate and optimal sensor nodes utilization. For that reason, metaheuristics algorithms that try to achieve near optimal solutions within reasonable sensor nodes and time are applied. Ant colony system is one of the metaheuristics algorithms that has been solving many optimization problems such as packet routing, scheduling, node localization, and the TSP. Local pheromone update and global pheromone update that are applied in ACS to balance between exploitation and exploration during the searching process are suitable for application in WSNs. These approaches can balance the energy among sensor nodes in order to prolong the network lifetime while increasing the packet delivery success rate and reducing latency. However, in a huge instances problem where the search space is very big, ACS always has insufficient resources to produce a quality solution with minimal computational time. In such a case, the number of ants in the ACS needs to be increased in order to move and explore many arcs and nodes in the system. It is also noted that increasing the number of ants will lead the search process to be slow as each ant will develop its own solution. Thus, it is crucial to control the number of ants so that there are enough to produce good solutions without causing overhead to the WSN system.

The exploration and exploitation concepts that are applied by ACS also have a significant impact on the whole system. As ants in many ant-based algorithms move randomly, any wrong selection will affect the quality of the solution in terms of cost and time. Therefore, the best exploration rate should be determined in order to control the exploration to the new potential solution or exploitation of the optimal solution. This study proposed enhanced ACS and TS algorithms to solve the packet routing problem in WSNs. The TS algorithm that can achieve faster convergence in a reasonable time is suitable to be combined with the ACS algorithm in solving the huge instances problem. Ant colony system in this proposed routing algorithm is also not based on random selection but has optimal balance between exploration and exploitation to achieve the highest result.

CHAPTER 3

RESEARCH FRAMEWORK AND METHODOLOGY

This chapter presents the research framework and methodology that have been used for this research work. Section 3.1 elaborates all the activities in the research framework that were implemented in the research methodology while the comparative measure that indicates the experimental design is presented in the Section 3.2. Lastly, the summary of the chapter is presented in Section 3.3.

Available algorithms that were applied in the WSN were analysed to determine the potential problems and gaps. Deficiencies of available ACO approaches are studied, for example where the ACO algorithm only considers the throughput and energy consumption of each sensor node without considering the load balancing problem. This may lead to hotspot problems in WSN systems which will occur when sensor nodes in such areas are under heavy traffic load. The main reason for this heavy traffic load is when the same sensor node is heavily assigned to forward the majority of the packets to a destination node. Therefore, the energy of these sensor nodes will quickly deplete and the network lifetime of the system will be reduced. The local optima problem also makes an impact on network lifetime when the ant is trapped in a blind alley and cannot continue the searching process. The problems and potential solutions were determined throughout this research work. For example, when the hotspot and local optima problem are clearly identified from previous algorithms, several potential approaches are also identified to overcome these problems. The hybrid approach between ACO and other algorithms such as TS was proposed as a potential solution to reduce the local optima problem.

The way EACS(TS) improve the routing technique in WSNs is discussed in detail in the research framework presented in Section 3.1. The modification of the ACO approach in terms of the nodes selection phase, local search phase, exploration control phase, and exploitation control phase were determined as subjects to be researched to reduce the attraction to a single solution. These four phases provide significant impact to an ant's decision-making process to organize the attraction of ants towards exploration of a new routing path or exploitation to the previously constructed solution.

3.1 Research Framework

The proposed research framework consists of four phases: node selection, local search, exploration control, and exploitation control. The research framework that depicts the relationship between the framework, method and outcome is presented in Figure 3.1. The research method or scientific method in computer science can be divided into three categories: theoretical, experimental and simulation (Dodig-Crnkovic, 2002). However, one research work may be categorized into one or more of these methodological areas (Moret & Shapio, 2001). The research method used to establish the research framework, also presented in Figure 3.1, focuses on the simulation and experimental categories which aim to improve the existing routing algorithms. Several methods are applied in order to support the research framework to improve the EACS(TS) algorithm. Details of methods used in this research are elaborated in the following subsection.

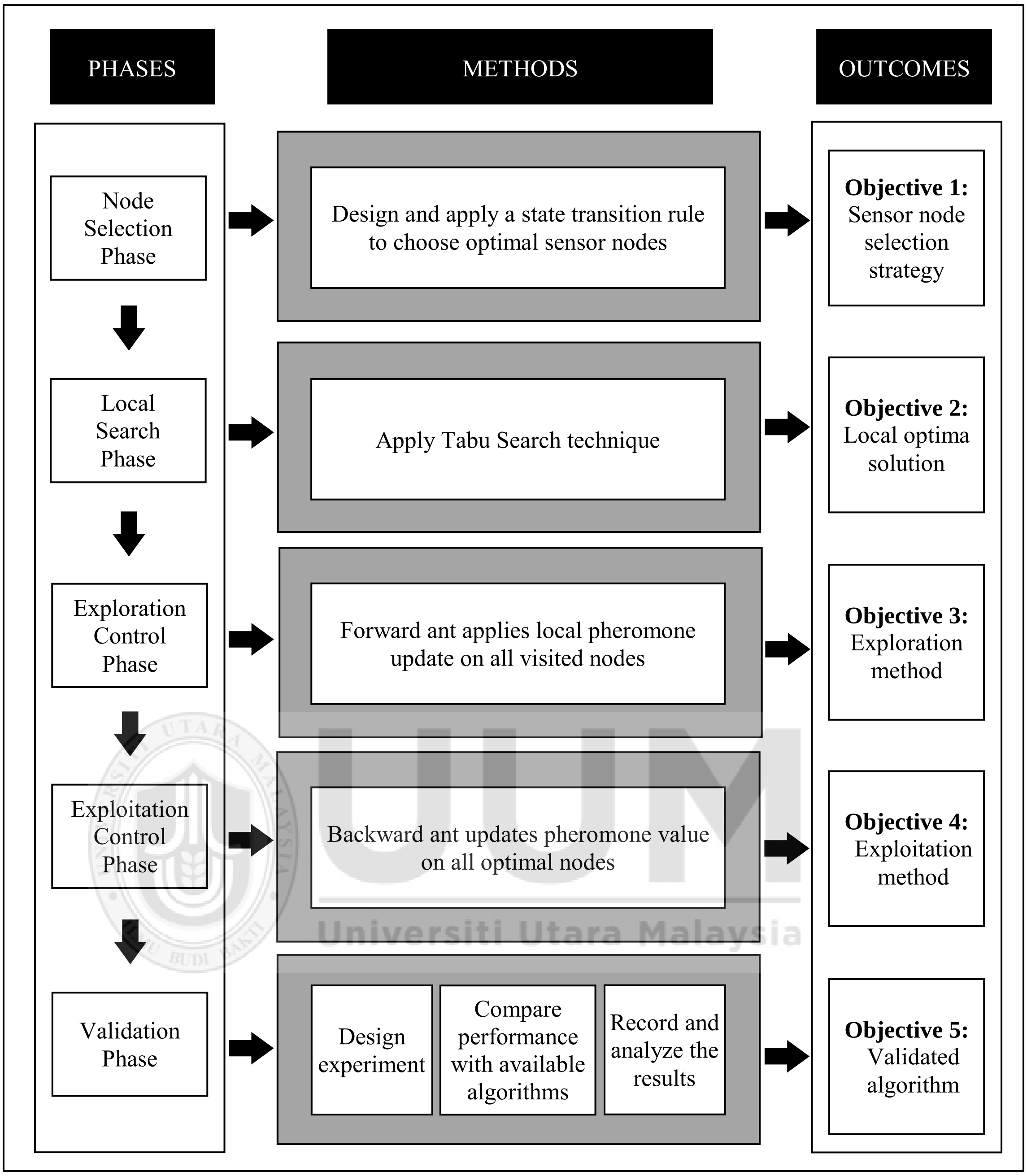


Figure 3.1. Research framework

3.1.1 Node Selection Phase

The node selection phase is an important part in selecting the optimal sensor nodes to forward packets to destination nodes. In this phase, ants will choose either to exploit the previously used sensor node or explore the new sensor node to be used in forwarding packets. The state transition rules, proposed by Dorigo and Gambardella

(1997a), were adopted and adapted in this phase in order to help ants identify the optimal sensor nodes to forward packets to the destination node. The modification has been done by EACS(TS) in terms of heuristics value where the remaining energy of sensor nodes is considered during node selection phase. Ants will decide to exploit the previously used nodes or explore new sensor nodes based on the state transition rule. Ants will move randomly in the WSN environment and the capability of each neighbour node will be evaluated using state transition rules based on their pheromone and heuristics value. Pheromone information is obtained from the routing table of the ants' previous experience while heuristics information is a priori information of the goodness of a solution. This modification ensures that the energy efficiency of sensor nodes can be increased during routing process where sensor nodes with higher remaining energy have higher possibilities to be selected as compared to the nodes with lower remaining energy. In this research, the state transition rule is equivalent to the fitness function of the proposed algorithm in selecting optimal nodes to ensure the packet routing process is efficient and with high possibility of success. The sensor node selection strategy was produced at the end of this phase.

3.1.2 Local Search Phase

The objective of the local search phase is to prevent the local optima problem during the packet routing process. Local optima problems occur when the forward ant is trapped in a blind alley. The ant has no possibility to reach the destination node when trapped in a blind alley during the searching process. This is because the only available nodes are visited nodes and the ant cannot continue the searching process to the destination node. The EACS(TS) algorithm was developed at this phase where the TS algorithm, which is the local search algorithm, was adopted and adapted from Orojloo and Haghighat (2016) to support EACS(TS) in preventing local optima problems. By using a TS approach, the forward ant must check the status of the next sensor node from the routing table. If the next node is already visited, this node will be captured in the Tabu List and excluded from selection by the next ant. In this situation, the forward ant will return to the previous sensor node and select other potential sensor nodes. And if no other potential sensor nodes are available, the forward ant will continue with further backward movement and repeat the process of identifying other potential sensor nodes that are not visited or not in the Tabu List. The previous TS algorithms store the best known solution and better solutions will overwrite previous best known solutions in the Tabu list whereas in EACS(TS), the known bad solutions will be captured in the Tabu list so that they will not be re-selected.

3.1.3 Exploration Control Phase

The exploration control phase is an important part in balancing load on all sensor nodes. A local pheromone update is applied by the forward ant to control the pheromone value on each visited node. This phase is important in encouraging ants to explore new sensor nodes to prevent hotspot problems in WSNs. Hotspot problems occur where the energy at certain sensor nodes is quickly depleted due to the heavy load. Such problems can be prevented by applying local pheromone updates on each visited node to reduce the pheromone value while encouraging the exploration of other potential sensor nodes. This approach can balance the energy consumption and energy depletion of each sensor node so that they operate at almost similar rates. EACS(TS) modified the traditional local pheromone update formula by considering the initial energy value and remaining energy value of each sensor nodes. By considering both values, the energy level of sensor nodes in the system can be balanced, and is one of the key components in preserving the network lifetime of the whole system. The exploration of new sensor nodes is important to increase the network lifetime and to prevent some sensor nodes from being over utilized and drastically drained of their energy compared to other unutilized sensor nodes.

3.1.4 Exploitation Control Phase

The exploitation control phase is an important part in encouraging the ant in the next iteration to exploit the optimal selected sensor nodes. The global pheromone update is applied on the optimal sensor nodes by the backward ant to increase the pheromone intensity on sensor nodes to be selected by the ant in the next iteration. A forward ant will be transformed into a backward ant once it reaches the destination node and all information from the forward ant will be transferred to the backward ant. The backward ant will move back to the source node by using the same sensor nodes used by the forward ant by referring to the ant memory. During the journey back to the source node, a global pheromone update will be applied by the backward ant in encouraging the ant in the next iteration to exploit the optimal path. EACS(TS) improved the existing global pheromone update by considering the current pheromone value of nodes and number of visited nodes. This modification can increase the exploitation of optimal sensor nodes that leads to reduction of searching time and increases the possibility of successful submission of packets to the destination node. This approach can also reduce the latency and packet loss rate during packet submission.

3.1.5 Validation Phase

A computer simulation was developed to simulate the EACS(TS) algorithm in simulated WSN environments. Experiments were carried out by the EACS(TS) algorithm and the other comparison algorithms by using a Routing Modelling Application Simulation Environment (RMASE) applied as an application in the Probabilistic Wireless Network Simulator (Prowler). This simulator was selected because it is easily embedded into optimization algorithms and simulation parameters are easily adjusted with the input file to simulate real WSN environments. The simulation parameters such as network topology, packet characteristics, sensor node capacity which include source node and destination node were determined at this stage. Details about experimental set up are discussed in Section 3.2.

3.2 Simulation Design and Implementation

The proposed EACS(TS) algorithm was implemented in the RMASE which is applied as an application in Prowler. Prowler is a complete framework that is written and run by using Matlab for simulating WSN environments. The Prowler simulator has been chosen by many researchers (Camilo et al., 2006; Zungeru et al., 2012a; Zungeru et al., 2012b; Despaux, 2015) to simulate and evaluate their research because it offers a simple and fast way to prototype applications with nice visualization capabilities for experimental and comparison purposes. This simulator is also selected because it was designed to be easily embedded into an optimization algorithm where it can incorporate an arbitrary number of nodes on arbitrary dynamic topologies.

Prowler is an event-driven network simulator that consists of a MAC layer model and radio propagation model. The radio propagation model takes into account the strength of transmitted signal from all transmitters at a particular point. A combination of deterministic propagation function and random disturbances will determine the strength of the signal sent from a transmitter to a receiver. From this information, collisions can be detected based on the strength of the signal. Different types of actions and events that queue during simulation processes will be redirected between application layer and MAC layer. Figure 3.2 shows the connection between application layer and MAC layer in the Prowler simulator. Based on the figure, *b_t* and *w_t* refers to back-off time and waiting time, respectively, which is associated with random time delays while *packet_length* stands for transmission time given as bit time units. Details of EACS(TS) implementation using Prowler are discussed in Section 4.4.

Prowler also consists of a topology model and application model that give a chance to the user to set up the simulation environment. Network topology is built according to standard specifications in order to simulate a real WSN environment. The topology model in the Prowler simulator allows a user to specify the topology's components in order to arrange along parallel lines, triangular grids and random networks environment. The specifications to build a network topology such as the size of the system, position of each sensor node, and distance between sensor nodes can be controlled by the topology model.

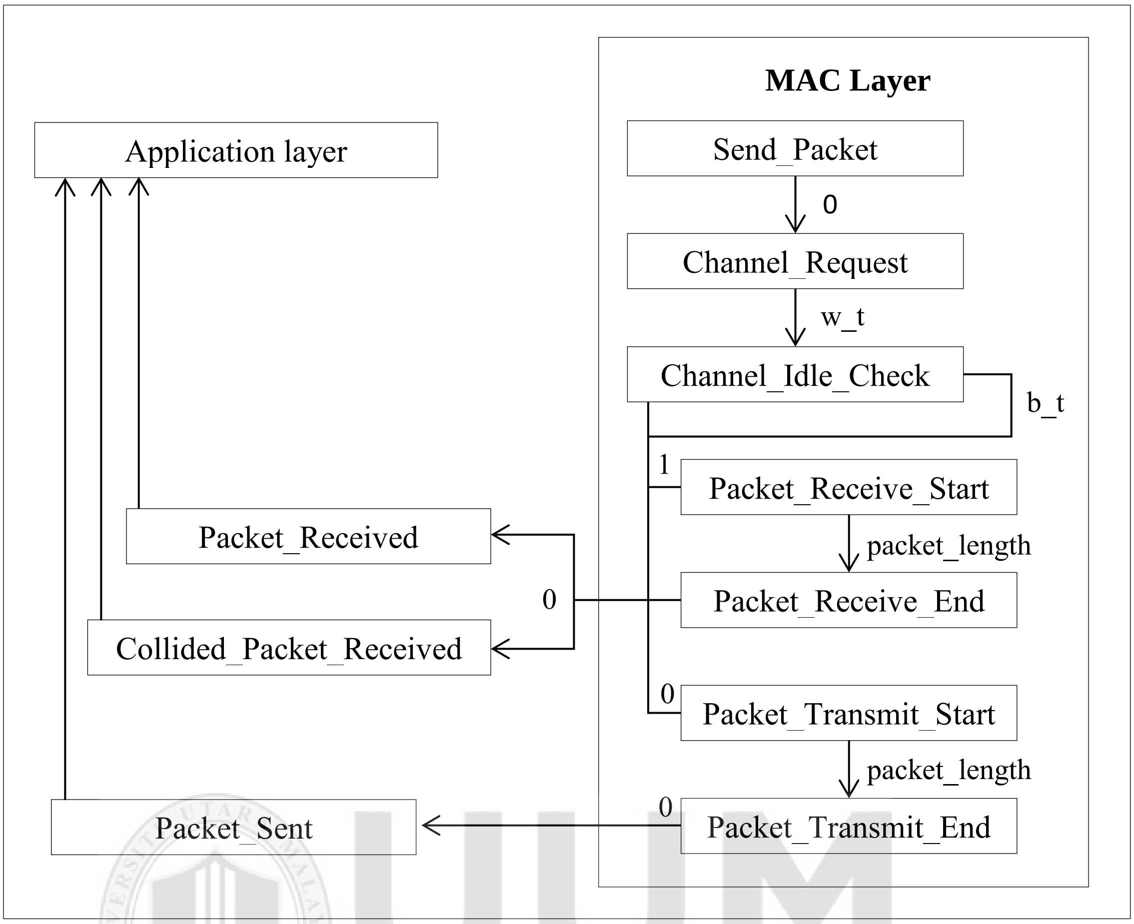


Figure 3.2. Events between application layer and MAC layer

The basic components of a topology model include grid size, grid distance, grid shift, grid density, grid offset, holes and alive rate (failure rate). Distance, density and number of grid points in x and y directions are important parts that need to be considered during topology development to make a submission of packets easily.

The application model is the main part in organizing the characteristics of each sensor node in a WSN system. Routing application scenarios which are peer-to-peer (one-to-one), multicast (one-to-many), and converge cast (many-to-many) can be specified by the application model. Peer-to-peer is a static communication scenario where the source node and the destination node are set to be unique and static during the simulation process. In the multicast scenario, there are multiple destination nodes with a unique source node that contrast with the converge cast scenario that has multiple source nodes with a unique destination node.

The type of source node and destination node can be specified to be static, dynamic or mobile with the configurations of their types, centre, radius, percentage, uniqueness and velocity. Source rate is the measurement of frequencies of events which is the number of packets sent from the source node per second while the destination rate is the measurement of frequencies of queries which is the number of packets requested per second by the destination node. In addition, the number of source packets is defined as total number of packets sent from each sensor node in the WSN system. Details of topology model and application model applied by EACS(TS) are discussed in Section 5.2.

3.3 Comparative Measures

Experiments were conducted to determine how different parametric measurements such as β, ρ, α and q_o will affect the performance of the EACS(TS) algorithm. The best values of these parametric measurements were selected and used in the rest of the experimental process. β is used as a heuristic value in state transition rule while q_o is the value that determines the exploration and exploitation of sensor nodes. Meanwhile, ρ is the coefficient value that controls the pheromone range in the local pheromone update and α is the evaporation rate to be used in global pheromone update.

Experiments were also conducted to test the performance of the EACS(TS) algorithm in routing packets from source node to destination node in WSNs. The purposes of the experiments are to evaluate and compare the performance of EACS(TS) algorithm with the other single and hybrid algorithms in terms of success rate, packet loss rate, throughput, latency, energy consumption and energy efficiency. High success rate and low latency value indicate a good throughput value where a large number of packets can be submitted within the period. At the same time, a good load balancing routing algorithm can be measured by a low latency and energy consumption (Zhou et al., 2017; Yousif et al., 2018).

Most researchers only compared and evaluated their proposed algorithms in one simulation environment (as mentioned in Section 2.5.1 and Section 2.5.2) such as Zungeru (2013) that performed the experiment using 9 sensor nodes and Gupta (2018) that only considered the energy aspects when using energy consumption and residual energy as a performance metrics. At the same time, many hybrid algorithms only compared and evaluated their performances with single algorithm. These situations limit the evaluation of the overall performance of the routing algorithms. In order to solve these problems, several set of experiments have been done to evaluate the performance of EACS(TS) in various simulation environments such as different number of sensor nodes, different size of packets, different simulation time, and different energy level of sensor nodes. The performance of EACS(TS) has also been compared with other single and hybrid swarm intelligence algorithms.

Two sets of experiments were performed to compare the performance of EACS(TS) with the single swarm intelligence approach in WSN. In the first set of experiments, EACS(TS) was compared with traditional swarm intelligence algorithms which are EEABR, BeeSensor and Termite-hill where packets are submitted to the destination node by using different number of sensor nodes in 300 seconds. Meanwhile, the effect of simulation time to the performance of routing algorithms was investigated in the second set of experiments where the performance of each algorithm is captured at 20, 40, 60, 80 and 100 seconds. In this set, the performance of EACS(TS) was compared with IEEABR, EEABR, BeeSensor and Termite-hill.

Four set of experiments were also conducted to compare the performance of EACS(TS) with the hybrid swarm intelligence algorithms in WSN. Various simulation parameters were used in all experiments to observe the effect on the whole system such as the number of sensor nodes, simulation time, energy level and size of submitted packets. The performance of EACS(TS) was compared with other hybrid swarm intelligence approaches such as FSACO, ICSCA, BeeSensor-C, and PSO-C.

The routing process was repeated until all packets were successfully submitted to the destination node or until all sensor nodes had died. The main aim is to show the strength and weakness of the proposed routing algorithm. The results of the other algorithms were taken from the experiments and reliable, published literature for validation purposes. The performance of all tested algorithms was recorded and analysed in order to observe the strengths and weaknesses among algorithms. The experimental results were recorded and transformed into graph form for better understanding and readability.

3.4 Summary

The main objective of this research is to develop an efficient WSN routing algorithm that combines enhanced ACS and TS algorithms. The proposed framework developed contains four main phases, namely, node selection phase, local search phase, exploration control phase, and exploitation control phase that are used to establish the proposed EACS(TS) algorithm.

The node selection phase that aims to produce the sensor node selection strategy is the element in the framework that helps ants to decide either to explore new sensor nodes or exploit previously used sensor nodes in forwarding packets to the destination node. This decision is based on the capabilities of sensor nodes that are calculated by a state transition rule.

The exploration control phase in the research framework helps EACS(TS) in preventing hotspot problems during the routing process. Hotspot problems occur when the load on the system is not equally distributed, leading to certain sensor nodes being under heavy traffic load. This phase is controlled by the local pheromone update that is applied by the forward ant to reduce the pheromone intensity of the visited sensor nodes to encourage exploration to other potential sensor nodes.

The exploitation control phase can increase the exploitation of previously used sensor nodes. This phase is implemented by a backward ant by increasing the pheromone intensity of optimal sensor nodes by applying a global pheromone update. This approach can reduce the searching time of ants in the next iteration while reducing the packet loss rate and delay.

CHAPTER 4

ENHANCED ANT COLONY SYSTEM AND TABU SEARCH ALGORITHM

This chapter presents the proposed enhanced ACS with TS algorithm, which is called EACS(TS). The optimal routing path that uses less time and energy is the main consideration of EACS(TS) during packet submission. Balancing the distribution of sensor nodes is also considered in the proposed algorithm to prevent hotspot problems during the routing process. Section 4.1 discusses in detail the enhancement of ACS called EACS in terms of node selection strategy and pheromone update technique. The objectives of EACS are to increase the throughput and energy efficiency of sensor nodes. Tabu search implementation and how TS works in preventing the forward ant from getting trapped in a blind alley during the node selection strategy and, at the same time, reducing the delay and packet loss problem is discussed in Section 4.2. Section 4.3 describes the details of the proposed EACS(TS) approach that combines EACS as the main algorithm and the TS algorithm as the subordinated algorithm in searching the optimal path in WSNs. The design and implementation of EACS(TS) are discussed in Section 4.4 while the summary of this chapter is presented in Section 4.5.

4.1 Ant Colony System Implementation

In this section, the enhancement of ACS algorithm in WSNs, namely EACS, is discussed. EACS is based on the ACS proposed by Dorigo and Gambardella (1997a, 1997b) as the first variation of AS to improve the performance of ant routing in TSP and some other problems. EACS uses the concept of ACS in node selection strategy which includes exploration and exploitation based on state transition rules. The calculation to update the pheromone value of the selected path in the proposed algorithm is based on the combination of local pheromone update and global pheromone update techniques. At the same time, EACS also adopted and adapted the concept used by the EEABR algorithm, proposed by Camilo et al. (2006), in terms of node selection strategy to submit packets from the source node to the destination node in WSNs. EEABR that is based on the AS algorithm uses the concept of the forward ant to find the destination node and backward ant to go back to the source node before submitting packets.

Figure 4.1 depicts an overview of a network used by ants in a WSN system. Sensor nodes are represented by a circle and ants that carry packet information move from one sensor node to another in finding the destination node. At this moment, local optima problem may happen when searching ant gets trapped in a blind alley during node searching process. The proposed algorithm is expected to reduce the local optima problem by preventing ants from visiting previously visited nodes and nodes that do not have active neighbours. Further discussion about local optima is presented in Section 4.2 and illustrated in Figure 4.2, Figure 4.3, and Figure 4.4.

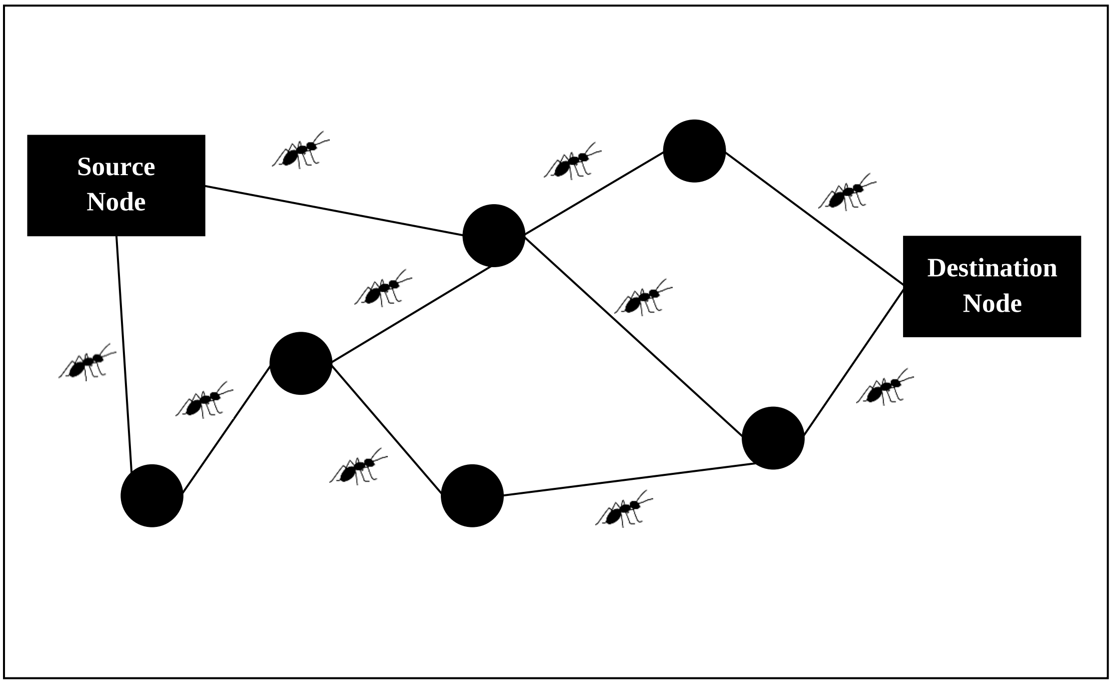


Figure 4.1. Ants in a WSN system

The forward ant is launched from the source node to find an optimal path to the destination node. The information of visited sensor nodes is stored in a memory carried by the ant. The state transition rule adopted and adapted from the ACS algorithm proposed by Dorigo and Gambardella (1997a, 1997b), is used in selecting the next sensor nodes. The modification has been done in terms of heuristics value where EACS(TS) considered the remaining energy of sensor nodes during node selection phase to increase the energy efficiency. Sensor nodes with higher remaining energy have high possibilities to be selected as compared to the nodes with lower remaining energy. State transition rule is calculated using the following equation:

$$P^k{}_{(r,s)} = \begin{cases} argmax\,\{[\tau_{(r,s)}][E_v]^{\beta}\} & if\ q \le q_0 \quad (exploitation) \\ S & otherwise \quad (exploration) \end{cases} \tag{4.1}$$

where $P^k{}_{(r,s)}$ is the probability value of ant k that chooses to move from node r to node s. $\tau_{(r,s)}$ is the pheromone trail of the edge between node r and node s. E_v is the heuristics value given by $\frac{1}{E_r}$, where E_r is the remaining energy of node s. β is an important parameter in controlling the heuristics information in the state transition rule. On the other hand, q is a random variable ranging from 0 to 1, q_o $(0 \leq q_o \leq 1)$ is a parameter to control the possibility of exploration or exploitation and S is a random variable based on the probabilistic decision rule using the following equation:

$$S^k{}_{(r,s)} = \frac{[\tau_{(r,s)}][E_v]^\beta}{\sum[\tau_{(r,s)}][E_v]^\beta} \tag{4.2}$$

The local pheromone update is applied by the forward ant on each visited node before the destination node to reduce the pheromone intensity on this node while encouraging the use of alternative nodes to the destination node. This approach would reduce the hotspot problem by encouraging exploration of other sensor nodes so that the energy depletion happens at an almost similar rate which will eventually increase the network lifetime. If the exploration is not encouraged, some sensor nodes will be over-utilized and drastically drain their energy as compared to other un-utilized sensor nodes. The proposed local pheromone update is defined by:

$$\tau_{(r,s)} = (1-\rho) * \left(\tau_{(r,s)}\right) + \rho(E_{i-}E_{r}) \tag{4.3}$$

where ρ is the coefficient value ($0 \leq \rho \leq 1$) that can control the range of pheromone values, E_i is the initial energy of sensor node, and E_r is the remaining energy of sensor node. The previous local pheromone update formula as stated in Section 2.3

only considers the pheromone value whereas in EACS(TS), the pheromone value and energy level of sensor nodes become the important part in calculating the local pheromone update. Both parameters are used to identify and reduce the pheromone value on over-utilized sensor nodes so that the packets can be distributed fairly throughout the system.

The forward ant will be transformed into a backward ant once it reaches the destination node. The backward ant is responsible for performing a global pheromone update to the optimal path which consists of nodes that it traversed from source node to destination node. This update is done through backward movement to increase the pheromone value so that the path becomes more attractive to following ants. The number of visited nodes and pheromone value are considered in calculating the global pheromone update formula. The global pheromone update is adopted and adapted from Dorigo and Stűtzle (2004) and defined by:

$$\tau_{(r,s)} = (1-\alpha) * \tau_{(r,s)} + \alpha\left(\Delta\tau_{(r,s)}\right) \tag{4.4}$$

where α *(0<α<1)* is the evaporation rate value and $\Delta\tau_{(r,s)}$ is defined by the following formula:

$$\Delta\tau_{(r,s)} = \frac{1}{N_r} \tag{4.5}$$

where N_r is the number of visited nodes from node r until the destination node. As compared to the global pheromone update formula stated in Section 2.3 which only considers the length of the shortest path, EACS(TS) calculates the pheromone based on the number of visited nodes from the destination node to the current node. This approach can update the pheromone value based on the position of each sensor nodes where the nodes near the destination node receive more pheromone value compared to the nodes that are far from the destination node. This will reduce the dead node problem of over-utilized node near the destination node and at the same time encourage the optimal sensor nodes to be selected again in the next iteration.

4.2 Tabu Search Implementation

The concept of the TS algorithm in this proposed algorithm is adopted and adapted from the research by Orojloo and Haghighat (2016) where it is used to store the best known solution and better solutions will overwrite previous best known solutions in the Tabu list whereas in EACS(TS), the known bad solutions will be captured in the Tabu list so that they will not be re-selected. The implementation of the TS algorithm consists of four parts:

a) Initial Solution

In this research, the initial solution in the TS is passed from the bad solution found by the ant in the EACS algorithm to improve the ant routing scheme by avoiding known bad solutions which will eventually improve the performance of this hybrid approach.

b) Objective Function

The objective function of this TS approach is to prevent the ant from getting trapped in a bind alley. There are situations where the forward ant is not able to reach the destination node and the remaining neighbour nodes are already visited (Yoshikawa

& Otani, 2010). To achieve this objective, the ant will put the node with mentioned conditions in the Tabu list and return to the previous node to find the potential neighbour node.

c) Move

Searching the neighbour of the current sensor node in order to identify the potential neighbour node and Tabu node is a critical part in the TS algorithm. This research work applied two techniques to find the next neighbours which are move-backward-insert and move-forward. The move-backward-insert technique is invoked when the current node will lead to local optima, either no potential neighbour nodes or the only available neighbour node is already visited. After detecting these criteria, the ant will perform a backward movement to its previous node and the current node will be inserted into the Tabu list. The expiration function is based on the position where the current Tabu list will be moved adjacently to free up the first position and the last position will be overwritten with the second last position before insertion is performed. On the other hand, the move-forward technique is performed when the current node is not recorded in the Tabu list and has potentially unvisited neighbour nodes. The ant will deposit its pheromone and make a move to the next node. By using these techniques, the path that will not lead to local optima will likely be constructed by the ant and could become the optimal path to the destination node.

d) Neighbourhood Search

The neighbourhood search is performed by the forward ant checking the routing table at the current node to find active and non-visited neighbour nodes. As shown in

Figure 4.2, when the ant reaches Node F, it will check if there are other unvisited neighbour nodes to move forward and only visited Node B is available.

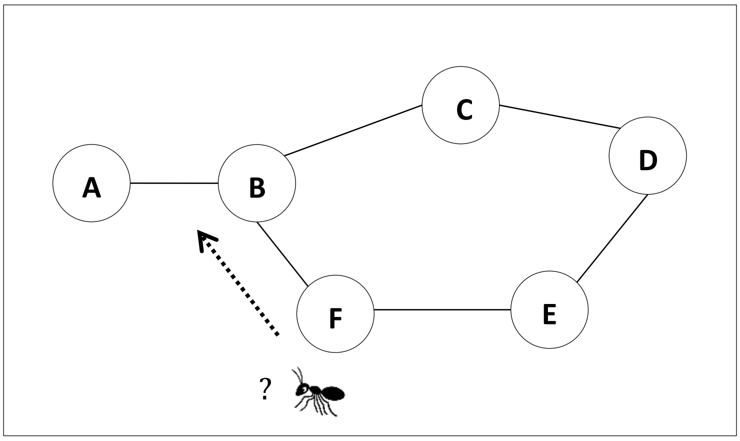


Figure 4.2. Situation where forward ant is trapped in the blind alley

The ant will check if Node B has been visited and when the condition is true, Node F is inserted into the Tabu List and the forward ant will initiate a backward movement to the previous node as shown in Figure 4.3.

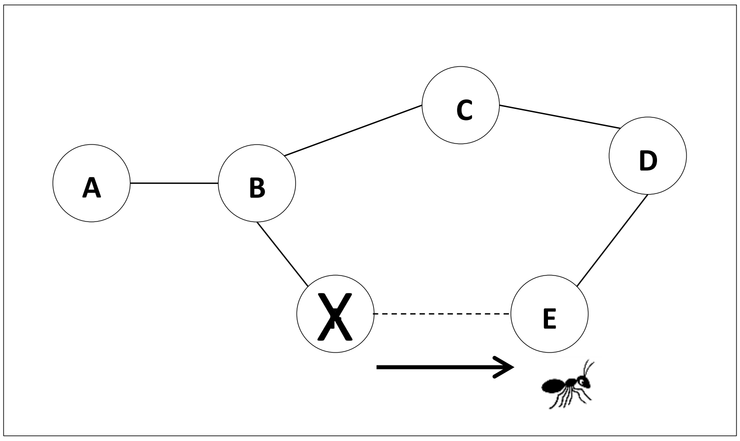


Figure 4.3. Situation where node F is excluded by forward ant from moving destination candidate

As depicted in Figure 4.4, the same process will be repeated by the ant at Node E and when the same condition as Node F is true, Node E is inserted into the Tabu List and backward movement is initiated until the ant can find other unvisited nodes.

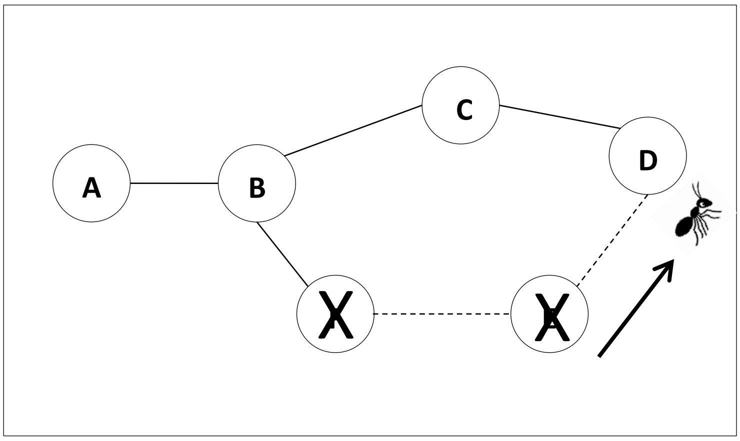


Figure 4.4. Situation where forward ant returns to the previous node

e) Aspiration Criteria

There are two criteria that control the Tabu movement in this proposed algorithm. First, if the movement of the forward ant to the previous sensor node can find the potential neighbour node, then accept the movement. Otherwise, if the previous sensor node has no moving candidate, the neighbour node as well as the previous solution, the ant will put the previous sensor node into the Tabu list and move further to the previous sensor node. These processes will be repeated until the potential neighbour node appears and the destination node is found.

f) Termination Rule

In this study, fixed simulation time is used as the stopping criteria. To make a fair comparison with other algorithms, the simulation parameters are standardized among algorithms.

4.3 Enhanced Ant Colony System and Tabu Search Algorithm

The concept of hybridizing swarm intelligence with local search has been inspired from the research done by Stützle and Hoos (1999) that combined MMAS with local search to improve the performance of traditional MMAS. Although there is no hybrid algorithm that combines ACS and Tabu search in WSN, the concept in hybridizing these two metaheuristics algorithms have been successfully applied in the other research domain such as TSP (Yoshikawa & Otani, 2010), grid computing (Alobaedy, 2015), and cloud computing (Liu, Zhang, Cui, & Li, 2017). In EACS(TS), low level hybridization is applied which combines ACS with TS in constructing the optimal path to submit packets from the source node to the destination node. Tabu search that is based on a systematic process (Suzuki & Cortes, 2016) is suitable to be combined with ACS in enhancing the exploration mechanism.

In this approach, both algorithms will interchange their inner procedures where ACS is the main algorithm and TS is the subordinated algorithm. At first, ACS will perform exploration or exploitation to construct the path, followed by the local pheromone update on each visited node. During the routing process, ACS will initiate the TS where the bad solution produced by ACS is captured in the Tabu list which will be used by TS to reduce the possibility of ACS to explore the known bad solutions. By doing this, the energy will be well utilized by giving opportunity for the next ants to explore potential good solutions rather than trying to leverage known bad solutions. In this case, good solutions from ACS will be complemented with avoidance of known bad solutions captured by TS to preserve the energy as well as ensure the next ant exploration has higher possibility of constructing a good or better solution. Once the good solution is constructed, ACS will perform the global pheromone update using backward movement to every node within the path until the source node. The updated pheromone in each node will be referenced by the next ants. Figure 4.5 represents the flow chart while Figure 4.6 represents the pseudocode of the proposed EACS(TS) algorithm.

In Figure 4.5, the forward ant is created at the source node and forwarded as an agent to find an optimal path from source node to destination node. The task for the forward ant is to move from one sensor node to another with the aim of discovering the best sensor node to be assigned to forward the packet with minimal possibility of packet loss. The information of every visited node is saved in the ant's memory throughout the journey. Before the forward ant moves from the source node, the routing table will be initialized, and the Tabu List updated by the previous ant will be loaded into the ant's memory. The condition of each neighbour node will be evaluated by the ant using information in the routing table.

The ant will choose the next best node based on the pheromone and heuristics value using the state transition rule (refer to Equation (4.1) and Equation (4.2)) with reference to the routing table in each sensor node. The neighbour node with the highest probability value will be selected in the exploitation rule while random neighbour nodes will be selected in the exploration rule. When the node is selected, the ant will check if the node has been visited or exists in the Tabu List. If any such condition is true, the ant will repeat the node selection step using the state transition rule until the best node is found. If the current node is not a destination node, the ant will check if the node has an active neighbour in the routing table. If no active neighbour is found, the node will be inserted into the Tabu list and the ant will initiate a backward movement to the previous node to repeat the node selection step. Then, the ant will make a move to the selected node and perform local pheromone update using Equation (4.3) to the node. The purpose of the local pheromone update is to make the visited sensor node less attractive to the following ants as well as encourage the exploration of other sensor nodes. This approach can prevent the hotspot problem on certain sensor nodes and increase the network's lifetime.

As the ant tries to construct the path to the destination, each visited node will be saved in its memory until the destination node is found. Once found, the forward ant will be transformed into the backward ant which will back trace the path constructed by the forward ant until the source node. The backward ant is responsible to update the pheromone of each node visited by the forward ant. Global pheromone update is performed using Equation (4.4) to all previously traversed nodes in the optimal path to increase pheromone value so that the path becomes attractive to following ants. The backward ant will carry the Tabu list and routing path which will be stored at the source node before being terminated. In this last stage, the path from source node to destination node should have been constructed and the data packet can be forwarded via the optimal path constructed. At regular intervals, the next initiated ant will refer to the Tabu list and routing table updated by the previous ant to either exploit the current optimal path or explore alternative paths if the current optimal path is no longer active due to a dead node or low remaining energy.

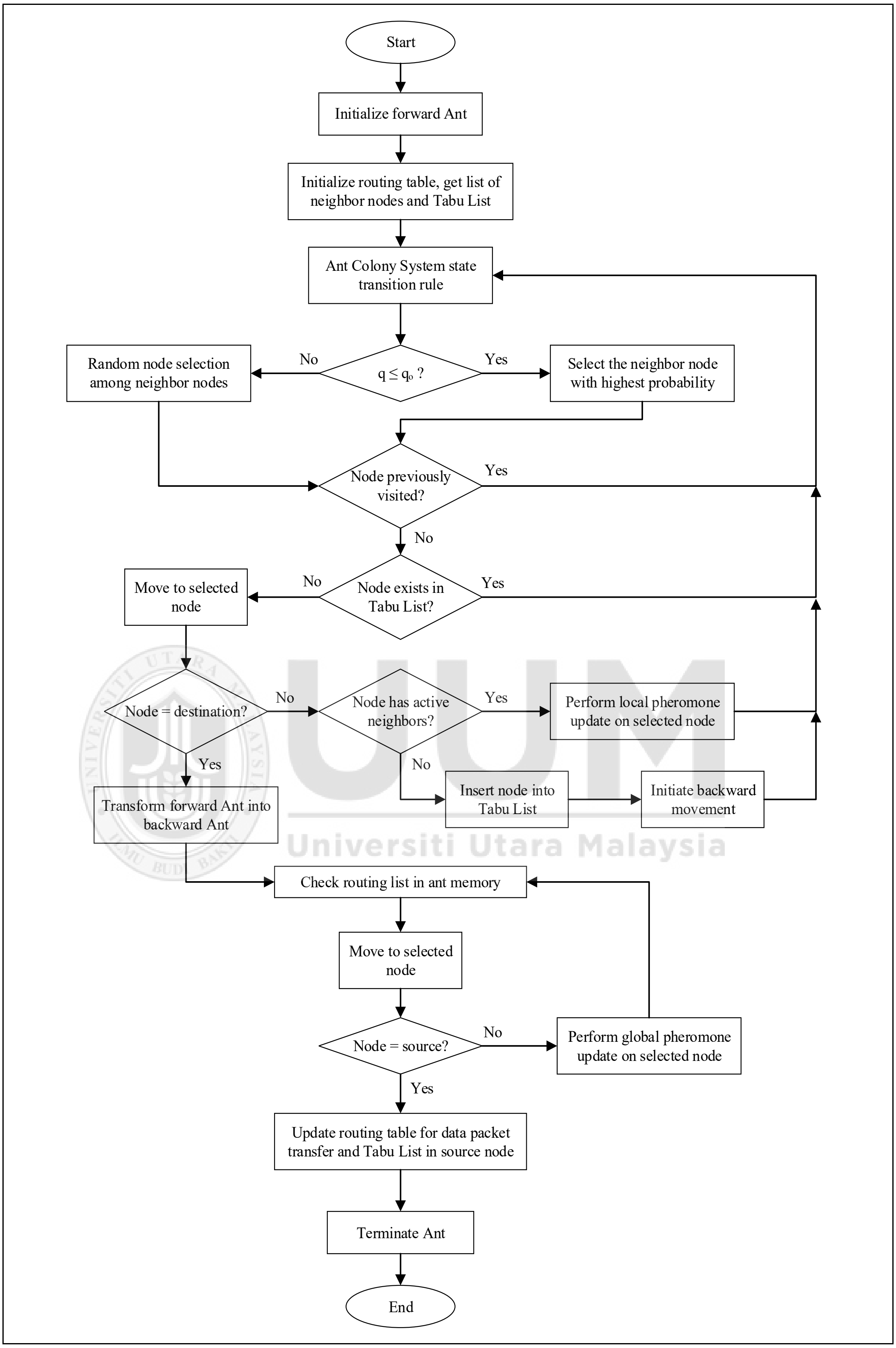


Figure 4.5. Flowchart of EACS(TS) algorithm

```
Initialize forward ant;
Initialize routing table;
Get list of neighbour nodes and Tabu list;
Apply state transition rule;
IF (q ≤ q0)
        THEN Select the neighbour node with highest probability;
        ELSE Random node selection among neighbour nodes;
ENDIF;
IF (Node previously visited)
        THEN Find another node;
        ELSE
        IF (Node exists in Tabu list)
                THEN Find another node;
                ELSE Move to selected node;
        ENDIF;
ENDIF;
IF (Node = Destination)
        THEN Transform forward ant into backward ant;
        ELSE
        IF (Node has active neighbors)
                THEN Perform local pheromone update on selected node;
                ELSE Insert node into Tabu list;
                Initiate backward movement;
        ENDIF;
ENDIF;
Check routing list in ant memory;
Move to selected node;
IF (Node = source)
        THEN Update routing table for data packet transfer and Tabu list in source node;
        Terminate ant;
        ELSE Perform global pheromone update on selected node;
ENDIF;
```

Figure 4.6. Pseudocode of EACS(TS) algorithm

EACS(TS) differs from other hybridization algorithms (Yoshikawa & Otani, 2010; Liu et al., 2017) that only used Tabu list during node searching process where EACS(TS) used the Tabu list to store bad sensor nodes and routing table to store the pheromone value. Both elements are important as reference to the ant in the next iteration to prevent the local optima problem and to reduce the latency during node selection phase. EACS(TS) also proposed the expiration function to update the Tabu list by freeing up the oldest sensor nodes where the newest sensor nodes are added to the Tabu list. In addition to that, the parameters to calculate the state transition rule and pheromone update formula are also different. Research by Yoshikawa and Otani (2010) used pheromone value and distance while Alobaedy (2015) used pheromone value, number of task and number of machines to calculate these elements. On the other hand, EACS(TS) considered the pheromone value, energy level and path length of visited sensor nodes with the objective to improve the network lifetime by reducing the latency and energy consumption.

4.4 EACS(TS) Evaluation Design and Implementation

In this study, the EACS(TS) algorithm was integrated in the Prowler to simulate and validate the performance of the algorithm in WSN environments. Figure 4.7 illustrates the high-level sequence of the simulation process which shows how the EACS(TS) routing algorithm works. At first, a forward ant is generated at the source node and it will find the list of neighbour nodes at the current node regardless of whether the node is a source node or not. The returned list contains basic information such as distance, unique node identifier, pheromone value, and remaining energy. Based on these information and verification of node existence in Tabu list, the best node will be selected. Once confirmed, local pheromone update will be applied and ant will move to the confirmed node. Once moved, the ant will check whether the new node is a destination node or normal node. However, in the event where the current node does not have active neighbours or the only active neighbour is already visited, backward movement will be initiated as illustrated in Figure 4.7. Each movement will be recorded in ant's memory to guide the backward movement. If it is a normal node, the next node selection process will be repeated. On the other hand, if the node is a destination node, the forward ant will be transformed into backward ant to move back to the source node using the path constructed by the forward ant. During this movement, global pheromone update will be applied to increase the pheromone intensity of nodes along current optimal path. Once the backward ant reaches at the source node, it will save its memory that contains the Tabu list. Finally, data packet will be forwarded from the source to the destination node using optimal path as depicted in Figure 4.8.

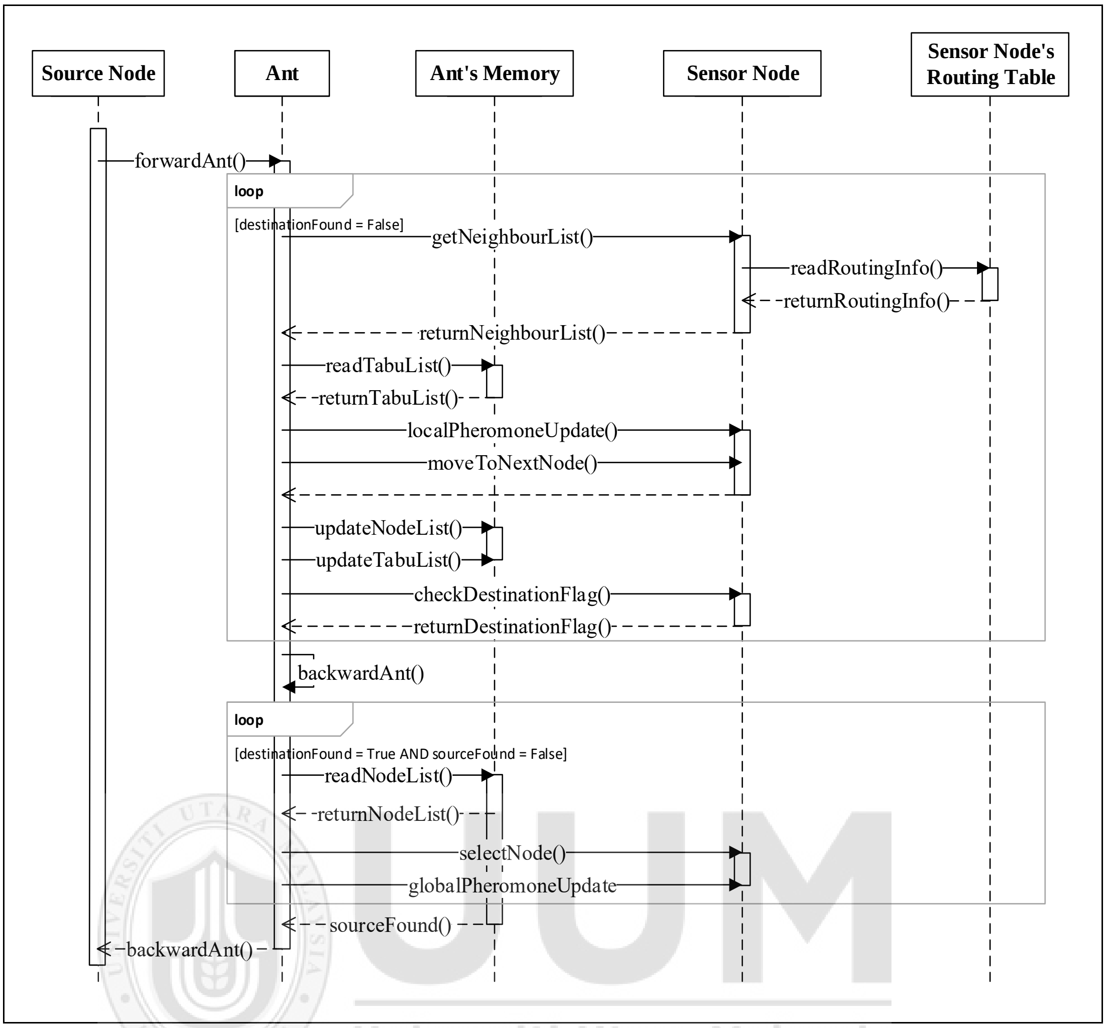


Figure 4.7. High-level workflow of EACS(TS) algorithm in constructing routing path

As illustrated in Figure 4.7, the backward movement by the forward ant is performed only when the current node does not have active neighbours or the only active neighbour is already visited. This process is crucial to avoid the ant from getting trapped in a dead loop. In addition to typical node selection process, the node that meets these criteria will be marked as Tabu node and stored in Tabu list carried by the ant before backward movement is performed. Tabu nodes will not be selected permanently or temporarily during node selection process.

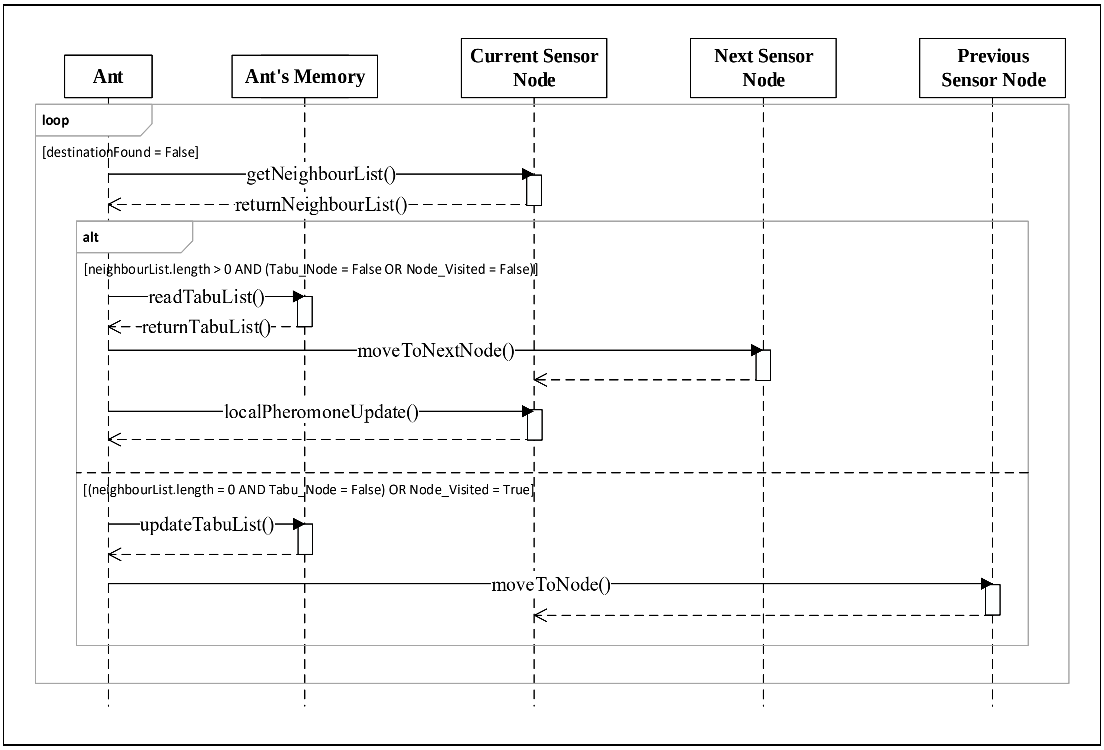


Figure 4.8. High-level workflow of EACS(TS) algorithm in performing backward movement

Data packet will be forwarded as soon as the backward ant reaches at the source node. As shown in Figure 4.8, the source node will refer to its routing table to identify the neighbour node with the highest pheromone updated by the backward ant. Once identified, data packet will be forwarded to the next node and the same process to identify the next optimal node from the routing table will be repeated until data packet reaches at the destination node. Each individual node has its own routing table which is updated by the ant through global and local pheromone update. It is possible that multiple data packets may be forwarded simultaneously in which each individual data packet will follow the same process.

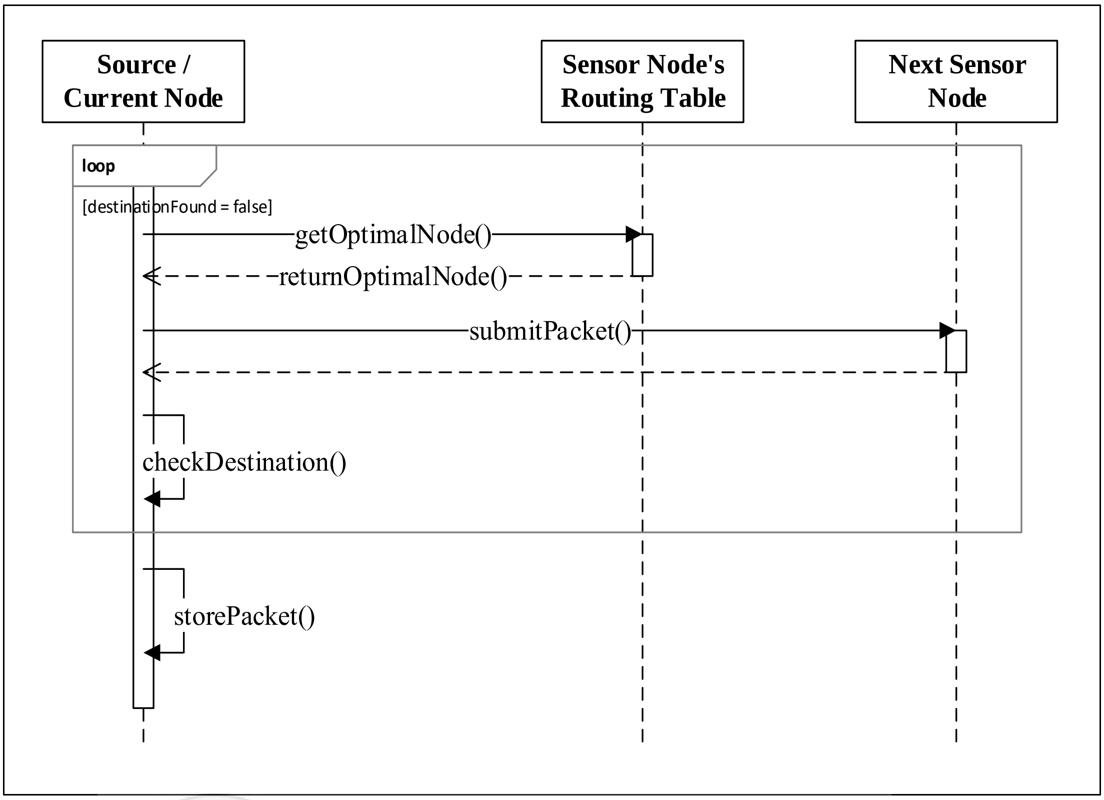


Figure 4.9. High-level workflow of EACS(TS) algorithm in submitting data packet

The detailed steps of the simulation process are as follows:

1. Prowler is responsible to trigger events that consist of simulation parameters and routing layer to the WSN system which will invoke the EACS(TS) layer.
2. EACS(TS) will create a forward ant to explore and find the best sensor nodes in making an optimal routing path in the WSN system.
3. Ants will read a routing table and calculate the state transition rule of sensor node based on the pheromone value and heuristic value.
4. Based on the state transition rule, the next sensor node to be adopted by the ant will be selected either by exploitation or exploration.
5. The sensor node will be selected if it is not captured in the Tabu list and not yet been visited. Otherwise, the forward ant will find another potential sensor node.

6. If the sensor node is not a destination node and has active neighbour nodes, a local pheromone update will be performed to improve the load balancing among potential sensor nodes. Otherwise, this sensor node will be put into the Tabu list and backward movement will be initiated.
7. When the ant has found the destination node, the forward ant will be transformed into a backward ant.
8. The backward ant will move back to the source node by referring to the routing list in the ant's memory and, at the same time, the pheromone value of previously visited sensor nodes in the optimal path will be updated by the global pheromone update.
9. After the backward ant reached at the source node, the routing table and Tabu list at the source node will be updated.
10. EACS(TS) will initiate the packet based on the path constructed and saved by the backward ant in the routing table at the source node.
11. Routing results will be sent back to the Prowler for compilation and output.

EACS(TS) performed based on the low hybridization approach between EACS and TS is a new variation of the ant-based routing algorithm in WSNs. EACS acts as the main algorithm while TS is the subordinated algorithm in this hybrid proposed algorithm. The aims of EACS(TS) are to reduce the energy consumption of sensor nodes, minimize the submission time of packets to destination nodes, increase the packet delivery success rate and, at the same time, avoid local optima during the routing process.

4.5 Summary

Routing packets in WSNs are complicated due to the heterogeneous nature and distribution of sensor nodes. An EACS(TS), that is a hybridization of the ACS and TS algorithm, is proposed to improve the routing path construction by increasing the possibility of the optimal sensor nodes to be used in transmitting the packets from the source node to the destination node with the aim of reducing packet loss rate, latency, and energy consumption. There are two types of ant in EACS(TS), the forward ant that is responsible to find the path from the source node to the destination node, and the backward ant that performs backward movement from the destination node to the source node while increasing the pheromone intensity of the path. EACS(TS) applies a state transition rule in the node selection strategy which includes exploration (randomly explore potential neighbour nodes) and exploitation (leverage neighbour node with the highest probability influenced by the pheromone and visibility). Pheromone and visibility of neighbour nodes are stored in the routing table at each node which is updated by the ant based on information broadcasted by the neighbour nodes.

TS is applied to prevent local optima problems during the path construction process. Nodes with no potential neighbour node will be put into the Tabu list which is used by the ant to avoid becoming trapped in a blind alley. Local pheromone update is applied by the forward ant to each visited node to reduce the pheromone intensity on the node to encourage the exploration of other potential sensor nodes. The forward ant will then be transformed into a backward ant once it reaches the destination node. The backward ant will move back to the source node by referring to the list of traversed nodes in the ant's memory. Global pheromone update is applied by the backward ant to all previously traversed nodes within the optimal path to increase the pheromone intensity so that the path becomes attractive to the following ants in the next iteration.

CHAPTER 5

EXPERIMENTAL RESULTS

This chapter presents the experimental results of the proposed EACS(TS) algorithm with various controlled parameters on a WSN environment. Section 5.1 discusses the performance evaluation criteria that are used to evaluate the performance of the proposed algorithm and other benchmark algorithms while the experimental design is presented in Section 5.2. Section 5.3 covers the parameter tuning experiment to identify the optimal values for β, ρ, α and q_o to be used by EACS(TS). Experimental results and analysis are presented in Section 5.4 that consists of the swarm intelligence approaches and hybrid approaches in WSNs. Finally, the summary of the chapter is concluded in Section 5.5.

5.1 Performance Evaluation Criteria

The performance evaluation criteria are the main elements in evaluating the performance of the proposed algorithm. Minimizing the utilization of energy on each sensor node and reducing the time taken to submit packets from source node to destination node are the main objectives of WSN routing algorithms. A good routing algorithm also considers the packet loss and throughput value of submitted packets. Performance evaluation criteria used for the whole experiments comprise of success rate, packet loss rate, throughput, latency, energy consumption, and energy efficiency. The purposes of applying different measures in the different experiments are to evaluate the stability of EACS(TS) in various conditions and at the same time compare its performance with other swarm intelligence algorithms.

Success rate is measured by the number of successful data packets received by destination node per number of packets sent from the source node. Success rate is measured based on equation 5.1.

$$\text{Success Rate} = \frac{Number\ of\ \ successful\ received\ packets}{Number\ of\ submitted\ packets} \tag{5.1}$$

Packet loss rate is calculated based on the number of unsuccessful received packets per number of submitted packets by source node as shown in Equation 5.2.

$$\text{Packet Loss Rate} = \frac{Number\ of\ unsuccessful\ received\ packets}{Number\ of\ submitted\ packets} \tag{5.2}$$

Throughput measures the number of data packets or messages that successfully pass through from the source node to the destination node at a specific unit of time. It is an important performance metric to show how fast the packets can flow in the WSN system. In the experiments, the throughput is measured by the number of packets successfully passed to the destination node per second of simulation time.

Latency is the time taken to send a message from source node to destination node where it can be measured per each individual destination node or per the whole WSN system. Length of transmission queues, number of hops and random delays at the MAC layer are considered in measuring the latency value. A good load balancing routing algorithm can be measured by a low latency and low energy consumption during packet transmission (Zhou et al., 2017; Yousif et al., 2018).

Energy consumption is measured by the total used energy of all sensor nodes while transmitting and processing submitted packets from the source node to the destination node (Duarte-Melo and Liu, 2002). The energy used for communication which includes idling, transmitting and receiving packets is used in measuring the energy consumption. Low level energy consumption during packets transmission can prolong the network lifetime of the whole system.

According to the research done by Zungeru et al. (2012a), energy efficiency is the ratio of total number of successful packets received at the destination node per total energy used in the whole system. In this formula, the higher the value of energy efficiency, the longer the network lifetime of the system and vice versa. Energy efficiency is measured based on Equation 5.3.

$$\text{Energy Efficiency} = \frac{Number\ of\ successful\ received\ packets}{Total\ energy\ used} \tag{5.3}$$

There is also a different formula to calculate energy efficiency as applied by Saleem et al. (2012) and Cai et al. (2015) where energy efficiency is measured by the energy consumed in submitting 1000 bits data packets to a destination node. By applying this formula, the lower the energy efficiency, the longer network lifetime of the system and vice versa. Since both equations use the same information but differ in terms of the position of numerator and denominator, it can be concluded that the objective of energy efficiency measurement is the same. Both of the energy efficiency formulas were applied in this research work.

5.2 Experimental Design

Experiments were conducted to evaluate the parameters that result in the optimal performance of the proposed EACS(TS) algorithm. The effects of different values of EACS parameter were investigated in Section 5.3 to obtain the best value for β, ρ, α and q_o to be used in the subsequent experiments.

The performance of the EACS(TS) algorithm was compared with the other bio-inspired algorithms comprised of EEABR (Camilo et al., 2006), Termite-hill (Zungeru et al., 2012a) and Tabu Search (Orojloo & Haghighat, 2016). The performance evaluation criteria, as explained in Section 5.1, include success rate, packet loss rate, throughput, latency, energy consumption, and energy efficiency. The basic simulation parameters used in all the experiments are highlighted in Table 5.1.

Table 5.1

Simulation Parameters

Data traffic	**Constant bit rate**
Data rate	250Kbps
Maximum hops	Infinity
Nodes energy	50J
Packet length	512 bits

The characteristics of the source node and destination node are shown in Table 5.2 and Table 5.3.

Table 5.2

Source node characteristics

Source type	**Static**
Centre type	Random
Radius	1
Rate	4
Random rate	0

Table 5.3

Destination node characteristics

Destination type	**Static**
Centre type	Random
Radius	1
Rate	0.5
Random rate	0

Table 5.4 shows the parameter values for the TS algorithm that were applied in this research work.

Table 5.4

TS parameter values

Parameters	**Value**
Search method	Move-backward-insert, move-forward
Tabu size	5

5.3 EACS(TS) Parameter Tuning

A set of experiments was conducted to discover the effects of parameters on the EACS(TS) algorithm. The objective of these experiments is to find the best value for β, ρ, α, and q_o used in the EACS(TS) algorithm before it can be compared with other algorithms. Experiments have been done iteratively by investigating one parameter for each set of experiments. First, the best value for β to be used in the state transition rule is obtained followed by the q_o value that determines the probability of exploration and exploitation of sensor nodes. Then, the best value for ρ to be applied in local pheromone update is investigated followed by the best value for α to be used in global pheromone update. The success rate, throughput, latency, energy consumption, and energy efficiency are evaluated for each controlled parameter. In this section, the Zungeru et al. (2012a) formula was applied in measuring the energy efficiency of sensor nodes in the system.

5.3.1 Effect of β value

The β value is an important element that can influence the heuristics information in the state transition rule. Based on Dorigo and Gambardella (1996), the ideal value of β must be more than 1. Experiments were completed to evaluate the best value of β ranging from 1 to 10 that will give the best success rate, throughput, latency, energy consumption and energy efficiency in EACS(TS). The results of the experiments are shown in Table 5.5 which shows that the best value for β for all performance metrics is 4.

Table 5.5

Effect of β value to the EACS(TS) algorithm

	***β* value**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Packet Received	311	303	312	**333**	285	300	278	298	272	271
Success Rate	0.79	0.77	0.79	**0.84**	0.72	0.76	0.70	0.75	0.69	0.68
Packet Loss Rate	0.21	0.23	0.21	**0.16**	0.28	0.24	0.30	0.25	0.31	0.32
Latency	0.06	0.08	0.06	**0.03**	0.12	0.06	0.09	0.06	0.11	0.06
Throughput	3.14	3.06	3.15	**3.37**	2.88	3.03	2.81	3.01	2.75	2.74
Energy Consumption	22.3	26.10	22.15	**17.73**	34.35	21.13	28.22	20.52	30.61	20.97
Energy Efficiency	13.95	11.61	14.08	**18.78**	8.30	14.20	9.85	14.52	8.89	12.92

The detailed results of Table 5.5 are provided in Appendix I for success rate, Appendix II for throughput, Appendix III for latency, Appendix IV for energy consumption, and Appendix V for energy efficiency. The optimal β produced the highest number of packets received with low energy consumed which will eventually lead to higher energy efficiency to the whole network.

5.3.2 Effect of q_o value

The state transition rule, also known as pseudo-random proportional rule, is the key element in deciding the next sensor node to be selected by the moving ant. The best sensor node with high pheromone value and high energy level can be obtained by the state transition rule. Exploitation of the previous best sensor node occurs when $q \leq q_o$, otherwise the ant will randomly explore the new possible sensor node. q is a random number distributed within 0 and 1, and q_o is a parameter $(0 \leq q_o \leq 1)$.

The value of q_o ranging from 0 to 1 was investigated by a set of experiments. Based on Table 5.6, the best value of q_o to be used by EACS(TS) is 0.5. The results in Table 5.6 are also displayed for all performance metrics which comprise success rate, throughput, latency, energy consumption and energy efficiency in Appendix VI, Appendix VII, Appendix VIII, Appendix IX, and Appendix X, respectively. There is a huge different in value in terms of energy efficiency when q_o is 0.5.

Table 5.6

Effect of q_o value to the EACS(TS) algorithm

	q_o value										
	0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
Packet Received	260	306	283	305	302	**345**	302	296	318	301	310
Success Rate	0.66	0.77	0.71	0.77	0.76	**0.87**	0.76	0.75	0.80	0.76	0.78
Packet Loss Rate	0.34	0.23	0.29	0.23	0.24	**0.13**	0.24	0.25	0.20	0.24	0.22
Latency	0.12	0.06	0.09	0.06	0.06	**0.03**	0.06	0.09	0.06	0.09	0.06
Throughput	2.63	3.09	2.86	3.08	3.05	**3.49**	3.06	2.99	3.22	3.04	3.14
Energy Consumption	32.26	21.92	27.78	21.30	20.74	**14.79**	20.97	28.76	20.89	26.84	21.83
Energy Efficiency	8.06	13.96	10.19	14.32	14.56	**23.32**	14.40	10.29	15.22	11.22	14.20

5.3.3 Effect of ρ value

The aim of local pheromone update that is applied during solution construction is to reduce the attractiveness of the visited sensor nodes to the following ants. This will encourage the exploration of the unvisited nodes to improve the load balancing among sensor nodes. Based on Stützle et al. (2011), the ρ value in the local pheromone update formula is an important element in reducing the pheromone value of sensor nodes where $0 \le \rho \le 1$. A set of experiments was undertaken to investigate the best ρ value to be used by EACS(TS) in the local pheromone update formula.

Based on Table 5.7, the best value for ρ is 0.3 where the results are displayed in Appendix XI for success rate, Appendix XII for throughput, Appendix XIII for latency, Appendix XIV for energy consumption, and Appendix XV for energy efficiency. These results indicate that the best ρ value gives significant improvement in terms of energy efficiency and packet delivery ratio compared to the other ρ values.

Table 5.7

Effect of ρ value to the EACS(TS) algorithm

	ρ value										
	0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
Packet Received	303	307	287	**326**	312	314	308	287	313	310	301
Success Rate	0.77	0.78	0.72	**0.82**	0.79	0.79	0.78	0.72	0.79	0.78	0.76
Packet Loss Rate	0.23	0.22	0.28	**0.18**	0.21	0.21	0.22	0.28	0.21	0.22	0.24
Latency	0.06	0.06	0.09	**0.03**	0.06	0.06	0.06	0.11	0.06	0.09	0.06
Throughput	3.07	3.10	2.90	**3.29**	3.16	3.17	3.11	2.90	3.16	3.13	3.04
Energy Consumption	21.92	21.61	27.29	**16.19**	21.00	21.08	22.39	30.71	20.81	27.13	22.63
Energy Efficiency	13.83	14.21	10.52	**20.13**	14.86	14.90	13.75	9.35	15.04	11.42	13.30

5.3.4 Effect of α value

The aim of the global pheromone update applied by the backward ant is to make sensor nodes within the optimal path more desirable to the following ants. The function of the α value in the global pheromone update is to avoid unlimited accumulation of the pheromone trails on certain areas where $0 < \alpha < 1$. Experiments were completed to investigate the best value of α to be used by EACS(TS) in the following experiments. Table 5.8 indicates that 0.2 is the best value of α in all tested performance metrics. The results in Table 5.8 are presented in Appendix XVI (success rate), Appendix XVII (throughput), Appendix XVIII (latency), Appendix XIX (energy consumption), and Appendix XX (energy efficiency).

Table 5.8

Effect of α value to the EACS(TS) algorithm

	α value								
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Packet Received	299	**345**	317	312	286	252	299	312	286
Success Rate	0.755	**0.871**	0.801	0.788	0.722	0.636	0.755	0.788	0.722
Packet Loss Rate	0.245	**0.129**	0.199	0.212	0.278	0.364	0.245	0.212	0.278
Latency	0.061	**0.031**	0.07	0.062	0.094	0.149	0.067	0.061	0.092
Throughput	3.022	**3.487**	3.203	3.152	2.893	2.552	3.021	3.153	2.89
Energy Consumption	20.41	**14.79**	24.68	22.56	28.03	36.76	24.87	20.64	28.23
Energy Efficiency	14.65	**23.32**	12.84	13.83	10.2	6.855	12.02	15.11	10.13

5.4 Experimental Results of EACS(TS)

This section discusses the performance of EACS(TS) when compared with the other algorithm in terms of success rate, packet loss rate, throughput, latency, energy consumption, and energy efficiency. Experiments consist of several subsections that include manipulation of number of sensor nodes and simulation time. Section 5.4.1 covers detailed comparison among single swarm intelligence algorithms and hybrid algorithms are covered in Section 5.4.2. Even though EACS(TS) is a hybrid algorithm, the comparison has been made with the most significant single swarm intelligence algorithms in WSN such as EEABR, BeeSensor, and Termite-hill. These algorithms has been adopted and adapted by many single and hybrid routing algorithm as a fundamental algorithm to improve the routing process in WSN (Zungeru, 2013; Cai et al., 2015).

5.4.1 Experimental Results on Single Swarm Intelligence Algorithms

In this first set of experiments, the performance of EACS(TS) with different numbers of sensor nodes are compared with EACS (Nasir, Ku-Mahamud, & Kamioka, 2018), EEABR, BeeSensor and Termite-hill based on the parameter specifications described in Table 5.9 which were adopted from Saleem et al. (2012). In this experiment, packets are submitted to the destination node by 49, 64, 81, and 100 sensor nodes in 300 seconds. The objective of this experiment is to determine the effect of the number of sensor nodes in terms of success rate, packet loss rate, latency, and energy efficiency when routing packets in the same simulation time.

Table 5.9

Scheduling parameters to investigate the performance of the algorithms for different numbers of sensor nodes

Parameters	**Values**
Routing algorithm	EACS(TS), EACS, EEABR, BeeSensor, Termite-hill
Number of nodes	49, 64, 81, 100
Nodes energy	50 J
Simulation Time	300 seconds
Performance Metric	Success rate, packet loss rate, latency, energy efficiency

The ultimate objective of most optimization algorithms is to achieve the highest success rate when submitting packets from the source nodes to the destination node. Figure 5.1 shows that EACS(TS) achieved the highest success rate when using 100 sensor nodes and slightly lower than BeeSensor and Termite-hill when using 49, 64, and 81 sensor nodes in submitting packets. EACS which is another ACS variant also achieved good results during experiments. The combination of local pheromone update and global pheromone update in EACS encourages the exploration and exploitation of optimal routing path as compared to EEABR that only applied global pheromone update to increase the exploitation of the previous optimal path. At the same time, EACS does not consider the local optima problem which has a lower success rate as compared to EACS(TS) when routing packets in large size of network. This proved that the absence of Tabu search approach in EACS gives a significant impact to the number of packets received when routing packet in large area. EEABR only considered the attraction to the optimal sensor nodes without taking into consideration the hotspot and local optima problem that has the lowest success rate value compared to the other algorithms. The absence of local pheromone update in EEABR that could balance the packet distribution among sensor nodes gives a huge impact to the success rate value. In addition to representing the performance in the form of success rate, representation using loss rate is also possible, as shown in Figure 5.2.

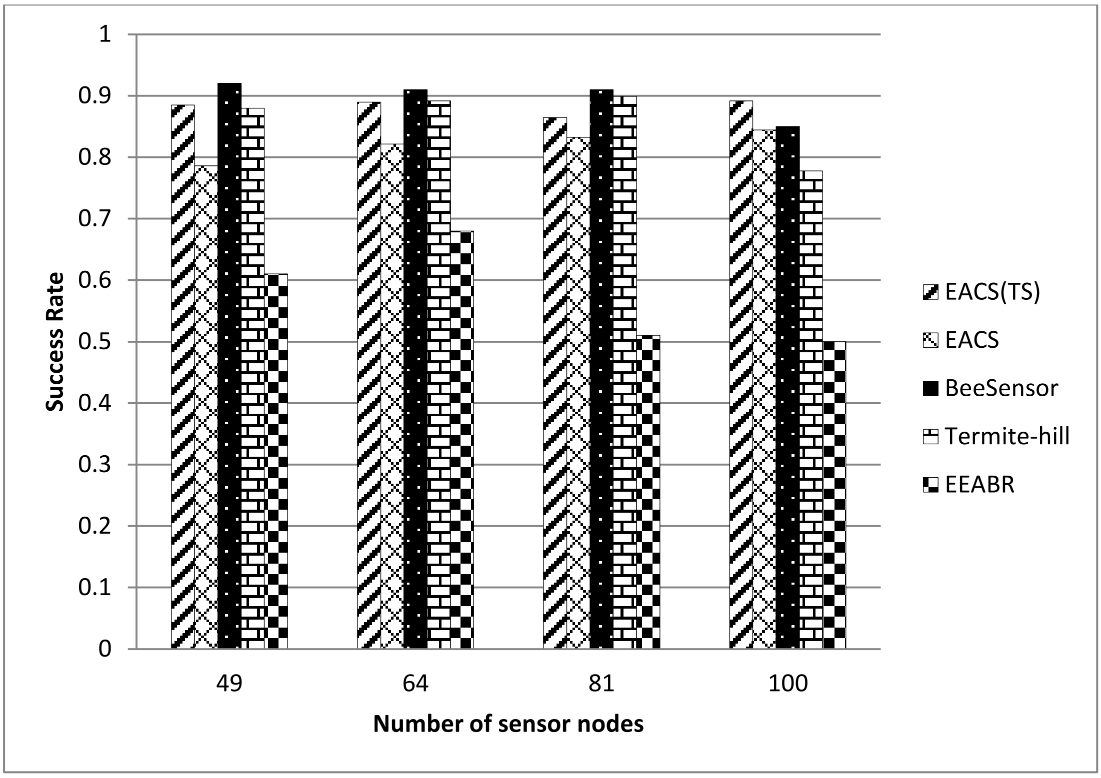


Figure 5.1. Success rate of different algorithms by using different numbers of sensor nodes in 300 seconds

Figure 5.2 shows the packet loss rate that contradicts the results in Figure 5.1 in which EACS(TS) has the lowest packet loss rate as compared to the others when using large numbers of sensor nodes. These results prove that EACS(TS) is suitable to be used in a large sized network. This is expected because in the large network, EACS(TS), that combines techniques in ACS and TS, is meant to avoid the local optima problem. This approach can prevent the ant from getting trapped in a blind alley by storing the node that has no potential neighbour node in the Tabu list. All the nodes stored in the Tabu list are marked as Tabu and cannot be used again in the future routing process. EACS(TS) takes high consideration of other potential neighbour nodes that can increase the possibility of packets' arrival at the destination node while reducing packet loss. On the other hand, EEABR and Termite-hill do not effectively tackle the local optima problem in large sized network.

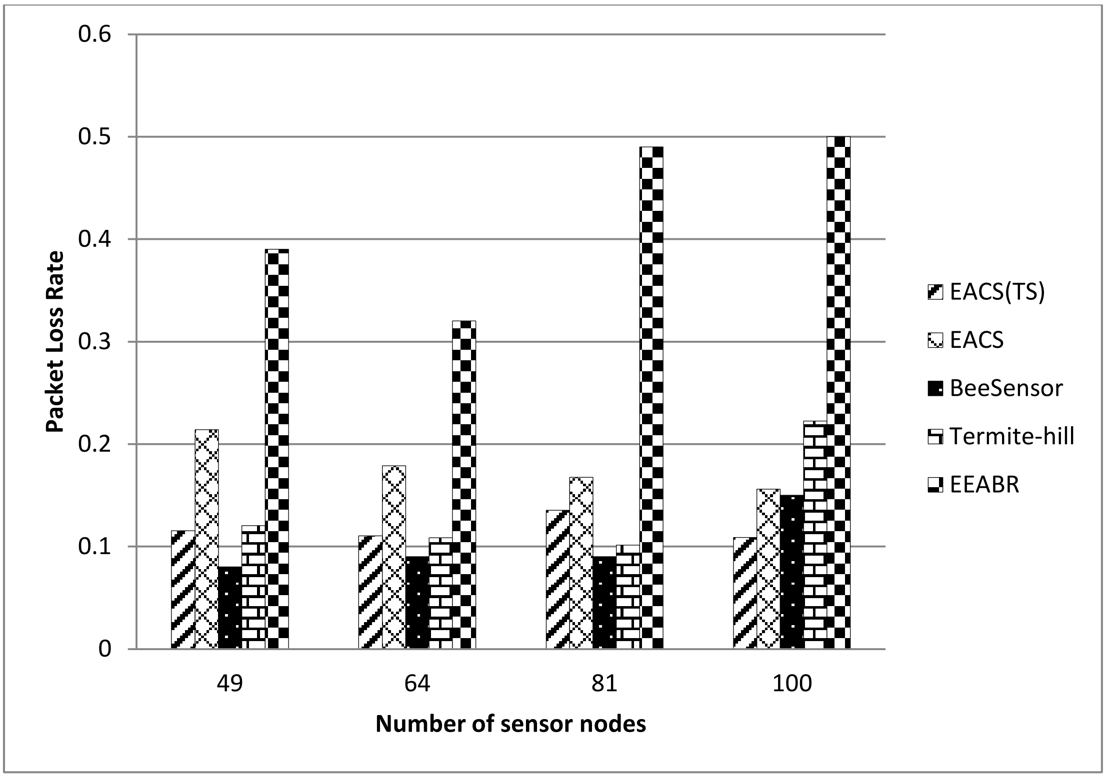


Figure 5.2. Packet loss rate of different algorithms by using different numbers of sensor nodes in 300 seconds

Latency, which is the difference between the packet submission time and arrival time, is one of the key criteria to measure how well the algorithm can reduce stagnation problems. Figure 5.3 shows the comparison of latency between EACS(TS), BeeSensor, Termite-hill, and EEABR. As can be seen in the figure, EACS(TS) has the lowest latency value while Termite-hill has the highest latency for all numbers of sensor nodes. This is anticipated because EACS(TS) reduces the search and submission time by referring to the information stored in the routing table and Tabu list. The routing table stores all the information of the visited sensor nodes from the previous iterations such as node ID, pheromone value and residual energy while Tabu list stores the list of Tabu nodes. Both tables are used by EACS(TS) in reducing the search time whereby EACS(TS) will select the node with the highest pheromone value and residual energy from the routing table and check its existence in the Tabu list. The selection of node is ultimately controlled by the Tabu list regardless of the value of pheromone and residual energy. From these experiments, it can be concluded that even though BeeSensor achieved the second highest success rate, it still needs improvement in terms of time taken to submit the packets. This is because BeeSensor has the second highest latency value, which indicates the submission time of packets to destination node is not as effective as EACS(TS) and EEABR. On the other hand, since Termite-hill does not consider exploration to other potential sensor nodes during routing process, it has the highest latency value in all experiments. The termite in the next iteration will choose previous optimal sensor nodes without exploring to other potential sensor nodes. This will lead to the congestion problem during packet submission. Furthermore, it will consume more time to submit packets from source node to destination node during congestion.

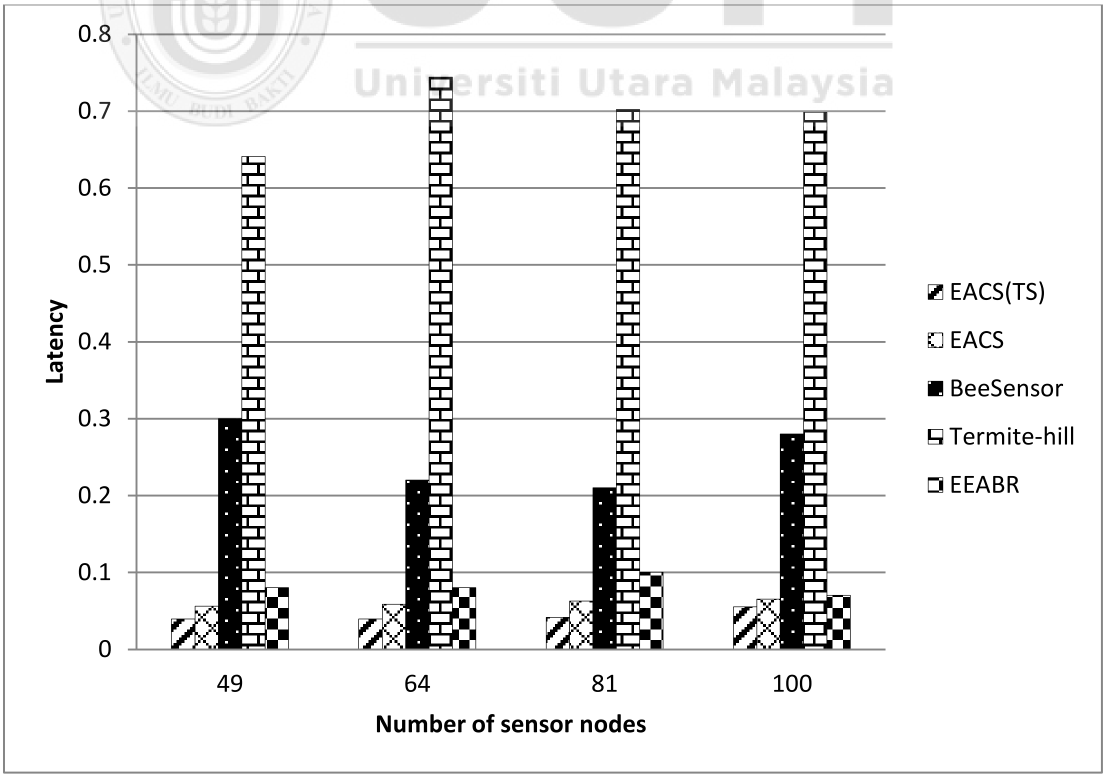


Figure 5.3. Latency of different algorithms by using different numbers of sensor nodes in 300 seconds

Energy efficiency is crucial to measure as it shows which algorithms can achieve the highest network lifetime. Figure 5.4 shows a comparison of energy efficiency between EACS(TS), EEABR, Termite-hill, and BeeSensor calculated based on the formula by Cai et al. (2015). This formula calculates energy efficiency based on the total energy consumed to submit 1000 bits of data to destination node where lower energy efficiency is better than higher energy efficiency. Based on the figure, it clearly shows that EACS(TS) has the lowest energy efficiency while Termite-hill has the highest energy efficiency in all scenarios. These results are expected because EACS(TS) can reduce the energy consumption of each sensor node by fairly distributing packets among potential sensor nodes. Exploitation of optimal sensor nodes from previously constructed routing paths and exploration of new potential sensor nodes is balanced effectively to better preserve the energy consumption in the system. In contrast, Termite-hill, which consumes the most energy during packet submission and has low packet received value, achieved the highest energy efficiency among others, that clearly indicating the least network lifetime.

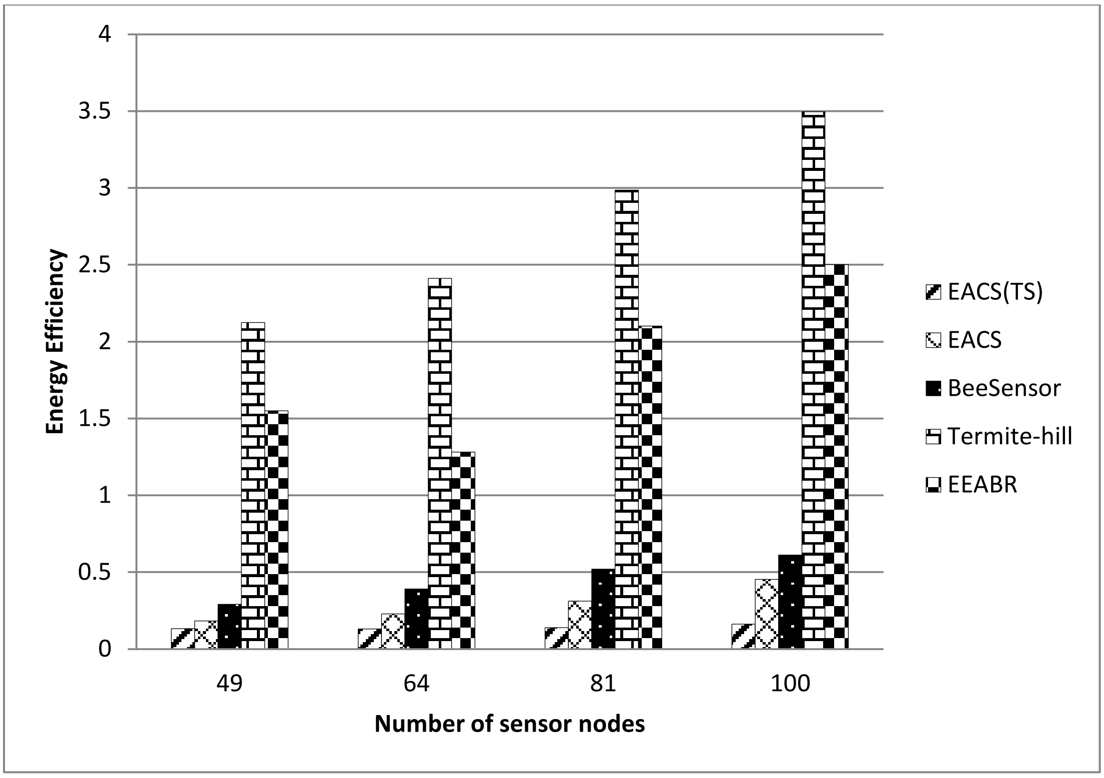


Figure 5.4. Energy efficiency of different algorithms by using different numbers of sensor nodes in 300 seconds

The effect of simulation time to the performance of routing algorithms was investigated in this second set of experiments by adopting the simulation parameters from Zungeru (2013). These experiments were done to study the pattern of each routing algorithms throughout the time when submitting packets using the same number of sensor nodes. The performance of EACS(TS) was compared with IEEABR, EEABR, BeeSensor, and Termite-hill in terms of success rate, throughput, latency, and energy efficiency based on the parameter specifications in Table 5.10. In this experiment, nine sensor nodes are distributed in the system and the performance of each algorithm is captured at 20, 40, 60, 80, and 100 seconds.

Table 5.10

Scheduling parameters to investigate the performance of the algorithms for different simulation time

Parameters	Values
Routing algorithm	EACS(TS), IEEABR, EEABR, BeeSensor, Termite-hill
Number of nodes	9
Nodes energy	30 J
Simulation Time	20, 40, 60, 80, 100 (seconds)
Performance Metric	Success rate, throughput, energy efficiency, latency

Figure 5.5 shows that the increase of simulation time does not give significant affect to the success rate value of all algorithms. Despite the insignificancy, EACS(TS) achieved the highest and almost constant success rate from 20 seconds until the end of simulation at 100 seconds, followed by EEABR. This is expected as EACS(TS) applies a state transition rule and probabilistic decision rule that can balance the exploitation and exploration during the node searching process. These approaches guarantee that node selection is undertaken fairly and effectively to ensure the packet can reach the destination node successfully and eventually increase the throughput value. In addition to that, BeeSensor and Termite-hill suffer from performance degradation beyond 80 seconds due to some heavy loaded nodes starting to die and being incapable of routing the packet any longer. Another significant finding is that EEABR, which is also based on the ant algorithm, have the lowest success rate in the beginning but it increased from 40 seconds until the end of simulation. Since EEABR only applies probabilistic decision rule in selecting sensor nodes without avoiding potential bad nodes, it suffers from higher possibility of packet loss when constructing optimal path in the beginning of execution.

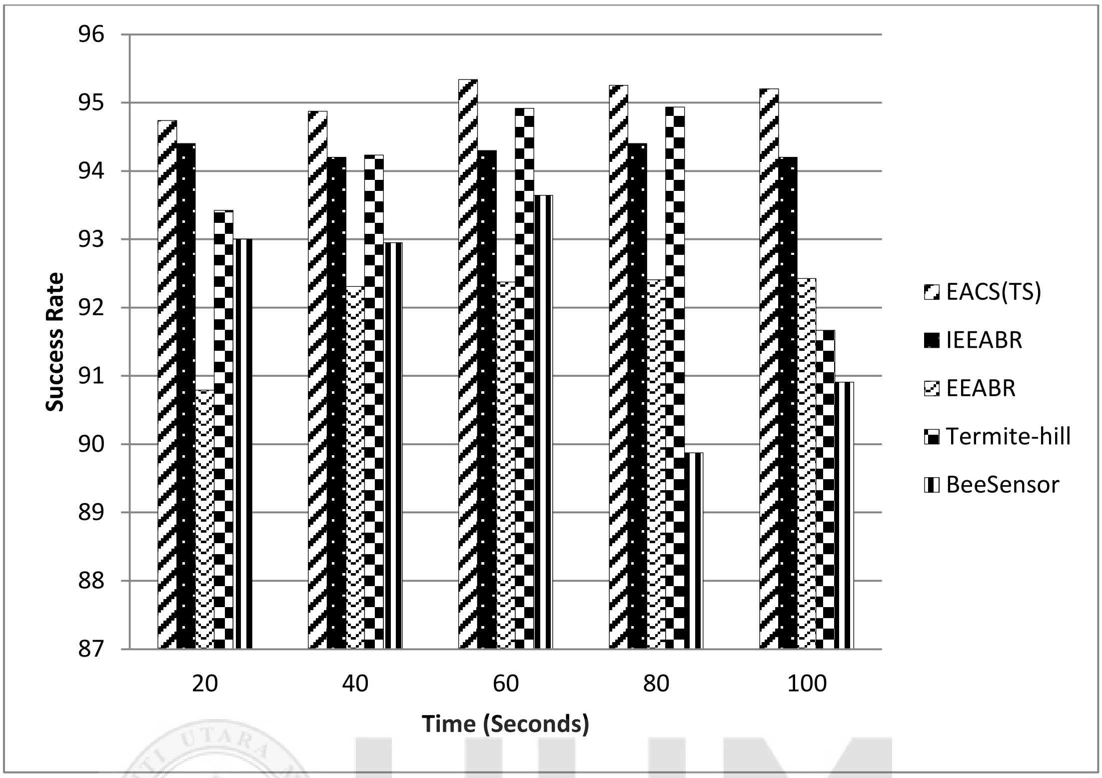


Figure 5.5. Success rate of different algorithms by using nine sensor nodes in 100 seconds

Throughput measures the number of packets successfully transmitted to the destination node per second. As presented in Figure 5.6, the pattern of results is almost identical to the success rate in Figure 5.5 since the denominator in the formulation is the same for both. Thus, it can be concluded that the higher the success rate, the higher the throughput of the system.

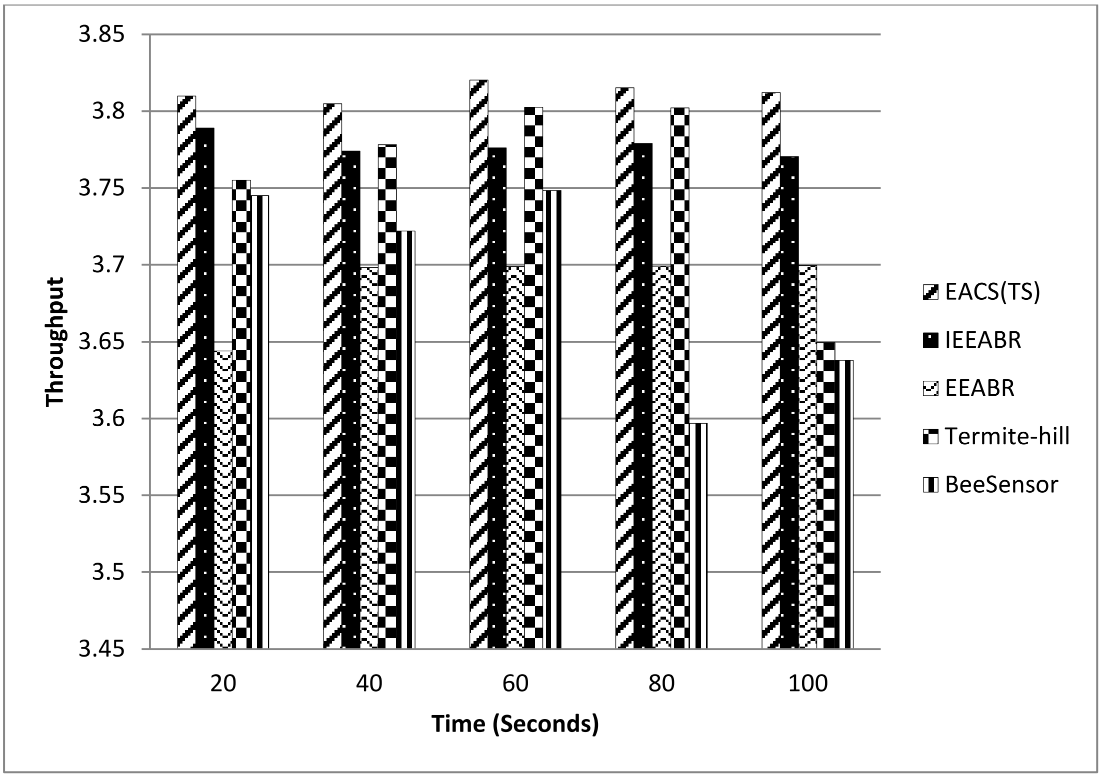


Figure 5.6: Throughput of different algorithms by using nine sensor nodes in 100 seconds

Figure 5.7 depicts the energy efficiency of all algorithms that is calculated by the number of packets received per total energy used (Zungeru et al., 2012a) where the higher energy efficiency is better than the lower energy efficiency. This formula contradicts the formula used by Cai et al. (2015) but both formulas are acceptable to be used to calculate the energy efficiency. EACS(TS), EEABR, and IEEABR have slightly lower energy efficiency in the beginning but become stable after the 20 seconds mark. In contrast, the energy efficiency for Termite-hill and BeeSensor has a significant spike in the beginning but decreases along with time, with some instability. However, these results are not sufficient to prove which algorithm will achieve the highest energy efficiency but can be used to show which algorithm will achieve more stable energy efficiency. As can be seen in Figure 5.5 and Figure 5.6, even though EACS(TS) has a slightly lower energy efficiency than IEEABR throughout the whole time, it still has the highest success rate and throughput value during experiments. This can be related to the energy used by EACS(TS) in submitting a large number of packets throughout the time.

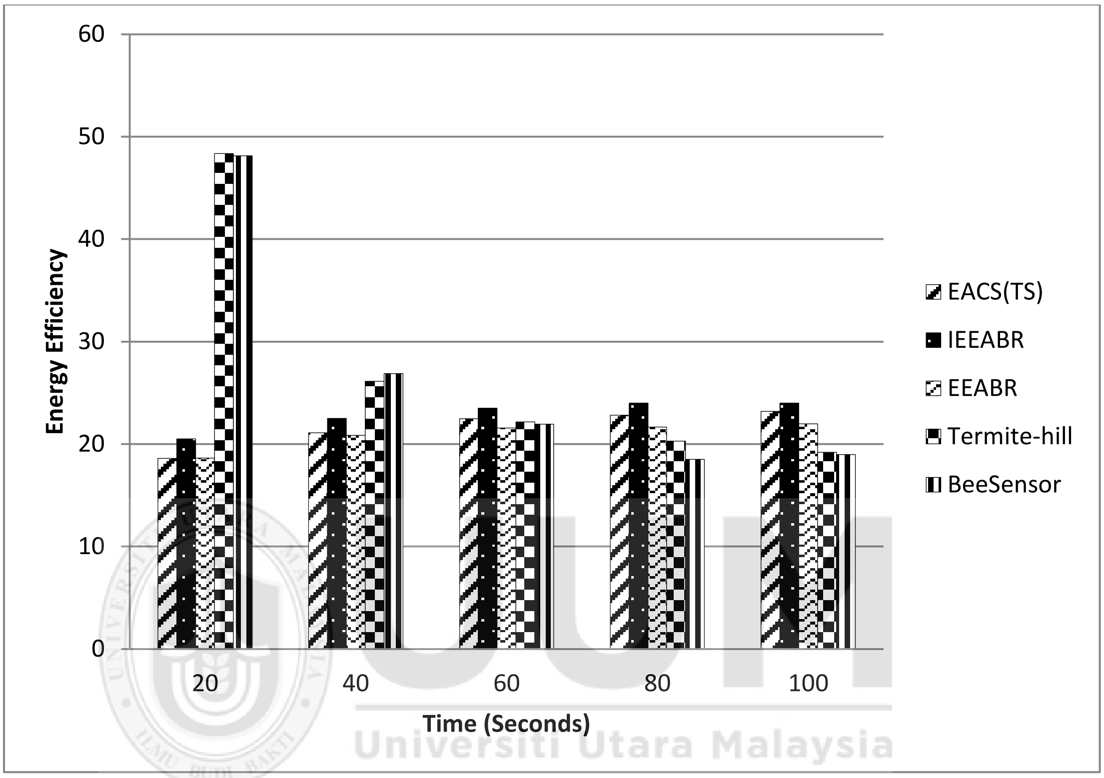


Figure 5.7. Energy efficiency of different algorithms by using nine sensor nodes in 100 seconds

Figure 5.8 shows the latency value of all algorithms when simulated in 20, 40, 60, 80 and 100 seconds. As can be seen, BeeSensor has the highest latency value followed by Termite-hill in all simulation time which indicates that both algorithms require more time to submit packets from source node to the destination node. On the other hand, EACS(TS) has the lowest latency value during the experiment followed by IEEABR and EEABR. These results not only show which algorithm has the lowest latency but, rather, prove that ant-based algorithms which use routing tables to store the visited and optimal path to be used by ants in the next iteration can save significant amounts of time to perform node searching and optimal path construction.

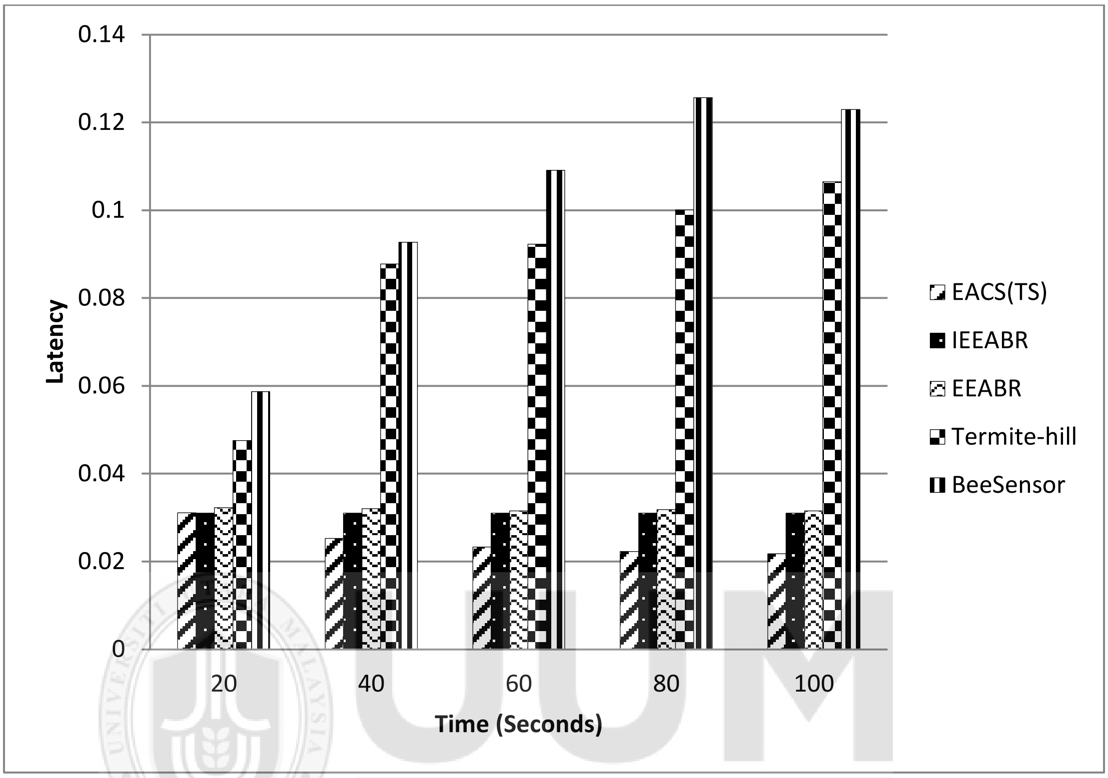


Figure 5.8. Latency of different algorithms by using nine sensor nodes in 100 seconds

5.4.2 Experimental Results of Hybrid Algorithms

The third set of experiments was conducted to evaluate the performance of ACO variants in terms of throughput, energy consumption, and energy efficiency based on the parameter specifications in Table 5.11 which were adopted from Li et al. (2018). These experiments were done by using hybrid ACO algorithms which are EACS(TS) and FSACO, and also single ACO algorithms such as EEABR, IACO and SensorAnt.

Table 5.11

Scheduling parameters to investigate the performance of ACO variant algorithms for different simulation time

Parameters	**Values**
Routing algorithm	EACS(TS), FSACO, EEABR, IACO, SensorAnt
Number of nodes	30, 60, 90, 120, 150, 180, 210, 240, 270, 300
Nodes energy	1000 J
Simulation time	300 seconds
Performance metric	Throughput, energy consumption, energy efficiency

Figure 5.9 depicts the throughput value of all algorithms when routing packets by using 100 sensor nodes in 300 seconds. Throughout the simulation process, EACS(TS) and FSACO which are hybrid based ant routing algorithms achieved higher throughput than EEABR, IACO, and SensorAnt which are single ant based routing algorithms. These results prove that by avoiding local optima, higher throughput can be achieved because data packets can be transferred more quickly with lesser possibility of packet loss.

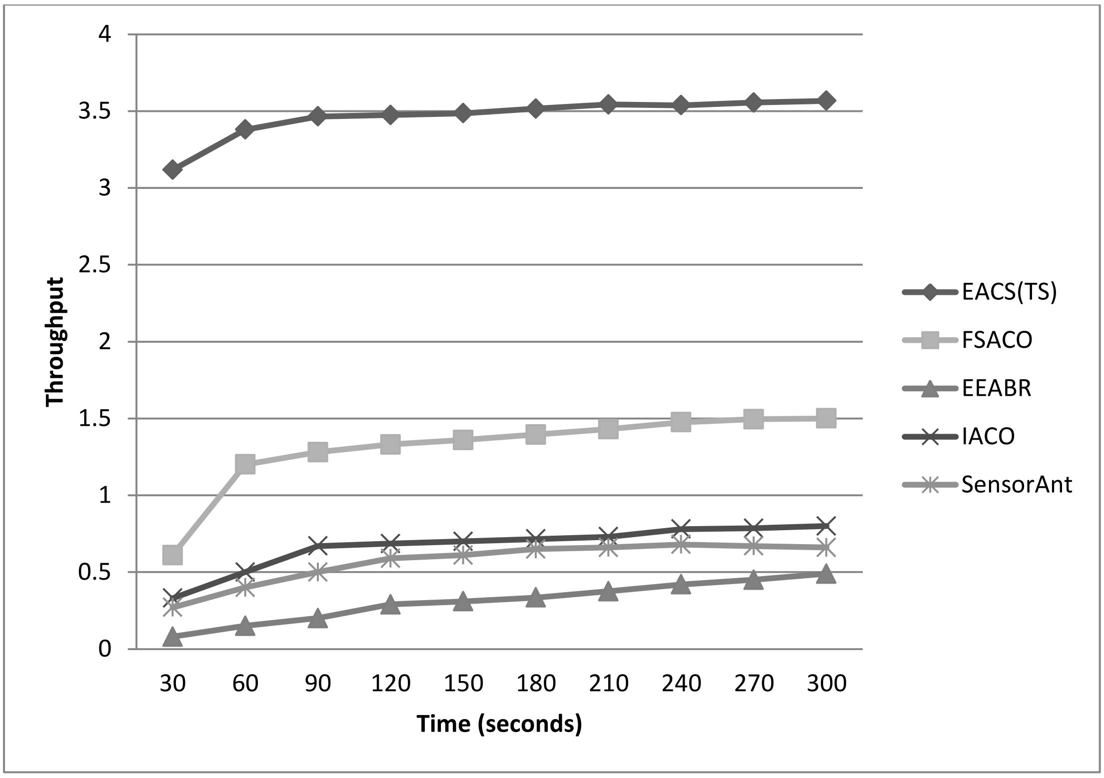


Figure 5.9. Throughput of different ACO variant algorithms by using 100 sensor nodes in 300 seconds

Statistical test was performed to calculate the average and standard deviation of throughput value as presented in Table 5.12. As shown in the table, the average throughput of EACS(TS) is the highest among all algorithms and the standard deviation of EACS(TS) is comparable to EEABR. These results showed that EACS(TS) is stable and consistent during experiments as compared to FSACO that has the second highest throughput but the lowest standard deviation.

Table 5.12

Statistical test for the throughput values of the third experiment

	EACS(TS)	FSACO	EEABR	IACO	SensorAnt
Average	3.464	1.308	0.310	0.670	0.569
Standard Deviation	0.134	0.264	0.133	0.147	0.138

Figure 5.10 shows the energy consumption of EACS(TS), FSACO, EEABR, IACO, and SensorAnt in routing packets in 300 seconds. As illustrated, the energy consumption of all algorithms show a linear growth in response to the increasing of simulation time. However, EACS(TS) and FSACO have slightly lower energy consumption as compared to the other algorithms. Even though the differences are not very significant, these results are in alignment with the throughput as shown in Figure 5.9. In addition to that, by balancing the load using local and global pheromone update, the algorithm

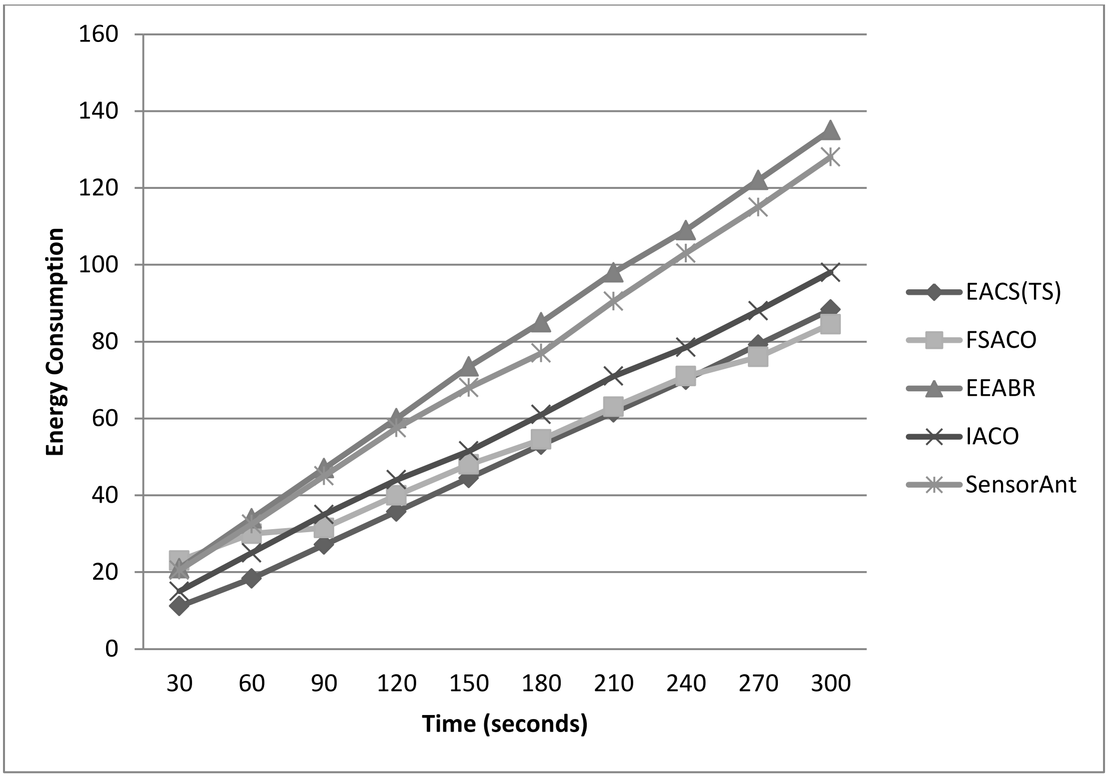


Figure 5.10. Energy consumption of different ACO variant algorithms by using 100 sensor nodes in 300 seconds

Energy efficiency of all algorithms in routing packets by using 100 sensor nodes is depicted in Figure 5.11. These results obtained by using the same formula of results presented in Figure 5.7 where the energy efficiency is measured by the number of packets received per total energy used (Zungeru et al., 2012a). Despite EACS(TS) and FSACO having almost similar energy consumption throughout the simulation, EACS(TS) outperformed FSACO in terms of energy efficiency. This fact is driven by the high throughput value and low energy consumption achieved by EACS(TS) during experiments. Nevertheless, EACS(TS) achieved the highest energy efficiency because in addition to having a mechanism to avoid local optima just like FSACO, it also has a mitigation process where the backward movement will be performed when the ant is trapped in local optima. This also ensures that less energy will go to waste when the ant or data packet are dropped during transmission.

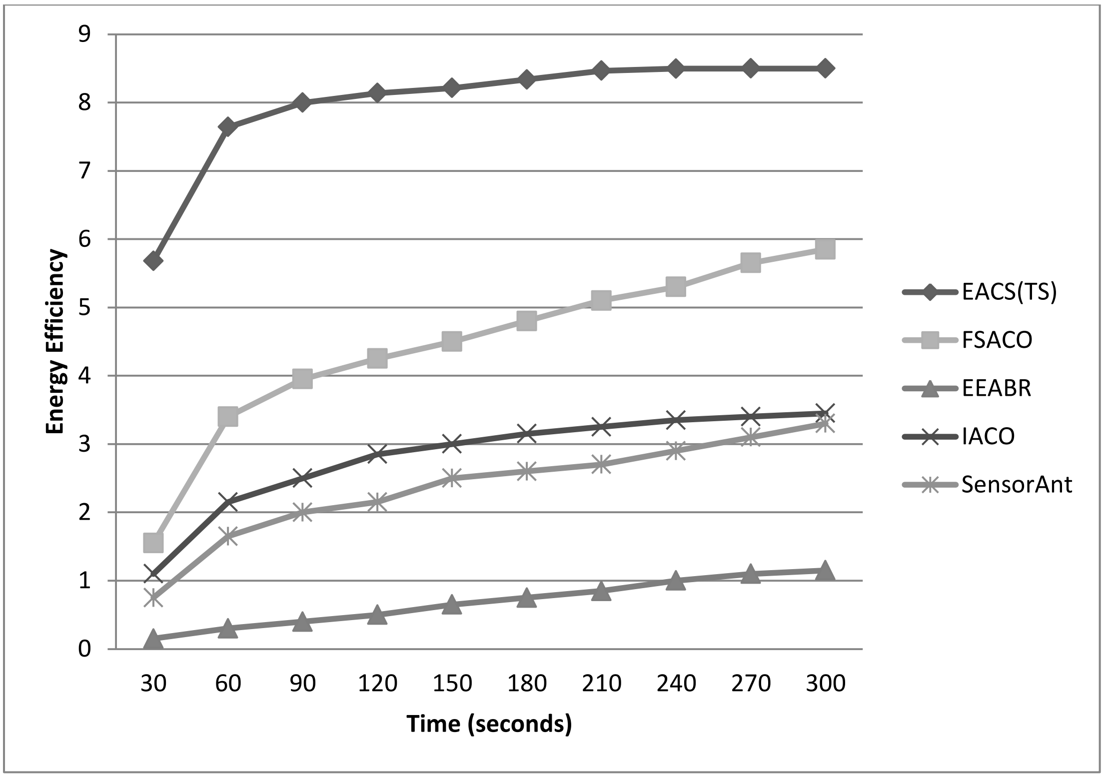


Figure 5.11. Energy efficiency of different ACO variant algorithms by using 100 sensor nodes in 300 seconds

The fourth set of experiments was conducted to evaluate the performance of hybrid algorithms which include EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH in terms of success rate, packet loss rate, and energy consumption based on the parameter specifications in Table 5.13 which were adopted from Elhabyan and Yagoub (2014). Different numbers of sensor nodes with high energy levels were used during the experiments to submit packets in 5000 seconds.

Table 5.13

Scheduling parameters to investigate the performance of hybrid algorithms for different numbers of sensor nodes

Parameters	**Values**
Routing algorithm	EACS(TS), PSO-HC, PSO-C, LEACH-C, LEACH
Number of nodes	20, 40, 60, 80,100
Nodes energy	18720 J
Simulation time	5000 seconds
Performance metric	Success rate, no. of packets received, energy consumption

Figure 5.12 and Figure 5.13 depict how the high energy sensor nodes and large simulation time will affect the success rate and packet delivery value of all algorithms, respectively. The overall results show that all algorithms successfully delivered packets in the long simulation time with significantly high success rates and packet delivery value. It is important to ensure that each sensor node is assigned a high energy value to avoid dead nodes before the end of long simulation times. It is also shown in both figures that EACS(TS) achieved the most consistent success rate and packet delivery value in all numbers of sensor nodes. It is proven that by applying a TS technique whereby nodes that have no potential neighbour node stored in the Tabu list will be avoided during path construction which eventually reduces the possibility of packet loss and local optima. PSO-HC, which is a hybrid algorithm, that combines the PSO and clustering technique, is the improved version of traditional LEACH, LEACH-C, and PSO-C that maximizes the network coverage and cluster link quality. These approaches can reduce the un-clustered sensor nodes to ensure packets can successfully arrive at the destination node. However, PSO-HC, PSO-C, and LEACH only consider the development of cluster but not the local

optima problem during the routing process. This will lead to the searching agent getting trapped in local optima when submitting packets from the cluster head to the destination node.

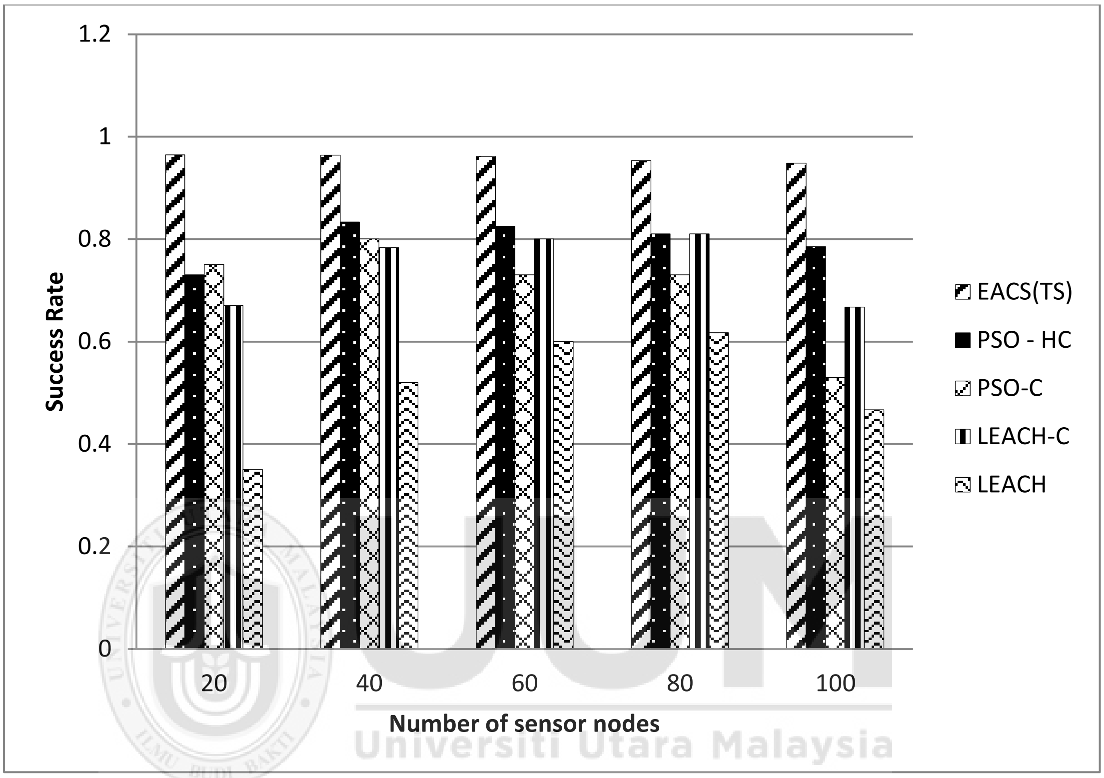


Figure 5.12. Success rate of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy sensor nodes in 5000 seconds

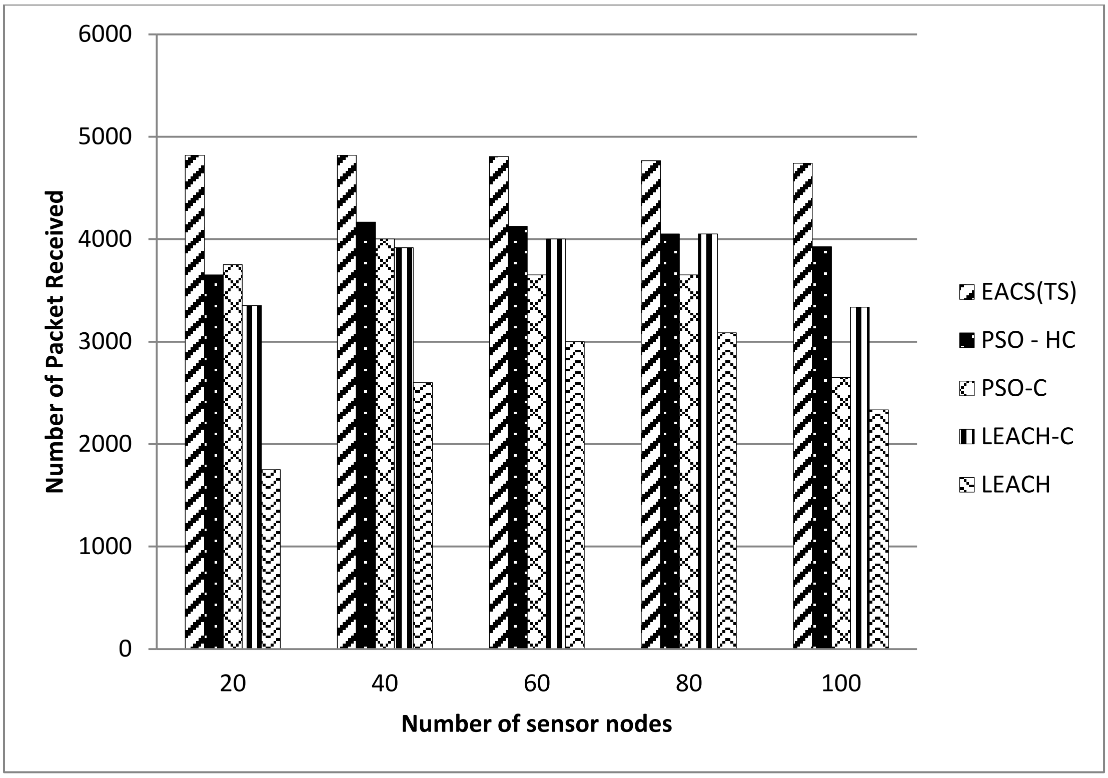


Figure 5.13. Number of packet received of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy sensor nodes in 5000 seconds

Figure 5.14 shows the energy consumption of EACS(TS), PSO-HC, PSO-C, LEACH-C and LEACH when submitting packets in 5000 seconds. Even though EACS(TS) has a significantly high success rate and packet delivery value, it consumes a lot of energy. LEACH, LEACH-C, PSO-HC, and PSO-C that are based on the clustering technique can reduce energy consumption when using the cluster head as an agent to submit the packets to the destination node. All sensor nodes that are divided into several clusters will submit their packets to the cluster head that has high energy to be forwarded to the destination node. This approach can minimize the energy usage of all cluster members in each cluster. However, even though EACS(TS) used a lot of energy when compared to the other hybrid algorithms, its performance in term of success rate and throughput are still superior. Thus, it is important to ensure that in the actual WSN implementation, the battery or energy source of each sensor node provides energy that can last longer than the sensor node's expected operating period.

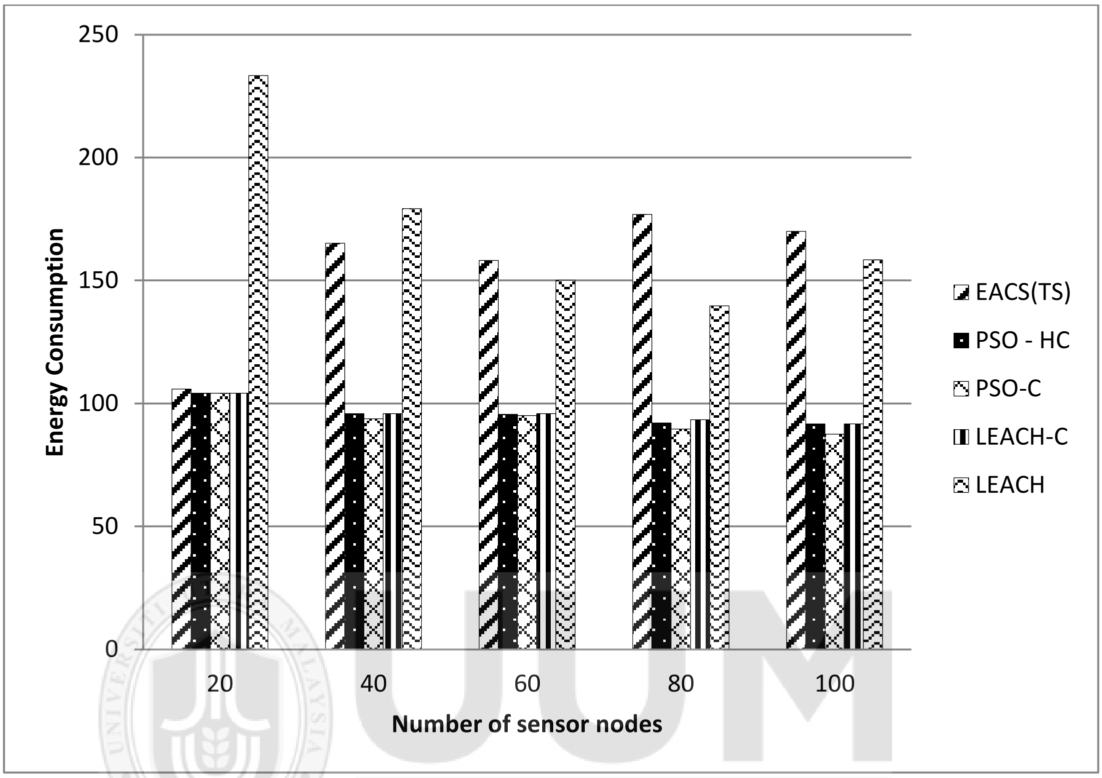


Figure 5.14. Energy consumption of EACS(TS), PSO-HC, PSO-C, LEACH-C, and LEACH by using high energy of sensor nodes in 5000 seconds

The fifth set of experiments was carried out to evaluate the performance of EACS(TS) with IEEABR, BeeSensor and BeeSensor-C in large sized networks. In this experiment, packets were submitted by using 100, 200, 300, and 400 sensor nodes in 200 seconds. The specifications and parameter settings are listed in Table 5.14, which was adopted from Cai et al. (2015) to ensure that the experiments are done against the results that are officially validated and published by previous research work.

Table 5.14

Scheduling parameters to investigate the performance of hybrid algorithms for large numbers of sensor nodes

Parameters	Values
Routing algorithm	EACS(TS), IEEABR, BeeSensor-C, BeeSensor
Number of nodes	100, 200, 300, 400
Nodes energy	50 J
Simulation time	200 seconds
Performance metric	Latency, energy efficiency

Figure 5.15 depicts the latency value of EACS(TS), IEEABR, BeeSensor and BeeSensor-C when submitting packets using 100, 200, 300, and 400 sensor nodes. The figure shows that BeeSensor has the highest latency which is not favourable in terms of performance. However, the enhanced version of BeeSensor, which is BeeSensor-C proposed by Cai et al. (2015), shows a huge improvement in the latency aspect. BeeSensor-C is a hybrid algorithm that combines the traditional BeeSensor and clustering technique with the aim to reduce collision that affects the submission time of packets to the destination node. The larger the size of network and simulation time, the higher the possibility of collision in the system, which leads to higher latency. EACS(TS) that applies the TS algorithm in preventing local optima can minimize the time taken to submit packets to the destination node. The forward ant in EACS(TS) can select the most potential sensor nodes that can guarantee the submission packets to the destination node by referring to the routing table and Tabu list. Both tables can help ants to select the sensor nodes with the highest pheromone value and have the lowest possibility of leading to local optima.

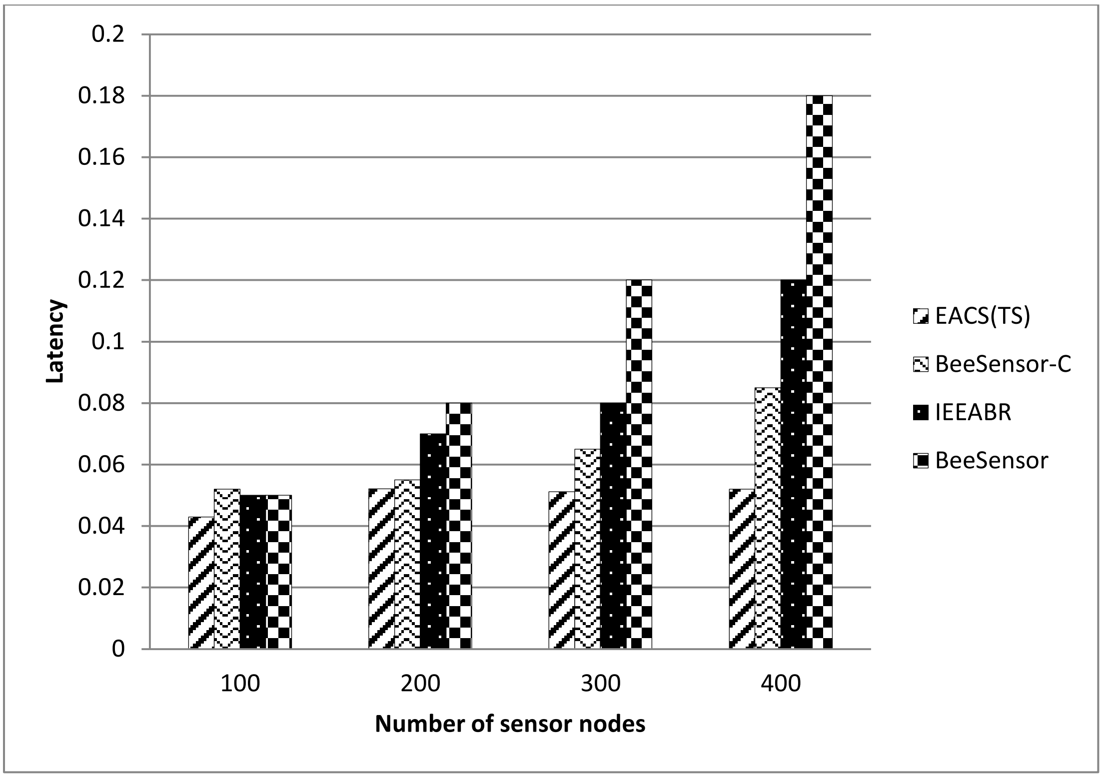


Figure 5.15. Latency of EACS(TS), IEEABR, BeeSensor-C and BeeSensor by using large numbers of sensor nodes in 200 seconds

Figure 5.16 shows the energy efficiency of algorithms in submitting packets by using 100, 200, 300, and 400 sensor nodes in 200 seconds that was calculated based on the formula by Cai et al. (2015) where lower energy efficiency is better than higher energy efficiency. BeeSensor-C has the lowest energy efficiency when the number of nodes is 100 and 200 respectively. However, as the number of nodes increases to 300 and 400, EACS(TS) overtakes the lowest energy efficiency from BeeSensor-C. These results suggest that the clustering technique is one of the best techniques to preserve the energy since packet routing to the destination node is done by the cluster head, the node with the highest residual energy, and the rest of the sensor nodes can utilize their energy simply to forward the packets within the cluster. However, in large size of network, cluster head uses a high amount of energy to consolidate packets from the cluster node and forward the collected packets to the destination node that affects the energy efficiency during routing process. On the other hand, EACS(TS) that is based on hybrid technique is also favourable in minimizing the energy efficiency in the large network by overcoming the local optima problem. The rest of non-hybrid algorithms have significantly large energy efficiency as they are prone to local optima and unbalanced distribution of packets.

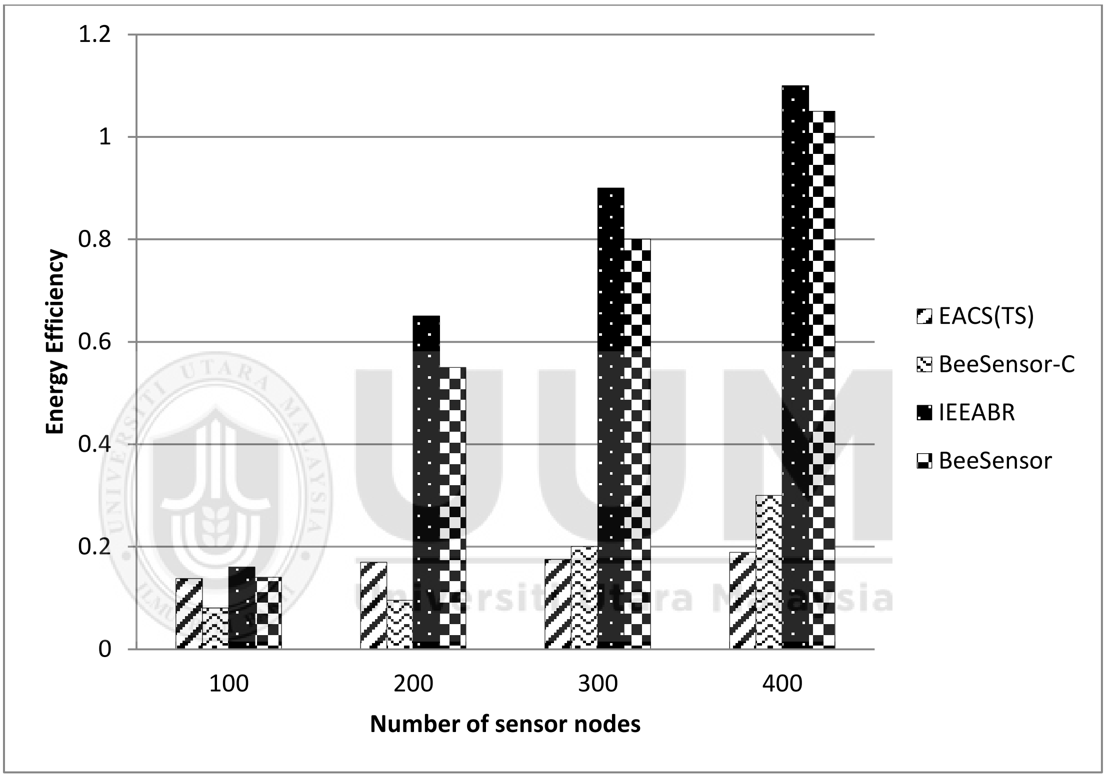


Figure 5.16. Energy efficiency of EACS(TS), IEEABR, BeeSensor-C and BeeSensor by using large numbers of sensor nodes in 200 seconds

The effect of packets size and number of sensor nodes to the performance of routing algorithms was investigated in this sixth set of experiments by adopting the simulation parameter from Gupta (2018). The performance of EACS(TS) were compared with ICSCA, PSO-ECHs, and LEACH in terms of energy consumption based on parameter specifications in Table 5.15. In this experiment, large size of packets was submitted in network qwhich consists of 100, 150, 200, and 250 sensor nodes in 100 seconds.

Table 5.15

Parameters to investigate the performance of the algorithms in submitting large size of packets

Parameters	Values
Routing algorithm	EACS(TS), ICSCA, PSO-ECHs, LEACH
Number of nodes	100, 150, 200, 250
Nodes energy	200 J
Simulation Time	100 seconds
Packet size	4000 bits
Performance metric	Energy consumption

Figure 5.17 shows the energy consumption of EACS(TS), ICSCA, PSO-ECHs, and LEACH while submitting large sized packets in 100 seconds. In this experiment, packets with 4000 bits were submitted using 100, 150, 200, and 250 sensor nodes. The results show that EACS(TS) attained the least energy consumption as compared to the other algorithms in all numbers of sensor nodes. This is due to the state transition rules applied during node selection that can balance the exploration and exploitation of sensor nodes. ICSCA, that balances the energy level of each cluster head and the best host nest during the iterative process, achieved the second-best algorithm after EACS(TS). In contrast, PSO-ECHs and LEACH have the most energy consumption because both algorithms only focus on selecting the best cluster head with high energy but not the energy of the whole system.

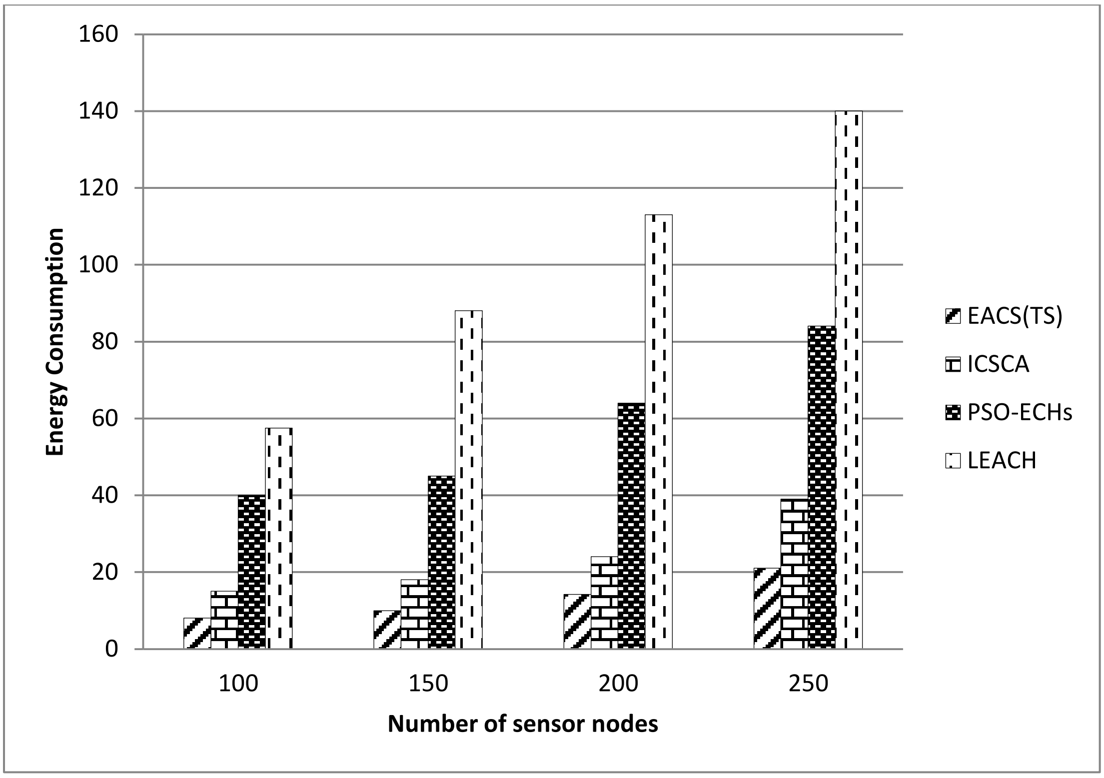


Figure 5.17. Energy consumption of EACS(TS), ICSCA, PSO-ECHs, and LEACH by using different numbers of sensor nodes in submitting large size of packets

5.5 Summary

In this chapter, experiments were undertaken to determine the best parameters to be used by EACS(TS) to route packets in a WSN. The best value for β, ρ, α and q_o were investigated in terms of success rate, throughput, latency, energy consumption, and energy efficiency. From the experimental results, the best value for β to be used in probabilistic decision rule is 4 while the best value for q_o as a parameter to control the exploration and exploitation in the state transition rule is 0.5. The best value for ρ which is an element to calculate the local pheromone update is 0.3 and 0.2 is the best value for α in global pheromone update calculation.

The next experiments were undertaken to evaluate the performance of EACS(TS) by comparing it with the single swarm intelligence algorithm. The effect of sensor nodes number to the routing algorithm were investigated where the performance of EACS(TS) in terms of success rate, packet loss rate, latency, and energy efficiency was compared with EEABR, BeeSensor, and Termite-hill. Experimental results show that EACS(TS) performed better in all performance metrics when compared with selected single swarm intelligence approaches.

The effect of time to the routing algorithms has also been investigated where packets were routed with the same number of sensor nodes in all experiments but with different simulation time. Results from experiments were captured at every 20 seconds which were set at 20, 40, 60, 80, and 100 seconds and evaluated in terms of success rate, throughput, latency, and energy efficiency. The experiment results show that EACS(TS) and IEEABR based on improved EEABR performed better than EEABR, BeeSensor, and Termite-hill in all tested performance metrics.

EACS(TS) which is the hybrid algorithm has also been compared with the other hybrid routing algorithms. The effect of simulation time and energy level of sensor nodes on EACS(TS) was compared with PSO-HC, PSO-C, LEACH-C, and LEACH in terms of success rate, energy consumption, and number of packets received by the destination node. Packets were routed by 20, 40, 60, 80, and 100 high energy sensor nodes in 5000 seconds. EACS(TS) performed better than the other algorithms in terms of numbers of packets received and success rate. Even though EACS(TS) can reduce the packet loss problem, it consumes more energy to route packets to the destination node as compared to others.

The effect of the large number of sensor nodes to the routing process has also been investigated. Packets were submitted to the destination node by using 100, 200, 300, and 400 sensor nodes within 200 seconds. The experiment results show that EACS(TS) achieved the lowest latency value by becoming the best after BeeSensor-C in terms of energy efficiency.

Experiments were also completed to examine the effects of packet size to the performance of routing algorithms. Large sized packets were routed using 100, 150, 200, and 250 sensor nodes in 100 seconds. The experiment results show that EACS(TS) used less energy when compared with ICSCA, PSO-ECHs, and LEACH.

From all the experiments undertaken, it can be concluded that the proposed hybrid EACS(TS) has achieved the best performance when compared with single swarm intelligence routing algorithms. EACS(TS) can overcome the problems that occur by a single routing algorithm and improve the performance in terms of success rate, packet loss rate, latency, energy consumption, and energy efficiency. When compared with hybrid swarm intelligence routing algorithms, EACS(TS) showed a good performance in several performance metrics. Even though EACS(TS) consumed more energy when compared to the other hybrid routing algorithms, it achieved better performance in other performance metrics such as success rate, throughput, and latency when routing packets using different simulation parameters. These advantages can ensure EACS(TS) in reducing the packet loss problem and, at the same time, can minimize the submission time of packets to the destination node. EACS(TS) also used less energy when compared with other hybrid swarm intelligence algorithms in submitting large sized packets using high energy sensor nodes, thus maximizing the network lifetime of the WSN.

CHAPTER 6

CONCLUSION AND FUTURE WORK

EACS(TS), as an improvement of the ACO algorithm, offers the opportunity to improve the results of ACO algorithms reported in the literature. The idea of improving the performance of ACO algorithms has been a great success. The results of EACS(TS) show that this approach can be superior to the best known ACO algorithms in WSNs like AS and MMAS.

Five (5) research questions were considered in Section 1.1 and five (5) corresponding research objectives that answered these questions are included in Section 1.2. The main objective of the research was to develop an enhanced ACS and TS algorithm in WSNs which can route packets to the suitable sensor nodes, minimize the forwarding time of packets to the destination node, minimize the energy consumption of sensor nodes, balance the workload of entire sensor nodes, prevent local optima problems during the routing process, and improve the network lifetime of the WSN.

The first objective was to formulate a state transition rule that could be used to evaluate the neighbour nodes' capabilities during the node selection phase. The state transition rule is calculated based on the pheromone value and heuristics value of each sensor node. This formula can be used as a benchmark in deciding whether to explore the new potential sensor nodes or exploit the previously used sensor nodes as a medium to transfer the packets from source node to destination node.

Applying a TS algorithm during the node selection phase to prevent the local optima problem was the second objective of the research. Ants always become trapped in a blind alley during the searching phase where the only available neighbour nodes are the visited nodes. In order to solve this problem, the visited sensor nodes will be put in the Tabu list and the ant will move backwards to find other potential sensor nodes. A detailed explanation of how the TS algorithm works with the ACS algorithm in preventing local optima problem was presented in Section 4.3. EACS(TS) that considers the local optima problem successfully increased the throughput and energy efficiency value by reducing the packet loss during routing process. Searching ant in EACS(TS) was able to discover the optimal routing path to submit large number of packets in short time and less energy usage.

The third objective was to develop an extended local pheromone update that can balance the load on all available sensor nodes and encourage the exploration of new potential nodes in the searching process. Local pheromone update is calculated based on the formula in Section 4.3 in order to support the exploration control phase discussed in Section 3.1.2. This formula can decrease the pheromone intensity of the visited sensor nodes in order to encourage the ant in the next iteration to explore the new potential sensor nodes in balancing loads in the WSN system. Experimental results showed that EACS(TS) performed better than traditional ant approach which is IEEABR in terms of energy efficiency. EACS(TS) that combined local pheromone update and global pheromone update will encourage exploration and exploitation during routing process compared to IEEABR that only applied global pheromone update to exploit optimal sensor nodes. By considering the exploration to the other potential sensor nodes, EACS(TS) can balance the energy usage of the whole system.

Developing an extended global pheromone update to establish the exploitation control phase in Section 3.1.3 was the fourth objective of the research. This formula was calculated based on the formula in Section 4.4 with the objective of encouraging the exploitation of selected optimal paths while reducing delays and packet loss problems in the WSN. Global pheromone update will increase the pheromone value on the selected sensor nodes, so that the ant in the next iteration will save searching time by simply referring to the previously optimal path in the routing table.

The last objective of the research is to develop a simulation model that can be used to evaluate the performance of the proposed algorithm. The performance of EACS(TS) was evaluated using RMASE which is applied as an application in Prowler. Based on the discussion in Section 3.2 and Section 4.4, Prowler was selected because it offers a simple and fast way to prototype applications with a good visualization capability for experimental and comparison purposes. This simulator was also selected because it is designed to be easily embedded into optimization algorithms. Finally, the results obtained in Chapter 5 support the idea that EACS(TS) performs better in terms of success rate, packet loss rate, delay, throughput, energy consumption, and energy efficiency when compared with other swarm intelligence algorithms such as Termite-hill, BeeSensor, PSO, EEABR algorithm.

6.1 Contribution of the research

The main contribution of the research is the way in which an ant tries to find the optimal path in submitting packets from source node to destination node. In order to achieve this objective, the state transition rule is adopted and adapted to choose the best neighbour node with high pheromone value and energy level. The ant will decide either to explore unvisited sensor nodes or exploit previously visited sensor nodes by using the state transition rule. This ensures packets arrive safely at the destination node and increases the throughput of the whole system.

The proposed EACS(TS) algorithm that combines ACS and TS algorithm has been proven to reduce local optima problem during path construction process. The ant may become trapped in a blind alley where it fails to reach the destination node and the only available nodes are visited nodes. This problem can lead to high energy consumption of the ant which will affect the network's lifetime. The TS algorithm was applied in this proposed routing algorithm to solve this problem by determining the visited sensor nodes and putting them on the Tabu list. If there is no available unvisited sensor node, the ant will move backward to the previous sensor node and continue the searching process. This technique will speed up the searching time while increasing the energy efficiency of the whole system.

EACS(TS) aims to reduce the hotspot problem and increase the load balancing among sensor nodes in the WSN. This problem is controlled by applying a local pheromone update during the node searching process to reduce the pheromone intensity of the visited sensor nodes. The effect of this approach is to make an already chosen sensor node less desirable for the following ant and encourage the exploration of unvisited sensor nodes to increase the load balancing of the whole system.

EACS(TS) also aims to reduce delay and packet loss rate during the packet submission process by enhancing the global pheromone update to increase the pheromone intensity of the optimal sensor nodes. The pheromone value that is updated by the backward ant will be stored in the routing table to be used by the ant in the next iteration. This information can help ants in the next iteration to save the searching time that will reduce the delay and packet loss problem.

The proposed EACS(TS) algorithm also has a great potential in solving the routing problem in other NP-complete problems such as TSP, vehicle routing problem, and sequential ordering problem. EACS(TS) that combines both metaheuristic algorithms which are EACS and Tabu search is suitable to solve the routing problem in other research domains in terms of success rate, packet loss rate, delay, throughput, energy consumption, and energy efficiency.

6.2 Future Work

In WSN environments, packets that are submitting from source node to destination node have different size and priority. Packets will be submitted by a multi-hop technique where the ant moves from one sensor node to another until it reaches the destination node. Future works can enhance the proposed EACS(TS) to consider the multiple path routing technique in order to consider the packet priority during packet submission. There will be two different paths, the priority path and normal path, to submit packets in a specified time. Therefore, this approach will ensure that high priority packets arrive on time without the need to queue for a long time to be submitted into the system.

The second future study will enhance the EACS(TS) algorithm in solving the fault tolerance problem. Sensor nodes in WSNs have limited capabilities in terms of power, storage and memory that will lead to dead nodes during packet submission. Submitted packets always drop during packet submission due to the dead node problem. Therefore, by applying a fault tolerance mechanism, the packet loss problem will be reduced while optimizing the lifetime of the network system.

REFERENCES

Aarts, E., & Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization*. Princetonn: Princeton University Press.

Abdel-Basset, M., Manogaran, G., El-Shahat, D., & Mirjalili, S. (2018). A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. *Future Generation Computer Systems, 85*, 129-145.

Aijaz, A. (2018). ENCLOSE: An enhanced wireless interface for communication in factory automation networks. *IEEE Transactions on Industrial Informatics*, 1-13.

Akkan, C., & Gülcü, A. (2018). A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem. *Computers & Operations Research, 90*, 22-32.

Alazzawi, L., & Elkateeb, A. (2009). Performance evaluation of the WSN routing protocols scalability. *Journal of Computer Systems, Networks, and Communications*, *2008,* 1-9.

Ali, A., Ming, Y., Chakraborty, S., & Iram, S. (2017). A comprehensive survey on real-time applications of WSN. *Future Internet, 9*(4), 1-22.

Aliouat, Z., & Aliouat, M. (2013). Improving wireless sensor networks robustness through multi-level fault tolerant routing protocol. In *Modeling Approaches and Algorithms for Advanced Computer Applications* (pp. 115-124). Switzerland: Springer.

Alizadeh, M., Edsall, T., Dharmapurikar, S., Vaidyanathan, R., Chu, K., Fingerhut, A., ... & Varghese, G. (2014). CONGA: Distributed congestion-aware load balancing for datacenters. In *ACM SIGCOMM Computer Communication Review* (Vol. 44, No. 4, pp. 503-514). New York: ACM.

Almshreqi, A. M. S., Ali, B. M., Rasid, M. F. A., Ismail, A., & Varahram, P. (2012). An improved routing mechanism using bio-inspired for energy balancing in wireless sensor networks. In *2012 International Conference on Information Networking* (pp. 150-153). Bali: IEEE.doi: 10.1109/ICOIN.2012.6164367.

Amagata, D., Hara, T., Sasaki, Y., & Nishio, S. (2017). Efficient cluster-based top-k query routing with data replication in MANETs. *Soft Computing, 21*(15), 4161-4178.

Angelo, J. S., Bernardino, H. S., & Barbosa, H. J. (2015). Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint. *Advances in Engineering Software, 80*, 101-115.

Anjali & Kaur, N. (2013). Routing based ant colony optimization in wireless sensor networks. *Global Journal of Computer Science and Technology Network, Web & Security, 13*(4), 34-40.

Arafath, M. S., Khan, K. U. R., & Sunitha, K. V. N. (2018). Pithy review on routing protocols in wireless sensor networks and least routing time opportunistic technique in WSN. In *Journal of Physics: Conference Series, 933*, 1-8.

Azami, M., Ranjbar, M., Rostami, A. S., & Amiri, A. J. (2013). Increasing the network life time by simulated annealing algorithm in WSN with point coverage. *International Journal of Ad Hoc, Sensor and Ubiquitous Computing, 4*(2), 1-8.

Barto, L., & Kozik, M. (2014). Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM, 61*(1), 3:1-3:19.

Benlic, U., & Hao, J. K. (2013). Breakout local search for the quadratic assignment problem. *Applied Mathematics and Computation, 219*(9), 4800-4815.

Beşikci, U., Bilge, Ü., & Ulusoy, G. (2015). Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research, 240*(1), 22-31.

Bhaskar, U., Cheng, Y., Ko, Y. K., & Swamy, C. (2016). Hardness results for signaling in Bayesian zero-sum and network routing games. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (pp. 479-496). ACM.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239-287.

Bierwirth, C., & Kuhpfahl, J. (2017). Extended GRASP for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research, 261*(3), 835-848.

Biswas, S., Das, R., & Chatterjee, P. (2018). Energy-efficient connected target coverage in multi-hop wireless sensor networks. In *Industry Interactive Innovations in Science, Engineering and Technology* (pp. 411-421). Singapore: Springer.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews, 2*(4), 353-373.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *Journal of ACM Computing Surveys, 35*(3), 268-308.

Bodirsky, M., Martin, B., & Mottet, A. (2018). Discrete temporal constraint satisfaction problems. *Journal of the ACM, 65*(2), 1-41.

Bosek, B., Dębski, M., Grytczuk, J., Sokół, J., Śleszyńska-Nowak, M., & Żelazny, W. (2018). Graph coloring and Graham's greatest common divisor problem. *Discrete Mathematics, 341*(3), 781-785.

Bouarafa, S., Saadane, R., & Rahmani, M. D. (2018). Inspired from ants colony: Smart routing algorithm of wireless sensor network. *Information*, *9*(1), 1-11.

Boucetta, C., Idoudi, H., & Saidane, L. A. (2016). Hierarchical cuckoo search-based routing in wireless sensor networks. In *2016 IEEE Symposium on Computers and Communication,* (pp. 996-998). Messina: IEEE. doi: 10.1109/ISCC.2016.7543866

Bougleux, S., Brun, L., Carletti, V., Foggia, P., Gaüzère, B., & Vento, M. (2017). Graph edit distance as a quadratic assignment problem. *Pattern Recognition Letters, 87*, 38-46.

Bukhari, S., Ku-Mahamud, K. R., & Morino, H. (2017). Fault tolerance grid scheduling with checkpoint based on ant colony system. *Journal of Computer Science, 13*(8), 363-370.

Cai, X., Duan, Y., He, Y., Yang, J., & Li, C. (2015). Bee-sensor-C: an energy-efficient and scalable multipath routing protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks, 11*(3), 1-14.

Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. (2006). An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Ant Colony Optimization and Swarm Intelligence* (pp. 49-59). Springer Berlin Heidelberg.

Chakraborty, A., Mitra, S. K., & Naskar, M. K. (2011). A genetic algorithm inspired routing protocol for wireless sensor networks. *International Journal of Computational Intelligence Theory and Practice, 6*(1), 1-10.

Chatzimilioudis, G., Cuzzocrea, A., Gunopulos, D., & Mamoulis, N. (2013). A novel distributed framework for optimizing query routing trees in wireless sensor networks via optimal operator placement. *Journal of Computer and System Sciences, 79*(3), 349-368.

Chen, Y., Hao, J. K., & Glover, F. (2016). An evolutionary path relinking approach for the quadratic multiple knapsack problem. *Knowledge-Based Systems, 92*, 23-34.

Cheng, D., Rao, J., Jiang, C., & Zhou, X. (2015). Resource and deadline-aware job scheduling in dynamic hadoop clusters. In *2015 IEEE International Parallel and Distributed Processing Symposium*, (pp. 956-965). Hyderabad: IEEE.

Cheng, L., Niu, J., Luo, C., Shu, L., Kong, L., Zhao, Z., & Gu, Y. (2018). Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks. *Computer Networks, 134*, 66-77.

Chong, C. Y., & Kumar, S. P. (2003). Sensor networks: evolution, opportunities, and challenges. In *Proceedings of the IEEE* (Vol. 91, No. 8, pp. 1247-1256). IEEE. doi: 10.1109/JPROC.2003.814918

Chung, K. M., Pettie, S., & Su, H. H. (2017). Distributed algorithms for the Lovász local lemma and graph coloring. *Distributed Computing, 30*(4), 261-280.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceedings of European Conference on Artificial Life* (Vol.142, pp. 134-142). Paris: Elsevier Publishing.

Costa, D. (1994). An evolutionary tabu search algorithm and the NHL scheduling problem. *Journal of INFOR: Information Systems and Operational Research*, *33*(1), 161–178.

Crawford, B., Soto, R., Astorga, G., & García, J. (2018). Constructive metaheuristics for the set covering problem. In *International Conference on Bioinspired Methods and Their Applications* (pp. 88-99). Springer, Cham.

Crawford, B., Soto, R., Riquelme-Leiva, M., Peña, C., Torres-Rojas, C., Johnson, F., & Paredes, F. (2015). Modified binary firefly algorithms with different transfer functions for solving set covering problems. In *Software Engineering in Intelligent Systems* (pp. 307-315). Switzerland: Springer.

Cui, Y., Liu, W., & Zhao, Z. (2015). Wireless sensor network route optimization based on improved ant colony-genetic algorithm. *International Journal of Online Engineering*, *11*(9), 4-8.

Cui, Y., Médard, M., Yeh, E., Leith, D., Lai, F., & Duffy, K. R. (2017). A linear network code construction for general integer connections based on the constraint satisfaction problem. *IEEE/ACM Transactions on Networking, 25*(6), 3441-3454.

Czubak, A. (2013). A note on the local minimum problem in wireless sensor networks. In *International Conference on Computer Networks* (pp. 64-75). Springer.

Dai, H. N. (2009). Throughput and delay in wireless sensor networks using directional antennas. In *2009 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (pp. 421-426). Melbourne: IEEE.doi: 10.1109/ISSNIP.2009.5416826

Dam, S., Mandal, G., Dasgupta, K., & Dutta, P. (2018). An ant-colony-based meta-heuristic approach for load balancing in cloud computing. In *Applied Computational Intelligence and Soft Computing in Engineering* (pp. 204-232). IGI Global.

Dehghan, M., Seetharam, A., Jiang, B., He, T., Salonidis, T., Kurose, J., ... & Sitaraman, R. (2015). On the complexity of optimal routing and content caching in heterogeneous networks. In *2015 IEEE Conference on Computer Communications* (pp. 936-944). Kowloon: IEEE.

Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research, 255*(1), 1-20.

Delorme, M., Iori, M., & Martello, S. (2018). BPPLIB: a library for bin packing and cutting stock problems. *Optimization Letters, 12*(2), 235-250.

Derbew, Y., & Libsie, M. (2014). A wireless sensor network framework for large-scale industrial water pollution monitoring. In *2014 IST-Africa Conference Proceedings*, (pp. 1-8). Mauritius: IEEE.doi:10.1109/ISTAFRICA.2014.6880619

Derr, K., & Manic, M. (2015). Wireless sensor networks—Node localization for various industry problems. *IEEE Transactions on Industrial Informatics, 11*(3), 752-762.

Desai, U. B., Jain, B. N., & Merchant, S. N. (2007). Wireless sensor networks: Technology roadmap. In *Workshop on Wireless Sensor Networks* (Vol. 20, pp. 1-121). Bombay.

Despaux, F. (2015). *Modeling and evaluation of the end-to-end delay in wireless sensor networks* (Doctoral dissertation). Université de Lorraine, France.

Dhivya, M., & Sundarambal, M. (2012). Lifetime maximization in wireless sensor networks using tabu swarm optimization. *Procedia Engineering, 38*, 511-516.

Dodig-Crnkovic, G. (2002). Scientific methods in computer science. In *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia* (pp. 126-130).

Dokeroglu, T. (2015). Hybrid teaching–learning-based optimization algorithms for the quadratic assignment problem. *Computers & Industrial Engineering, 85*, 86-101.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms* (Ph.D Thesis). Politecnico di Milano, Italy.

Dorigo, M., & Gambardella, L. (1997a). Ant colonies for the travelling salesman problem. *BioSystems, 43*(2), 73-81.

Dorigo, M., & Gambardella, L. (1997b). Ant colony system: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.

Dorigo, M., & Gambardella, L. M. (1996). A study of some properties of Ant-Q. In *International Conference on Parallel Problem Solving from Nature* (pp. 656-665). Springer.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization.* Cambridge, Massachusetts, London, England: MIT Press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991a). *Positive Feedback as a Search Strategy* (Technical Report No. 91-016), Politecnico di Milano, Italy.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics– Part B, 26*(1), 29–41.

Dorigo, M., V. Maniezzo & A. Colorni (1991b). *The Ant System: An Autocatalytic Optimizing Process*. (*Technical Report* No. 91-016 Revised), Politecnico di Milano, Italy.

Duarte-Melo, E. J., & Liu, M. (2002). Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In *Global Telecommunications Conference* (Vol. 1, pp. 21-25). Taipei: IEEE.

Eberhart, R. C., & Kennedy, J. (1995). Particle swarm optimization. In *Proceeding of IEEE International Conference on Neural Network* (pp. 1942-1948). Perth: IEEE. doi: 10.1109/ICNN.1995.488968.

El Rhazi, A., & Pierre, S. (2009). A Tabu search algorithm for cluster building in wireless sensor networks, *IEEE Transactions on Mobile Computing, 8*(4), 433-444.

Elhabyan, R. S., & Yagoub, M. C. (2014). Particle swarm optimization protocol for clustering in wireless sensor networks: A realistic approach. In Proceedings of 2014 IEEE 15th International Conference on Information Reuse and Integration (pp. 345-350). Redwood City: IEEE.

Elhoseny, M., Yuan, X., Yu, Z., Mao, C., El-Minir, H. K., & Riad, A. M. (2015). Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm. *IEEE Communications Letters, 19*(12), 2194-2197.

Engmann, F., Katsriku, F. A., Abdulai, J. D., Adu-Manu, K. S., & Banaseka, F. K. (2018). Prolonging the lifetime of wireless sensor networks: A review of current techniques. *Wireless Communications and Mobile Computing*, *2018,* 1-23.

Ennaji, R., & Boulmalf, M. (2009). Routing in wireless sensor networks. In *International Conference on Multimedia Computing and Systems* (pp. 495-500). Ouarzazate: IEEE. doi:10.1109/MMCS.2009.5256646

Ez-Zaidi, A., & Rakrak, S. (2017). A new approach for storage balancing in wireless sensor networks. In *2017 International Conference on Electrical and Information Technologies* (pp. 1-6). Rabat, Morocco: IEEE.

Ezzat, A. (2013). *Ant colony optimization approaches for the sequential ordering problem (*Master dissertation). American University in Cairo, Egypt.

Ezzat, A., Abdelbar, A. M., & Wunsch, D. C. (2014). A bare-bones ant colony optimization algorithm that performs competitively on the sequential ordering problem. *Memetic Computing, 6*(1), 19-29.

Fathima, K. S. A., & Sindhanaiselvan, K. (2013). Ant colony optimization based routing in wireless sensor networks. *International Journal Advanced Networking and Applications, 4*(4), 1686-1689.

Fister, I., & Fister Jr, I. (Eds.). (2015). *Adaptation and Hybridization in Computational Intelligence* (Vol. 18). Switzerland: Springer International Publishing.

Frey, H., Rührup, S., & Stojmenović, I. (2009). Routing in wireless sensor networks. *Guide to Wireless Sensor Networks* (pp. 81-111). London: Springer.

Frotzscher, A., Wetzker, U., Bauer, M., Rentschler, M., Beyer, M., Elspass, S., & Klessig, H. (2014). Requirements and current solutions of wireless communication in industrial automation. In *2014 IEEE International Conference on Communications Workshops*, (pp. 67-72). Sydney: IEEE. doi: 10.1109/ICCW.2014.6881174

Fu, Y., Ding, J., Wang, H., & Wang, J. (2017). Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in Industry 4.0-based manufacturing system. *Applied Soft Computing, 68*, 847-855.

Gambardella, L. M., Montemanni, R., & Weyland, D. (2012). An enhanced ant colony system for the sequential ordering problem. In *Operations Research Proceedings* (pp. 355-360). Springer Berlin Heidelberg.

Gendreau, M., & Potvin, J. (2010). *Handbook of Metaheuristics*. New York: Springer.

Gendreau, M., & Potvin, J. (2014). Tabu search. In E. K. Burke & G. Kendall (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 243–263). Boston: Springer.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Journal of Computers and Operations Research*, *13*(5), 533 – 549.

Glover, F. W., & Kochenberger, G. A. (Eds.). (2006). *Handbook of metaheuristics* (Vol. 57). Springer US.

Glover, F., & Laguna, M. (1997). *Tabu Search*. Boston: Kluwer Academic.

Goksal, F. P., Karaoglan, I., & Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering, 65*(1), 39-53.

Gómez Santillán, C., Cruz Reyes, L., Meza Conde, E., Schaeffer, E., & Castilla Valdez, G. (2010). A self-adaptive ant colony system for semantic query routing problem in P2P networks. *Computación y Sistemas, 13*(4), 433-448.

Gonçalves, J. F., & Resende, M. G. (2013). A biased random key genetic algorithm for 2D and 3D bin packing problems. *International Journal of Production Economics, 145*(2), 500-510.

Goss, S., Aron, S., Deneubourg, J., & Pasteels, J. (1989). Self-organized shortcuts in the Argentine ant. *Journal of Naturwissenschaften, 76*(12), 579-581.

Gülcü, Ş., Mahi, M., Baykan, Ö. K., & Kodaz, H. (2018). A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. *Soft Computing, 22*(5), 1669-1685.

Gupta, G. P. (2018). Improved cuckoo search-based clustering protocol for wireless sensor networks. *Procedia Computer Science*, *125*, 234-240.

Gupta, K., & Sikka, V. (2015). Design issues and challenges in wireless sensor networks. *International Journal of Computer Applications*, *112*(4), 26-32.

Hanane, H., & Fouzia, B. (2014). Improving resource discovery and query routing in peer-to-peer data sharing systems using gossip style and ACO algorithm. In *The Ninth International Conference on Systems and Networks Communications* (pp. 99-106).

Hladik, C., Schalles, J., & Alber, M. (2013). Salt marsh elevation and habitat mapping using hyperspectral and LIDAR data. *Remote Sensing of Environment, 139*, 318-330.

Holzinger, A., Plass, M., Holzinger, K., Crişan, G. C., Pintea, C. M., & Palade, V. (2016). Towards interactive machine learning (iML): Applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In *International Conference on Availability, Reliability, and Security* (pp. 81-95). Springer International Publishing.

Hu, F., & Cao, X. (2010). *Wireless sensor networks: principles and practice*. CRC Press.

Jangra, A., Awasthi, A., & Bhatia, V. (2013). A study on swarm artificial intelligence. *International Journal of Advanced Research in Computer Science and Software Engineering, 3*(8), 259-263.

Javaid, N., Rasheed, M. B., Imran, M., Guizani, M., Khan, Z. A., Alghamdi, T. A., & Ilahi, M. (2015). An energy-efficient distributed clustering algorithm for heterogeneous WSNs. *EURASIP Journal on Wireless Communications and Networking, 2015*(151), 1-11.

Jourdan, L., Basseur, M., & Talbi, E. G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, *199*(3), 620–629. doi:10.1016/j.ejor.2007.07.035.

Kaegi, S., & White, T. (2003). Using local information to guide ant based search. In *Proceedings of the 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems* (pp. 692-701).

Karaboga, D. (2005). *An Idea Based on Honey Bee Swarm for Numerical Optimization* (Technical report-tr06). Erciyes university.

Karim, L., Nasser, N., Taleb, T., & Alqallaf, A. (2012). An efficient priority packet scheduling algorithm for wireless sensor network. In *2012 IEEE International Conference on Communications* (pp. 334-338). Ottawa, Canada: IEEE.

Karthikeyan, T., & Subramani, B. (2014). QoS based optimal routing in WSN using hybrid intelligent bee colony agent. *International Journal of Computer Science and Information Technologies, 5*(4), 5785-5792.

Katyara, S., Izykowsk, J., Chowdhry, B. S., Musavi, H. A., & Hussain, R. (2018). WSN-based monitoring and fault detection over a medium-voltage power line using two-end synchronized method. *Electrical Engineering, 100*(1), 83-90.

Kaur, J., & Gangwar, R. C. (2015). Improved tabu search based energy efficient routing protocols for wireless sensor networks. In *2015 International Conference on Green Computing and Internet of Things,* (pp. 637-642). Noida: IEEE.

Keskin, M. E., Altınel, İ. K., & Aras, N. (2015). Combining simulated annealing with Lagrangian relaxation and weighted Dantzig–Wolfe decomposition for integrated design decisions in wireless sensor networks. *Computers & Operations Research*, *59*, 132-143.

Ketshabetswe, L. K., Zungeru, A. M., Mangwala, M., Chuma, J. M., & Sigweni, B. (2019). Communication protocols for wireless sensor networks: A survey and comparison. *Heliyon*, *5*(5), 1-43.

Khan, A., Javaid, N., Ahmad, A., Akbar, M., Khan, Z. A., & Ilahi, M. (2018). A priority-induced demand side management system to mitigate rebound peaks using multiple knapsack. *Journal of Ambient Intelligence and Humanized Computing*, 1-24.

Khan, F., & Portmann, M. (2017). Backhaul, QoS, and channel-aware load balancing optimization in SDN-based LTE networks. In *2017 11th International Conference on Signal Processing and Communication Systems*, (pp. 1-10). Gold Coast: IEEE. doi: 10.1109/ICSPCS.2017.8270500

Khan, M. I., Gansterer, W. N., & Haring, G. (2013). Static vs. mobile sink: The influence of basic parameters on energy efficiency in wireless sensor networks. *Computer Communications, 36*(9), 965-978.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. S*cience*, *220*(4598), 671-680.

Kuhpfahl, J., & Bierwirth, C. (2016). A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective. *Computers & Operations Research, 66*, 44-57.

Kumar, V., Jain, A., & Barwal, P. N. (2014). Wireless sensor networks: Security issues, challenges and solutions. *International Journal of Information and Computation Technology, 4*(8), 859-868.

Kuo, R. J., & Zulvia, F. E. (2017). Hybrid genetic ant colony optimization algorithm for capacitated vehicle routing problem with fuzzy demand—A case study on garbage collection system. In *2017 4th International Conference on Industrial Engineering and Applications*, (pp. 244-248). Nagoya: IEEE.

Lalla-Ruiz, E., Expósito-Izquierdo, C., Taheripour, S., & Voß, S. (2016). An improved formulation for the multi-depot open vehicle routing problem. *OR Spectrum, 38*(1), 175-187.

Laouid, A., Dahmani, A., Bounceur, A., Euler, R., Lalem, F., & Tari, A. (2017). A distributed multi-path routing algorithm to balance energy consumption in wireless sensor networks. *Ad Hoc Networks, 64*, 53-64.

Levendovszky, J., Tornai, K., Treplan, G., & Olah, A. (2011). Novel load balancing algorithms ensuring uniform packet loss probabilities for WSN. In *2011 IEEE 73rd Vehicular Technology Conference* (pp. 1-5). Yokohama, Japan: IEEE.

Li, S., Lim, A., & Liu, C. (2010). Improving QoS-based routing by limiting interference in lossy wireless sensor networks. *International Journal of Wireless & Mobile Networks, 2*(4), 44-58.

Li, X., Keegan, B., & Mtenzi, F. (2018). Energy efficient hybrid routing protocol based on the artificial fish swarm algorithm and ant colony optimisation for WSNs. *Sensors, 18*(10), 1-18.

Li, Z., & Shi, Q. (2013). An QoS algorithm based on ACO for wireless sensor network. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing* (pp. 1671-1674). Zhangjiajie: IEEE. doi: 10.1109/HPCC.and.EUC.2013.236.

Li, Z., Chen, B., Liu, X., Ning, D., Duan, W., Qiu, X., & Xu, C. (2017). QoS-aware parallel job scheduling framework for simulation execution as a service. In *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications* (pp. 208-211). Rome: IEEE.

Lim, C. P., & Jain, L. C. (2009). Advances in swarm intelligence. In *Innovations in Swarm Intelligence* (pp. 1-7). Springer Berlin Heidelberg.

Lissovoi, A., & Witt, C. (2015). Runtime analysis of ant colony optimization on dynamic shortest path problems. *Theoretical Computer Science, 561*, 73-85.

Liu, M., Xu, S., & Sun, S. (2012). An agent-assisted QoS-based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications, 35*(1), 29-36.

Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018). An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Transactions on Evolutionary Computation, 22*(1), 113-128.

Liu, X., Zhang, X., Cui, Q., & Li, W. (2017). Implementation of ant colony optimization combined with tabu search for multi-resource fair allocation in heterogeneous cloud computing. In *2017 IEEE 3rd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, and IEEE International Conference on Intelligent Data and Security* (pp. 196-201). Beijing, China: IEEE.

Loganathan, M., Sabapathy, T., Ghazali, N. H., Ahmad, R. B., & Osman, M. N. (2017). Energy efficient routing protocols for wireless sensor networks: Comparison and future directions. In *MATEC Web of Conferences* (Vol. 140, pp. 1-7). EDP Sciences.

Lu, S., Zhou, P., Wang, X., Liu, Y., Liu, F., & Zhao, J. (2018). Condition monitoring and fault diagnosis of motor bearings using undersampled vibration signals from a wireless sensor network. *Journal of Sound and Vibration, 414,* 81-96.

Lule, E., & Bulega, T. E. (2015). A scalable wireless sensor network (WSN) based architecture for fire disaster monitoring in the developing world. *International Journal of Computer Network and Information Security*, *2015*(2), 40-49.

Luo, L., & Li, L. (2012). An ant colony system based routing algorithm for wireless sensor network. In *2012 International Conference on Computer Science and Electronics Engineering* (Vol. 2, pp. 376-379). Hangzhou: IEEE. doi: 10.1109/ICCSEE.2012.145

Maity, S., Roy, A., & Maiti, M. (2018). A rough multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem. *Granular Computing*, 1-18.

Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied Soft Computing, 19*, 93-101.

Masrom, S., Abidin, S. Z., & Omar, N. (2012). A taxonomy of low-level hybridization in metaheuristics algorithms. In *2012 IEEE Fifth International Conference on Advanced Computational Intelligence* (pp. 441-446). Nanjing: IEEE.

Masrom, S., Abidin, S. Z., Omar, N., & Nasir, K. (2014). Taxonomy of low-level hybridization (LLH) for PSO-GA. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).

Mathiyalagan, P., Suriya, S., & Sivanandam, S. N. (2010). Modified ant colony algorithm for grid scheduling. *International Journal on Computer Science And Engineering, 2*(02), 132-139.

Mavrovouniotis, M. & Yang, S. (2018) Ant colony optimization for dynamic combinatorial optimization problems. In Tan, Y. (Ed.), *Swarm Intelligence - From Concepts to Applications* (pp. 1-26). IET.

Mavrovouniotis, M. (2013). *Ant colony optimization in stationary and dynamic environments* (Doctoral dissertation). University of Leicester, England.

Mittal, M., & Kumar, K. (2015). Energy efficient homogeneous wireless sensor network using self-organizing map (SOM) neural networks. *African Journal of Computing & ICT, 8*(1), 179-184.

Mittal, V., Gupta, S., & Choudhury, T. (2018). Comparative analysis of authentication and access control protocols against malicious attacks in wireless sensor networks. In *Smart Computing and Informatics* (pp. 255-262). Singapore: Springer.

Mohammadi, R., & Noghabi, H. B. (2016). SAT: Simulated annealing and tabu search based routing algorithm for wireless sensor networks. *International*

Journal of Computer Networks and Communications Security, *4*(10), 286-293.

Mohindru, V., & Singh, Y. (2018). Node authentication algorithm for securing static wireless sensor networks from node clone attack. *International Journal of Information and Computer Security*, *10*(2-3), 129-148.

Moret, B. M. E., & Shapiro, H. D. (2001). Algorithms and experiments: The new (and old) methodology. *Journal of Universal Computer Science, 7*(5), 434-446.

Moschakis, I. A., & Karatza, H. D. (2015). Multi-criteria scheduling of Bag-of-Tasks applications on heterogeneous interlinked clouds with simulated annealing. *Journal of Systems and Software*, *101*, 1-14.

Mostafaei, H. (2018). Energy-efficient algorithm for reliable routing of wireless sensor networks. *IEEE Transactions on Industrial Electronics*, *66*(7), 5567-5575.

Muñoz, G., Espinoza, D., Goycoolea, M., Moreno, E., Queyranne, M., & Letelier, O. R. (2018). A study of the Bienstock–Zuckerberg algorithm: applications in mining and resource constrained project scheduling. *Computational Optimization and Applications, 69*(2), 501-534.

Nasir, H. J. A., & Ku-Mahamud, K. R. (2016). Wireless sensor network: A bibliographical survey. *Indian Journal of Science and Technology*, *9*(38), 1-21.

Nasir, H. J. A., Ku-Mahamud, K. R., & Kamioka, E. (2017). Ant colony optimization approaches in wireless sensor network: Performance evaluation. *Journal of Computer Science*, *13*(6), 153-164.

Nasir, H. J. A., Ku-Mahamud, K. R., & Kamioka, E. (2018). Enhanced ant colony system for reducing packet loss in wireless sensor network. *International Journal of Grid and Distributed Computing*, *11*(1), 81-88.

Okafor, F. O., & Fagbohunmi, G. S. (2013). Energy efficient routing in wireless sensor networks based on ant colony optimization. *West African Journal of Industrial and Academic Research*, *8*(1), 102-109.

Okdem, S., & Karaboga, D. (2009). Routing in wireless sensor networks using an ant colony optimization (ACO) router chip. *Sensors*, *9*(2), 909-921.

Oldewurtel, F., & Mahonen, P. (2010). Metrics for performance prediction of wireless sensor networks. In *2010 IEEE 72nd Vehicular Technology Conference-Fall* (pp. 1-5). Ottawa, Canada: IEEE.

Orojloo, H., & Haghighat, A. T. (2016). A tabu search based routing algorithm for wireless sensor networks. *Wireless Networks*, *22*(5), 1711-1724.

Ouaarab, A., Ahiod, B., & Yang, X. S. (2014). Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications, 24*(7-8), 1659-1669.

Owais, M., Osman, M. K., & Moussa, G. (2016). Multi-objective transit route network design as set covering problem. *IEEE Transactions on Intelligent Transportation Systems, 17*(3), 670-679.

Paquete, L., & Stützle, T. (2018). Stochastic local search algorithms for multiobjective combinatorial optimization: A. review. In T. F. Gonzalez (Ed.), *Handbook of Approximation Algorithms and Metaheuristics: Methodologies and Traditional Applications* (Vol. 1). CRC Press.

Parenreng, J. M., & Kitagawa, A. (2017). A model of security adaptation for limited resources in wireless sensor network. *Journal of Computer and Communications, 5*(03), 10-23.

Perretto, M., & Lopes, H. S. (2005). Reconstruction of phylogenetic trees using the ant colony optimization paradigm. *Genetics and Molecular Research*, *4*(3), 581-589.

Phillips, A. E., Walker, C. G., Ehrgott, M., & Ryan, D. M. (2017). Integer programming for minimal perturbation problems in university course timetabling. *Annals of Operations Research, 252*(2), 283-304.

Piana, S., Klepeis, J. L., & Shaw, D. E. (2014). Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Current Opinion in Structural Biology, 24*, 98-105.

Pintea, C. M. (2014). *Advances in Bio-Inspired Computing for Combinatorial Optimization Problems*. Springer Berlin Heidelberg.

Qiu, C., Shen, H., & Chen, K. (2017). An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in WSNs. *IEEE Transactions on Mobile Computing, 17*(6), 1247-1259.

Rahim, S., Khan, S. A., Javaid, N., Shaheen, N., Iqbal, Z., & Rehman, G. (2015). Towards multiple knapsack problem approach for home energy management in smart grid. In *2015 18th International Conference on Network-Based Information Systems* (pp. 48-52). Taipei: IEEE.

Rao Y. C., & Rani, S. (2015). Energy efficiency and maximizing network lifetime for WSNs using ACO algorithm. *International Journal of Innovative Technology and Exploring Engineering, 5*(2), 15-20.

Rathod, V., & Mehta, M. (2011). Security in wireless sensor network: A survey. *Ganpat University Journal of Engineering and Technology*, *1*(1), 35-44.

Rothlauf, F. (2011). *Design of Modern Heuristics Principles and Application*. Heidelberg: Springer.

Sahni, V., Bala, M., & Sharma, R. (2016). Tabu search based cluster head selection in stable election protocol. *International Journal on Recent and Innovation Trends in Computing and Communication, 4*(8), 90-94.

Saleem, M., Ullah, I., & Farooq, M. (2012). BeeSensor: An energy-efficient and scalable routing protocol for wireless sensor networks. *Information Sciences*, *200*, 38-56.

Santana, F. S., Costa, A. H. R., Truzzi, F. S., Silva, F. L., Santos, S. L., Francoy, T. M., & Saraiva, A. M. (2014). A reference process for automating bee species identification based on wing images and digital image processing. *Ecological Informatics, 24*, 248-260.

Sarkar, A., & Murugan, T. S. (2019). Cluster head selection for energy efficient and delay-less routing in wireless sensor network. *Wireless Networks*, *25*(1), 303-320.

Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research, 245*(3), 704-718.

Shah, B., Iqbal, F., & Khattak, A. M. (2016). Fuzzy query routing in unstructured mobile peer-to-peer networks. In *2016 IEEE Tenth International Conference on Semantic Computing* (pp. 154-161). Laguna Hills: IEEE.

Shekofteh, S. K., Yaghmaee, M. H., Khalkhali, M. B., & Deldari, H. (2010). Localization in wireless sensor networks using tabu search and simulated annealing. In *2010 The 2nd International Conference on Computer and Automation Engineering,* (Vol. 2, pp. 752-757). Singapore: IEEE. doi: 10.1109/ICCAE.2010.5451779

Shin, T. H., Chin, S., Yoon, S. W., & Kwon, S. W. (2011). A service-oriented integrated information framework for RFID/WSN-based intelligent construction supply chain management. *Automation in Construction, 20*(6), 706-715.

Shobaki, G., & Jamal, J. (2015). An exact algorithm for the sequential ordering problem and its application to switching energy minimization in compilers. *Computational Optimization and Applications, 61*(2), 343-372.

Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. *Cluster Computing, 18*(2), 829-844.

Singh A., & Behal S. (2013). Ant colony optimization for improving network lifetime in wireless sensor networks. *International Journal of Engineering Sciences*, *June 2013*(8), 1-12.

Singh, R., Singh, D. K., & Kumar, L. (2010). Swarm intelligence based approach for routing in mobile ad hoc networks. *International Journal of Science and Technology Education Research, 1*(7), 147-153.

Skinderowicz, R. (2017). An improved ant colony system for the sequential ordering problem. *Computers & Operations Research, 86*, 1-17.

Song, X., Sun, L., & Cao, Y. (2010). Study on the convergence of converse ant colony algorithm for job shop scheduling problem. In *2010 Sixth International Conference on Natural Computation* (Vol. 5, pp. 2710-2714). IEEE.

Soria-Alcaraz, J. A., Ochoa, G., Swan, J., Carpio, M., Puga, H., & Burke, E. K. (2014). Effective learning hyper-heuristics for the course timetabling problem. *European Journal of Operational Research, 238*(1), 77-86.

Stützle, T., & Hoos, H. (1999). The max-min ant system and local search for combinatorial optimization problems. In S. Voß, S. Martello, I. H. Osman, & C. Roucairol (Eds.), *Meta-heuristics* (pp. 313-329). Boston: Springer.

Stützle, T., & Hoos, H. (2000). MAX-MIN ant system. *Future Generation Computer Systems, 16*(9), 889-914.

Stützle, T., López-Ibánez, M., Pellegrini, P., Maur, M., De Oca, M. M., Birattari, M., & Dorigo, M. (2011). Parameter adaptation in ant colony optimization. In *Autonomous Search* (pp. 191-215). Springer.

Sun, Y., Dong, W., & Chen, Y. (2017). An improved routing algorithm based on ant colony optimization in wireless sensor networks. *IEEE Communications Letters, 21*(6), 1317-1320.

Sundaran, K., Ganapathy, V., & Sudhakara, P. (2017). Energy efficient multi-event based data transmission using ant colony optimization in wireless sensor networks. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies* (pp. 998-1004). Kannur: IEEE. doi: 10.1109/ICICICT1.2017.8342703

Suppiah, Y., & Shen, K. P. (2015). Minimizing total weighted tardiness on single machine. *International Journal of Research in Science Engineering and Technology, 2*(5), 12-18.

Suryadevara, N. K., Mukhopadhyay, S. C., Kelly, S. D. T., & Gill, S. P. S. (2015). WSN-based smart sensors and actuator for power management in intelligent buildings. *IEEE/ASME Transactions on Mechatronics, 20*(2), 564-571.

Sutar, U. S., & Bodhe, S. K. (2010). Energy efficient topology control algorithm for multi-hop ad-hoc wireless sensor network. In *3rd IEEE International Conference on Computer Science and Information Technology* (Vol. 3, pp. 418-421). Chengdu: IEEE. doi: 10.1109/ICCSIT.2010.5564137

Suzuki, Y., & Cortes, J. D. (2016). A tabu search with gradual evolution process. *Computers & Industrial Engineering*, *100*, 52-57.

Tall, H., & Chalhoub, G. (2017). ABORt: Acknowledgement-based opportunistic routing protocol for high data rate multichannel WSNs. *Journal of Sensor and Actuator Networks, 6*(4), 1-15.

Tan, W. F., Lee, L. S., Majid, Z. A., & Seow, H. V. (2012). Ant colony optimization for capacitated vehicle routing problem. *Journal of Computer Science*, *8*(6), 846-852.

Tewari, M., & Vaisla, K. S. (2014). Optimized hybrid ant colony and greedy algorithm technique based load balancing for energy conservation in WSN. *International Journal of Computer Applications*, *104*(17), 14-18.

Thesen, A. (1998). Design and evaluation of tabu search algorithms for multiprocessor scheduling. *Journal of Heuristics, 4*(2), 141–160.

Tritschler, M., Naber, A., & Kolisch, R. (2017). A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research, 262*(1), 262-273.

Tuani, A. F., Keedwell, E., & Collett, M. (2017). H-ACO: A heterogeneous ant colony optimisation approach with application to the travelling salesman problem. In *Artificial Evolution: International Conference on Artificial Evolution* (pp. 144-161). Springer International Publishing.

Tubiello, F., Poehls, L., Webber, T., Marcon, C., & Vargas, F. (2018). A path energy control technique for energy efficiency on wireless sensor networks. In *2018 IEEE 9th Latin American Symposium on Circuits & Systems* (pp. 1-4). Puerto Vallarta: IEEE.

Turley, M. D., Bilotta, G. S., Arbociute, G., Chadd, R. P., Extence, C. A., & Brazier, R. E. (2017). Quantifying submerged deposited fine sediments in rivers and streams using digital image analysis. *River Research and Applications, 33*(10), 1585-1595.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research, 257*(3), 845-858.

Varsha, V., Singh, J., & Bala, M. (2017). Tabu search based energy efficient clustering protocol for wireless sensor networks. *Global Journal of Computers & Technology, 5*(2), 302-309.

Vijayalakshmi, K., & Anandan, P. (2018). A multi objective tabu particle swarm optimization for effective cluster head selection in WSN. *Cluster Computing*, 1-8.

Vob, S. (2001). Meta-heuristics: The state of the art. In *Proceedings of the Workshop on Local Search for Planning and Scheduling* (pp. 1–23). Heidelberg: Springer.

Wajgi, D., & Thakur, N. V. (2012). Load balancing algorithms in wireless sensor network: A survey. *International Journal of Computer Networks and Wireless Communications, 2*(4), 456-460.

Wang, D., & Ni, Y. (2012). Wireless sensor networks for earthquake early warning systems of railway lines. In *Proceedings of the 1st International Workshop* *on High-Speed and Intercity Railways* (pp. 417-426). Springer, Berlin, Heidelberg.

Wang, Q., & Balasingham, I. (2010). Wireless sensor networks - An introduction. *Wireless Sensor Networks: Application-Centric Design* (pp. 1-13). InTech Open Access Publisher.

Wang, S. (2018). Improved swarm intelligence algorithm for protein folding prediction. *Cluster Computing*, 1-10.

Wang, Y., Chen, H., Wu, X., & Shu, L. (2016). An energy-efficient SDN based sleep scheduling algorithm for WSNs. *Journal of Network and Computer Applications, 59*, 39-45.

Wang, Z. X., Zhang, M., Gao, X., Wang, W., & Li, X. (2017). A clustering WSN routing protocol based on node energy and multipath. *Cluster Computing*, 1-13.

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *265*(3), 843-859.

Xenakis, A., Foukalas, F., & Stamoulis, G. (2016). Cross-layer energy-aware topology control through simulated annealing for WSNs. *Computers & Electrical Engineering, 56*, 576-590.

Xhafa, F., Alba, E., Dorronsoro, B., Duran, B., & Abraham, A. (2008). Efficient batch job scheduling in grids using cellular memetic algorithms. *Journal of Mathematical Modelling and Algorithms*, *7*(2), 217–236.

Xhafa, F., Gonzalez, J. A., Dahal, K. P., & Abraham, A. (2009). A GA(TS) hybrid algorithm for scheduling in computational grids. In *Proceedings of the 4th*

International Conference on Hybrid Artificial Intelligence Systems (pp. 285–292). Salamanca. doi:10.1007/978-3-642-02319-4_34.

Xhafa, F., Kolodziej, J., Barolli, L., & Fundo, A. (2011). A GA+TS hybrid algorithm for independent batch scheduling in computational grids. In *Proceedings of the 14th International Conference on Network-Based Information Systems* (pp. 229–235). Tirana. doi:10.1109/NBiS.2011.41.

Xiu-li, R., Hong-wei, L., & Yu, W. (2008). Multipath routing based on ant colony system in wireless sensor networks. In *2008 International Conference on Computer Science and Software Engineering,* (Vol. 3, pp. 202-205). Wuhan: IEEE. doi: 10.1109/CSSE.2008.1140

Yan, J. F., Gao, Y., & Yang, L. (2011). Ant colony optimization for wireless sensor networks routing. In *2011 International Conference on Machine Learning and Cybernetics* (Vol. 1, pp. 400-403). Guilin: IEEE. doi: 10.1109/ICMLC.2011.6016670

Yang, C. H., Lin, Y. S., Chuang, L. Y., & Chang, H. W. (2017). A particle swarm optimization-based approach with local search for predicting protein folding. *Journal of Computational Biology, 24*(10), 981-994.

Yang, J., Xu, M., Zhao, W., & Xu, B. (2010). A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks. *Sensors, 10*(5), 4521-4540.

Yang, X. S. (2014a). *Nature-Inspired Optimization Algorithms*. Amsterdam: Elsevier.

Yang, X. S. (2014b). Swarm intelligence based algorithms: A critical analysis. *Evolutionary Intelligence*, *7*(1), 17-28.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing* (pp. 210-214). IEEE.

Yang, Z., Ping, S., Aijaz, A., & Aghvami, A. H. (2018). A global optimization-based routing protocol for cognitive-radio-enabled smart grid AMI networks. *IEEE Systems Journal, 12*(1), 1015-1023.

Ye, Z., & Mohamadian, H. (2014). Adaptive clustering based dynamic routing of wireless sensor networks via generalized ant colony optimization. In G. Lee (Ed.), *IERI Procedia* (Vol. 10, pp. 2-10). Elsevier. doi:10.1016/j.ieri.2014.09.063

Yildiz, H. U., Bicakci, K., Tavli, B., Gultekin, H., & Incebacak, D. (2016). Maximizing wireless sensor network lifetime by communication/computation energy optimization of non-repudiation security service: Node level versus network level strategies. *Ad Hoc Networks, 37*(2), 301-323.

Yoshikawa, M., & Otani, K. (2010). Ant colony optimization routing algorithm with tabu search. In P*roceedings of the International Multiconference of Engineers and Computer Scientists* (Vol. 3, pp. 17-19).

Yousif, Y. K., Badlishah, R., Yaakob, N., & Amir, A. (2018). An energy efficient and load balancing clustering scheme for wireless sensor network (WSN) based on distributed approach. *Journal of Physics: Conference Series, 1019,* 1-8.

Zakaria, Y., & Michael, K. (2017). An integrated cloud-based wireless sensor network for monitoring industrial waste water discharged into water sources. *Wireless Sensor Network, 9*, 290- 301.

Zapfel, G., Braune, R., & Bogl, M. (2010). *Metaheuristic Search Concepts a Tutorial with Applications to Production and Logistics*. Heidelberg: Springer.

Zeng, B., & Dong, Y. (2016). An improved harmony search based energy-efficient routing algorithm for wireless sensor networks. *Applied Soft Computing, 41*, 135-147.

Zhang, H., Zhang, S., & Bu, W. (2014). A clustering routing protocol for energy balance of wireless sensor network based on simulated annealing and genetic algorithm. *International Journal of Hybrid Information Technology*, *7*(2), 71-82.

Zhao, H., Pei, Z., Jiang, J., Guan, R., Wang, C., & Shi, X. (2010). A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony. *Advances in Swarm Intelligence* (pp. 558-565). Springer Berlin Heidelberg.

Zhou, F., Trajcevski, G., Tamassia, R., Avci, B., Khokhar, A., & Scheuermann, P. (2017). Bypassing holes in sensor networks: Load-balance vs. latency. *Ad Hoc Networks, 61*, 16-32.

Zhu, X., Dai, L., & Wang, Z. (2015). Graph coloring based pilot allocation to mitigate pilot contamination for multi-cell massive MIMO systems. *IEEE Communications Letters, 19*(10), 1842-1845.

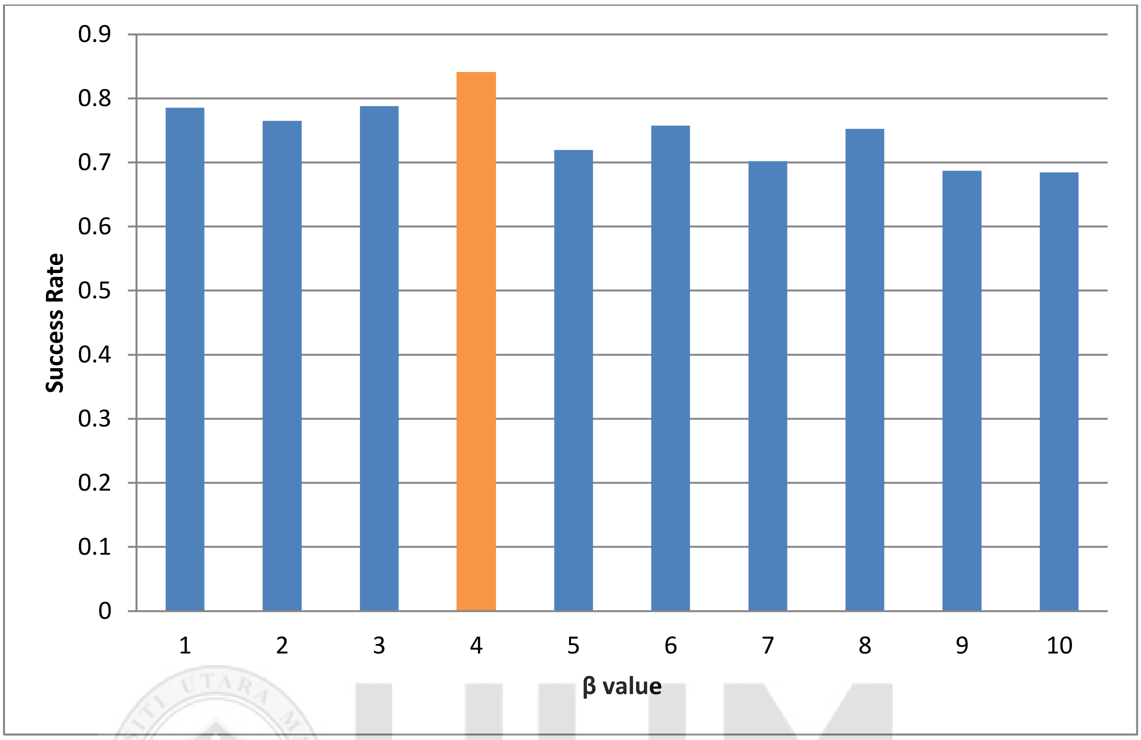
Zou, Z., & Qian, Y. (2018). Wireless sensor network routing method based on improved ant colony algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1-8.

Zungeru, A. M. (2013). *Energy-Efficient Routing Algorithms Based on Swarm Intelligence for Wireless Sensor Networks* (Doctoral dissertation). University of Nottingham, England.

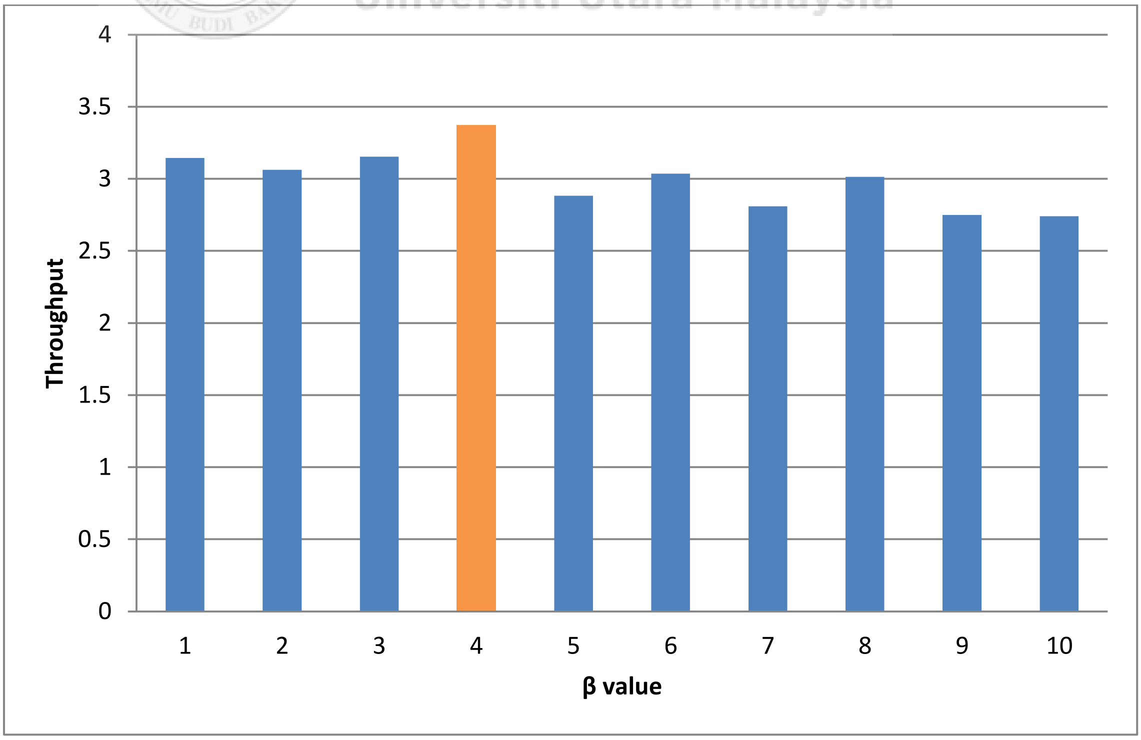
Zungeru, A. M., Ang, L. M., & Seng, K. P. (2012a). Performance evaluation of ant-based routing protocols for wireless sensor networks. *International Journal of Computer Science Issues, 9*(3), 388-397.

Zungeru, A. M., Ang, L. M., & Seng, K. P. (2012b). Termite-hill: Performance optimized swarm intelligence based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications*, *35*(6), 1901-1917.

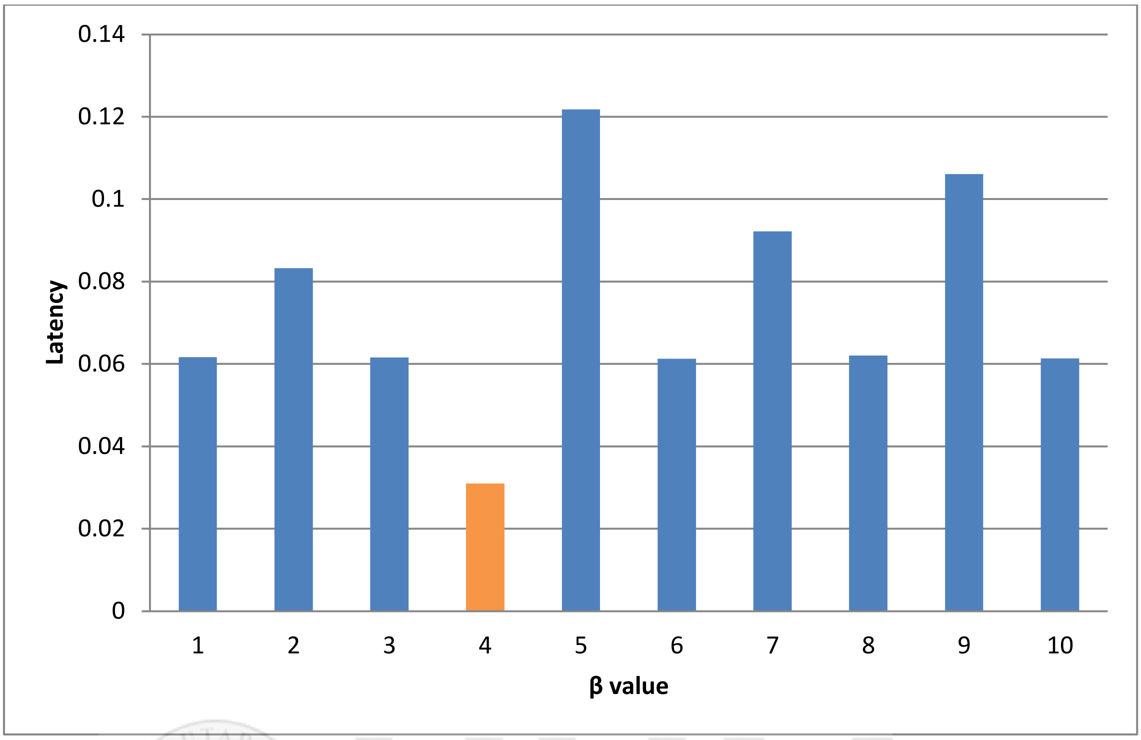
APPENDICES



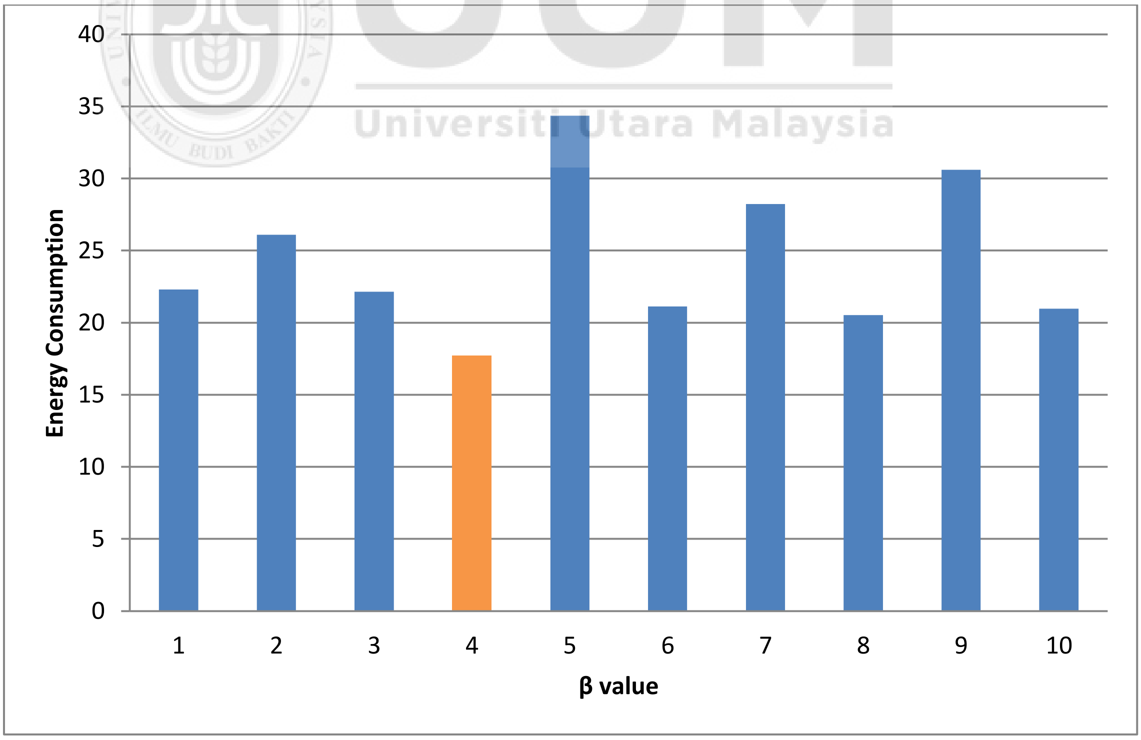
Appendix I: Effect of β value to the success rate of EACS(TS) algorithm



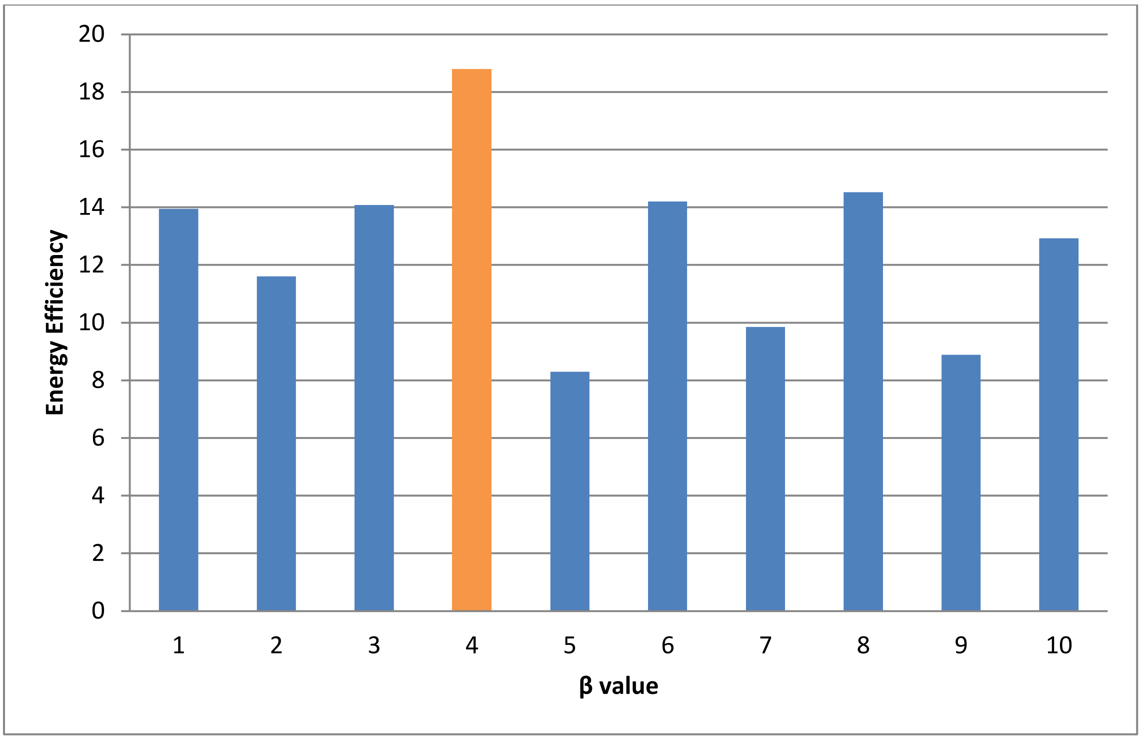
Appendix II: Effect of β value to the throughput of EACS(TS) algorithm



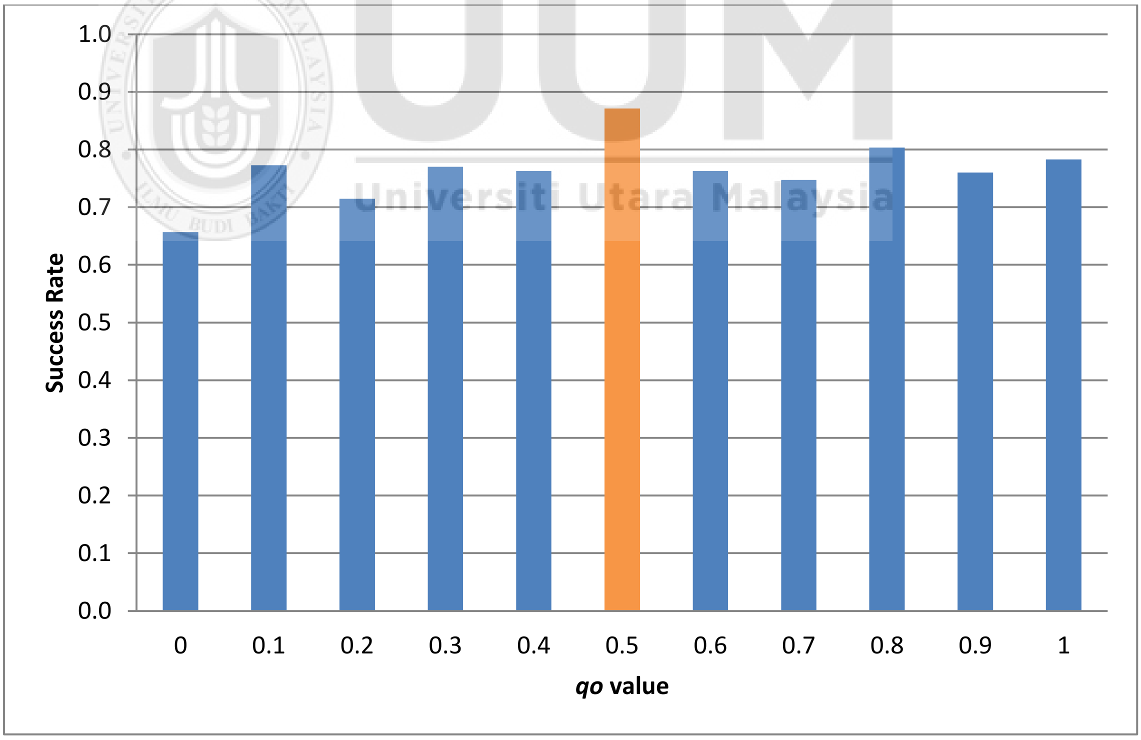
Appendix III: Effect of β value to the latency of EACS(TS) algorithm



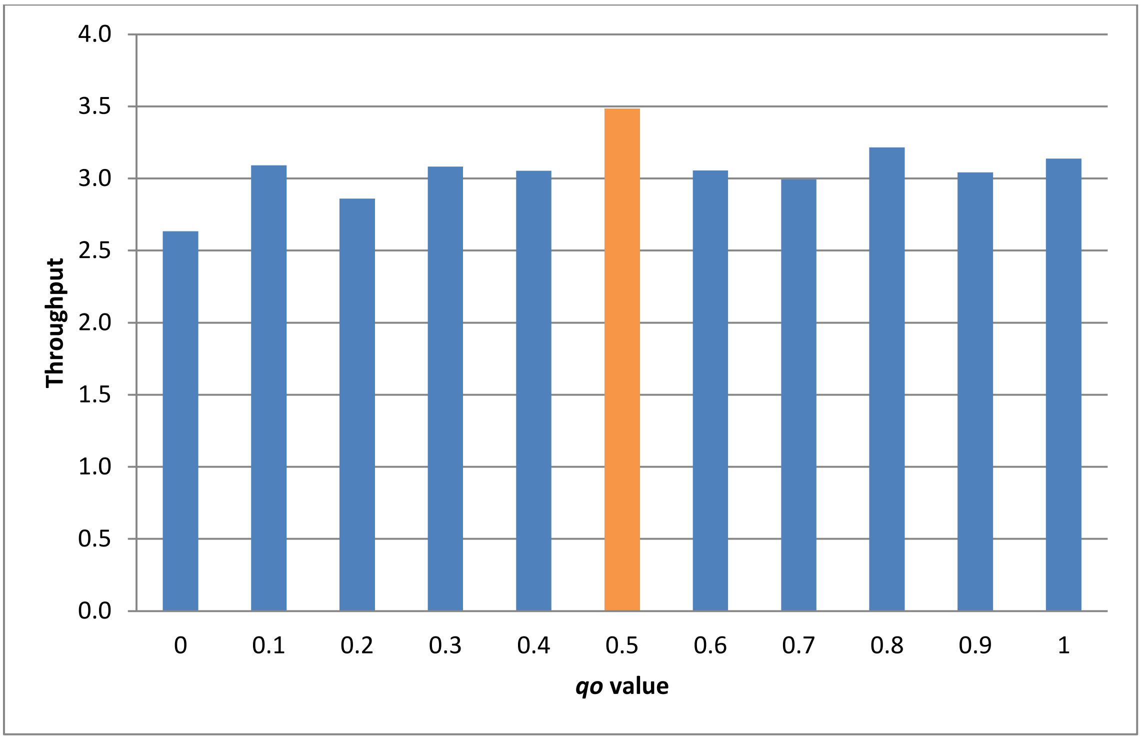
Appendix IV: Effect of β value to the energy consumption of EACS(TS) algorithm



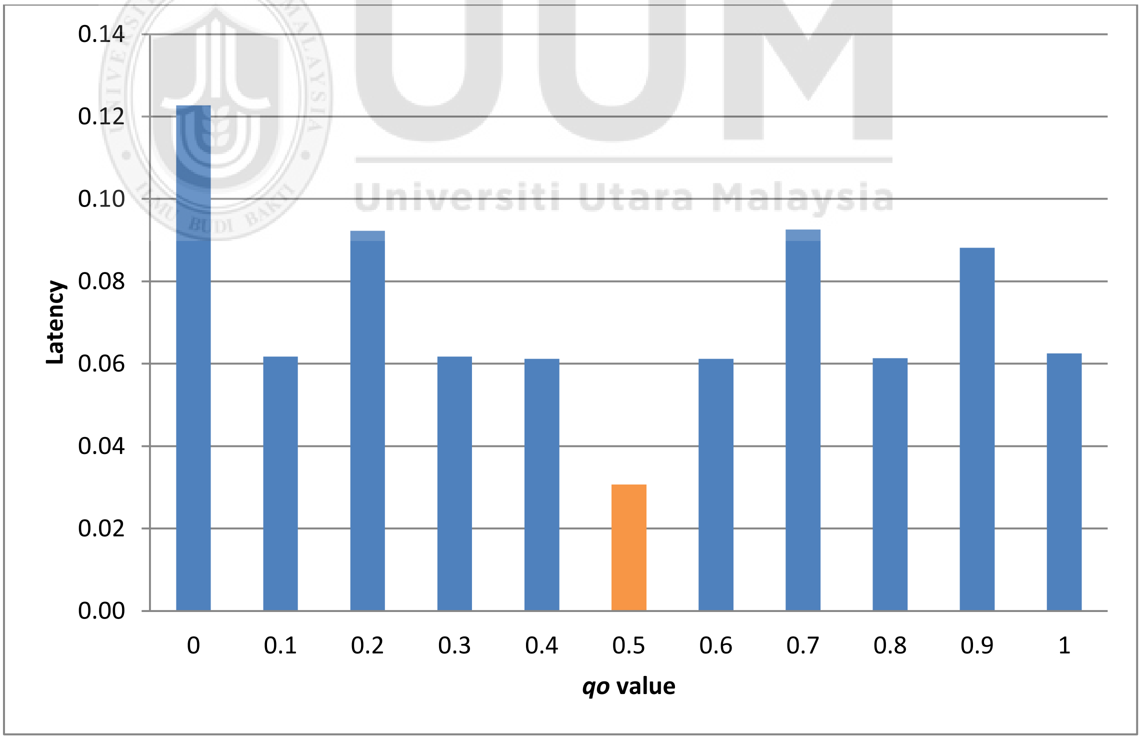
Appendix V: Effect of β value to the energy efficiency of EACS(TS) algorithm



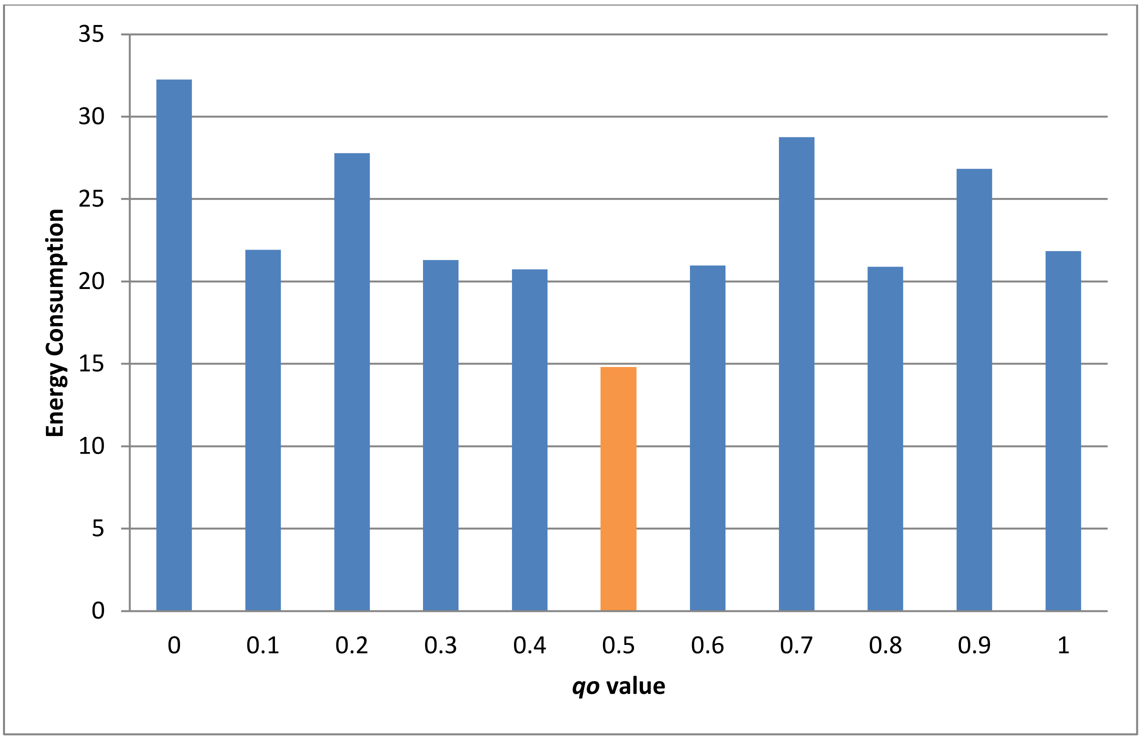
Appendix VI: Effect of q_o value to the success rate of EACS(TS) algorithm



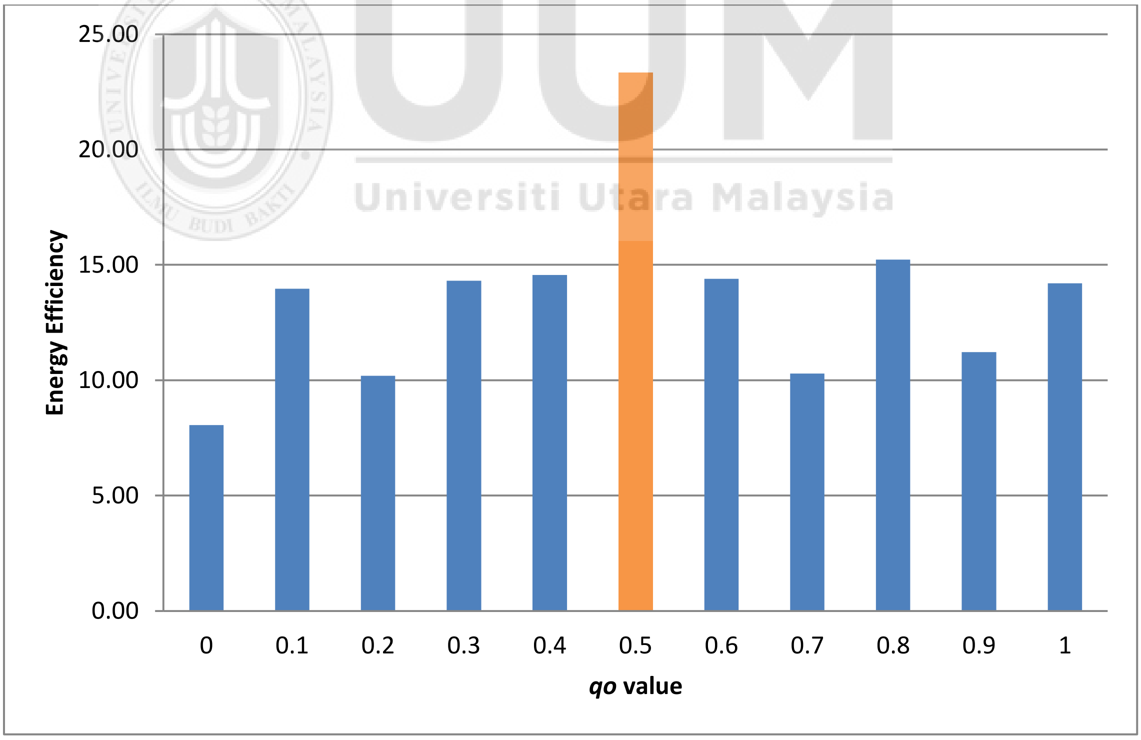
Appendix VII: Effect of q_o value to the throughput of EACS(TS) algorithm



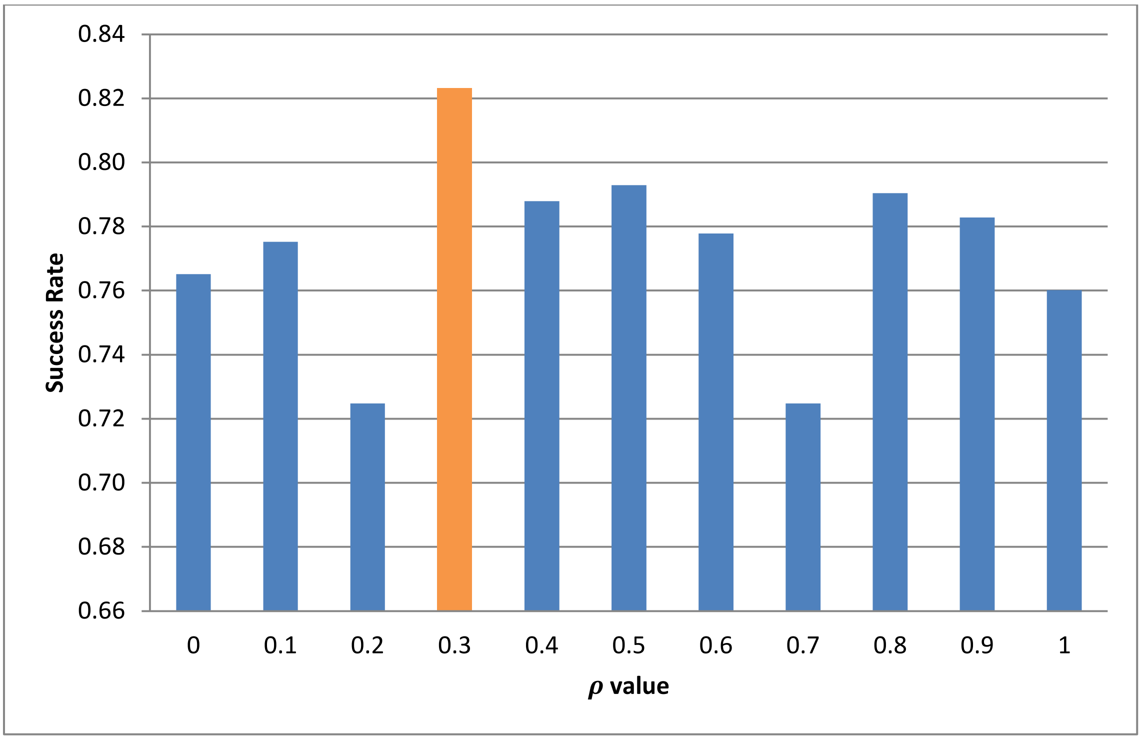
Appendix VIII: Effect of q_o value to the latency of EACS(TS) algorithm



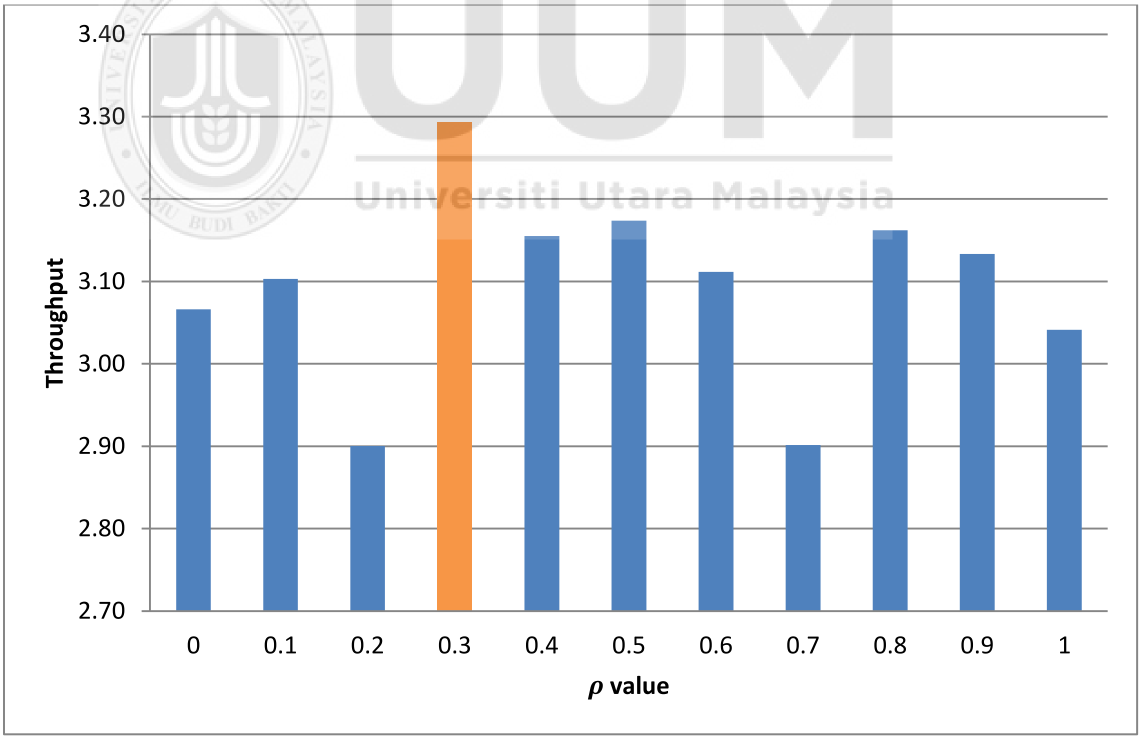
Appendix IX: Effect of q_o value to the energy consumption of EACS(TS) algorithm



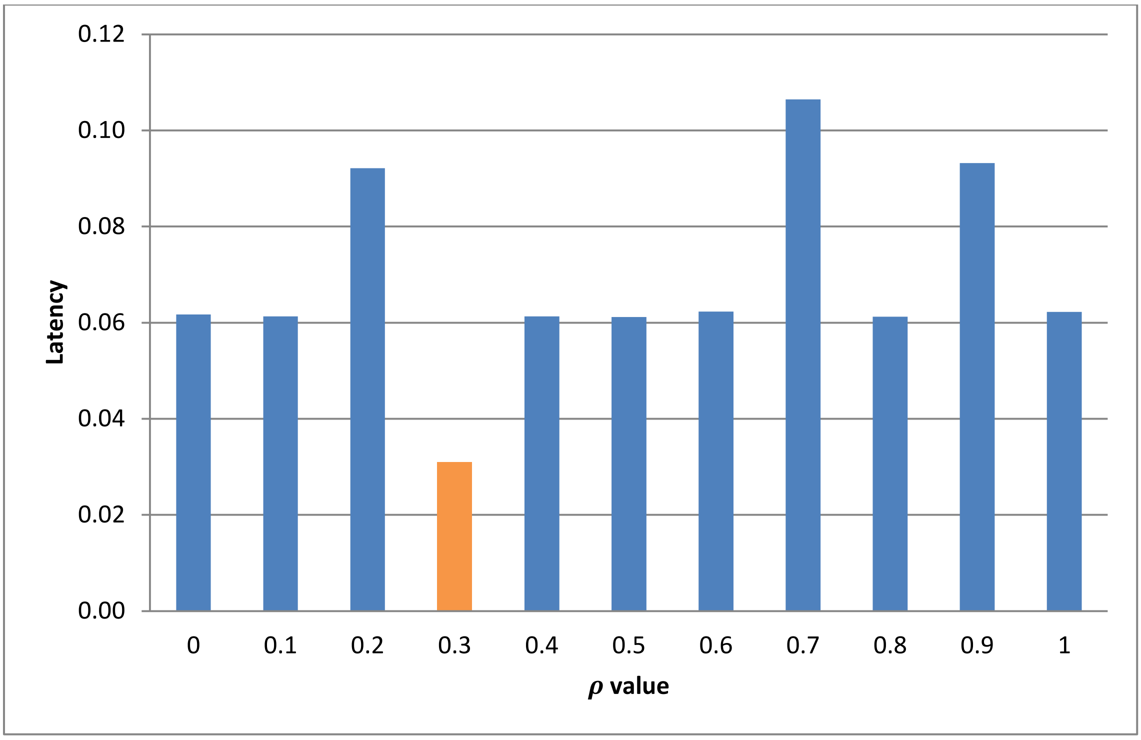
Appendix X: Effect of q_o value to the energy efficiency of EACS(TS) algorithm



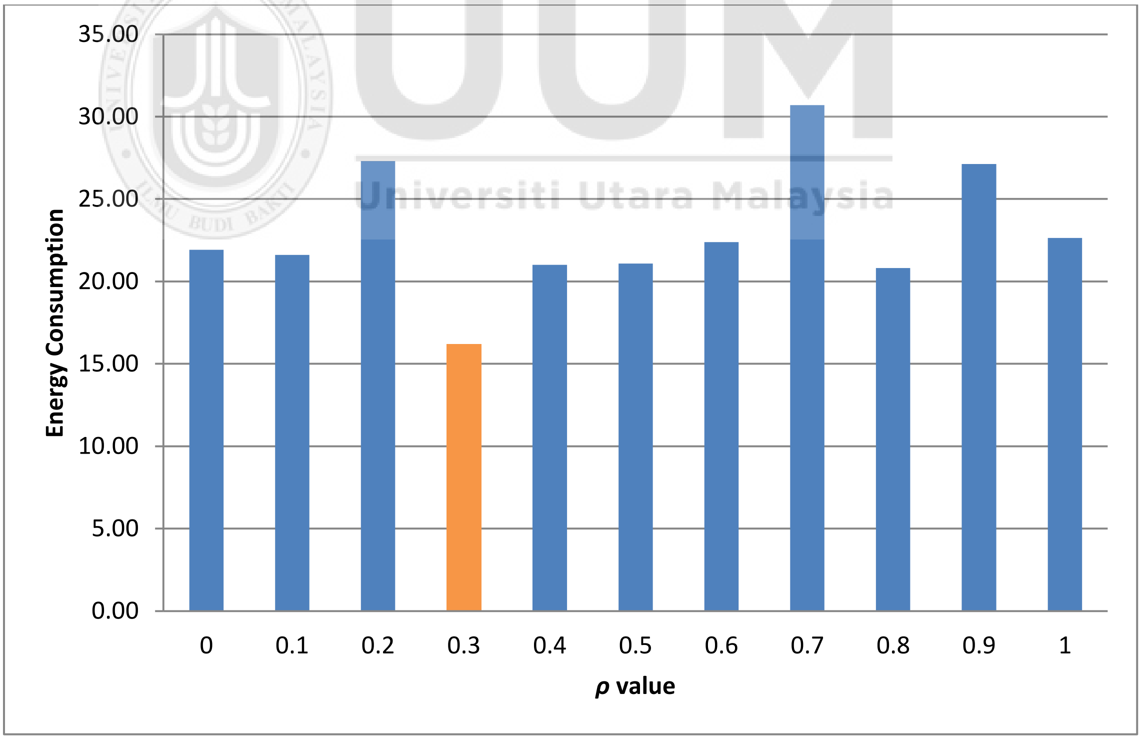
Appendix XI: Effect of ρ value to the success rate of EACS(TS) algorithm



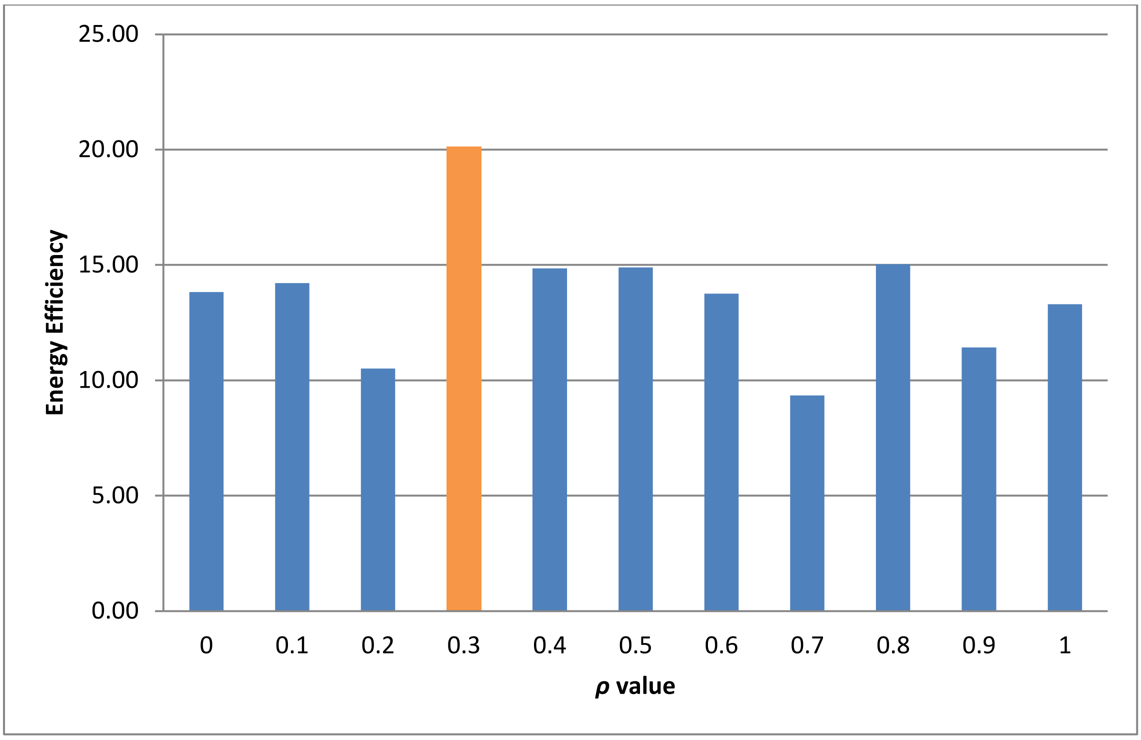
Appendix XII: Effect of ρ value to the throughput of EACS(TS) algorithm



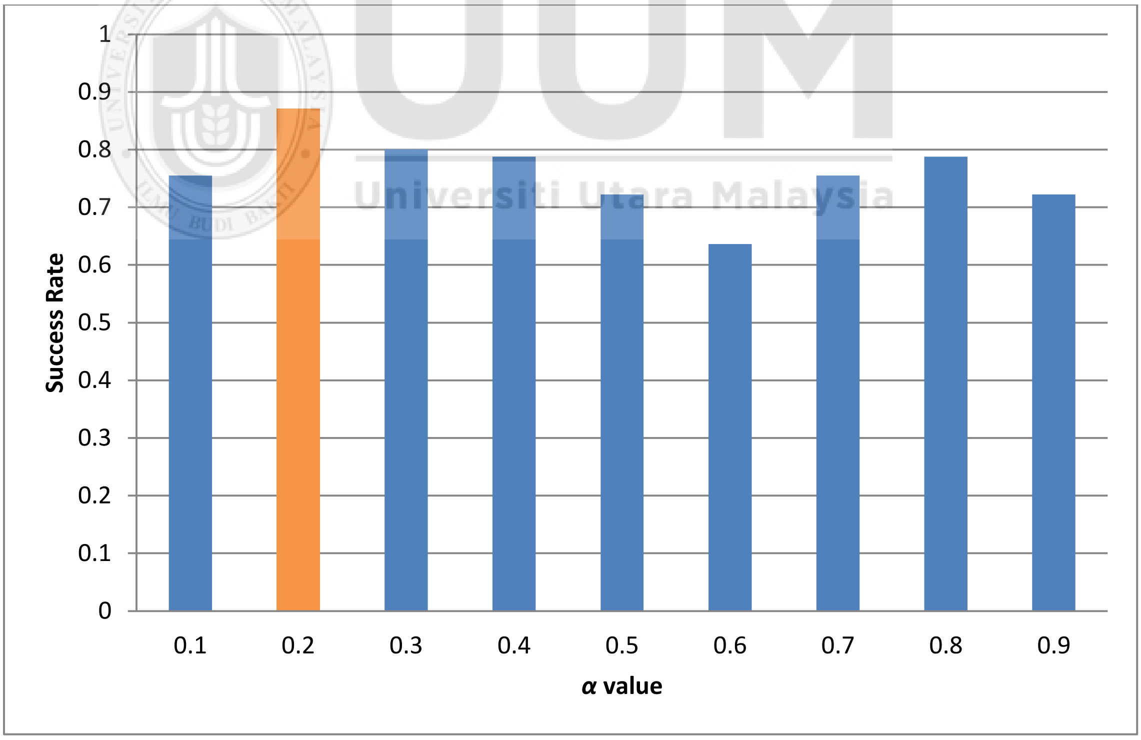
Appendix XIII: Effect of ρ value to the latency of EACS(TS) algorithm



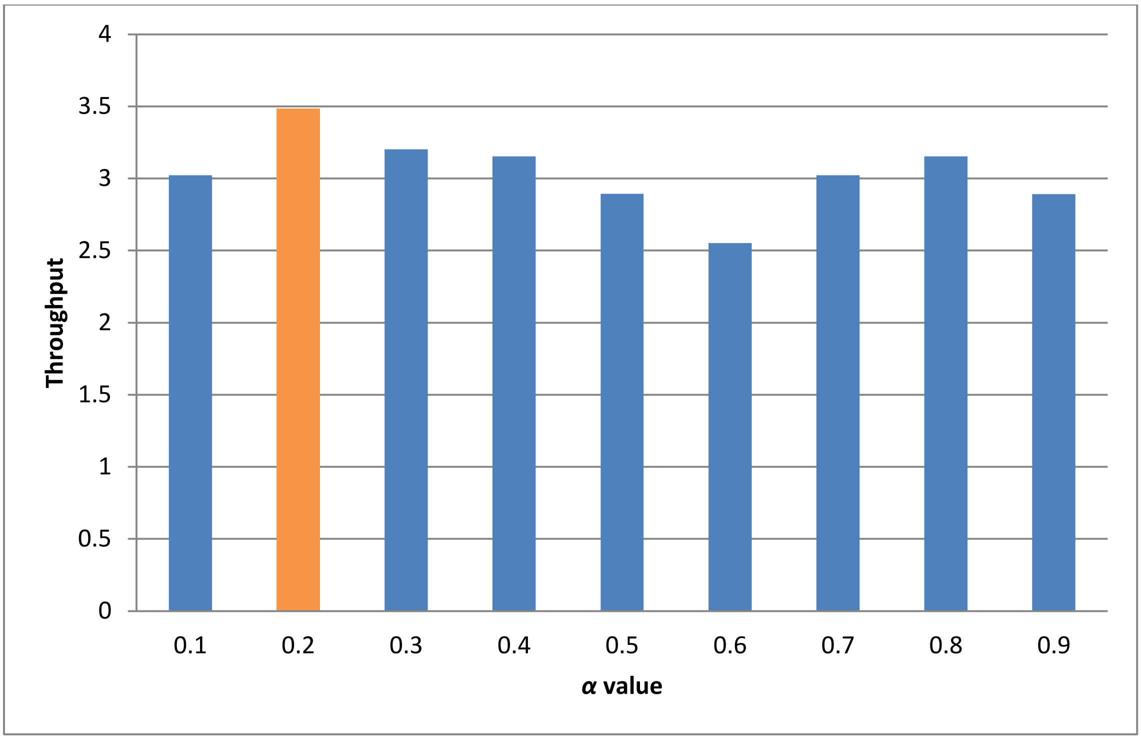
Appendix XIV: Effect of ρ value to the energy consumption of EACS(TS) algorithm



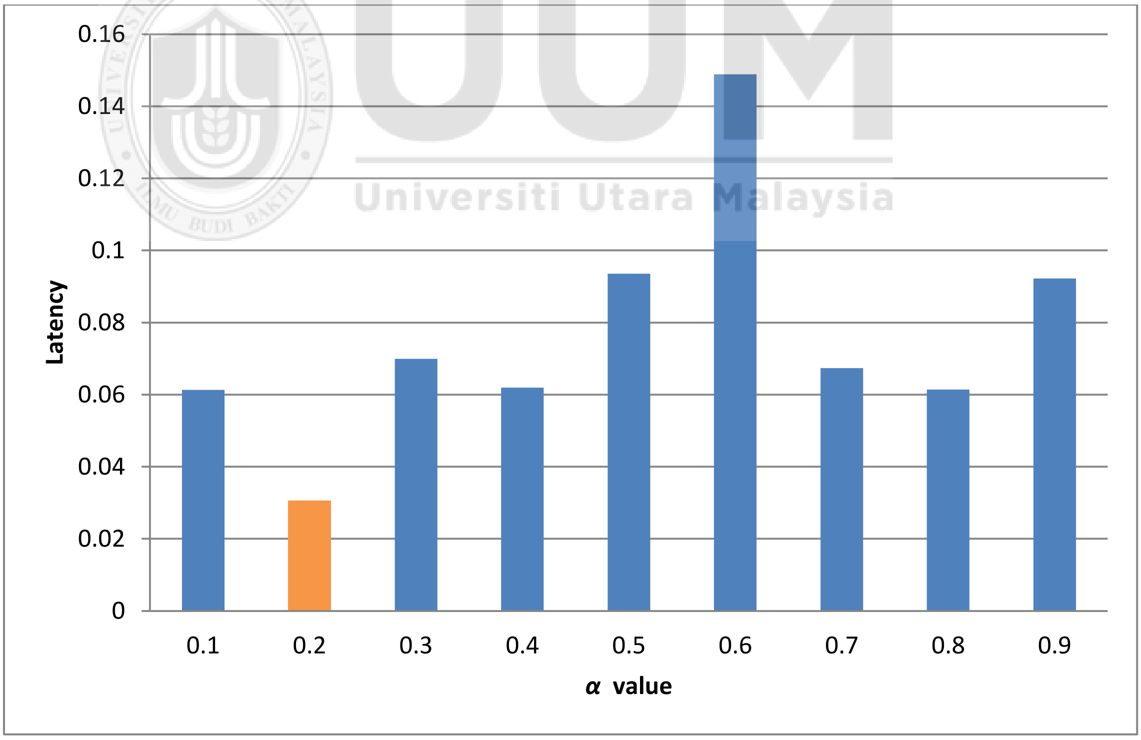
Appendix XV: Effect of ρ value to the energy efficiency of EACS(TS) algorithm



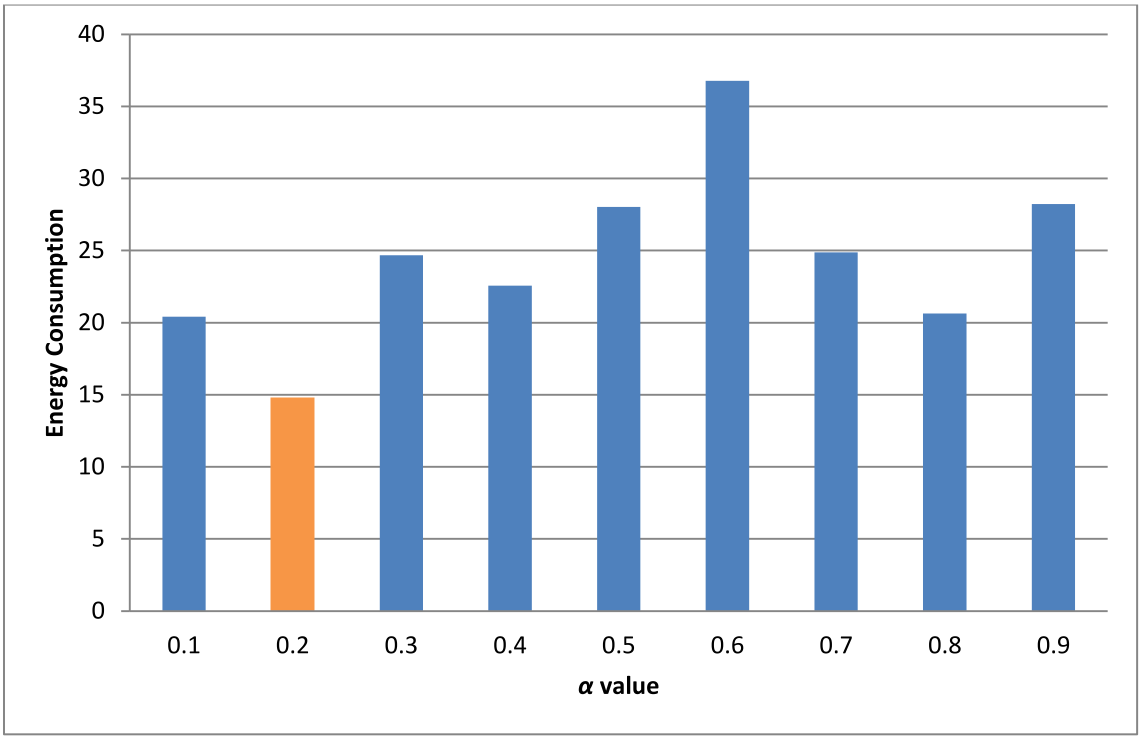
Appendix XVI: Effect of α value to the success rate of EACS(TS) algorithm



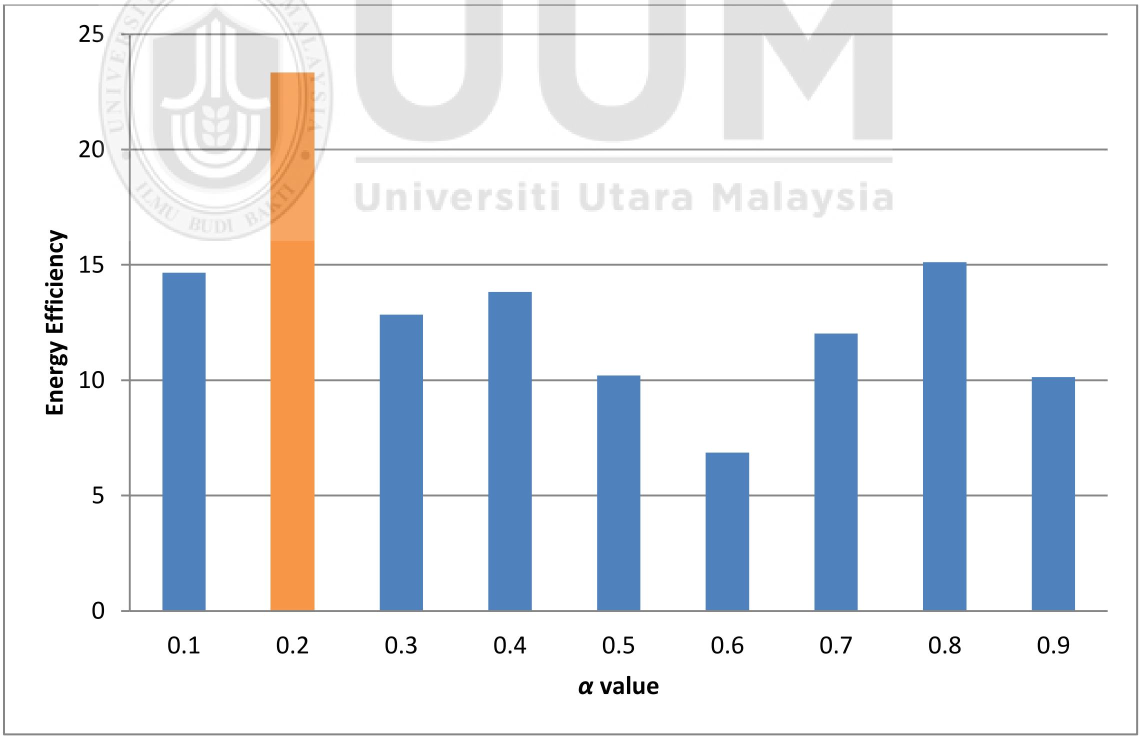
Appendix XVII: Effect of α value to the throughput of EACS(TS) algorithm



Appendix XVIII: Effect of α value to the latency of EACS(TS) algorithm



Appendix XIX: Effect of α value to the energy consumption of EACS(TS) algorithm



Appendix XX: Effect of α value to the energy efficiency of EACS(TS) algorithm