IMPROVING ROBOT DARWINIAN PARTICLE SWARM OPTIMIZATION USING QUANTUM-BEHAVED SWARM THEORY FOR ROBOT EXPLORATION AND COMMUNICATION

DUAA MEHIAR

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

IMPROVING ROBOT DARWINIAN PARTICLE SWARM OPTIMIZATION USING QUANTUM-BEHAVED SWARM THEORY FOR ROBOT EXPLORATION AND COMMUNICATION

DUAA MEHIAR

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of the candidate: Duaa Mehiar

Registration/Matric No: WHA150031

Name of the Degree: Doctor of Philosophy

Title of Dissertation: Improving Robot Darwinian Particle Swarm Optimization using Quantum-Behaved Swarm Theory for Robot Exploration and Communication

Field of Study: Artificial Intelligence (Robotics)

I do solemnly and sincerely declare that:

(1) I am the sole author /writer of this work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealings and any expert or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship has been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyrights to this Work to the University of Malaya (UM), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained actual knowledge;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether internationally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature: Date: 10/2/21

Subscribed and solemnly declared before,

Witness Signature: Date: 17/2/21
Name:
Designation:

IMPROVING ROBOT DARWINIAN PARTICLE SWARM OPTIMIZATION USING QUANTUM-BEHAVED SWARM THEORY FOR ROBOT EXPLORATION AND COMMUNICATION

ABSTRACT

Despite the significance of the Robotic Darwinian Particle Swarm Optimization (RDPSO) algorithm on swarm-robot exploration and communication, there remain notable gaps such as premature and slow convergence, collisions between robots, and communication breaks and constraints. The quantum computing theory has several advantages that can improve the searching capabilities of PSO-based algorithms. However, there has yet an attempt to adopt quantum behaviour onto robot-based system such as the RDPSO. In this study, a new algorithm called the Quantum Robot Darwinian Particle Swarm Optimization (QRDPSO) is contributed with the hypothesis that quantum behaving particles can address the RDPSO main gaps; i.e. to improve the exploration and communication performance of a swarm robotic system. In terms of convergence time, the experiment done shows the QRDPSO algorithm is faster to reach an optimal solution than the RDPSO algorithm. The QRDPSO algorithm also shows tolerance to premature convergence compared to RDPSO. This study also contributed a distributed swarm navigation strategy that allows the QRDPSO robots to communicate directly with other robots in the swarm. Two popular communication schemas over wireless sensor network have been adopted and tested on the QRDPSO, the Multi-hop Routing Algorithm with Low Energy Adaptive Clustering Hierarchy (MR-LEACH) and the mobile ad hoc communication network (MANET). The QRDPSO algorithm with MR-LEACH consumes less power with energy consumption at 48% compared to the QRDPSO with MANET at 63%. Less power allows the MR-LEACH to increase lifetime for the nodes more than MANET while reducing interruptions between robots but not faster to reach the optimal solution than the QRDPSO algorithm with MANET. The QRDPSO with MANET needs 180 iterations, while the QRDPSO with MR-LEACH needs 202 iterations. The predecessor, RDPSO, needs 210 iterations for comparison to reach a victim. Given the importance of a swarm's sustainability; swarm not losing robots, able to conserve energy and explore farther, the MR-LEACH schema is proposed to complement the QRDPSO communication.

Keywords: Swarm robotics, particle swarm optimization (PSO), quantum behaving particles, robot exploration, robot communication

MEMPERBAIKI ROBOT *DARWINIAN PARTICLE SWARM OPTIMIZATION* MENGGUNAKAN TEORI KAWANAN DENGAN KELAKUAN KUANTUM UNTUK PENJELAJAHAN DAN KOMUNIKASI ROBOT

ABSTRAK

Walaupun algoritma *Robotic Darwinian Particle Swarm Optimization (RDPSO)* berkepentingan dalam penjelajahan dan komunikasi kawanan robot, masih terdapat jurang-jurang yang ketara seperti penumpuan pramatang dan lembab, perlanggaran sesama robot, serta kekangan dan gangguan komunikasi. Teori pengkomputeran kuantum mempunyai beberapa kelebihan yang dapat meningkatkan keupayaan pencarian bagi algoritma berasaskan *PSO*, namun kini masih tiada percubaan untuk menyesuaikan perilaku kuantum ke atas sistem berdasarkan robot seperti *RDPSO*. Dalam kajian ini, satu algoritma baru yang dikenali sebagai *Quantum Robot Darwinian Particle Swarm Optimization (QRDPSO)* telah disumbangkan dengan hipotesis bahawa zarah berkelakuan kuantum dapat menangani jurang utama *RDPSO*; iaitu untuk meningkatkan prestasi penjelajahan dan komunikasi satu sistem kawanan robotik. Dari segi masa penumpuan, eksperimen yang telah dilakukan menunjukkan algoritma *QRDPSO* lebih pantas untuk mencapai penyelesaian yang optimum berbanding algoritma *RDPSO*. Algoritma *QRDPSO* juga menunjukkan toleransi kepada penumpuan pramatang berbanding dengan *RDPSO*. Kajian ini juga turut menyumbang satu strategi navigasi yang mengagihkan kawanan robot supaya robot-robot *QRDPSO* dapat berkomunikasi secara langsung dengan robot-robot lain di dalam kawanan. Dua skema komunikasi popular untuk rangkaian sensor tanpa wayar, *Multi-hop Routing Algorithm with Low Energy Adaptive Clustering Hierarchy (MR-LEACH)* dan *mobile ad hoc communication network (MANET)* telah diterima pakai dan diuji ke atas *QRDPSO*. Algoritma *QRDPSO* dengan *MR-LEACH* menunjukkan penggunaan tenaga yang lebih rendah iaitu sebanyak

48% berbanding algoritma *QRDPSO* dengan *MANET* sebanyak 63%. Penjimatan kuasa ini membenarkan *MR-LEACH* memanjangkan jangka-hayat nod-nod berbanding *MANET,* serta pengurangan gangguan komunikasi antara robot tetapi lebih lembab untuk mencapai penyelesaian yang optimum berbanding algoritma *QRDPSO* dengan *MANET*. Algoritma *QRDPSO* dengan *MANET* memerlukan 180 lelaran, sementara *QRDPSO* dengan *MR-LEACH* memerlukan 202 lelaran. Sebagai perbandingan, algoritma *RDPSO* yang terdahulu memerlukan 210 lelaran. Memandangkan pentingnya kemampanan satu kawanan; kawanan tidak kehilangan robot, dapat menjimatkan tenaga dan berupaya menjelajah lebih jauh, skema *MR-LEACH* adalah dicadangkan untuk melengkapi komunikasi *QRDPSO*.

Kata kunci: Kawanan robot, *particle swarm optimization* (PSO), zarah dengan kelakuan quantum, penjelajahan robot, komunikasi robot

ACKNOWLEDGEMENTS

With the name of ALLAH (most gracious, most merciful), I am grateful to Him in blessing me with the knowledge, giving me the courage to tackle the problems and always help me in each step of my life.

In successful completion of this work, I would like to express my deepest appreciations to my supervisors. Firstly, to Dr Zati Hakim Azizul Hasan for her critical input in the discussions and her valuable time and dedication in shaping this thesis. Secondly, to Prof Dr Loo Chu Kiong for his advice, encouragement and support. I cannot find the right words to express my admiration and sincere gratitude to my supervisors, who allowed me to be a part of this work. They helped, encouraged and motivated me in every step of research, during the time I have spent with them. Their suggestions were always valuable, and their technical comments lead to the completion of this research project. I would also like to thank my thesis examiners for their critical feedback have improved the work's writing and presentation.

Finally, my deepest thanks are expressed to all my family members for their encouragement and support, especially my kind mother and beautiful sister Dima.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

RDPSO	**Robotic Darwinian Particle Swarm Optimization**
PSO	**Particle Swarm Optimization**
t	Discrete time
v_n	Velocity vector of robot n
x_n	Position vector of robot n
w_n	Inertial influence on robot n
ρ_1	Local best (cognitive) coefficient
ρ_2	Global best (social) coefficient
ρ_3	Obstacle avoidance coefficient
ρ_4	network communication coefficient
χ_1	Best position of the robot regarding its local (cognitive) solution.
χ_2	Best position of the robot regarding its global (social) solution
χ_3	Best position of the robot regarding its obstacle avoidance
χ_4	Best position of the robot regarding its network topology
r_i	Random vectors, i=1,2,3,4
$f(x_n[t])$	Sensed solution of robot n at position xn
$dmax$	Maximum communication distance
q_{min}	Minimum signal quality
R_w	Obstacle sensing radius
N_T	Population of robots
N_s	Total number of robots in subgroup s,
N_I	The initial number of robots in a subgroup
N_{min}	Minimum number of robots required to form a subgroup

QPSO	**Quantum Particle Swarm Optimization**
N	Space dimensions
M	The number of particles.
P^{j}_{n}	centre of the N-Dimension Hilbert space with a δ potential well.
C	*mbest* positions vector
µ	Random vectors
QRDPSO	**Quantum Robotic Darwinian Particle Swarm Optimization**
$\sigma(x_i(t))$	standard deviation value of the current position of the current particle
$im_{i,j}(t)$	sensing function
MR-LEACH	**MR-Low Energy Adaptive Clustering Hierarchy**
p	percentage of nodes to be elected as cluster heads
r	current round
G	set of nodes that have not been cluster heads in the last 1/P rounds
AODV	Ad hoc On-demand Distance Vector
MRS	Multi-Robot System
MRSim	Multi-Robot Simulator

CHAPTER 1: INTRODUCTION

1.1 Background

Over the past decades, many scientists and engineers have studied nature's best and time-tested patterns and strategies. Within the existing biological architectures, swarm societies revealed that relatively unsophisticated agents with limited capabilities, such as ants or birds, could accomplish complex tasks necessary for their survival cooperatively. Those simplistic systems embrace all the conditions necessary to survive, thus embodying cooperative, competitive and adaptive behaviours. In the never-ending battle to advance artificial human-made mechanisms, computer scientists simulated the first swarm behaviour designed to mimic birds' flocking behaviour in the late eighties.

Ever since, many other fields, such as robotics, have benefited from the fault-tolerant mechanism inherent to swarm intelligence. Flocks and swarms are intrinsically cooperative and even competitive, behaviours observed in birds and most insects which survives natural evolution. The way flocks and swarms cope and adapt to social life difficulties has fascinated robotics researchers to embrace biologically inspired computational algorithms into robot evolution. One common observation which inspires the fascination includes how birds worked together to spot food. The probability is high for the flock of birds to find a location with the highest amount of food in the area following a trajectory which combines three directions (Floreano & Mattiussi, 2008):

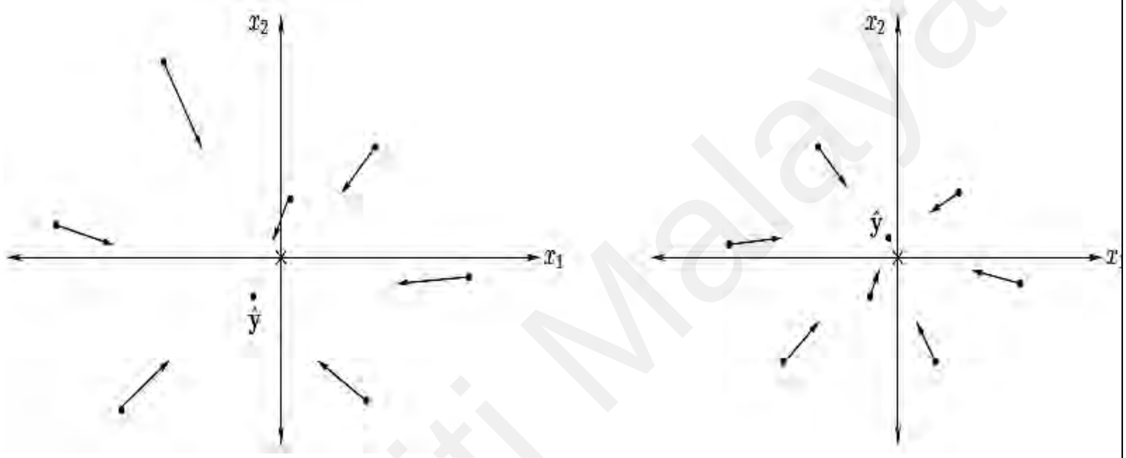
a) flying in the same way,
b) Return to the location where the bird found a lot amount of food, and
c) Move towards the neighbouring bird that cries when food is abundance.

Those principles and how nature copes with life's difficulties brought forth the research towards nature-inspired, more widely known as biologically inspired, humanmade designs. From biologically inspired robots mimicking birds' kinematics (Couceiro et al., 2012) to complex collective aggregation of robots mimicking swarms of insects, robotics has benefitted the most from biologically inspired evolution over the past few years. Kennedy & Eberhart (1995) proposed the PSO as a population-based algorithm which contributes to global optimisation over continuous search spaces.

In real life, global optimisation could mean how swarms conform a cooperative way to find food, by imitating a member or another that reaches the food during a particular search. The swarms move around and create a dynamic search pattern with each member changing speed and direction to emulate a particular swarm member that became the optimal solution. Each PSO swarm member is represented as a particle, where its velocity and position updated at each iteration. Therefore, the changes to a particle within the swarm are influenced by its interconnected neighbours' learning and experiences.

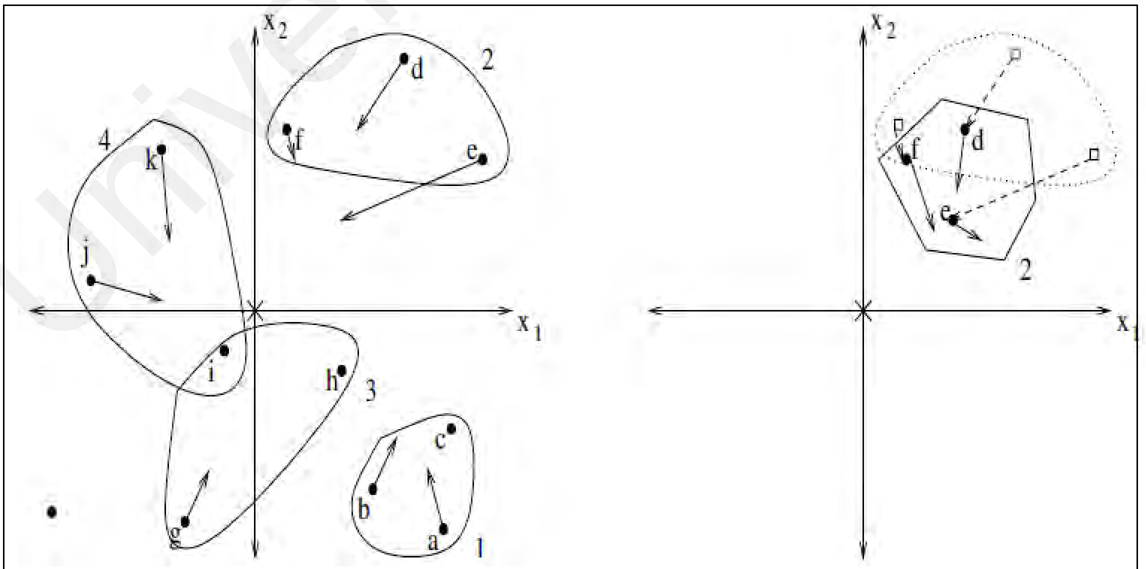
The PSO gained popularity among the robotics community due to ease of implementation as the algorithm has few parameters to adjust. It is also computationally inexpensive because the spatial ordering of particles in the neighbourhood is not required. Two versions of the PSO algorithm are usually discussed: the global best PSO (*gbest*) and the local best PSO (lbest). For the *gbest* PSO, each particle's neighbourhood is the entire swarm, which leads to high interconnectivity between particles. For the lbest PSO, smaller neighbourhoods are defined for each particle, offering diversity to the swarm. The *gbest* and lbest PSOs differ mainly due to their convergence characteristics.

Due to larger interconnectivity in *gbest* PSO, the swarm may converge faster than the lbest PSO, albeit with less diversity, leading to the immature selection of a less accurate solution. With more considerable diversity and careful attention given to the search space sub-parts, the lbest PSO is less susceptible to being trapped in local minima like the *gbest* PSO but takes longer time to converge. Figure. 1.1 and Figure. 1.2 show illustrations of *gbest* and lbest PSOs and Figure. 1.3 shows the PSO general process.



1. At time $t = 0$ 2. At time $t = 1$

Figure 1.1: Global best PSO illustration. Taken from Engelbrecht (2007)



1. Initial swarm 2. Second swarm

Figure 1.2: Local best PSO illustration. Taken from Engelbrecht (2007)

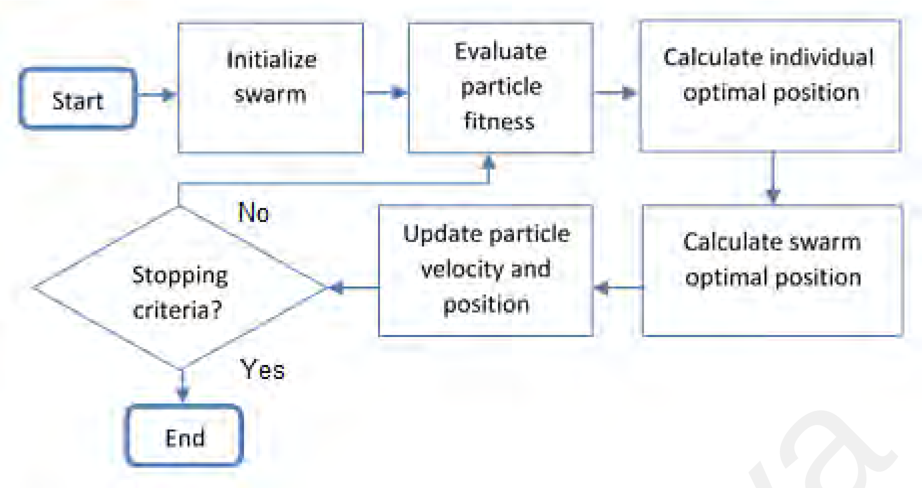


Figure 1.3: Flowchart of the PSO algorithm

The PSO consequently is a representation of the symbiotic cooperative model. It is stochastic by nature and by optimizing the mathematical functions, propose a fast convergence rate. The PSO is, however, limited in several crucial areas. The PSO is most likely to fall into one local extreme for problems with several local extremes and achieve the inaccurate result. According to Wang et al. (2018), this premature convergence happens when two factors appear, the swarm having less optimized functions, and losing particles' diversity quickly. The PSO will also achieve an inaccurate result when historical information calculated for the individual and swarm is not fully considered during updates. There is no guarantee the PSO will always achieve global optimization, although fundamentally, the algorithm provides the probability of global search.

The most significant limitation of the PSO for adoption into robotic applications has to be the nature of the particles' motion themselves. In theory, the particles are allowed movement in any direction and with any velocity permitted. The particles' interconnectivity also posed high communication traffic for the swarm. On the other hand, the robots have a limited range of mobility and individually, requires the capacity to perform a different kind of measurements according to their sensors. In terms of communication traffic, handling excess messages (broadcasts from each robot to all other robots in the swarm) at every time interval will not propose a practical update. Therefore, adjusted versions of the PSO have been proposed by robotics researchers focusing on various areas including modification of various parameters configurations, studying the topological structures to improve communication, and studying the discrete or multi-objective optimization within the PSO algorithm.

1.2 Motivation

The robotics community benefitted from modifying the PSO algorithm to suit many applications such as finding robot search paths, optimizing robot localization, and motion planning (Tang et al., 2017; Collinsm & Shen, 2017). In the industry, the PSO is used to find the optimal movement of robotic arms to improve productivity while in aerial robotics, swarms of drones are controlled to reach specific target using the PSO (Tarmizi et al., 2016; Djaneye-Boundjou et al., 2016; Spanogianopoulos et al., 2017; Alejo et al., 2014). One widespread impact of a swarm robotic system is in search and rescue (SaR) problems where the swarm is deployed to find the global best (victim), and the swarm performance is measured on how fast the victim is discovered.

Central to SaR are two key components, (1) a platform for distributed architecture and (2) a venue for exclusive inter-robot communication. A distributed architecture redefines the robotics swarms without a central task allocator. Gasser & Huhns (2014) proposed that swarm architectures are expensive computationally in a real-world application due to the central task allocator having to handle the number of dynamic behaviours generated by the high number of robots. In terms of inter-robot communication, the paradigm to use wireless communication as a medium so robots can openly exchange information within a network path is recommended (Farooq et al., 2010).

In particular, the SaR problem can happen in a hostile environment, disaster recovery, battlefields and space, for example, where communication infrastructure may be damaged or missing. Communication protocols such as the Mobile Ad Hoc Network (MANET) and Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy (MR-LEACH) can reduce interruptions to reduce team performance risk.

One extension to the PSO algorithm that conforms to SaR systems' requirements is the Darwinian PSO (DSPO) algorithm (Tillet et al., 2005). In DPSO, the Darwinian's survival of the fittest principle defines how the lbest PSO is organized. When a sub-group of the swarm gets stuck in a local optimum during a search, that sub-area is discarded. Another sub-area is searched instead. A reward and punish system is proposed to reinforce the swarms' learning. At every interval, sub-groups with increasing mobility get rewarded with new particles or an extra life. Sub-groups that are stagnant have the risk of losing particles or reduced swarm life. Fitness of all sub-group particles is calculated before the neighbourhood, and each particle's individual best positions are updated. Internally, particle teammates from the same sub-group cooperate while externally, different sub-groups compete to achieve global optimization, a concept termed as coopetition (Tsai, 2002).

What is missing from the DPSO are parameters that can handle the swarm robots' physical characteristics, i.e. for obstacle avoidance and maintaining communication, during real-time exploration. A series of articles discussed how the DPSO is extended to develop and test the Robotic Darwinian PSO (RDPSO) (Couceiro et al., 2011; Couceiro et al., 2012; Couceiro et al., 2015; Dadgar et al., 2017; Kumar et al., 2017; Sanchez et al., 2018).

The RDPSO is reported to overcome the limitation of convergence on multiple targets. Unlike the PSO and DPSO with its particles falling into sub-optimal solutions and unlikely to escape, the RDPSO has a mean to push its particles to wander and avoid the trapping altogether in the first place. An allocator is required in the PSO and DPSO to divide particles into sub-groups. In the RDPSO, a mechanism to generate sub-groups within the swarm is enabled. Traffic control is then introduced, so the number of messages shared between the robots is not overwhelming.

In a SaR scenario, the robots are released to explore the search space at random. Depending on the swarm initialization parameters, the robots move in different directions until a signal is received. A signal means a robot has found a potential victim. The higher the intensity of the signal, the closer a robot is to a potential victim. When the robot receives a signal, it checks the list of broadcasted signals to single out a signal with the highest intensity. The robot with maximum intensity is considered the best performing robot at that interval and gets rewarded by other robots in the swarm. For robots that cannot broadcast signal or transmitted only weak signal, they may face exclusion by the RDPSO algorithm.

The RDPSO has been tested on the Robot Operating System (ROS) framework to simulate a swarm of autonomous mobile robots for exploration and communication in SaR situation. Each robot is pre-programmed with initial position and initial velocity and information about the search space to be explored. The robots do not have prior knowledge regarding the location of the victims. The environment setup includes multiple static targets and obstacles. A universal grid map library is utilized to compute an intensity map to guide the robots' navigation autonomously. The swarm's performance for the SaR use case with multiple targets is measured in terms of speed and convergence accuracy.

The main aim of the RDPSO algorithm is to improve the efficiency of the PSO algorithm to allow the search to take place at a faster rate. The RDPSO extends the PSO algorithm using evolution to reduce overlapping search area, so the robots move away from provincial (local) optima. Despite the significance of the RDPSO algorithm for multi-robot exploration, there remain essential gaps for ideal searching. The RDPSO shows the fastest convergence in the global best search than the PSO or the DPSO, but the technique is not free of premature convergence (Couceiro et al., 2011; Couceiro et al., 2012). When obstacle avoidance and social inclusion/exclusion are introduced, the RDPSO can choose to reduce the number of robots dividing the whole swarm population into multiple sub-groups. The division means the RDPSO is scalable to large populations of robots in the search space. However, collisions between robots still pose an issue (Couceiro et al., 2015).

Regarding reducing overlapping search area (and avoid local optima), Dadgar et al. (2017) proposed the repulsion mechanism between similar robots. This method keeps the robot search more stable, but it compensated a much slower convergence rate. The robots can escape from sub-optimal solutions; however, the ROS framework test shows the RDPSO inability to detect multiple targets and avoid collisions (Kumar et al., 2017). Recently, in an underwater exploration use case, the RDPSO shows a higher level of robustness and enhanced exploration speeds. Still, there is a high dependency on factors such as swarm size and the rate of commands (Sanchez et al., 2018).

For the RDPSO performance for swarm inter-communication, several works have reported strategies to decrease information exchange and reduce the communication overhead within swarm sub-groups. For example, the MANET communication protocol is applied to prevent network splits so each RDPSO robot gets multi-hop connected MANET over time (Couceiro et al., 2014). The researchers focused on analysing the data packet structure shared between teammates (Couceiro et al., 2013). Nevertheless, communication breaks and constraints remain challenging to the RDPSO algorithm towards smooth swarm interconnectivity.

For optimized inter-communication, the critical feature is to lower the energy consumption in creating and maintaining network clusters and improving the wireless sensor network (WSN). Introducing methods to partition the network into different layers of clusters could be key to improve interconnectivity. Cluster heads in each layer collaborate with adjacent layers to transmit sensors' data to the base station. Ordinary sensor nodes can join the cluster head based on the Received Signal Strength Indicator (RSSI).

The MR-LEACH single-hop implementation (Farooq et al., 2010) could be an exciting adaptation which shows the cluster heads (sources) directly communicate with the sink. Multi-hop implementation is also interesting. However, if unequal clustering is proposed (Gong et al., 2008), such implementation will risk energy wastage because the Time Division Multiple Access (TDMA) is not present at the network level. Figure. 1.4 shows illustrations of single-hop and multi-hop communication.

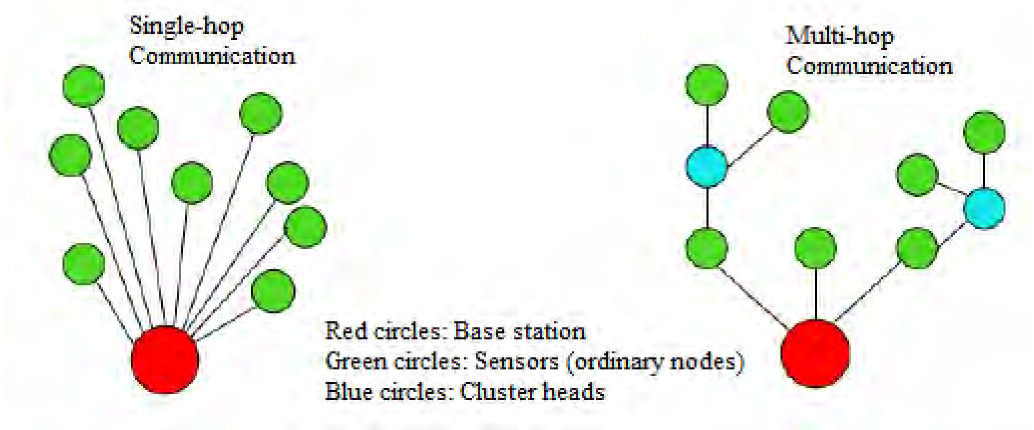


Figure 1.4: The single-hop and multi-hop source-to-sink wireless sensor network

In summary, revisiting the PSO algorithm and modifying the optimization (mathematical) functions could be key to improve any PSO-based algorithms for swarm robotics exploration. Interestingly, quantum computing theory has several advantages that can improve the searching capabilities of these PSO-based algorithms. Regarding inter-robot communication in the swarms sub-groups, communication protocols such as the MR-LEACH have shown promising results for node-based source-to-sink wireless sensor networks, but suitable for *sinkless* communication such as the swarm robotics, in which robots communicates directly with other robots.

1.3 Theoretical Considerations

In the PSO, the state of particles is described by their position and velocity, which eventually inadequate in avoiding premature convergence or local minima traps in all situations. The PSO also poses fast convergence problems in the early stage, slow convergence in the later stage, and randomness in parameter selection. A new paradigm to focus at the state of particles as wave function rather than position and velocity, inspired regulation of speed and direction, is proposed by the quantum mechanics. Sun et al. (2004) used a strategy based on a quantum delta potential well model to sample around the previous best points and introduced mean best position to the PSO algorithm. This innovative methodology is termed the Quantum-behaved PSO or in short QPSO. The core modification between the PSO and the QPSO is the iterative equation, besides describing states of particles using wave function. Interestingly, the QPSO has lesser parameters to adjust that the PSO, making it easier to implement.

Supplanting the particle descriptors of the PSO with wave function in the QPSO has significant consequences. With the wave function, the probability of a particle showing up in a particular position can be calculated from the partial differential equation corresponding to the potential field where the particle currently is. The individual particle's position can be updated by employing the Monte Carlo method. This exponential distribution usage to feed the delta potential well, rather than the normal distribution for the PSO, is an important characteristic that allows the QPSO to search in the broader space. Most importantly, the exponential distribution is less prone to premature convergence.

The other important characteristic of the QPSO implementation is the concept of wait mechanism. Implementing the mean best position (mbest), i.e. the average best position of the swarms' sub-groups, is essential to execute the concept. The *mbest* does not allow particles to converge to the global best (*gbest*) position until the position of neighbouring particles, or teammates, have been considered. In the QPSO, the distance between a particle's position (*pbest*) and *mbest* determines the particles' position distribution in the sub-group for the next iteration. When particles with respective *pbest* are far from the *gbest* position (often called lagged particles) while other particles are nearer, then the *mbest* position serves an innovative function.

The *mbest* position now becomes a landmark, so the lagged particles do not target *gbest* (too far) and have a way to catch up to their teammates. At each interval, the lagged particles will converge towards the *mbest* while the other particles proceed on course towards *gbest*. Slowly, the lagged particles will converge towards the *gbest* and particles which have converged explore globally around the *gbest* temporarily. The wait mechanism is intuitive, and in theory, the QPSO never have to abandon lagged particles behind. Most importantly, the utility of the *mbest* position enhances the QPSO global search ability over the PSO. In comparison, each of the PSO particles converges independently towards the global optima. There is no mechanism in the PSO to conserve the size of the swarm. Particles trapped in local optima are lost and removed from the swarm.

The QPSO algorithm has shown superior convergence speed and solution accuracy in continuous optimization problems. Researchers recently implemented the QPSO algorithm to solve the optimal power flow problem in a chaotic artificial bee colony algorithm (Yuan et al., 2015). The QPSO shows good performance in improving the global searchability in the practical application such as financial forecasting, except in a few instances where it falls in a local optimum. In another work, the QPSO is implemented to solve particle distribution and true pose (localization) in FastSLAM problems (Zuo et al., 2018). The estimated pose of a particle is reportedly closer to the true pose (accurate localization), and the QPSO can overcome particle depletion. The overall time consumption of the QPSO-FastSLAM is less than that of PSO-FastSLAM.

It is important to note that these works, however, are particle-based and not robot-based. One robot-based work implemented the QPSO to solve the searching process in mobile robots' path selection (Xue et al., 2017). Their results show high convergence speed and solution accuracy; nevertheless, the work focused on a free environment (without any obstacles) and likely unsuitable for SaR simulations.

1.4 Problem Statement

Despite the significance of RDPSO algorithm on multi-robot exploration and communication, there remain significant gaps such as:

- Premature and slow convergence.
- Collisions between robots.
- Communication breaks and constraints.
- power consumption

The quantum computing theory has several advantages that can improve the PSO-based algorithm's searching capabilities; however, reformulating the RDPSO is not a straightforward task. The MR-LEACH schema enhances a communication (robot communicate directly with other robots) such as lower power consumption to improve robot lifetime and reduce interrupt between robots.

1.5 Objectives of the Study

The objectives of this study are described as follows:

1. To formulate a searching strategy to reach global best in shorter time in existing RDPSO algorithm
2. To reduce the energy consumption of sensor (robot) nodes by using clustering hierarchy design
3. To test the performance of QRDPSO swarm with MR-LEACH schema in avoiding local optima and finding global best

1.6 Contribution

In this thesis, I proposed a new quantum-behaving model for swarm robotics to optimize the swarm behaviour in a simulation with SaR as the use case. This research's main contribution revolves around the extension of the RDPSO to enhance the swarm robots searching capabilities using the quantum aspects of the QPSO. This novel extension is denoted as the Quantum Robotic Darwinian PSO (QRDPSO). In theory, the proposed QRDPSO can be used to devise swarm robotics' applicability in various applications.

The following sub-areas are given careful attention in this work in deriving the algorithm to completion:

a) Individual robot-obstacle susceptibility rate,

b) Communication rate between the robots,

c) Individual robot connectivity rate,

d) Formulation of the QRDPSO,

e) Control architecture design, and

f) Parameters control

To maintain unbroken inter-robot communication in the swarm in solving SaR missions, the QRDPSO is also designed with a communication protocol such as the Low Energy Adaptive Clustering Hierarchy (MR-LEACH). The MR-LEACH is the most popular energy-efficient algorithms for wireless sensor network and its evolutionary properties. In this thesis, I designed the MR-LEACH with a sink less approach, making the QRDPSO the first algorithm for swarm robotics with a distributed network platform that takes advantage of the constant active partitioning of the entire robot swarm.

Central to the MR-LEACH implementation in this work are the following study:

a) The MR-LEACH interconnectivity between robots,

b) The MR-LEACH interruption handling throughout SaR mission,

c) Establishing multi-hop paths on a distributed network,

d) Fault-tolerance strategies to prolong MR-LEACH lifetime, and

e) Fault-tolerance strategies to prevent loss of connectivity

Despite the RDPSO algorithm's significance on swarm robotics exploration and communication, there remain significant gaps, including premature and slow convergence, collisions between robots and communication breaks and constraints. I hypothesize that the proposed QRDPSO algorithm with MR-LEACH has superiority over these gaps. Also, I believe that quantum-behaved swarm robots for exploration with enhanced communication will help researchers uncover critical areas in the robot dynamics and new strategies for efficient obstacle avoidance and improved swarm coordination.

1.7 A Guide to this Thesis

There are in total of five chapters to this thesis. Chapter 2 presents a thorough assessment of the relevant literature for the work done in the thesis. The review opens with a brief analysis of concepts supporting swarm robotics for search and rescue problems. Then, the review continues with the concept and formulation of the PSO, RDPSO and the QPSO. The review ends with discussing a few current methods from the robotic fields that use PSO in their systems for completeness.

Chapter 3 concentrates on the computation and implementation of the QRDPSO algorithm. Three research questions anchor the chapter:

1. How can one improve the maturity and rate of convergence for RDPSO during swarm-robot exploration?
2. How can one enhance the swarm communication for robot energy conservation and prolonged lifetime during exploration?
3. How does the MR-LEACH schema perform in avoiding local optima and finding global best compared to benchmark protocol such as MANET?

Answering these research questions lead to the following outcomes, respectively:

1. A novel QRDPSO algorithm improves convergence maturity and rate during swarm-robot exploration over RDPSO algorithm.
2. A coordinated swarm movement strategy conserves the robot's energy and extends the robot's lifetime during exploration.
3. The MR-LEACH schema performs better time over benchmark protocol such as MANET to avoid local optima and find global best (victims).

To show the algorithm's robustness, I examine the global best convergence and robot collision occurrences for different quantities of robots in a MATLAB simulation. Chapter 4 presents the results from these experiments. The discussion surrounding the results and an in-depth analysis of their implication is also presented for completion. Chapter 5 summarizes the thesis contributions emphasizing the lessons learned and ends with propositions for future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

In various existing societies attributed to cooperation, humans make researchers insert a lot of effort into the generation of robots that can do tasks. In this chapter, societies attributed to cooperation are evaluated based on their benefits, and multi-robots systems are presented with advancement models. Bio-logically inspired cooperative systems are where most of the interest is focused. The chapter presents the concept and formulation of the PSO series from the classical PSO, DPSO, RDPSO and the quantum behaving QPSO.

This review's attention is on the curiosity towards nature-inspired cooperative systems and how PSO series perform when challenged with tasks to optimize search, avoid obstacles, and maintain inter-communication in swarms, with or without robot. A comparison table depicting essential works in the PSO series is provided for completion. Insights from this comparison study provide a backdrop to work in this thesis. Partial to the chapter are reviews on two important communication protocols, the Mobile Ad Hoc Networks (MANETs) and the Multi-hop Routing Algorithm with Low Energy Adaptive Clustering Hierarchy (MR-LEACH) and their impact on enhancing swarm robotics communication. In the work of research, the key concepts will be used to enhance effectiveness.

2.2 Darwin Theory

Central to past discussion has always focused on “man is a natural animal and, inevitably, selfish”. The notion is considered since the science of “survival for the fittest” was introduced (Darwin, 1872). It shows how greed and power influence man’s survival since the early stages. Capitalism is another factor that man considers the most in is reign while in power. It turned out to be one of the essential concepts that Charles Darwin defended. The theory states that not all animals such as birds, insects, animals, or human being had the same survival ability.

Capitalism became a theory that Darwin reinforced in natural selection. Those who had high adaptive ability had high chances of surviving than those who do not possess the potential to do that in the new environment. Darwin uses the “law of the jungle” to refer to most of the attributes among animals, birds, insects, and humans. The law elaborated that only the brightest could survive and evolve in harsh environmental conditions. Darwin’s theory was applied at biological levels, which was done over the years and turned to be applied even social and economic competition (Maslow, 1943).

It is shown by the Darwinism that every hierarchy has the potential of dividing man’s need basing on the characteristics that the theory portrayed. The theory of motivation was developed by psychologist Abraham Maslow, who demonstrated a hierarchy of needs that a man needs to satisfy. Maslow’s pyramid is used to represent the requirements. Psychological and safety needs are to be satisfied by a man when illustrated by the previous model before entering the level of intergroup or interpersonal relations.

There are certain analogues in nature, and the attributes are overseen to be true. To get enough food, wolves' hunts in groups enhance their performance and supplement other groups with food. If the process of something dangerous occurs, the wolves disappear on their own so has they can survive. Before we can complete the tasks that we are assigned within our societies, human needs to recharge themselves by eating.

Figure 2.1: The Maslow's Pyramid

In the hierarchy of beings, men are considered to have needs that are divided by different anxieties. They have needs that are chronic, competitive and recurrent. Their needs tend to compete with each other to be satisfied first. Motivation theory is used in the most scenario to address the need for satisfying human needs (Maslow, 1943). The requirements for satisfying the needs were presented by Maslow using Maslow's pyramid, as shown in Figure 2.1.

Every level of Maslow's Pyramid has its requirement before the needs are satisfied. Most of the need is considered natural, such as the need to get water and food to satisfy the required tropic level. According to the model, men must meet their physiological and safety needs before interring the level of interpersonal and intergroup relations. There is a particular analogue in nature, and we can see that this makes senses. Existence of wild animals in a different environment is influenced by warmth, water and food. During the migration period of wild animals such as antelope, the migration leaves the group of wolves with the stand that only the strong will survive. Most of the wolves will undergo emaciation and most probably the little remaining flock will continue to survive when the migration is back.

2.3 The Summary of Cooperation

Darwin and Maslow might have faced a time difference as the main feature in their theories. Cooperation with others may be the key attribute to the survival of particular members in society. There are attributes of cooperation that do not fit within Darwin's concepts, and this aspect was brought in by Martin (Martin et al., 2013). He argues that assets are found in nature. Cooperation is considered to be every as shown vindicated by Grotuss in the year 2011. In multicellular organisms, cells are considered to work together and animals in society. Also, the genes have joined the in the genomes.

How cooperation can be essential is demonstrated in various societies in the world. The humble was studied by King Solomon, who was a student of nature in thousand years ago. He noted that "go the ant, it has not commanded, you sluggard; consider its ways and be wise, yet it stores its provision in summer has it has no overseer or ruler, and you gathers your food at harvest" (Dean, 1913) a perfect example of diligence, cooperation and order are portrayed by the ants. Ants seem to be able to find their paths regardless of their cooperation at work. Pathfinding can be going around an object or the nest where food is located and back. The ants can find the path despite being virtually blind. Chemical communication is used in this scenario, as indicated by several studies between the ants and the emergent caused by many ants. Stigmergy concept is portrayed in this scenario. There are computational logarithms that use this concept in most of the sceneries to evaluate the mechanism.

The case is showed by the heuristic principle Ant System used to control and stimulate the behaviours portrayed by the ant. Solving problems by ants is done based on a group of ants working in close conduct using simple communications (Dorigo et al., 2011). A swarm of robots uses brood sorting to effect on their activities. In the optimization logarithm, those and similar principles are implemented, including the well-known PSO, genetic algorithms, and the evolutionary theory proposed by Kennedy and Eberhart (Kennedy and Eberhart, 1995).

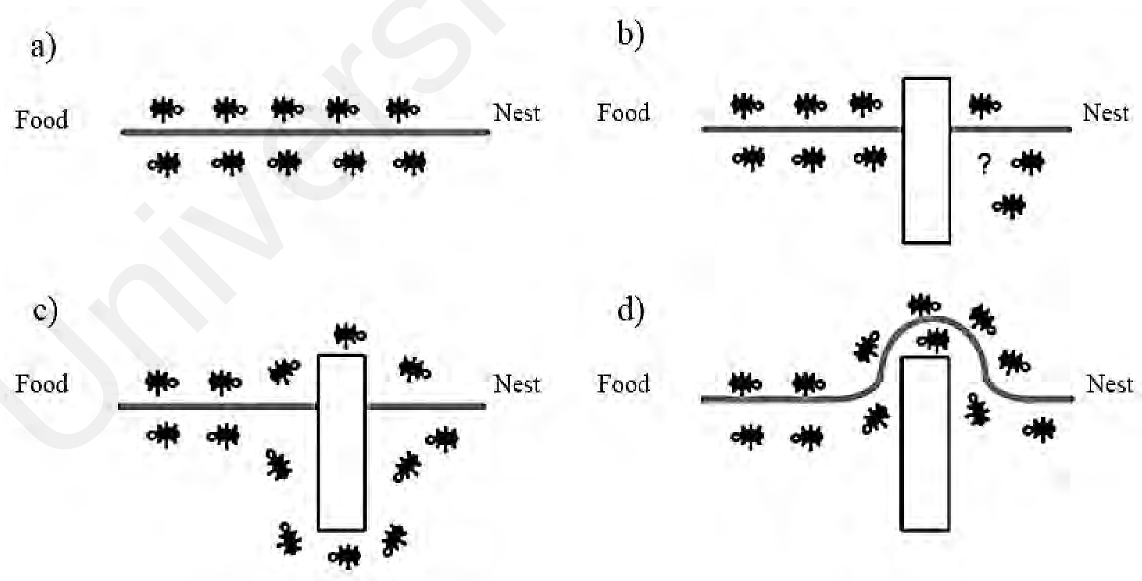


Figure 2.2: Ants' stigmergic behaviour to find the shortest path between food and nest a) travel an ants path directly, b) an obstacle interrupts the ant's path, c) the ants create two different paths to overcome the obstacle; d) a new ant's shortest path is created around the obstacle.

In the light of pelican is where another exciting engineering example is portrayed based on biological cooperation. It is evidenced that pelicans when flying in a group have high boost power compared to flying alone. A 15% reduction in the fly potential is portrayed when flying alone. Electronic equipment was used in an experiment that aimed to prove pelican flight's concept by enabling the pilot to keep the plane at a distance of 90m behind. In the concept of robotics, the plane experienced resistance of 20% lower and consumed 18 % of the fuel.

The concept becomes essential to be implemented in the military to improve the dynamics of flying robots used to govern forest fires. Biological spying robots are modified using this attribute (Couceiro M.S. et al., 2012). We can see mutual support from microorganisms to man or on related or different species or same species at all life levels. When cooperation is being addressed, we might link them to cooperation systems where some of the contributions cannot be classified quickly and the perception of all the contributions (Colman, 1995). The 3C Model of communication, coordination and cooperation adapted from Ellis et al. presented in Figure 2.3 (Ellis et al., 1991) shows an exemplary schematic of cooperative systems.

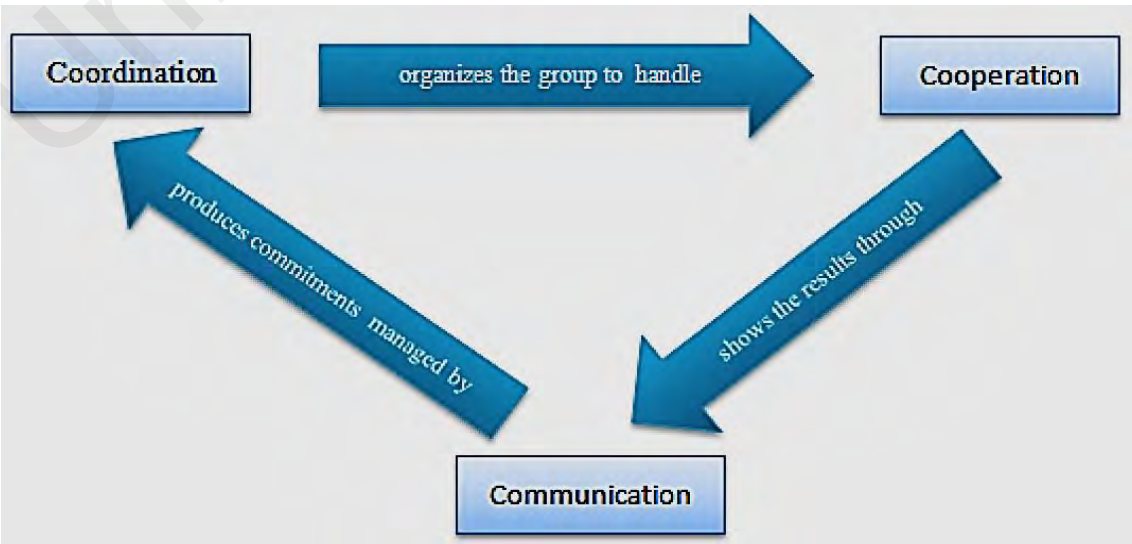


Figure 2.3: Adapted 3C Model

One of the essential tools included in a cooperation system requires cooperation as the primary attribute, leading to collaboration between different group members. Between the group members, cooperation, communication is considered the main focus to the group's success, and the communication strategies must be familiar with all the members. In cooperative systems, also coordination is considered to play an important role. It ensures that the task is performed together by eliminating communication barriers and losing efforts towards a task. It also helps to meet the constraints and objectives on time. In fields such as computer science have applied this model related to dynamics and cooperation (Andriessen et al., 2012).

2.3.1 Robots of a new society

In several biological societies, the results are inspired from this perspective of existing cooperation, i.e. humans, plants, bees, and ants. The need for creating robots that collaborate becomes a critical attribute that researchers and scientists consider. Can it be essential to evaluate the group of robots as a cooperate system or even society? Is the question that mostly arises in most of the scenarios? Therefore, communication will be a vital attribute required by the group of robots to ensure that they coordinate well to perform a task effectively.

The Multi-Robot System is a cooperative system to ensure that countless advantages are obtained as it is considered a key factor of success. Since the early days to the present, the time has any doubt, the most relevant variables to every singles earthling. Careful management is required because time doesn't stop; in the present societies, the loss of time is considered the greatest fear. The limitation related to the concept of time can be circumvented by performing simultaneous parallelism. This attribute is widely applied for both non-biological such as robots and biological creatures. Simultaneously, we can have multiple robots performing different tasks, and multiple tasks can be performed simultaneously, such as temporal distribution.

For robots to belong to MRS, they must pose a specific characteristic (Rocha et al. 2013). Some of the characteristics include; they should have the ability to interact with the dynamic environment, take deliberate action after reacting, without supervision tasks need to be done, and have features such as learning, adapting, and being sociable. The next view is examining of MRS as a society. The state at which individuals can interact, share tasks, and have some form of organization when interacting with each other can be described as a society. Figure 2.4 shows an example of cooperation concept in robotics where more than one robot work to build a 3D map of the environment.

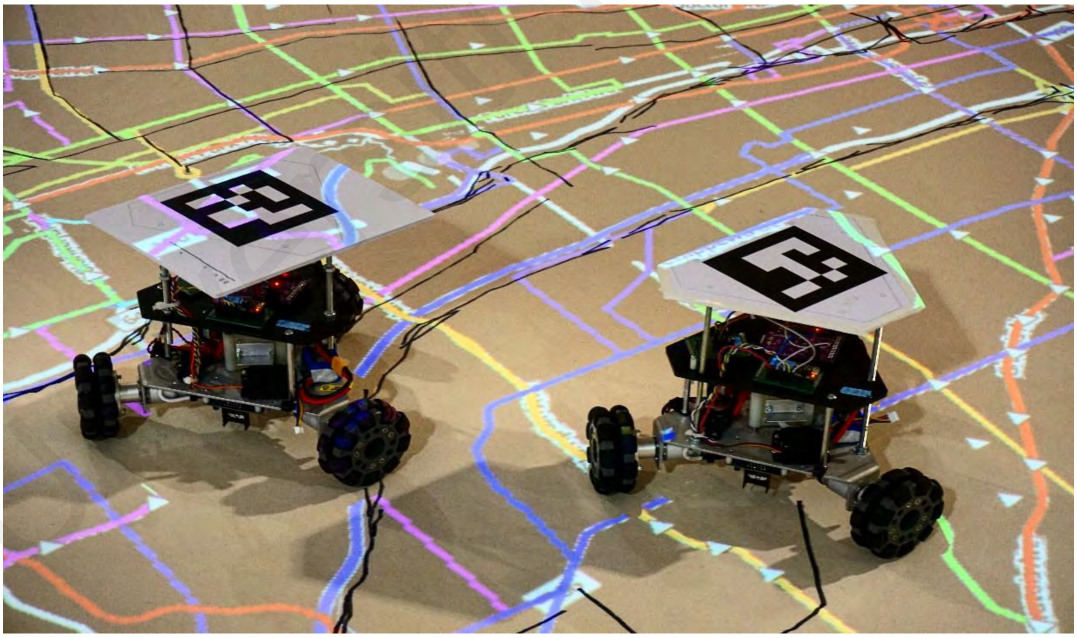


Figure 2.4: 3D mapping cooperation in robotics

Different societies can be represented by groups such as wolves, men, birds, ants, or robots. Isaac's fiction book brought about the concept of robot society and the robot living in society (Asimov, 1982). It was considered it gained its popularity in real-world and roboticists believe that the robots could emerge in the next few years maybe 5 or 10. The cooperation between humans and robots as a concept has been central to many scientific studies. The likes of research include human-robot interaction and human-machine interaction. The science also focused on SaR operations that were good at showing HMI potentialities within the human-interaction framework.

Suppose the heterogeneous primary human team's deployment in a SaR task consists of human robots (legs), UGV, and UAV, as shown in Figure 2.5. Different levels of mobility allow multiple platforms to achieve a greater degree of freedom. Mobility significantly increases mission success. For example, human robots can walk over the debris identified by the large number of low-cost UGVs. On the other hand, the swarms can interact directly with the first responders through HMI (right portion of the image).

Drones can increase the rescue operation's coverage area and reach places as high as buildings (left portion of the image). However, for collaboration to emerge, a shared wireless medium like WiFi is required between all agents. Furthermore, this medium cannot rely on a pre-existing communications infrastructure, as it may be absent, or it may not exist in such scenarios. In other words, all agents, robots, and humans must simultaneously act as MANET end nodes and routers. Figure 2.5 shows an example of an application that uses a heterogeneous robot team to accomplish a SaR mission with the first responders.

Electronic payment robots allow low-cost micro-robots to explore the environment taking advantage of automated robot algorithms. Human robots represented by NAO platforms can walk on rubble due to their high movement on wheeled platforms and build bridges between human first responders and robotic factors using HMI algorithms (Baca et al., 2011). The drone operating as a quadcopter can dramatically increase the rescue operation's coverage area. Drones can reach places where UGV vehicles cannot enter, increasing the mission's success (Julian et al., 2012).

Figure 2.5: HMI potentialities within the human-interaction framework

A better sense of the impact of technological changes on the environment, the need of studying the market on how frothy robotic speculation will be used in the future to attain human goals being set by different individuals becomes an essential measure to be considered. Their responsibility needs to be higher than a human; they need to have the ability to differentiate between their colleagues' enemies. The fact that collaborative and independent mobile robot teams can provide human teams with an extension of detection, inference, and operational capabilities in hazardous areas is motivating.

Hazardous areas are where human activity should be avoided (for example, response areas to accidents, contaminated areas, dismantling of nuclear facilities, among others). Darwin's law seems to apply in this scenario due to competition between human beings and robots in different sectors. Most of the work required to be done by humans will be substituted by robots, leading to increased competition in the labour market (Kuswadi et al., 2017).

2.3.2 The role of communication

The act of sharing and receiving information is done through communication means among or between individuals. For our lives to run more smoothly, we need communication as we live in a more complex society. The glue that holds the society together can be referred to as communication. The nature of biological societies is considered different, such as for flocks of birds or insects' swarms. Communication can involve signal use as it comes in different forms, including motion, look, and sound. Every animal is attributed to a sense of communication as the ability to share information that will propel their cooperation which is considered a survival tool.

The lifestyles that humans portray is considered to be a very different animal. Also, animals communicate to obtain food, protect a territory, stay safe, or find a mate. In MRS, the main aim of communication is considered to accomplish specific tasks. Therefore for any sensation that is taking place, robots should share that information. More resources are required to cater for the attribute of information sharing. The coordination among member groups is determined by the amount of shared information when coordinating a task (Mukhija et al., 2010).

It is considered that interaction between two robots requires a joint action which may exist from different mechanisms and social process. Human has served a biological inspiration for robotic systems for many decades through the existence of natural capacity to solve various tasks. In the robotic manipulation of tasks, joint action is considered another human behaviour that effectively coordinates how different tasks are being conducted (Zieliński et al., 2014). Efficient solutions can sometimes be achieved through the interaction of robots lacking the complete data about the world, which is an attribute that is considered a fundamental assumption in MRS research.

Robots are required to obtain minimal information concerning their team members to ensure that they cooperate accordingly. At this end, robots need to benefit from each end. Three most techniques can be used to communicate among robots; implicit communication through the world, the effect of being teammates, is felt in robots' organization (Chen et al., 2013). Passive action recognition through robot sensors can observe the action of teammates. Direct communication is considered robots directly and intentionally communicate relevant information to their teammates.

Direct communication is the most effective technique due to its attribute of directness. Each robot can become aware of the reflex action conducted by teammates (Ahmad, 2014). Synchronizing action becomes the primary use of direct communication; it determines the ability to negotiate between robots and exchange information. The use of explicit communication evaluates hidden state problems has limited sensors cannot distinguish between different states of the world for the tasks to be done effectively.

Some of the limitations shown by explicit communication theory include fault-tolerance and reliability. It depends on a noisy communication that is limited-bandwidth the channel connecting with other team members of the robot crew becomes a problem (Shannon, 1949). All lost messages and communication failures need to be addressed by the use of the explicit model. A direct channel is essential at ensuring that all the sort of voice and other communication mechanisms are highly required under this model. Most robots are designed with Bluetooth like a system used with communication with others, increasing the communication model's effectiveness. Most of the systems that are developed in robots are meant to function on a one stand task as they have integrated in a task-dependent adaptive manner.

Key mechanisms are established for robots to collaborate when performing different tasks. The interaction is used to determine the foundation of future cooperation object manipulation approach. In conclusion, effective communication and cooperation between robots are meant to achieve different tasks that are considered to exist in a dynamic environment. The use of joint action does objective manipulation in the cooperative robotic system. In humans, the cooperation of robots related to the growth of automated warehouses is used in the establishment cooperative mechanism based on human behaviour (Potkonjak et al., 2012).

2.3.3 The particular domain of swarm robotics

The range of MRS applicability is increased in cases where robots are endowed with communication capabilities. Sustaining cooperation improves MRS architecture for practical SaR application (Kuchwa et al., 2018). The swarm robotics have gained popularity after MRS has begun to branch into several other domains potentially. Swarm robotics is widely treated as a grave computational problem called swarm intelligence.

Optimization problems such as MRS within SaR application requires the attributes to face the same dilemma; exhaustive techniques. Exhaustive techniques are fitting for robots performing searching imitating biological agents (Kuchwa, 2018). The birds, bees, or ants societies have local control rules to stochastically search the scenario (Kar et al., 2016).

The use of MRS to overcome real-world issues are considered over the years as a common trend. One wonders if it is possible to achieve the required biological behaviour in the real world mission? This is the question that areas in more sceneries. A final decision hasn't been reached even though the amount of research being conducted to answer the question. Murphy and Suarez presented more than 50 papers on animal foraging, making SaR application an analogue (Murphy & Suarez, 2011).

The whole environment needs to be divided into patches by robots whenever compared to animals such as bees. An environment that fires can occur becomes challenging to define patches with unknown sceneries as animal foraging is different from the motivation model. At first, for SaR robots, using a biologically inspired optimization method is considered unsuitable. The use like can be foreseen by implementing several applications that can be used to control the situation in fire areas (Figure 2.6).

a) b)

Figure 2.6: SaR real application in monitoring a) a basement garage, and b) underground fire of a shopping centre.

The most popular catastrophe in urban areas is considered urban fires that need immediate response due to life endangerment for densely populated areas. It can spread to parked cars and buildings. The SaR application becomes challenged in a state whereby fire in urban centres spread to rooms that contain inflammable materials such as a garage of has shopping mall. SaR is challenged due to the existence of a confined environment nature. The spaces become rapidly filled with smoke as fire evolves as the following attributes arise, first, the toxic atmosphere is generated, an unbreathable situation occurs, and visibility is reduced. Both the responders and victims at the scene are endangered. In less than 20 minutes, victims in such a scenario may be unable to survive.

Regardless of the biological situation, two domains can be obtained after obtaining alternatives: swarm algorithm and non-swarm algorithms. As observed in non-swarm societies, the difference between the concepts is considered to have the same differences (Dorigo, et al., 2011). Individuals from non-swarm societies have complex agents, making them be independent most sceneries. There is a limitation in swarm agents. The communication systems are considered flawed in a scenario where there are robotic differences in sensory and actuation.

The population of non-swarms is considered smaller compared to swarm societies (Kar et al., 2016). The significant performance outcome is not affected by the attribute of addition or removal of members from the group. Collective performance might be reduced when a member of non-swarm society is reduced, such as robotics. Architecture-simple local control rules are used to coordinate swarms as they lack a centralized agent to command others on several tasks that are to be conducted by the group. Centralization results in the emergence of the global behaviour of the system. In swarm strategies, the principle is not considered part of the distributed architecture (Dorigo, et al., 2011).

Within the application itself, the choice itself falls in either of the domains. Under practical application, the loss of robots as stated before due to harsh conditions interferes with the model. The use of rescue robots can be attributed to a rule of simpler is better. The case tries to elaborate that one agent of thegroup's failure shouldn't affect the performance of other members to achieving the stated mission (Julian et al 2012). As a real fulfilment of applications such as SaR, the work settles under swarm intelligence. Most of the phenomena are abounds in a collective swarm environment that ensures the need for increasing movement coordination to nest building.

All the collective biological phenomena are considered under principles of self-organization. They are considered as resulting functionalities and structures greatly exceeded in the cognitive, physical, and perceptual abilities to participate in specific tasks by organisms. Construction of beehives can be considered biological self-organization. It can also be seen in the regulation of colony life by some social insects and foraging strategies in ants. The collective works are emerging structures for an individual that exhibits simple behaviour as they don't possess a global plan for their actions. For swarming operations, they adapt to the principles of an army. Most of the properties in the real-world robotic applications are intended to be emulated from biological properties.

Obstacle avoidance-in the swarm robot society, obstacle avoidance is also considered to be an essential task. Most researchers have argued that the need for incorporating all sort of function within a robot becomes a key attribute that needs to be considered to ensure that all robots can detect obstacles on the way. The concept of real obstacle avoidance was introduced in 1986 by Khatib, whereby he used a time-varying artificial potential in the field of moving objects. The traditionally high-level planning was converted by the solution of robots being in the ability to detect objectives in a complex environment (Kar et al., 2016).

Schemes are some of the examples that are used. Several controllers have been developed with the robot system to perform different functions and interact with each other. The robots play a role in flying around the nucleus due to the generation of a virtual repulsive force similar to the mechanism in the atomic nucleus. A potential function is applied to avoid collision with the swarm. The formation and transition features are considered to be associated with geometrical features present in Delaunay diagrams. By using the proposed strategies, robots can select their neighbours by forming a topology that can connect the individuals.

The algorithm shows some flexibility as it is vulnerable to some robustness (Kadry, 2018). A new method for avoiding obstacles by robots based on the second-order motion model in a dynamic environment. The proposed model focused on velocity, destination, and direction of the robot, which was consistent in applying a mathematical-based model, which was better than the use of PSO. Inspired from swarm intelligence algorithms-swarm robotic searching and swarm optimization algorithms share a lot of similarities. For instance, they are using a swarm of individuals to search the best points.

In the swarm robotic, particle swarm intelligence is mostly used to create similarity with the searching and flocking schemes. Great ability in scalability robustness and flexibility are considered attributes related to the use of a swarm intelligence approach that shows the ability to be applied in real-life situations. Introduction of these algorithms is done at the same time which becomes a limitation. RDPSO was proposed by Cruceiro et al. (2012) for solving issues related to the robotic configurations. Topologies are updated in several ways, hence encouraging several iterations based on a punishment mechanism and reward. Distance metrics are ignored due to the sub-swarm division as they can easily escape the local minimum at the cost of the global coordinating and communication system. Swarm intelligence is accompanied by three methods; optimizing the parameters, modelling the individual behaviours and mixing the two methods (Cruceiro, 2014).

Optimizing the parameters is considered the first type of searching algorithm used in inspiring strategies from other related approaches with several parameters that are considered challenging to optimise. The optimization of these parameters is done through the implementation of swarm intelligence algorithms. The searching of the random building block is considered adequate as part of a collective construction task. Information exchange is done based on employing virtual pheromone responsible for task allocation for cooperative transportation in swarm robotic. The implementation of PSO model does the noisy problem of unsupervised learning in robots. Modelling individual behaviours is done regarding particle or agent response to swarm intelligence algorithm (Stirling et al., 2012).

To attain the target, the swarm uses fitness values after obtaining the searching environment. The use of PSO in a robotic environment is used to design practical algorithms that allow the swarm of robots to carry tasks together. At the abstracted level is where the analysis of the parameters and robotic setups are presented. Mixing of the two models-optimizing the parameters and using the swarm intelligence is sometimes combined with the use of different algorithms that control unmanned mobile robots in a target tracking application. In the inner layers of PSO is where schemas control the robots. A force-based algorithm determines every robot's behaviour under a level of threats of hostile attacks (Conner et al. 2012).

2.4 Swarm Robots

Swarm robots, or sometimes known as multi-robot systems (MRS) represent a collection of assembling robots that associate themselves with tackling a task such as locating and navigating towards a target, as a group. The key concept in the MRS development proposition is to organize and instruct a team of numerous autonomous robots in situations that could be tricky or dangerous for the human (Al-Khawaldah et al., 2014). In performing such a task, a task allocator will distribute the load for each of the swarm robots.

If the swarm is controlled by an algorithm that singles out a robot to carry out the task for the swarm, then the entire swarm community risk failure if that particular robot does not perform. Therefore, to ensure swarm success, every robot in the swarm should be equipped with similar fitness (i.e. sensors, devices, and/or protocols) to insist cooperation and competency in task completion (Dai et al., 2016). When each of the swarm community robots is competent, the swarm can thrive in active exploration and consequently minimize searching time. Collectively, the swarm becomes flexible and robust against failures (Deng et al., 2017; Ducatelle et al., 2010).

Realizing the benefit of the swarm robot concept, one can diagram the typical pattern in the search and rescue operation for robotic research. Figure. 2.7 shows the procedures. In general, there are three significant activities expected to prepare SaR robotic teams (Li et al., 2014):

1. The initial deployment of swarm robot in the unknown environment,
2. The execution of mission like target searching by the swarm of autonomous agents, and

3. The open communication for information gathering to update each robot in the swarm regarding the swarm's progress on the task. The information should also be aggregated externally to human operators such as first responders.

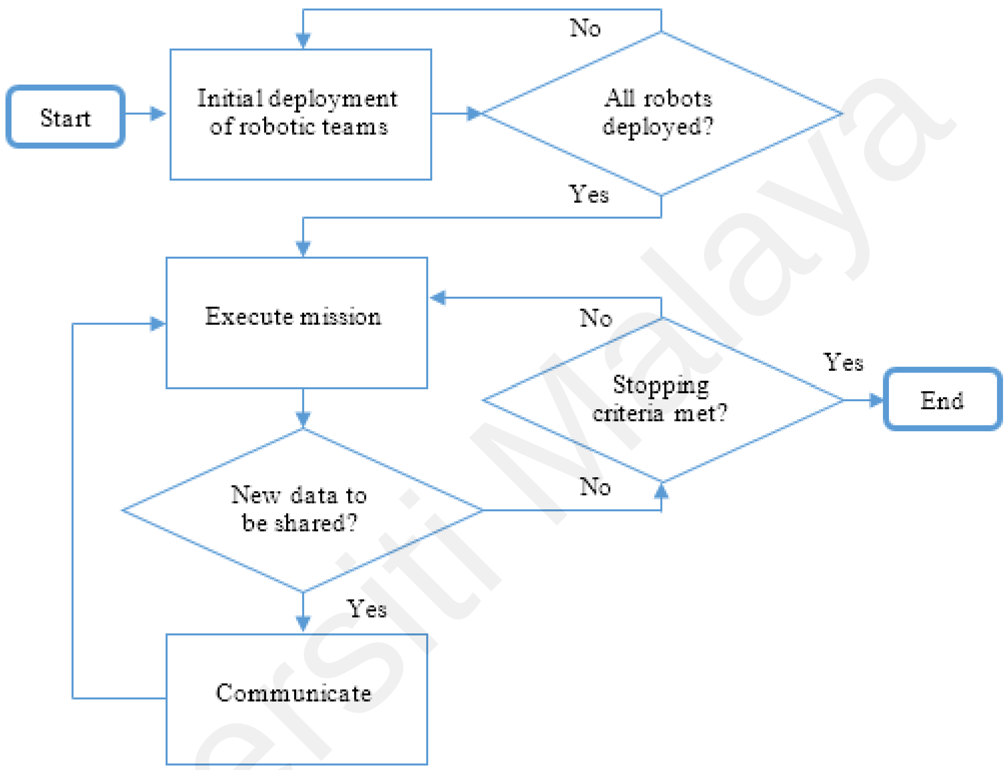


Figure 2.7: General flowchart of a SaR (robotic) operation

The growth of swarm robotics is attributed mainly to the popularity and influence of bio-inspired algorithms such as the genetic algorithm and ant colony optimization. These algorithms, taking inspiration from how ants, birds and bees flock and forage for food, execute target search by calculating approximate target position towards optimized solutions (Hereford et al., 2010). One other important algorithm that inspires an exchange to this domain is the particle swarm optimization (PSO).

The PSO algorithm's contribution is the proposition towards stochastic optimization, i.e. the notion of having a random probability distribution or pattern that may be analysed statistically. Still, it may not be predicted precisely (Kadry, 2018), which drove the social characteristics in foraging bees and schooling fish (de Sá et al., 2016). Apart from the exchange with nature's optimization, the PSO benefits too from the humanmind's theory (Dadgar et al., 2016). The PSO has attracted a large number of widespread researchers. The following sections describe how researchers execute swarm missions to optimize target searching, avoid obstacles and conserve inter-robot communication.

2.5 Particle Swarm Optimization (PSO)

The PSO has various improvements to further searching convergence and safeguards against communication as an algorithm that spreads swarm particles to optimize target searching (Cecconi et al., 2011). For instance, one robotic practice proposed a technique that listens and picks up periodic messages from the swarm to get position updates from each of the robots in the swarm. Thus, at each step, it happened that a robot can send and receive messages with only another robot from different sub-group of the swarm, creating a dynamic neighbourhood topology (Akat et al., 2010;). It is also common for swarm robotics to pursue searching tasks while interacting with its teammates or neighbours under limited local communication ability (Smith et al., 2018).

To describe the PSO equation, consider candidate solutions (e.g., ants and bees) as particles. These swarm of particles are spread through a multidimensional space for target searching, by continually listening and picking up information from their neighbour particles, in the hope that one of them obtains an individual best (local best solution). So, at each time step of the search, one individual best solution is updated as the global best solution, so the entire swarm always has the best solution.

This way, particles can always have information about the potential field locations where success (e.g. food) is likely found. These successes then guide the swarm search pattern. Algorithm 2.1 shows how a fitness function is used to evaluate particle success. Each particle n moves in a multidimensional space according to the position ($x_t{}^n$) and velocity ($v_t{}^n$) values which are highly dependent on the local best ($\breve{x}_t{}^n$) and global best ($\breve{g}_t{}^n$) information (Dadgar et al., 2016):

$$v_{t+1}{}^n = wv_t{}^n + \rho_1 r_1(\breve{g}_t{}^n - x_t{}^n) + \rho_2 r_2(\breve{x}_t{}^n - x_t{}^n) \quad (2.1)$$

$$x_{t+1}{}^n = x_t{}^n + v_{t+1}{}^n \quad (2.2)$$

Coefficients w, ρ_1, ρ_2, it is assigned weights to the inertial influence, the global best and the local best when determining the new velocity, respectively. By default, the inertial influence is set to a value lesser than 1. ρ_1, ρ_2 are constant integer values, representing the *cognitive* and *social* components. However, different results can be obtained by assigning different influences for each component. For example, many works do not think about neighbour best. The parameters r_1, r_2 are random vectors with each component generally a uniform random number between 0 and 1 (Couceiro et al., 2010). According to Cai et al. (2013a), the aim is to “multiply a new random component per velocity dimension, rather than multiplying the same component with each particle’s velocity dimension”. Tuning these parameters towards specific problem or application is required to increase the odds towards better performance.

At the swarm initialization, particles’ velocities are defined as zero. They are given random positions as long as they remain within the search space. The complexity of the task influences the local, neighbours, and global bests’ parameterisations. The more complicated the situation, the worst their values are set. An example would in designing the parameters for a cost problem where the aim is to minimize the fitness function. The algorithm also requires modification to several other parameters such as:

1. Population size. This is important to optimize to achieve overall reasonable solutions within an acceptable time
2. Stopping criteria. One way is to pre-program the number of iterations when the swarm stopped getting better results or other criteria depending on the problem

Three outstanding works describe the PSO algorithm's utility to optimize target searching for the swarm. The first work proposed an intelligent method for swarm robotic by adjusting parameters to manage premature convergence to reduce the swarm searching time (Cai et al., 2013a). In the second work, the researchers adopt parameterizations from the genetic algorithm (GA) and combine with the PSO for improved algorithm performance (Shi et al., 2013). The third work, focuses on a decentralized approach, with a strategy to solve path planning by coordinating the numerous robots at the global level (Cai et al., 2013b). These works paved the way for variation of extension to the PSO algorithm emphasizing on swarm coordination.

Wang et al. (2015) were interested with a self-organizing approach to study the direction and angle of moving target based on information perceived through the inter-robot local communication. They introduce and promote the concept of cooperative hunting by the robotic swarm. Nakisa et al. (2015) showed how local search solves premature convergence for robots tested in an unknown environment with static obstacles. Another work showed how PSO improves the localization problem for robots' swarm (de Sá et al., 2016). For Dadgar et al., (2015), they showed how the PSO with distributed algorithm perform target searching and successful obstacle avoidance. The PSO is also tested in a dynamic environment where the robots are challenged with multiple targets (Couceiro et al., 2012). More recently, Cai et al. (2016), introduce fuzzy logic to improve swarm cooperation towards better convergence.

The PSO method's popularity in solving swarm target searching and intercommunication is immense because of ease of implementation where only a few parameters require adjustment. However, the approach poses disadvantages such as the inability to resolve particle scattering and premature convergence. It is also prone to inexact regulation of speed and direction (Cai et al., 2013b; Cai et al., 2016). In short, the main problem of PSO is being stuck in local best. In some cases, the PSO technique is ideal to avoid sub-optimal solutions but interestingly perform poorly in others. The Darwinian PSO (DPSO) as an enhancement to the PSO can improve with sub-optimal solution problem and is discussed next.

2.6 Darwinian Particle Swarm Optimization (DPSO)

Tillett et al. (2005) first introduced the Darwinian Particle Swarm Optimization (DPSO) as an extension to the PSO algorithm to improve the natural selection model. The DPSO is inspired by the notion that the swarm can have as many sub-groups (mini swarms) existing at any given time. These sub-groups represent clusters of robots which populate the entire swarm. Each sub-group can have members depending on the maximum allocation defined before the search.

In the DPSO, each mini can have similar competency just like any sub-group in classical PSO, except for several rules designed to simulate the Darwinian's nature principle, the *survival-of-the-fittest*. The goal is to allow the PSO to escape from sub-optimality, an issue discussed in Section 2.3. The idea with the DPSO is to treat each sub-group like a mini swarm, in which each mini swarm runs their own PSO algorithms, on the same test problem. Sub-grouping means the DPSO runs multiple parallel PSO algorithms for a single search problem. A simple selection cooperative competition mechanism has to be applied to organize the search. For example, the swarm should pay attention to the area where sub-groups progress well in avoiding sub-optimal solution and discard areas where sub-groups get stuck (Couceiro et al., 2012).

To reinforce the learning, at each step, sub-groups that are not stagnant and show active movement get rewards such as extra life (new particles) or an extended lifetime. On the contrary, sub-groups that are slow and not progressing face punishments such as losing a life (delete particle) or reduced life extension. The state of each sub-group can be analysed by evaluating the fitness or cost function of all particles. The individual best position of each of the particles can be updated. Whenever a new global best is discovered, a new particle is spawned. A particle is removed whenever a sub-group gets trapped, even after many iterations are executed. Algorithm 2 describes this process. Some rules are followed in deciding when to remove a sub-group or particles, and when a sub-group or particles can have a new spawn (Couceiro et al., 2013):

1. When the sub-group population falls under a minimum bound, the sub-group is removed, and
2. The worst performing particle in the subgroup is removed, when a maximum search counter (maximum threshold number of steps) without improving the fitness function is reached.

After a particle is removed, the counter resets to a value that has the threshold value, instead of defaulting to zero, according to:

$$SC_s = SC_{max}[1 - 1/N_s^{kill} + 1] \quad (2.3)$$

where N_s^{kill} holds the number of particles removed from the sub-group s over a period in which there was no progress in the fitness value. In the case of a successful sub-group, the sub-group must not have removed any particle prior and the maximum number of members must accord to the pre-defined value before a spawn is rewarded.

Similar to the PSO algorithm, the DPSO also require adjustment to a few parameters so the algorithm runs efficiently:

1. Initial subgroup population N_I;
2. Maximum and minimum subgroup population N_{max}, N_{min};
3. The initial number of subgroups N_s^I;
4. Maximum and minimum number of subgroups N_s^{max}, N_s^{min};
5. Stagnancy threshold SC_{max}.

The DPSO has achieved progress over the PSO; however, to realistically adapt the method onto robot architecture, the DPSO lacks parameters to avoid obstacles and maintain uninterrupted communication. A series of articles proposed how the DPSO is extended to develop and test the Robotic Darwinian PSO (RDPSO) and their contributions are discussed next.

2.7 Robot Darwinian Particle Swarm Optimization (RDPSO)

The RDPSO is an evolutionary algorithm benefitting from the DPSO algorithm's extension designed to augment the ability to avoid sub-optimal solutions. The RDPSO does so without significantly increasing the computational cost of the DPSO. In the DPSO simulation, sub-groups of interacting robots are deployed to perform optimized target searching. However, the MRS presents environmental constraints that need to be attended. For example, obstacles are unavoidable. In robot missions, the environment is often unstructured, and communication devices may risk damaged or missing. Therefore, communication protocol plays a vital addition to interconnected robots.

A series of articles between 2011 and 2014 by Couceiro et al. discussed the implementation and testing done on the RDPSO algorithm. Fundamentally, four general features are proposed to adapt the DPSO into RDPSO (Couceiro et al., 2013):

1. A punish and reward techniques, that used to delete and create the robots,
2. An obstacle avoidance algorithm to avoid static and dynamic obstacles,
3. An enforcing multi-hop network connectivity algorithm to ensure that the Mobile Ad-Hoc Networks continue connected throughout between robots, and
4. RDPSO methodology to build up the initial planar deployment of robots with the Mobile Ad-Hoc Networks connectivity while distributing the robots as much as possible.

In a robotic swarm, each robot moves in a multidimensional space. The discrete equation (DE) system is used to model the RDPSO. First, the equations from (2.1) and (2.2) need to be rearranged into the following equation (2.4) and (2.5) respectively:

$$v_n[t+1] = w_n[t]v_n[t]+\Sigma^4_{i=1}\rho_i r_i(\chi_i[t]-x_n[t]), \tag{2.4}$$

$$x_n[t+1] = x_n[t] + v_n[t+1], \tag{2.5}$$

wherein $w_n[t]$ and ρ_i, i=1,2,3,4, assign weights to the inertial influence, the local best (cognitive component), the global best (social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity, with ρ_i>0. Similarly, r_i are random vectors where each component is a uniform random number between 0 and 1. $[t]$ and $[t]$ represents the velocity and position vector of robot n, respectively. While $\|[t]\|$ is limited to the maximum allowed velocity of v_{max} for robots, i.e., $\|v_n[t]\|\leq v_{max}$, $x_n[t]$ depends on the scenario dimensions. $[t]$ represents the best position of the cognitive, social, obstacle and the communication protocol (e.g. MANET) matrix components.

The cognitive $\chi_1[t]$ and social components $\chi_2[t]$ are commonly presented in the classical PSO algorithm, as seen in equation (2.1). $\chi_1[t]$ represents the local best position of robot n, while $\chi_2[t]$ represents the global best position of robot n. The size of the vectors (ϖ) depends on the dimensionality $\mathbb{R}^\varpi$ of the physical space being explored, e.g., ϖ=2 for planar problems.

Couceiro et al. (2013) presented a stability analysis of the RDPSO to understand the relationship between the algorithm parameters and the robot's convergence. It is reported that the RDPSO algorithm converges to the optimal solution faster and more accurately than the other approaches, particularly over the traditional PSO. They argued that dynamic enhancement to the communication system's architecture and characteristics could increase the scalability and applicability of the RDPSO. Couceiro et al. (2014) integrate MANETs is a fault-tolerant distributed search to prevent communication network splits, to ensure full connectivity. Analysis of the data packet structure shared between communicating teammates shows communication overhead within the swarms of robots is reduced. When put under communication constraints, Couceiro et al. (2013) argued that to ensure stable ad hoc connectivity, the number of robots in the exploration task needs to be increased.

Before 2013, the RDPSO has also been analysed from several mathematical perspectives. For example, the fractional calculus theory supports the RDPSO as a distributed foraging algorithm evaluated on real low-cost mobile robots (Couceiro et al., 2012). The fractional-order RDPSO presents a significant influence in the convergence time because of its inherent memory property. Fuzzy techniques have also been introduced to improve the coefficient function making the RDPSO converge faster and susceptible to obstacles and communication constraints. This result further researchers' progress in considering obstacle avoidance and social exclusion and inclusion concepts first introduced in Couceiro et al. (2011).

More recently, Kumar et al., (2017) tested the RDPSO on a ROS framework by identifying victim using voice and locating fire source using temperature. They highlighted that the RDPSO was unable to detect multiple targets or avoid locations collisions. Sànchez et al. (2018) applied the RDPSO algorithm on a multi-robot strategy to explore unknown underwater environments. In their report, the RDPSO observes a higher level of robustness and enhanced exploration speeds.

In summary, the advantages of the RDPSO algorithm are interesting. It is scalable to large populations of robots, it can decrease the amount of required information exchange among robots, and it poses faster convergence with higher accuracy. However, revisiting the mathematical functions of the RDPSO is inevitable as several disadvantages, for example, collisions between robots and instability of the swarm movement can still occur. The RDPSO algorithm also faces communication issues, including connection ruptures between robots leading to premature convergence.

This section shows how the RDPSO is a useful algorithm for swarm robotics with potential for expansion as a concept and at a technical level, for the work proposed in this thesis. However, to further complete the understanding of the RDPSO, it is essential to include here several key features borrowed by observing the evolution of the society which inspired the shaping and derivation of the algorithm. Behaviourism of human and animal in the social hierarchy, particularly the punish-reward mechanism, the organization of groups and the emergence of sub-groups, and the exclusion of sub-groups are considered in the development of the RDPSO and are discussed next.

2.8 The punish-reward mechanism

The intention behind the punish-reward mechanism is attributed to the notion that good conduct leads to desirable results in the long run. In contrast, bad conducts will eventually accumulate to unfitting results. In robotics, the punishment-reward mechanism's adoption is on the rise (Elfwing & Seymour, 2017; Chen et al., 2018; Kobayashi et al., 2019; Keijsers et al., 2019). In swarm robotics, the punish-reward mechanism is favourable for collective learning and optimizing strategies and handling challenges for example in robot coordination, locomotion and navigation (Couceiro et al., 2013; Wang et al., 2016; Clayton & Abbass, 2019).

The common DPSO formulated by Tillet et al. (2005) defines the punishment mechanism as deleting particles and shrinking of swarms. In contrast, the reward mechanism as the spawning of new particles and swarms follows the order of natural selection. The RDPSO (Couceiro et al., 2013) adopted the DPSO onto mobile robotics by adjusting the parameterizations regarding the punish-reward mechanism. This is because robots, unlike particles, are complex and real-time swarm robotics tasks often require a searching capability which considers higher-dimensional problems. A distributed approach such as the RDPSO would outline that each robot in the swarm is obligated to share its current velocity with all other members in the swarm. Thus, the social exclusion-inclusion concept becomes appealing to model the punish-reward mechanism for swarm robotics.

The social exclusion-inclusion concept is essential in defining the subgroups behaviour when members in the group perform or underperform for particular robot tasks or *objective functions*. Improved objective functions and vice versa define robotic subgroups that are performing. The tracking of underperforming subgroups is done by counting the number of times a subgroup s had evolved when the subgroup did not have any improved objective. The SC_s, as defined in equation (2.3) is an example of the search counter. In equation (2.3), the N_s^{kill} represents the robot count in subgroup s if the subgroup fails to increase its objective function after some duration. To optimize their search strategy, a maximum dangerous threshold, SC_{max}, is often defined. Whenever the SC_{max} is reached, the subgroup s executes the punish mechanism by rejecting the lowest-performing robot.

The punished robot is then forced to join a socially excluded subgroup, a category of a subgroup that hosts underperforming robots, usually a lower-ranking subgroup. The function for members in the socially excluded subgroup is to wander in the searching environment aimlessly. This differs than when they were in subgroup s, where their main objective is to search for the global optimum. Having them wander and *move again* makes it less susceptible for the individual robot to get trapped at local optima, which improves the overall performance of the algorithm. Nevertheless, their solution and the socially excluded group's global solution remain with them, so if they accidentally moved to a potential solution, they may inform the swarm of their discovery.

The cycle for each subgroup s is repeated with every subgroup in the ranking system re-evaluates each member robot by comparing their objective function values, either increasing or, stagnant/decreasing. If the desired outcome is to minimize the fitness function, the subgroup will punish or reject the robot with the lower fitness value. There could likely be an underperforming subgroup that keeps losing a member due to the punishing mechanism. When it is no longer relevant to maintain a subgroup N, for example, the robot count has fallen lower than the minimum acceptable member, and the punish mechanism also dictates that the subgroup should be deleted. When a subgroup is deleted, all remaining members are forced to join the lower-ranked socially excluded subgroup.

The reward mechanism reverses the situation for these subgroups. The reward mechanism defines that if the subgroup s keep improving its objective function, over time, the subgroup deserves to *expand* with new members. Each new member is to be selected from the socially excluded subgroup, in which the best performing robot is nominated. Consequently, if the subgroup s receives more reward than punishment, the subgroup may get overpopulated. It could be promoted to spawn a new subgroup, a predefined number of robots N_I is required, which makes the probability p_{sp} of spawning, in the original punish-reward model such as in the DPSO algorithm, not very high.

To improve the chances of spawning, the RDPSO recommends that the probability p_{sp} should not be dependent on the total count of active subgroups. Taking into consideration that there is only intra-communication permissible between members of the same subgroup (i.e. inter-connectivity between neighbouring subgroups is cut off), the RDPSO proposed redefining the equation for p_{sp} as follows;

$$p_{sp} = r_{sp} \frac{Ns}{Nmax} \tag{2.6}$$

Where r_{sp} takes in any random number between 0 and 1, N_s is the total count of robots inside the same subgroup s, and N_{max} controls the ceiling of total robot count allowed per subgroup. Equation (2.6) is designed so that when a socially productive subgroup shows steady progress and becomes over-populated (i.e. N_s=N_{max}), it can decide to spawn. The subgroup ensures the new spawn to consist only strong well-performing candidates among members of the socially excluded subgroup.

The DPSO algorithm thrives on an approach in which the parameterizations define the network based on the behaviour of the entire population of particles, concentrated as a swarm. The RDPSO, on the other hand, proposed governance of multiple numbers of swarms using multiple numbers of networks respectively, a distributed approach, as more superior. For instance, the RDPSO network now only has to consider intra-communication between nodes or robots of the same network, a much smaller traffic constraint than the DPSO.

With the distributed approach, the RDPSO becomes scalable to handle larger robots' redefining communication for swarm robotics to avoid the sub-optimal solution. Table 2.1 shows a summary of the RDPSO model for their punish-reward rules.

Table 2.1: RDPSO "Punish-Reward" Rules (Couceiro et al., 2013)

PUNISH	REWARD
Suppose a socially active subgroup does not improve during a specific threshold $SCmax$ (stagnancy counter $SCs = SCmax$), and the number of robots is superior to $Nmin$ ($Ns > Nmin$). In that case, the subgroup is punished by socially excluding the worst-performing robot.	If a socially active subgroup improves and its current number of robots is inferior to $Nmax$ ($Ns < Nmax$) and there is, at least, one socially excluded robot, then it is rewarded with the best performing socially excluded robot.
If a socially active subgroup does not improve during a specific threshold $SCmax$(stagnancy counter $SCs = SCmax$) and the number of robots is $Nmin$ ($Ns = Nmin$), then the subgroup are social.	If a socially active subgroup is not stagnated (stagnancy counter $SCs = 0$) and there are, at least, N_i socially excluded robots, then it has a small probability P_{sp} of spawning a new socially active subgroup.

The RDPSO consider three scenarios for the new socially excluded robot when wandering and broadcasting its whereabouts:

1. If no robot picks up its broadcast, the socially excluded robot will continue to explore randomly and resume broadcasting its location.
2. If an active robot picks up its broadcast, the message will be forwarded to any available robot from the same sub-group in the surrounding vicinities.
3. If another socially excluded robot picks up its broadcast, the wandering robot will answer by sending its current location.

In case (b), the act of message forwarding by the active robot to its subgroup will signal the new excluded robot to counter with a message containing its current location. Message forwarding triggers the active subgroup to update the excluded robot of their whereabouts. In case (c), the act of message forwarding by the other excluded robot to its excluded subgroup will signal the new excluded robot to counter with a message containing its current location. The new excluded robot is then invited to join the excluded subgroup that responded to its message. It is worthy to note that this excluded subgroup may not have all excluded robots in the swarm. However, this is not an important issue as even though all excluded robots are connected, they are powerless to make any decision. The only robots that can make any decision belong to the active subgroups.

For the active subgroups, when the opportunity to spawn arrives, one of its members, likely the highest performing robot, has to announce or broadcasts its reward to earn a new member or spawn to a new subgroup following guidelines as presented in Table 2.1. In this case, three scenarios are considered:

1. If no robot picks up its broadcast, the socially active robot will continue its mission and resume broadcasting its reward.
2. If an active robot picks up its broadcast, the reward's message is forwarded to any available robot from the same sub-group in the surrounding vicinities.
3. If one of the socially excluded robots picks up its broadcast, the wandering robot will forward the message regarding the reward to other socially excluded robots.

In the case of the case (b), and an excluded robot reaches the message regarding the reward, case (c) is assumed. What comes next are highly determined by the condition of the reward. Suppose the reward is regarding the active subgroup looking for a new member. In that case, the highest performing robot from the socially excluded robot will become the new addition to the active subgroup. If the reward is regarding spawning into a new subgroup, then several of the highest performing robots N_I from the excluded subgroup will join for that particular purpose. However, the other condition needs verification. Only when the total count of excluded robots is the same or exceed N_I that spawning several of the excluded robots into the new subgroup will be considered.

Due to the distributed approach where messages are usually forwarded to others in the neighbouring areas, the highest performing lot from the excluded subgroup does not represent the best performing robots from all socially excluded subgroup. Consequently, in the event where an active subgroup advertising for new member gets punished at the same time for some reason, then the reward is considered null and void.

Despite the advantages over the DPSO algorithm in terms of susceptibility to avoiding local optima and the capacity to scale over many robots, the RDPSO poses issues in the communication parameterizations. Nevertheless, the entire robot population's dynamic partitioning into smaller subgroups and network is an exciting model for swarm robotics and should be explored further. The next section shows how quantum behaving particles can influence optimization algorithms such as the RDPSO.

2.9 Quantum Delta Potential Well Model of PSO (QPSO)

The QPSO derives the PSO algorithm with the differential equation that does not need velocity to lead a best global position for the swarm robots. The QPSO describes the state of particles, in a specific way to increase the odds for better global searchability. In quantum space-time, the quantum state of a particle is shown by wave function $\Psi(\bar{x}, t)$ rather than position $\bar{x}$ and velocity $\bar{v}$. The wave function describes dynamism of the particles' behaviours developing in a different direction from that in PSO, that the exact values of $\bar{x}$ and $\bar{v}$ cannot be determined.

It is estimated that the probability of particle s showing up in the position $\bar{x}$ from the partial differential equation $|\Psi(\bar{x}, t)|^2$ is dependent on the potential field where the particle is (Sun et al., 2004):

$$X_{i,n+1}^{j} = p_{i,n}^{j} \pm \alpha \left| X_{i,n}^{j} - p_{n}^{j} \right| \ln\left(\frac{1}{u_{i,n+1}^{j}}\right) \tag{2.6}$$

$$X_{i,n+1}^{j} = p_{i,n}^{j} \pm \alpha \left| X_{i,n}^{j} - C_{n}^{j} \right| \ln\left(\frac{1}{u_{i,n+1}^{j}}\right) \tag{2.7}$$

In equation 2.6 and 2.7, C is the m best positions vector, and n is defined as the number of iterations. j is the component of the position of particle i where $j^{th}(1 \leq j \leq N)$ for the particle $i(1 \leq i \leq M)$ at the $(n+1)$ position, where N is space dimensions, and M refers to the number of particles. P_n^j is the centre of the N-dimension Hilbert space with a δ potential well. It is the best previous position, the position giving the best objective function value of fitness value, of the particle i (also refers to as Personal Best).

$$p_{i,j} = \varphi . p_{i,j}(t) + (1 - \varphi). G_j(t) \tag{2.8}$$

In equation 2.8, $G(t) = (G_1(t), (G_2(t), \ldots, (G_D(t))$ describes the optimal position vector of the group's particle in space with dimension D (also refers to as Global Best). The φ refers to population size, $p_{i,j}$ is the local attractor of each particle, and μ is a uniformly distributed random number between 0 and 1, (Sun et al., 2016). Equation 2.8 can be rewritten as:

$$p_{i,j}(t+1) = G_j(t) + \varphi.(p_{i,j}(\mathrm{t}) - G_j(t)) \tag{2.9}$$
$$\text{where } (1 \leq i \leq N, 1 \leq j \leq D)$$

It can be observed from equation 2.6 and 2.9 that the local attractor $p_{i,j}(t+1)$ is associated with the difference between the best position in the swarm $G_j(t)$ and the best position of the current particle $P_{i,j}(t)$. Its position $x_{i,j}(t+1)$ is associated with the difference between the average positions of current particles $c_i(t)$ and the position of the particle itself $x_{i,j}(t)$. The usage of these position vectors ensures the QPSO a stable convergence between particles, promoting faster and stronger searchers.

This quantum behaved PSO algorithm is observed to solve premature convergence and improve global optimisation performance when applied in mangroves classification (Li et al., 2015). Extension to the QPSO such as the new artificial bee colony algorithm with quantum PSO theory (QCABC) is reported to solve optimal power flow problem (Yuan et al., 2015). In Yueqiang et al. (2014), a QPSO-based algorithm is applied on welding manipulator path planning to expand the search range of particles and keep the PSO algorithm's good operability.

In mobile robot path planning, the QPSO algorithm performed well to get high efficiency of the searching process and successful obstacle avoidance using first the local than the global path planning strategy (Tokgo et al., 2014a; Tokgo et al., 2014b). The robot visual measurement system also benefitted from the QPSO. In 2013, Wang et al. used the QPSO to calibrate the robot twist angles obtained from laser tracker. The faster convergence rate of the QPSO minimized travelling time and distance and optimized robot trajectory (Guo et al., 2010; Guo et al., 2012).

Although the QPSO offers stable and ideal convergence of the swarm particles, the QPSO is not equipped to describe multi-robot applications, unlike the RDPSO. For completion, the following section shows a comparison between PSO-based works, the testing environment and their performances.

2.10 Comparison between the PSO-based works

Table 2.2 lists attributes of the PSO-based works in the literature, including the RDPSOs and the QPSOs. The list includes a description of the environment setup where the PSO-based methods are tested and its purpose. A performance measurement describing the achievement of each work completes the list.

Table 2.2: Comparison between PSO-based works

No	Authors & Year	Methods	Testing Environment	Task	Performance Measurement
1	Couceiro et al., 2011	PSO/ RDPSO	complex environment (30 x 30 meters)	search/ avoid obstacles	the RDPSO can converge faster, robust and more effectively in the searching task than PSO
2	Wang et al., 2018	PSO	grid environment	Communication	effectively improve coverage rate and reduce energy consumption
3	Tang & Eberhard, 2011	PSO	complex environment	Search/ avoid obstacles/ communication	the PSO creates search behaviour well and investigates the feature of fault tolerance
4	Song et al., 2017	PSO/ MDPSO	a smooth global path for mobile robots	search	the MDPSO achieved the best performance enable a particle search behaviour to adaptively adjust during a search process and reduce trap in the local optima
5	Cai et al., 2013	PSO	unknown environments 50 × 50 units	avoid obstacles	the effectiveness of the fuzzy obstacle-avoidance strategy which the robot trajectory smoothness in search space
6	Masehian & Sedighizadeh, 2013	PSO	dynamic environment 10 × 10 grids	search	the new method better performance, robustness, and scalability in searching task compare with a traditional PSO
7	de Sá et al., 2014	PSO/ BSA	without obstacles in the environment 100×100 units	search/communication	the BSA solve the localization problem is effective that require no more than 5.75% of processing time than the PSO algorithm
8	Islam et al., 2014	PSO	Dynamic/ static environment 20x20 meters	avoid obstacles	the method is flexible that you can change any parameters and improved performance to control avoiding or moving toward the goal.
9	de Sa et al., 2016	PSO	100×100 measurement units	communication	the method entirely distributed nodes with a low number of anchors that reduce the average localization error by 84%.

10	Dadgar et al., 2016	PSO /ARPSO	complex, with/without obstacle in environments 250 x 250 squares	search/avoid obstacles	the ARPSO increase convergence rate and increases exploration rate in the large environments and good performance for the small population
11	Couceiro et al., 2011	RDPSO	complex environment 300x300 meters	communication	the RDPSO performance decreased under communication constraints when the number of robots increases, and the maximum communication range is decreased.
12	Couceiro et al., 2012	RDPSO	large environment 2.55 m × 2.45 m real, 600 × 600 m simulation	Search/ avoid obstacles/ communication	the adaptive RDPSO is achieved a higher exploration behaviour keeping a high level of exploitation compare with nonadoptive RDPSO
13	Couceiro et al., 2012	RDPSO	larger environment 2.45m to 2.55 m real,300 x 300 meters simulation	control	when changing the frictional coefficient between .632 and 0 the RDPSO stable but if it > .632 and < 1 the algorithm unstable
14	Couceiro et al., 2012	RDPSO	dynamic environment 2.45m to 2.55 m real,300 x 300 meters simulation	communication	the optimal solution is achieved in approximately 90% of the experiments that show some situation interrupt communication between robots
15	Couceiro et al., 2013	RDPSO	complex environment 10 × 20 meters	communication	in this algorithm, each robot ensures a multi-connected MANET over time but is less susceptible to robot failures.
16	Couceiro et al.,2 013	RDPSO	indoor environments 20 × 10 meters	communication	the methodology reduces the communication that improving the scalability and applicability of the RDPSO

17	Couceiro et al., 2014	RDPSO	dynamic and unstructured environments 2000 m^2 simulation/ 2.0×1.8m real	search/avoid obstacles	the RDPSO successfully exploring "faster and more accurately " approximately 80% than the (EPSO, PPSO, GSO and AFS) approaches
18	Couceiro et al., 2014	RDPSO	complex environment2.55 × 2.45 m real,300 × 300 m simulation	Search/ avoid obstacles/ communication	the RDPSO improved converges faster and more accurately, but the algorithm exhibits some collisions and communication ruptures between robots
19	Couceiro et al., 2014	RDPSO	outdoor/ indoor environment 300x300 meters simulation	Search/ communication	the algorithm converges faster and more accurately using 75% of the experiments under the EST approach over 50% under the random deployment strategy
20	Couceiro et al., 2014	RDPSO	indoor environment 20× 10 m	communication	the RDPSO-based AODV reduces around 20% the number of required hops to deliver a packet that improving the scalability and applicability of the RDPSO algorithm
21	Wang et al., 2015)	RDPSO/ FRDPSO	unknown environment 50m x 22m	Search/ avoid obstacles/ communication	the fuzzy adaptive FORDPSO was the better effect on the multi-robot environment exploration compared with other PSO algorithms
22	Meng et al., 2010	QPSO/ PSO	static environments	search	the QPSO improved accuracy and efficiency quicker convergence speed in searching task compared with PSO
23	Guo et al.,2010	QPSO	environment with obstacles	Search/ avoid obstacles	the QPSO realized obstacle avoidance and fast convergence ability of mobile robots compared with GA

24	Wang et al., 2013	QPSO	unknown environment600 mm	search	this method improved the accuracy of the Calibrate robot visual measurement system
25	Guo et al., 2012	QPSO	flying space	Search/ avoid obstacles	the QPSO is efficiency improved to minimizing travelling time and distance without collision in the flying workspace
26	Yang et al., 2015	QPSO	static environments	search	the algorithm improve efficiency in the industry
27	Tokgo et al., 2014	QPSO	complex environment 16m×16m	Search/ avoid obstacles	the QPSO is better performance in convergence speed for trajectory planning in random obstacles environments.
28	Li et al., 2017	QPSO/ PSO	outdoor environments (earthquake, rainfall)	search	QPSO-LSSVM has quickest search velocity and the best convergence performance compared with the PSO-LSSVM
29	Yuan et al., 2015	PSO-QCABC	static environments	search	the QCABC can effectively solve the OPF problem, the stability and optimization results are all the better than (PSO, GA, ABC)
30	Huang et al., 2016	AQPSO/ QPSO/ PSO	environment without obstacles 320 × 320	search	SNN+AQPSO better performance global exploration ability and faster convergence speed compared with QPSO and PSO

Table 2.3: Features and gaps of the classical PSO, RDPSO and the QPSO algorithm

Features	**Classical PSO**	**RDPSO**	**QPSO**
Collision Free	X	X	✓
Avoid obstacles	X	✓	X
Communication	X	✓	X
Coverage region	X	X	✓
Stability	✓	✓	✓
Run time	X	X	✓

Table 2.2 describes works utilizing the PSO-based algorithm. Significant gaps are elicited from Table 2.2 and summarized in Table 2.3. Next, the review on two communication protocols, i.e. MANETs and MR-LEACH, is presented.

2.11 Communication Protocols for Swarm Robotics

Effective cooperation in swarm robotics stems from maintaining the communication network among robots in the swarm. With communication protocol an important utility for the RDPSO algorithm, it is not surprising that only the RDPSO checks the obstacle avoidance and communication features in Table 2.3. For tasks such as the SaR mission, the swarm robots' requirement includes preserving interconnectivity even when the communication infrastructure risk interruptions. Preserving interconnectivity is important so that the robots can guarantee the continuous exchange of information within the multi-hop network paths to not unnecessarily restrict the team's range.

Communication networks are useful for coordination and cooperation between agents belonging to a given MRS. Communication networks do not require particular infrastructure for the setup and can be fixed anytime and anywhere. The flexible setup makes communication networks popular for unstructured situations such as military missions where soldiers communicate for instructions, autonomous robots dispatched and organized in unmanned space exploration, and situations requiring remote data collection. A communication network's key feature includes many distributed nodes (e.g. robots) that organize themselves into multi-hop wireless networks (Couceiro et al., 2014). These nodes are allowed to cooperate and direct messages to each other and perform the dual roles hosts and routers.

There are several characteristics required in the design and development of a communication network. A node often corresponds to a robot through an embedded processor or low-power radio, usually, battery operated. For the communication network to be cost-effective, the on-board processing, wireless communication capabilities and the battery supply of each robot are minimal. Subsequently, external conditions from the environment can factor in physical damage to the nodes, or internally, the battery can fail or die.

It is also observed that in real-world MRS applications, some robots are commonly dispatched in dangerous or environments with less access, making battery replacements very difficult. It is then near impossible to repair the robots or renew their energy. In exploring mobile robots, the distributed networks' architecture depends on time, and the connection strength can fluctuate, with intermittent signals or loss entirely over time (Couceiro et al., 2011).

In this section, two communication protocols are reviewed. One is the MANETs, popularly used on the RDPSO architecture by Couceiro's team and the other is the MR-LEACH. This review covers fundamentals and the characteristics of the protocols to support swarm robots in SaR mission.

2.11.1 Mobile Ad Hoc Network (MANET)

The MANET supports coordination and cooperation between MRS agents, contributing to the protocol's popularity in maintaining network interconnectivity. The MANET follows the design and architecture of a communication protocol and offers specific advantages as follows:

1. Flexibility: installation is easy and can set up in any place and situation, depending on the need (Bang et al., 2013), thus providing a significant advantage over other types of networks that need a prior infrastructure for correct operation.
2. Movement of the nodes: to fit into numerous situations in which this type of network is useful, it is recommended that the nodes have a free movement within its coverage to allow users to perform other tasks and remain connected (Bernal et al., 2017).
3. Decentralization: the devices within the network are autonomous nodes, which allows anyone to play the role of a router without the need for any device to rout the messages, as long as the failure to any link does not disrupt or collapse the setting of the communication as a whole (Kaur et al., 2013).
4. Scalability: adding new nodes to the network does not risk failure to establish communication for this type of network, since the nodes are in constant movement (Helen et al., 2014).
5. Economy: its installation cost is low because it does not depend on traffic administrators, such as switches and routers. Also, it does not need wiring nor a centralized administration (Bang et al., 2013).
6. Multi-hop: each of the nodes is attached with low power antennas, which limit coverage range. The MANET's multi-hop feature takes care of this problem by sending acknowledgement messages to their adjacent nodes until the message reaches a teammate within reach of the target device and achieves total communication (Kaur et al., 2013).
7. Autonomous formation: a peer-to-peer (P2P) standard allows the nodes to form their dynamic topology when connecting with the MANET. This

feature removes the need for human intervention since the nodes can always establish communication as long as they are within the coverage area (Singh et al., 2010).

8. Autonomous organization: with the availability of such a dynamic topology, organizing the nodes becomes plausible to the devices, and the devices are not affected by the entry or exit of terminals in the network (Bernal et al., 2017).

The MANET has been used widely in application with potential communication breaks and constraints. To maintain full network connectivity, Tardioli et al., (2010) proposed a multi-robot cooperative motion control technique based on a virtual spring-damper model to prevent communication network splits. They implemented a task allocation algorithm that takes advantage of the network link information to ensure autonomous mission and a network layer capable of sustaining hard real-time traffic and changing topologies.

Another solution to maintain full connectivity is intermittent connectivity, where the networks must get online at a pre-defined interval (Hollinger et al., 2018). The periodic connectivity strategy benefit in situations where it may be desirable to temporarily break the connectivity, consequently decreasing the number of robots and the explicit information exchanged between robots of the same sub-network. It is also observed that it is plausible to direct messages gathered by the sub-networks when the network regains connectivity. The periodic connectivity algorithm performs better than other MANET designs that require a continuous connection. However, in the experimentation, for example, in Singh (2010), the performance is not extended to unknown or dynamic environments.

Researchers are working to further MANET capacity by focusing on improving the nodes' connectivity resilience in the network. Their investigation includes:

1. Energy: power conservation is always an issue when deploying devices on batteries. With time, the nodes of the MANETs will have a reduced lifetime leading to broken communication links which affect connectivity (Bang et al., 2013).
2. New protocols: since the MANET proposed a dynamic topology, new protocols to discover and maintain communication routes, and to eliminate routes that are no longer feasible or favourable to the communication can be the solution to control CPU consumption and improve energy conservation (Singh et al., 2010).
3. Broken links: with constant random movement of the nodes within the network, it is likely that some may enter or leave the MANET coverage area, ultimately destroying established routes or the routes needs recalculation to re-establish connection (Helen et al., 2014).

Thus, the MANET can be considered a temporary self-organizing network of wireless mobile nodes that can stand on its own without pre-existing infrastructure, unlike conventional communication networks. This section covered the current state of MANET's art, including some challenges relating to routing, power management, location management, and multimedia over ad hoc networks. Since there is no fixed infrastructure available for MANET with mobile devices, routing becomes a critical issue. The following section describes another communication protocol, the MR-LEACH, including its applications and gaps for future work.

2.11.2 Multi-hop Routing Algorithm with Low Energy Adaptive Clustering Hierarchy (MR-LEACH)

Like MANET, the MR-LEACH offers communication protocol towards coordination and cooperation between agents belonging to the MRS. The MR-LEACH follows the design and architecture of a communication protocol described in Section 2.8. For example, it is also based on many distributed nodes, forming multi-hop wireless networks. Routing messages from one location to another, robots as nodes in MR-LEACH applications can act as hosts and routers. The MR-LEACH behaves differently over other communication networks because we cannot determine the connectivity and robustness as early as it is in MR-LEACH.

For this reason, enforcing the MR-LEACH protocol on the communication network can provide prevention from loss of connectivity. The protocol also serves well as a fault tolerance strategy. Most significantly, the MR-LEACH allows node redundancy which consequently turns the topology dynamics to multi-connectivity strategy. The definition of k-connectivity, or k-fault tolerance, $k \in N$, is the exclusive communication pairing of one robot to another robot. Each robot should be connected to at least k disjoint robot paths.

The availability of this *bi*-connectivity means that in the worst-case scenario, a k connected MR-LEACH requires the failure of k robots to get disconnected. Multi-connectivity is highly favourable for fault tolerance and boosts communication capacity. Establishing bi-connectivity is a critical contribution of MR-LEACH regarding MRS application and the approach is gaining popularity (Casteigts et al., 2010; Abdulla et al., 2012; Mahapatra & Yadav, 2015; Salem & Shudifat, 2019).

Development teams require autonomous communication between robots for surveillance missions. With MR-LEACH as protocol, the robots can maintain the direct exchange of messages in the multi-hop network, without restricting the team's range. In Baroudi et al. (2017), a wirelessly energy-charged (WINCH) protocol is proposed to maintain communication links with battery maintenance, combining low-energy adaptive clustering hierarchy-centralized protocol (LEACH-C) and the routing process wireless networks. The experimental results show the WINCH protocol has better energy consumption performance, network throughput and coverage, demonstrating effectiveness than traditional protocols.

The multi-hop routes are established and used between the nodes (Tardioli et al., 2010) to maintain the network's full connection. This way, the communication link's quality can be measured, useful in restricting the robot's movement. Afsar et al. (2014) introduced a LEACH upgrade to the communication protocol where at initialization nodes are randomly selected to become a cluster head (CH). When a node becomes CH, it is responsible for performing broadcasting advertisement message. Upon receiving this advertisement message, other non-cluster nodes will decide to join a specific CH depending on the Received Signal Strength (RSS).

The CH creates time-division multiple access (TDMA), i.e. a roster table, to transmit schedule for each node in the cluster. The CH then compiles or aggregates the data from various nodes inside the cluster and sends it to the base station. At every other step, a different node is selected as CH. Except for the beginning where nodes are randomly selected as CH, at every other step, the node with the highest energy concentration wins the selection. Equation 2.10 is the formula used by the CH to distribute load among all participating nodes.

$$p_{i,j}(t+1) = G_j(t) + \varphi.(p_{i,j}(\mathrm{t}) - G_j(t)) \tag{2.10}$$

$$\text{where } (1 \le i \le N, 1 \le j \le D)$$

$$T(n) = \begin{cases} \dfrac{p}{1 - p * \left(r \, mod \dfrac{1}{p}\right)} & if\ n\ \in\ G \\ 0 & otherwise \end{cases} \tag{2.11}$$

Where,

P is the percentage of nodes to be elected as cluster heads in the whole network,

r is the current round (or step), and

G is the set of nodes that have not been cluster heads in the last $1/P$ rounds.

Clustering is essential for wireless sensor networks, and in MR-LEACH, it is termed the multi-hop clustered algorithm. Jiang et al. (2016) proposed an energy-balanced unequal clustering (EBUC) communication protocol able to partition the sensor network and turns them into many uneven clusters. Hence, one source communicates to the base station via the multi-hop channel. This way, any node with the highest energy level can promote itself to become a CH. About equation 2.10, nodes which are CH in round r is limited from the selection in the next 1of P rounds. One issue associated with LEACH is that all CH must reach the base station in a single hop.

Depending on the signal transfer range between a sensor node and its receiver, a node can select a CH from the broadcasted list of available candidates. After the formation of CHs, different clusters will choose their base stations. The cycle is repeated until the completion of it. The varying size of clusters and the various cluster hierarchy levels in the same network are challenging for scheduling. It is observed that the absence of TDMA leads to wastage of energy.

For improvement, Lee et al. (2011) discussed the utility of an energy-efficient scheme call the Location-based Unequal Clustering Algorithm (LUCA). LUCA localizes the nodes and forms CHs depending on the cluster's distance from the targeted sink. The farther a cluster from its sink, the larger is the cluster size. Far clusters take more energy than those nearer to the base station.

Protocols such as the Hybrid Energy-Efficient Distributed (RHEED) is introduced (Mardini et al., 2014) to support efficient clustering routing. The main objective of RHEED is to spread out the energy consumption so the network lifetime can be extended and to minimize energy wastage during CH selection. RHEED can also reduce the control overhead of the network to a minimum. RHEED outperforms the HEED protocol by more than 20% in term of network lifetime and residual energy.

To summarize, the following are MR-LEACH advantages:

1. Scalability: CHs are supposed to manage the network by listening and picking up data from the neighbourhood's communication traffic. Clustering topology can improve CHs performance by first dividing the sensor nodes into different classes of clusters, each with a particular assignment. In a clustering routing scheme, the clustering topology can set up a route from inside the cluster so that the routing table in each node are not overwhelmed. In comparison to flat topology, the clustering topology is more compact and more comfortable to sustain. It is also more scalable with larger node community is present (Tan et al., 2017).

2. Data Aggregation/Fusion: The CHs are responsible for aggregate data from within its clusters and other CHs. So internally, a member in the cluster only have to direct messages to its CH. The CHs will compile the data and transmit to the sink or base station. Such organization removes redundancy immensely and highly effective in saving network energy. With the introduction of this clustering data aggregation technique, the CHs multi-hop, and form a tree structure for data transmission, significantly reducing energy wastage (Yuea et al., 2012).

3. Less Load: With the elimination of redundant data transmissions, the network is given a new vantage point of view to review the problem (target) from other perspectives (Izadi et al., 2015). The network can trace and make a better estimation as data is cleaner with less noise.

4. Less Energy Consumption: Performing clustering with intra-cluster and inter-cluster communications reduces the number of sensor nodes performing the task in long-distance communications, thus allowing less energy consumption for the entire network. To further save and conserve energy, Lee et al. (2011) allow only CHs to perform data transmission in the clustering routing scheme.

5. More Robustness: Clustering routing scheme is useful for network topology control and corresponds to network changes such as node increment, node mobility and unpredicted failures. A clustering routing scheme only needs to react with these changes locally leaving the entire network more robust and more convenient to manage. CHs are generally rotated among all the sensor nodes to avoid single point failure in clustering routing algorithms to share the CH task.

6. Load Balancing: Load balancing can prolong the network lifetime in WSNs. Even distribution of sensor nodes among the clusters helps accomplish cluster construction where CHs must perform data processing and intra-cluster management. Generally, equal-sized clusters can sustain the CHs and prevent premature energy exhaustion. For an alternative, the multi-path routing can also lead to achieving load balancing.

7. Fault-Tolerance: In dynamic scenarios, sensor nodes may suffer from energy depletion, transmission errors, hardware malfunction and malicious attacks. In some applications like hurricane modelling and vision tracking, many small sensor nodes are deployed with each sensor node's cost constrained. The cost constraints, quality of sensor nodes, and considering the hostile environment, the sensor networks are prone to failure. Fault tolerance is crucial to reduce data loss from key sensor nodes. Re-clustering is the most intuitive fault-tolerant method to recover from a cluster failure, albeit the mess created during on-going operation. Assignment of CH backup is a crucial aspect for recovery from a CH failure.

8. Latency Reduction: When a WSN is divided into clusters, only CHs perform data transmissions out of the cluster, avoiding collisions between the nodes. Collision avoidance subsequently reduces latency. Usually, data transmission is performed hop by hop and flooding in a flat routing scheme, but in clustering routing scheme, only CHs perform data transmission. This decrease hops from data source to the base station, leading to decrease latency.

Researchers have proposed many protocols such as LEACH, HEED, MR-LEACH to explore the energy-efficient protocol technology, and we had gone through different research papers. Few of them are given below

Table 2.4: Comparison between Hierarchical protocols

Authors	Protocols	Advantages
Lin, et al. (2015)	LEACH	It is a clustering-based technique, and the cluster head in the network directly communicates with the base station in a single hop. It has two phases, set-up and steady-state phase.
Mishra et al. (2012)	CTPA	It is a tree-based technique; it has a low consumption of energy compared to LEASE. It has two phases, Chain formation phase, Broadcasting phase. The lifetime of PEGASIS would be more if we compared it with LEACH
AnandRao, et al. (2018)	HEED	It is a clustering-based technique suitable for heterogeneous WSN It has three phases: initialisation, set-up, and steady. The lifetime of HEED is more compared to CTPA.
Vijayvargiya, et al. (2012)	TCDGP	The length of path form end, leaf node to root/chain node in TREEPSI is shorter than CTPA. The data will not send data for a long path. For In TREEPSI power consumption is less if we compared it with PEGASIS
Kim et al. (2010)	TBC	Nodes in a cluster form a tree with the root as the cluster-head, while the height of the tree is decided based on the distance of the member nodes to the cluster-head
Han, et al. (2014)	GSTEB	A General Self- Organized Tree-Based Energy- Balance routing protocol is used to achieve a more extended network lifetime. Each round BS assigns a root node and broadcasts this selection to all sensor nodes. each node selects its parent by considering only itself and its neighbour's details thus making a dynamic protocol

Table 2.5: Comparison between Communication protocols

Performance Metrics	**MANET**	**Leach**	**Leach-C**	**MR-Leach**
Routing principle	Flat	Hierarchical	Hierarchical	Hierarchical
Save Energy Consumption	Less	Limited	Maximum	Maximum
Efficiency	Poor	Medium	Poor	High
Network lifetime	Good	Good	Good	Very good
Mobility	Supported	Fixed BS	Fixed BS	Fixed BS
Clustering Method	N/A	Distributed	Centralized	Hybrid

Finally, this research focuses on swarm robotics, a domain that embodies swarm intelligence mechanisms into robotics. More specifically, this research proposes a complete, swarm robotic solution applied to real-world missions. The search and rescue (SaR) missions were considered a case study due to their inherent complexity level to test the proposed solution. Such operations often occur in highly dynamic and large scenarios, with harsh and faulty conditions, that pose several problems to swarm robot applicability. This study focuses on these problems raising new challenges that cannot be handled appropriately by a simple adaptation of state-of-the-art swarm algorithms, planning, control and decision-making techniques.

In summary, the advantages of the RDPSO algorithm are threefold. First, the RDPSO is scalable to large populations of robots. Second, RDPSO can decrease the amount of required information exchange among robots. Thirds, the RDPSO has faster convergence and more accurately than the other approaches. Additionally, the RDPSO takes advantages of the QPSO used position vector to guarantee stable convergence to stationary between robots, to avoid collisions, so it's fast and robust searchability. The disadvantages of RDPSO include collisions occurring between robots, communication ruptures between robots, and premature convergence.

This research's contribution revolves around extending the Robotic Darwinian Particle Swarm Optimization (RDPSO) to swarm robotics, using the Quantum-PSO (QPSO) features to enhance the searching capabilities of RDPSO. This novel extension is denoted as Quantum Robotic Darwinian Particle Swarm Optimization (QRDPSO). The QRDPSO is a distributed swarm robotic architecture that benefits from the dynamical partitioning of robots' whole swarm. The QRDPSO is proposed and can be used to devise the applicability of novel approaches.

The other contribution from this study is the utility of the Multi-hop Routing Algorithm with Low Energy Adaptive Clustering Hierarchy (MR-LEACH) schema to enhance the communication of the QRDPSO. The MR-LEACH is the most popular energy-efficient algorithms for Wireless sensor network and its evolutionary properties. Significantly, a quantum-behaved swarm robot for exploration with enhanced communication will help researchers to uncover critical areas in robot dynamics and investigate new strategies for obstacle avoidance and swarm coordination.

2.12 Chapter Summary

This chapter presented a review on the PSO series and two necessary communication protocols towards swarm optimization. Despite significant progress and achievements, there is still scoped to deliver a more efficient PSO variation with accurate search, obstacle avoidance without collision and communication ruptures between robots. Chapter 3 will show how I adapted the QPSO onto the RDPSO to derive a new PSO-based algorithm for swarm robotics application. Included in the chapter is adopting the MR-LEACH schema for free collisions full interconnectivity between robots.

CHAPTER 3: METHODOLOGY

3.1 Overview

This chapter is organized to present this thesis's central aim by methodically describing the Quantum Robotic Darwinian Particle Swarm Optimization (QRDPSO) and adopting the MR-LEACH schema as a communication protocol. However, it is important to acknowledge that even though the discussions are exclusive around the idea of QRDPSO, its approach, parameterization, and insights can, and should, be applied to other swarm robotic algorithms.

This thesis proposed three main contributions:

1. A novel QRDPSO algorithm that improves convergence speed rate during swarm-robot exploration over RDPSO algorithm,
2. A coordinated swarm movement strategy which conserves the robot's energy and extends the robot's lifetime during exploration, and
3. Adoption of the MR-LEACH schema towards robot interconnectivity and mobility.

The first contribution revolves around the derivation of the QRDPSO and is presented in Section 3.2. Contribution two and three are results from the adoption of the MR-LEACH schema onto the QRDPSO and are presented in Section 3.3.

3.2 Derivation of the QRDPSO

This algorithm aims to propose a new fitness or cost function that can drive robots towards global best and successfully perform obstacle avoidance. Maintaining communication is critical, so when a robot moves from any position to the target position, it can avoid both static and dynamic obstacles in the environment. In this work, the proposed QRDPSO approach is inspired by the success of the RDPSO contribution to mobile robots, mainly the adoption of three general features (from RDPSO) as follows:

1. To formulate a searching strategy to reach global best in shorter time in existing RDPSO algorithm,
2. Integration of an obstacle avoidance behaviour to avoid collisions, and
3. A way to enforce multi-hop clustering network connectivity to ensure that the MR-LEACH remains connected throughout the mission to enhance robots' lifetime.

Based on the inertia-weighted parameters for the QPSO algorithm which have been defined in equation (2.7), including new parameters $is_{ij}(t)$ and $im_{ij}(t)$ into the original term $(X_{i,n}^{j} - C_{n}^{j})$ of (2.7), will derive the QRDPSO as follows:

$$X_{i,n+1}^{j}(t+1) = P_{i,n}^{j} \pm \left(\alpha_1 \left|X_{i,n}^{j} - C_{n}^{j}\right| + \alpha_2 \left|X_{i,n}^{j} - im_{n}^{j}\right| + \alpha_3 \left|X_{i,n}^{j} - is_{n}^{j}\right|\right) \ln\left(\frac{1}{u_{i,n+1}^{j}}\right) \tag{3.1}$$

From equation (3.1), it can be observed that the values of im determine the robot's movement. Given the mounted sensors' readings and the communication signals' strength between the robots, each one of these robots has the option to move in search for a better objective function. Nevertheless, that movement is bounded within the limitations of communication constraints.

It is observed that, benefiting from the quantum-behaving particles' searching capabilities in QPSO, hypothetically speaking, the robots can avoid any optimal local solution and reach the optimal global solution within a shorter time with quantum behaviour. For obstacle avoidance, it is assumed in QRDPSO that every robot is equipped with sensors suitable for finding obstacle location within a finite sensing radius r_s. The sensing function is can now be defined in equation (3.1). This function describes the sensor data such as the distance from the robot to obstacles or detected objects from the surrounding.

3.2.1 Obstacle Avoidance Function

The significant numerical modification that differs the QRDPSO compared with the RDPSO is the proposition of having a cost or fitness function that can be minimized or maximized depending on the mission objective. Given a scenario, for instance, a gas leaking situation, the swarm of robots running the QRDPSO algorithm must try, at each time step, to maximize the sensed gas. At the same time, minimize the distance between the robot and the location of the leak.

In another situation where the objective function is to locate the fire outbreak, the QRDPSO robots must maximize the fire's visual and work to minimize the distance between the robot and the outbreak's location. Conceptually, this is how the cost or fitness function determines the QRDPSO swarm' behaviour.

During a search exploration, the swarm of robots dispersed in the search area at random. Each with an assignment to fulfil a target searching task. The robot intends always to move closer to an identified solution. However, depending on the situation, it is often the environment has obstacles. The complexity of the environment can drastically change to worst when dynamic obstacles are involved. These robots must not get stuck at any obstacles to ensure success in target searching. Such behaviour can be defined numerically. One can assume that each robot has sensors sufficient to perceive the environment, including obstacles, within a finite sensing radius r_s. In the QRDPSO, the perceiving or sensing function $q(x_i[t])$ can be defined. Sensor data feed the function with distance information between each robot and the obstacle(s) in their vicinity, respectively.

In the case of robots equipped with range finders such as the sonar and the laser, the range finders can provide the time-of-flight value recorded when they bounce between its transmitter and receiver. Each bounce will carry signal either in sound waves for the sonar or light for the laser. According to the speed=distance/time equation, the reflected signal can estimate the distance value between the robot and any solid objects (i.e. obstacles) from the robot surrounding. The value is updated through the sensing function to the QRDPSO equation at every interval.

The robot then adjusts its movement bearing its distance to an obstacle(s). When there is an obstacle occluding a robot to a target, the robot needs to get closer to the obstacle (consequently the target) while keeping out from impact by altering its angle. Depending on the number and distance of the obstacle(s), a robot may have to perform several turnings overtime to keep away at a safe distance. If there are no near obstacle(s) obstructing the robot's line of sight, the robot should maintain its direction in drive forward towards the target.

To model a robot's susceptibility for the obstacle avoidance scenarios, the QRDPSO requires standard deviations. Two values are proposed, one is for obstacle susceptibility $\sigma(q_i(t))$, and the other is to keep track of the current position of the robot $\sigma(x_i(t))$. Together, these standard deviation values can be used to calculate each robot's trajectories in the swarm, including when a robot is passing an obstacle. At any interval, the individual robot's susceptibility is defined as follows:

$$is_{i,j}(t) = is_{i,1}(t), is_{i,2}(t), \dots, is_{i,D}(t) \tag{3.2}$$

$$= \frac{\sigma\left(q_{i,1}(t)\right)}{\sigma\left(x_{i,1}(t)\right)}, \frac{\sigma\left(q_{i,2}(t)\right)}{\sigma\left(x_{i,2}(t)\right)}, \dots \frac{\sigma(q_{i,D}(t))}{\sigma(x_{i,D}(t))} (1 \leq i \leq N) \tag{3.3}$$

In modelling obstacle avoidance, ideally the value of $is_i(t)$ and the susceptibility should be directly proportional to each other, where $(0 < is_i(t) \leq 1)$. If the value of $is_i(t)$ is always equals to 1, it means the robot successfully avoided all obstacles. In other words, when the robot i reaches $\sigma\left(q_{i,1}(t)\right)/\sigma\left(x_{i,1}(t)\right) = 1$, it means the robot has been updating its positions while managing to avoid impact with any solid object in the vicinity. Readings from the sensors are joined on one robot during its movement and compared to the output value with a predefined threshold to improve the obstacle avoidance. Thus, the final value of (3.3) may lie in the interval $(0,1)$, which helps determine the robots' trajectories.

Obstacle avoidance is an important consideration in the derivation of the QRDPSO for practical swarm robotics applications. Another critical factor in improving the odds for successful swarm missions is the inter-robot communication quality. The following subsection presents a method to achieve inter-communication between one robot to another.

3.2.2 Communication rate between the robots

For the swarm robot to maintain communication, the connectivity between robots is described as a link matrix $L = \{l_{i,f}\}$ which can be calculated as functions of either distance d_{max} or signal quality z_{min} or both. Together, they form the adjacency matrix $A = \{a_{if}\}$ and can be defined as follows:

$$a_{ij} = \begin{cases} 1, nodes\ i\ and\ f\ connected \\ 0, no\ connection \end{cases} \tag{3.4}$$

In the QRDPSO, the multi-hop concept is applied where nodes can represent robots. In multi-hop routing, to exchange messages between the nodes, other nodes can be used as relays. A connectivity matrix can be defined to construct the multi-hop connectivity for the network of robots. This matrix's values, an adjacency matrix, can represent the hop distances, a strategy in WSNs to reduce long hop for node energy conservation. To calculate how many hops needed so a node can transmit a message to a particular node in the network, i.e. hop distance, any diagonal entries in the adjacency matrix with value zero can be modified. Nonadjacent values in the matrix denote a robot communicating with another that is far away.

According to Couceiro et al. (2011), a connectivity matrix can be defined as $C^k = \{c^k_{i,f}\}$, where the entry (i, f) denote the minimum number of hop count required to perform the connection between the nodes i and f. The parameter k is used to denote the iteration, which can change according to the amount of hop count the network can manage. The connectivity matrix can be defined as follows:

$$c^k_{ij} = \begin{cases} h, i\ connected\ to\ f\ by\ h \leq k\ hops \\ 0, otherwise \end{cases} \tag{3.5}$$

Some limits are required to control the positioning of each robot $X_{i,n+1}^{j}$ within the boundaries of the communication, range not to affect the link matrix. Apart from safeguarding the boundaries, an alternative towards total connection is to *force* each robot always to hop an adjacent neighbour that, exclusively, has not selected it as its nearest neighbour. This way, at each time step, total connectivity among exclusive pairs of robots can be promoted.

Due to the inter-communication between robots highly dependent on signal range and quality, defining the minimum or maximum value of each line of the adjacency matrix A, after excluding zeros and if the (i, f) pairs have been previously chosen, can enforce the link to be active. A connectivity function $m\left(x_i(t)\right)$ Can be defined. The connectivity rate of individual robots within a swarm is given by the standard deviation value of the connectivity function $\sigma(m_i(t))$ and the standard deviation value of the current position of the robot $\sigma(x_i(t))$, respectively. At any point, the individual robot connectivity can be determined according to:

$$\begin{aligned} im_{i,j}(t) - \left(im_{i,1}(t), im_{i,2}(t), \dots, im_{i,D}(t)\right) \\ = \frac{\sigma\left(m_{i,1}(t)\right)}{\sigma\left(x_{i,1}(t)\right)}, \frac{\sigma\left(m_{i,2}(t)\right)}{\sigma\left(x_{i,2}(t)\right)}, \dots, \frac{\sigma(m_{i,D}(t))}{\sigma(x_{i,D}(t))} (1 \le i \le N) \end{aligned} \tag{3.6}$$

For optimization, ideally the value of $im_i(t)$ and the susceptibility should be directly proportional to each other, where $(0 < im_i(t) \le 1)$. If the value of $im_i(t)$ remains 1, it means the swarm of robots successfully connected. In other words, when the particle i reaches $\sigma\left(m_{i,1}(t)\right)/\sigma\left(x_{i,1}(t)\right) = 1$, it means the robot is in a position where it has connectivity with its neighbour.

The position sets the connectivity constraints of the robots' movements, while at the same time reducing the calculation overhead so that the robots can plan their movements considering the communication constraint, based on one value only. Moreover, this equation is adaptable with different communication techniques to permit the robots to correspond.

3.2.3 QRDPSO Equation Numerical Evaluation

Following Sun et al. (2004 (December)), seven benchmark functions are applied to test the performance of the QRDPSO against the classical PSO algorithms. The algorithms, SPSO, QDPSO, RQPSO1 and RQPSO2 are used for comparison with all minimization functions defined with a minimum value of zero. The first function is called the Sphere, which has local minima. It is continuous, convex and unimodal and described by:

$$f(x) = \sum_{i=1}^{d} x_i^2 \tag{3.7}$$

Where its global minimum is $f(x) = 0$, at $x = 0$ and was used within the range $(-100, 100)$. The second function is the De Jong's, multimodal, with very sharp drops on a mainly flat surface. It is described by:

$$f(x) = \sum_{i=1}^{d} i \times x_i^4 \tag{3.8}$$

Where its global minimum is given $f(x) = 0$, at $x = (0, \ldots, 0)$ and is used within the range $(-100, 100)$.

The third function is the Rosenbrock which is unimodal, and the global minimum lies in a narrow, parabolic valley. However, even though this valley is easy to find, convergence to the minimum is difficult (Sun et al., 2011). It is described by:

$$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_1 - 1)^2] \tag{3.9}$$

Where its global minimum is defined as $f(x) = 0$, at $x = (1, \dots, 1)$ and is used within the range $(-5.12, 5.12)$. The fourth function is the Griewank which has many widespread local minima that are regularly distributed and described by:

$$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{3.10}$$

Where its global minimum is $f(x) = 0$, at $x = (0, \dots, 0)$ and is used within the range $(-5.12, 5.12)$. The fifth function is the Rastrigin function. It is an example of a non-linear multimodal function. Finding the minimum of this function is a somewhat difficult problem due to its large search space and its large number of local minima.

$$f(x) = \text{An} + \sum_{i=1}^{d} [x_i^2 - A\,cos(2\pi x_i)] \tag{3.11}$$

Where its global minimum is A=10, $f(x) = 0$, at $x = (0, \dots, 0)$ and is used within the range $(-5.12, 5.12)$. Finally, the Schaffer function is shown on a smaller input domain in the second plot to show detail.

$$f(x, y) = 0.5 + \frac{sin^2(x^2 - y^2) - 0.5}{[1 + 0.001(x^2 + y^2)]^2} \quad (3.12)$$

The function is usually evaluated on the square xi ∈ [-100, 100], for all i = 1, 2. with global minimum $f(x, y) = (0,0)$.

For the numerical evaluation, the test conditions for the used benchmarks set the population's size to 20, 40 and 80, while maximum generation is 1000, 1500 and 2000, corresponding to particle dimension set to 10, 20, and 30. When the α minimum values are set to 1.5, and the α maximum values are set to 2, it is observed that the μ decreases linearly from 1.5 to 1, when the algorithm is running. Under a similar test condition, the fitness functions are measured for every single accessible set of models and the proposed four benchmark functions. The mean of the best fitness values and the standard deviations for 50 times iteration for each function are shown in Tables 3.1 to Table 3.7.

Table 3.1: The Mean Fitness Value for the Sphere Function

M	D	$Gmax$	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	10	1000	1e-20	1e-25	1e-31	1e-40	1e-76
	20	1500	1e-11	1e-15	1e-20	1e-23	1e-32
	30	2000	1e-06	1e-08	1e-11	1e-16	1e-35
40	10	1000	1e-23	1e-41	1e-62	1e-64	1e-86
	20	1500	1e-14	1e-23	1e-32	1e-37	1e-43
	30	2000	1e-10	1e-14	1e-23	1e-26	1e -25
80	10	1000	1e-28	1e-61	1e-82	1e-85	1e-94
	20	1500	1e-17	1e-32	1e-50	1e-55	1e-50
	30	2000	1e-12	1e-19	1e-38	1e-38	1e-32

Table 3.2: The Mean Fitness Value for De Jong's Function

M	D	$Gmax$	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	10	1000	0.0000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000	0.0000
40	10	1000	0.0000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000	0.0000
80	10	1000	0.0000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3.3: The Mean Fitness Value for Rosenbrock Function

M	D	$Gmax$	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	10	1000	96.1715	14.2221	13.8377	10.8643	5.2111
	20	1500	214.6764	175.3186	116.0543	97.9443	35.4802
	30	2000	316.4468	242.3770	152.1783	135.8685	94.3286
40	10	1000	70.2139	15.8623	12.9653	10.2468	4.1218
	20	1500	180.9671	112.4612	52.9421	80.3842	74.0324
	30	2000	299.7061	76.4273	75.6933	69.2908	48.7336
80	10	1000	36.2954	36.3405	11.8327	9.8421	5.2525
	20	1500	52.2802	23.5443	19.7310	17.6420	15.6642
	30	2000	205.5596	71.9221	58.5165	53.6345	31.1417

Table 3.4: The Mean Fitness Value for Griewank Function

M	D	$Gmax$	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	10	1000	0.0919	0.1003	0.0078	0.0051	0.0000
	20	1500	0.0303	0.0051	0.0002	0.0002	0.0000
	30	2000	0.0182	0.0544	0.0011	0.0009	0.0000
40	10	1000	0.0512	0.0484	0.0009	0.0006	0.0000
	20	1500	0.0251	0.0004	0.0002	0.0002	0.0000
	30	2000	0.0127	0.0009	0.0001	0.0000	0.0000
80	10	1000	0.0760	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0288	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0128	0.0000	0.0000	0.0000	0.0000

Table 3.5: The Mean Fitness Value for Rastrigin Function

M	D	G_{max}	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
q	10	1000	5.5572	4.9698	4.5712	4.5823	3.5632
	20	1500	22.8892	17.0789	16.0244	15.2001	14.6727
	30	2000	47.2941	48.6199	35.2052	33.5101	30.6121
40	10	1000	3.5623	2.0328	2.0489	2.1459	1.9727
	20	1500	16.3504	10.9453	10.2717	9.2517	9.7568
	30	2000	38.5250	21.3712	23.4756	20.8164	18.8761
80	10	1000	2.5379	0.9232	0.8871	0.7298	0.6352
	20	1500	13.4263	6.9554	7.2781	6.4174	6.1462
	30	2000	29.3063	18.130	19.9324	17.3473	16.4763

Table 3.6: The Mean Fitness Value for Shaffers Function

M	D	G_{max}	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	2	2000	0.0012	0.0051	7.9437-e4	2.9951-e5	1.8524-e3
40	2	2000	0.0006	0.0018	1.5385-e5	2.8818-e7	1.5693-e4
80	2	2000	0.0002	0.0004	8.5111-e7	7.2959-e8	6.6832-e8

Table 3.7: The Mean Fitness Value for Rosenbrock valley Function

M	D	G_{max}	SPSO	QDPSO	RQPSO1	RQPSO2	QRDPSO
20	2	2000	0.0000	0.0000	0.0000	0.0000	0.0000
40	2	2000	0.0000	0.0000	0.0000	0.0000	0.0000
80	2	2000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3.1 demonstrates that the QRDPSO is the best for precision, and it is significantly steadier than the other four models regardless of the used values of N, D and the maximum iteration. Since the Sphere function is unimodal, it is usually used to test the algorithm's local search capabilities. The results show that the local search capabilities of the QRDPSO are superior to those of the SPSO, QDPSO, RQPSO1 and RQPSO2. The De Jong's function from Table 3.2 has a similar property with the Sphere function, so it can be noticed that applying the De Jong's function on the QRDPSO is much improved than that of the Sphere function.

On the other hand, for the Rosenbrock, the performance of the QRDPSO is measured lower than the performance of the other benchmark algorithms. Although the difference is relatively too small, the proposed QRDPSO algorithm proves to be more efficient and stable than the other benchmark algorithms. As for the Griewank function in Table 3.2, the proposed QRDPSO algorithm is observed as the only one of all the benchmark algorithms to reach the global best solution, which is zero in this case, for all the selected scenarios in a very stable behaviour. From these experimental results, it can be noticed that the performance of the QRDPSO is much better than the performance of the SPSO, QDPSO, RQPSO1 and RQPSO2 in both unimodal and multimodal test functions.

Analysis of quantum behaved PSO series performance such as the SPSO, QDPSO, RQPSO1 and RQPSO2 against the proposed algorithm QRDPSO using seven benchmark functions, the Sphere, De Jong's, Rosenbrock, Rosenbrock valley, Shaffers, Rastrigin and Griewank is presented. The experiment result shows the QRDPSO has the best convergence of individual particles of the whole swarm population. The global search of the QRDPSO also reaches the global best solution, which is zero, for all the selected scenarios, which indicates very stable behaviour. The result shows that the QRDPSO is much better than the SPSO, QDPSO, RQPSO1 and RQPSO2 in both unimodal and multimodal test functions. To minimize distance value between robots and victim to approaching zero.

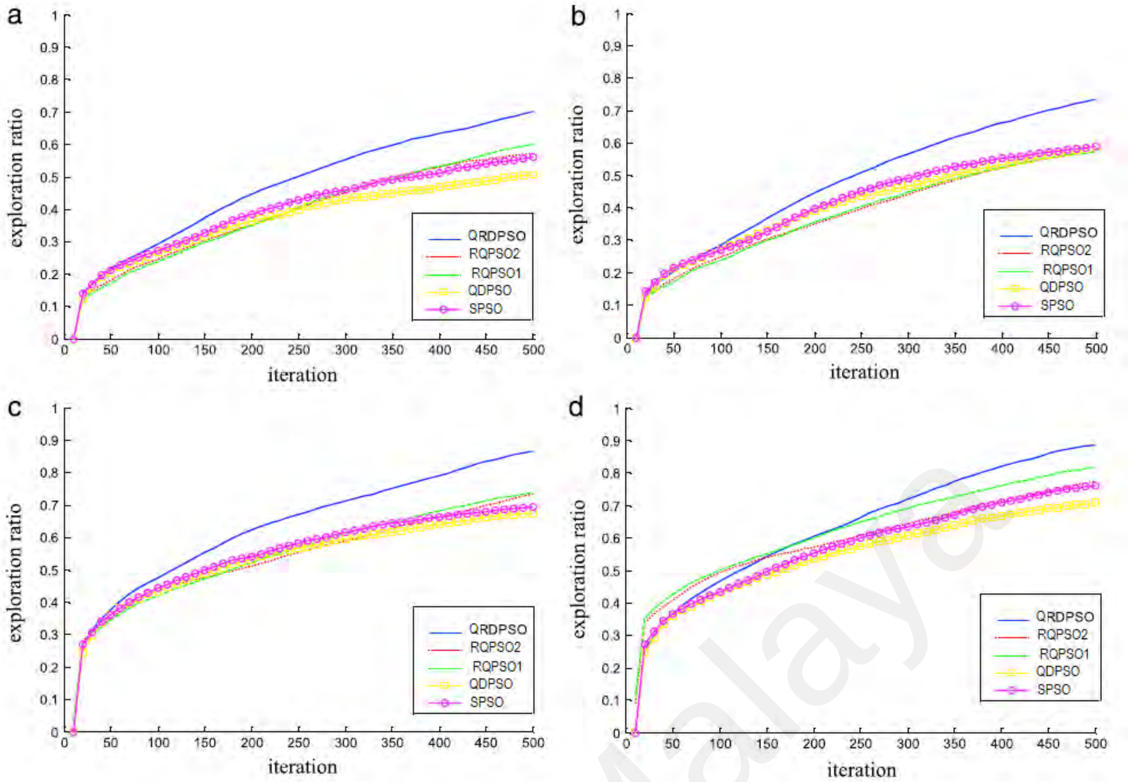


Figure 3.1: The exploration ratio over the 100 iteration for each method. (a) (N_T , d_{max}) = (10, 30 m); (b) (N_T , d_{max}) = (10, 100 m); (c) (N_T , d_{max}) = (20, 30 m).(d) (NT , dmax) = (20, 100 m);

As Figure 3.1 depicts, the median of the best solution over the 100 trials was taken as the final output for each (N_T,x) combination. As it is possible to observe, the QRDPSO outperforms the other methods for all (N_T,x) configurations tested. Nevertheless, the performance of RQPSO1 and the RQPSO2 decreases as robots population increases. For instance, for the configuration of (N_T,d_{max})=(20,100), i.e., Figure 3.1d, the RQPSO1 presents better performance than the RDPSO during the first iterations while the RQPSO2 closely follows the same performance as the QRDPSO. Table 3.8 shows the results when the QRDPSO is compared with the benchmark function that minimizes the robot and victim's distance.

Table 3.8: The QRDPSO against benchmark functions

M	D	Gmax	QRDPSO
20	10	1000	0.0000
	20	1500	0.0000
	30	2000	0.0000
40	10	1000	0.0000
	20	1500	0.0000
	30	2000	0.0000
80	10	1000	0.0000
	20	1500	0.0000
	30	2000	0.0000

As shown in Table 3.8, the test conditions for the experiment include swarm population of varying size of 20, 40 and 80, while the maximum iteration is 1000, 1500 and 2000, corresponding to particle dimension which is set to 10, 20, and 30, all of them are minimized to zero. Consideration of an excellent control architecture is proposed in the following subsection to apply the QRDPSO in practice.

3.2.4 The QRDPSO control architecture design

Control architecture usually defines the performance of an algorithm when being tested practically. In robotics application, in particular, works involving navigation and exploration, *how* the robot moves from one point to another can influence how the control architecture is designed in the first place. Studying a robot kinematic and dynamic features is important for any robot control design. Often, mobile robots are classified into either holonomic or non-holonomic. An omnidirectional drive system is an example of a holonomic robot that can drive in any given direction directed. Driving allows continuous translation and rotation when tracking the robot position from one point to another.

In the case of a non-holonomic robot, such as one with a differential drive system, the robot must often turn to face a target (i.e. rotation) first before driving forward (i.e. translation) to reach a searching target. A Low-Level Control (LLC) is selected to design the control architecture for the proposed QRDPSO to conceptualize the hardware control at a general level. In this work, it is assumed that all robots are non-holonomic for the LLC control architecture design. Each turn and move by the robots can be tracked.

Figure 3.2 shows the proposed control architecture design for the QRDPSO. In the architecture, the low-level control (LLC) receives the desired position $x_n^d[t+1]$ and computes the kinematic model. The rotation turn represents the output of the LLC $\theta_n[t+1]$ and the move forward distance $h_n[t+1]$. These outputs are useful so the robot can turn to face the target (robot-target spatial alignment) and move the distance calculated towards it. Whenever a new position is calculated, the current robot position $x_n[t+1]$ is updated. At each iteration, information on the updated position as well as the corresponding value from the objective function $f(x_n[t+1])$ needs to be shared between connected robots in the swarm so cooperation can emerge.

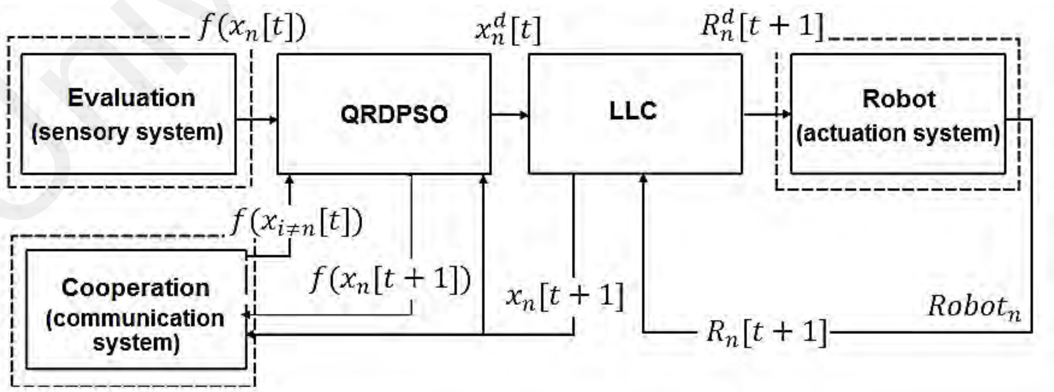


Figure 3.2: The proposed QRDPSO control architecture

Where $v_n^1[t+1]$ and $v_n^2[t+1]$ are elements which describe the vector $v_n[t+1]$ in equation (3.1). In equation (3.13), the function atan[2] returns the angle whose tangent is a given number and determines in which quadrant that angle $[t+1]$ is in the trigonometry. It can be observed that $[t]$ returns a value relative to equation (3.14) which at the beginning of the exploration initialized as zero (i.e. when the robot has yet to move). Finally, rotation $[t+1]$ and distance $h[t+1]$ return the values of the inverse kinematics (rotation and translation), so the robot can fix its orientation and cover the forward distance towards next ideal position.

In Figure 3.2, updates from the current robot position and objective function ($[t+1]$) must be communicated throughout the entire swarm (see Figure 3.2) through the multi-hop routing. Each robot pairs up with another based on the minimum hop distance denoted by the adjacency matrix. Repeating this activity at each iteration allows the robots to relay to other robots and cooperate towards swarm convergence. The rotation $\tau_n^{d1}[t+1]$ and the forward movement $\tau_n^{d2}[t+1]$ of the differential-drive robot are defined by:

$$\tau_n^{d1}[t+1] = \tau_{rev}.\frac{\theta n[t+1]}{2\pi}.\frac{Rrobot}{Rwheel} \qquad (3.16)$$

$$\tau_n^{d2}[t+1] = \tau_{rev}.\frac{hn[t+1]}{2\pi}.\frac{1}{Rwheel} \qquad (3.17)$$

Where τ_{rev} is the total number of steps or pulses per revolution. The radius of the robot and the wheels are defined by R_{robot} and R_{wheel}, respectively. A rotational threshold θ_T was introduced to improve the time response of the robot and the smoothness of its movement. Rotations $[t+1]$ inferior to θ_T are then ignored, and only the forward distance $h[t+1]$ is considered. Bearing in mind this assumption, and since a possible loss of steps or pulses may occur while executing the commands, i.e., $\tau_{n1}[t+1] \neq \tau_n^{d1}[t+1]$ or

$\tau_{n2}[t+1] \neq \tau_n^{d2}[t+1]$, a new real position is then recalculated and considered as the current position $x_n[t+1]$ of the robot.

This new position and the corresponding value of the objective function ($[t+1]$) defined in this position (*i.e.*, sensed by the sensory system) needs to be shared between robots (see Figure 3.2) so that cooperation can emerge. To that end, this information is sent directly to the robots in the neighbourhood (one-hop nodes) and relayed to other robots based on a multi-hop ad hoc networking paradigm.

3.2.5 Parameterization and Adaptability Behaviour

When facing dynamic complex problems, some draws arises when using parameterized algorithms. There is a change that is exhibited when problems related to sub-optimal issues arise. The QRDPSO and PSO are considered the main variants in adjusting and setting the parameters under challenging sceneries. They also ensure that the search capability is maintained and improved for constrained problems or higher-dimensional problems. Persistence should be portrayed in search of victims when using SaR applications which must use a chance for rescuing them.

Punish-reward rules have been applied in various sceneries to avoid stagnation in most circumstances. A sub-optimal solution can detect when robots are stuck or a transition of the solution over a certain period. Setting and adjusting PSO parameters are considered one of the solutions that can be used to solve issues in stabilising analysis regarding algorithm. A generalized model was developed by Blum et al. (2012). They analysed individual particle trajectory which was useful has it contained a set of coefficients to control convergence that occurs in a different system.

Absolute stability is the only attribute presented by the author's analysis, hence the aspect of ignorance in optimizing the optimal solution not considered (Liu et al. 2011). The numerical stability analysis model was presented, which focused on the feedback and reflex activities used to control intensification and diversification during a search. By employing the stable and unstable PSO regions, the author stated it is controlling the swarm. In the real world, robots are designed to operate where their obstacles and dynamics need to be accounted for missing communication infrastructure in specific systems. The need to consider the self-spreading of automates mobile nodes becomes difficult as it will increase the problem's complexity.

Basing on the contextual information within the surrounding should be used to change a robot's behaviour. Adopting swarms and robot behaviour requires contextual knowledge while considering agent-based, environmental context, and mission-related context (Turner, 2013). Hence, a robotic system's performance in a search, Li et al. presented a context-based approach used to enhance rescue missions (Li et al., 2012). The difficulty levels attained in the process are used to set metrics levels used as inputs concerning victim detection and mobility. Based on the set of context-based evaluation metrics, the algorithm adapting the robots' behaviour is what was proposed by the authors.

To adjust the RDPSO parameters, the need for metrics to input a fuzzy system becomes critical. The metrics are used to determine communication constraints, improve convergence rates of the system, and susceptibility to obstacles (Cueceiro et al., 2012). The threshold is considered as the area where QRDPSO parameters are used to group robots when looking for an optimal solution when considering or avoiding any obstacle that will come on the way and ensure that connectivity is attained for effective communication.

To calculate the parameters, influence within the QRDPSO algorithm, we used a swarm robot in MATLAB simulator to understand the relation between the coefficient's and robot's convergence behaviour within the QRDPSO. As described in Figure. 3.2, the behaviour of the swarm is susceptible to changes in μ. When μ = 2, (in 248 positions it takes 788 iterations), the swarm path is linear and presents good exploration behaviour. It indicates that the swarm is stable and converges to an optimal solution.

When μ is high, i.e. μ = 2.5, (in 270 positions it takes 788 iterations), the trajectory of the swarm is represented by linear convergence toward the global solution. Thus, having a high exploration but instability extends the time required to find the optimal solution. Moreover, when μ is small, i.e. μ =1.5, (in 250 positions it takes 550 iterations) the swarm moves slowly, finds it difficult to converge on a solution and gets stuck in a sub-optimal solution. In Figure 3.3, the best value is 2 to reach the optimal solution.

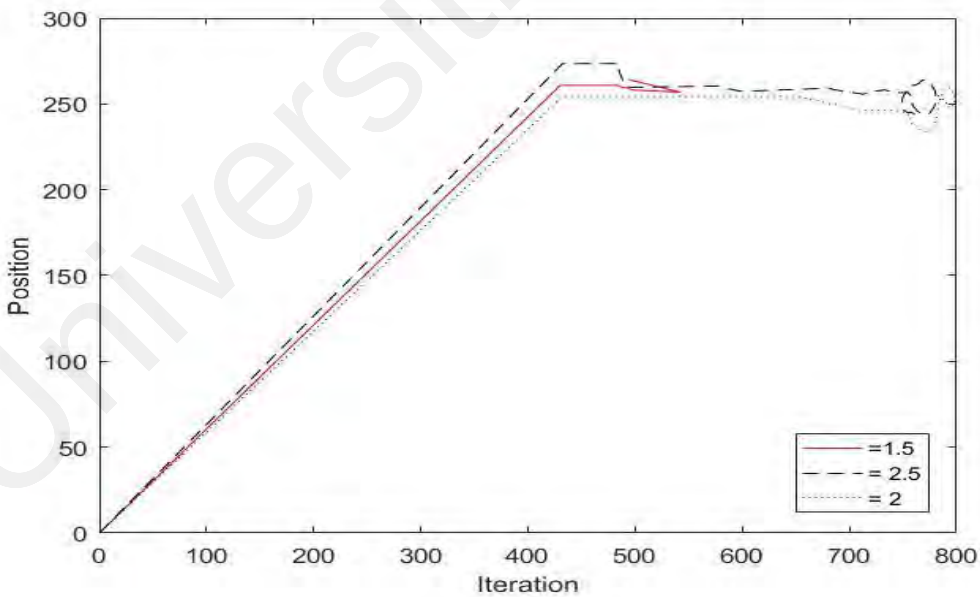
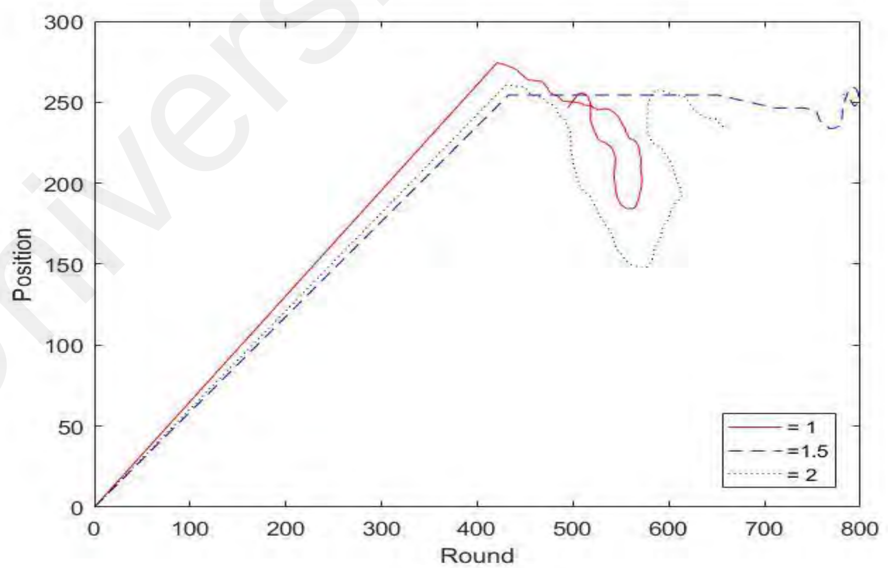


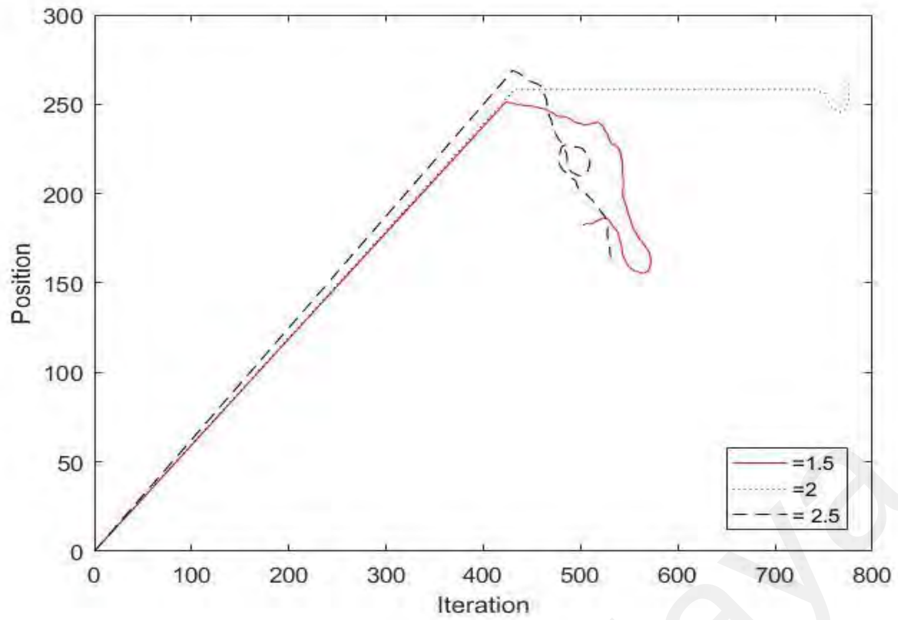
Figure 3.3: Analysis of trajectory swarm robots in MATLAB to evaluate μ

In Figure 3.4(a), the cognitive coefficient α_1 = 1.5 is the best value (250 robot positions stop in 780 iterations to reach the optimal solution). It gives a superior performance of speed of convergence and sub-optimal solution avoidance. When increasing the (α1), i.e. α_1=2, (in 260 robot position) the robots take times to find the victim (optimal solution) because of unstable robot movement. It completes the mission but not directly. When decreasing the (α1), i.e. α1=1, (280 robot positions stop at 850 iterations) the robots might be unable to complete their mission as it takes time to find a victim.

Figure 3.3(b) shows that when (α_2 =2.5), it is the worst-performing robot with an inadequate trajectory. It takes time to reach the victims (optimal solution) or not reach it because it is trapped in a sub-optimal solution. When (α_2 =1.5), the robot is moving chaotically and cannot avoid obstacles. However, when (α_2 =2), the robot can surround the obstacles, thus reaching the optimal solution directly. So, it is the best value.



(a)



(b)

Figure 3.4: Evaluating the robot's trajectory to reach optimal solution a) to convergence α_1, and b) to avoid obstacles α_2

In between the swarm, the robots should distribute around the area to augment the algorithm's convergence rate. Hence, we need to find a suitable coefficient (α_3) to move and maintain the MANET connectivity between robots. Regarding the MANET connectivity wherein each robot is a network node, to overcome the lack of interaction among them, the required position, i.e., $xn\ [t + 1]$, must be controlled influences the adjacency matrix A. The adjacency matrix depends on the maximum interaction range $dmax$ or minimum signal quality represented by the link matrix $L = \{lij\}$ for an N-node network. Each entry represents the link between robot i and j. So, when ($\alpha_3 = 2.5$), the robots take time reaching the optimal solution. It is close to the solution but cannot reach it. Figure 3.5 shows the robot's trajectory in finding optimal solution.

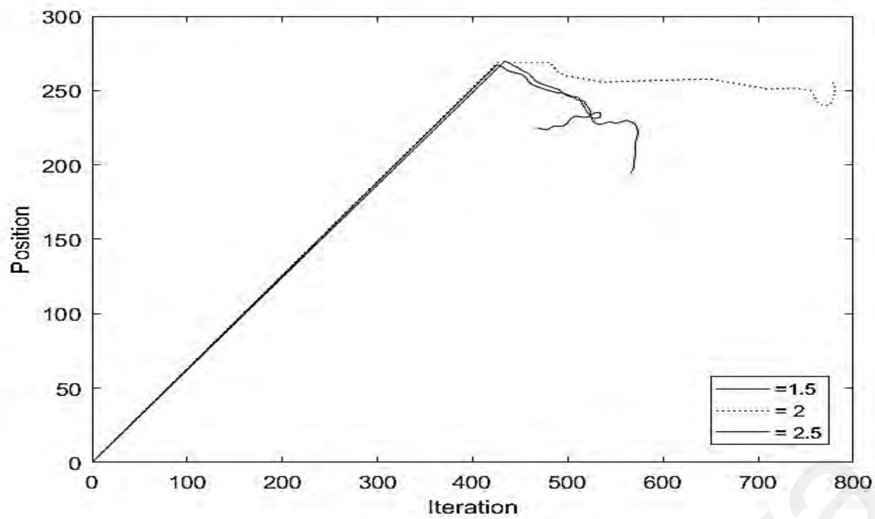


Figure 3.5: Evaluating the robot's trajectory to find a suitable coefficient (α_3) when reaching the optimal solution

Moreover, when (α_3 = 1.5), robots have an unstable trajectory and are stuck in a sub-optimal solution. When (α_3 = 2) (in 260 robot positions and stop in 780 iterations to reach the optimal solution), the robots move directly to the optimal solution with connectivity between them. Thus, it is the best value. However, given the above convergence analysis, the QRDPSO can be extended to control the primary mission's swarm susceptibility, obstacle avoidance, and communication constraint. In this line of thought, it is based on the fuzzy approach (Couceiro et al. 2012), introduced in this section, that I will evaluate the performance and adaptively adjust the parameters of the QRDPSO. Algorithm 3 describes the QRDPSO base mechanisms.

Algorithm 3: The QRDPSO Algorithm

Wait for information about initial pose ⟨[0],φ_n[0]⟩ and $swarmID$

Loop:

If $swarmID \neq 0$ // it is not an excluded robot

Evaluate its individual solution $h[t]$

If $h(x_n[t])>h_{best}$ // robot has improved

$h_{best}=h(x_n[t])$

$\chi_1[t]=x_n[t]$

Exchange information with teammates about the individual solution $h[t]$ and current position $x_n[t]$.

Build a vector $[t]$ containing the individual solution of all robots within $swarmID$

If max$[t]>H_{best}$ // subgroup has improved

$H_{best}=$max$[t]$

$\chi_2[t]=x_n[t]$

If $SC_s>0$

$SC_s=SC_s-1$ // stagnancy counter

If $SC_s=0$ // the subgroup can be rewarded

If $N_S<N_{max}$ and ( )1/N_s^{kill}+1>$rand$( ) // small probability of calling a new robot

Broadcast the need for a new robot to any available excluded robot

If $N_s^{kill}>0$

$N_s^{kill}=N_s^{kill}-1$ // excluded robots counter

If ( )*$N_s/N_{max}>ra$( ) // small probability of creating a new subgroup

Broadcast the possibility of creating a new subgroup to any available excluded robot

If $N_s^{kill}>0$

$N_s^{kill}=N_s^{kill}-1$ // excluded robots counter

Else // subgroup has not improved

$SC_s=SC_s+1$ // stagnancy counter

If $SC_s=SC_{max}$ // punish subgroup

If $N_S>N_{min}$ // it is possible to exclude the worst-performing robot

$N_s^{kill}=N_s^{kill}+1$ // excluded robots counter

$SC_s=SC[1-1/N_s^{kill}+1]$ // reset search counter

If $h_{best}=\min[t]$ // this is the worst-performing robot

$swarmID=0$ // exclude this robot

Else // delete the entire subgroup

$swarmID=0$ // exclude this robot

If $([t])\geq g_{best})$ // maximize distance to obstacle

$g_{best}=g(x_n[t])$

$\chi_3[t]=x_n[t]$

$is_{i,n}(t) = g(x_i[t]) = \frac{\sigma(g_{i,1}(t))}{\sigma(x_{i,1}(t))}, \frac{\sigma(g_{i,2}(t))}{\sigma(x_{i,2}(t))}, \ldots, \frac{\sigma(g_{i,n}(t))}{\sigma(x_{i,n}(t))}$ // represented as the relation between the analogy output voltage of distance sensors and the distance to the detected object.

$[L_n, index_n]=sort\_asc(L_n,1:N_S)$ // sort the elements of line n from link matrix L in ascending order

For $i=1:N_S$

If $ind(i)$ has not yet chosen it as its nearest neighbour

$\chi_4[t]=x_i[t]+d_{max}\ x_i[t]-x_n[t]/\|x_i[t]-x_n[t]\|$ // the position of the nearest neighbour increased by d_{max} toward $x_n[t]$

Communicate to robot i that it was chosen by robot n

$im_{i,n} = m(x_i[t]) = \frac{\sigma(m_{i,1}(t))}{\sigma(x_{i,1}(t))}, \frac{\sigma(m_{i,2}(t))}{\sigma(x_{i,2}(t))}, \ldots, \frac{\sigma(m_{i,n}(t))}{\sigma(x_{i,n}(t))}$ // standard deviation value of the connectivity function and the standard deviation value of the current position of the current particle

break from For

$$X_{i,n+1}^{j}(t+1) = P_{i,n}^{j} \pm \left(\alpha_1 \left|X_{i,n}^{j} - C_n^{j}\right| + \alpha_2 \left|X_{i,n}^{j} - im_n^{j}\right| + \alpha_3 \left|X_{i,n}^{j} - is_n^{j}\right|\right) \ln\left(\frac{1}{u_{i,n+1}^{j}}\right)$$

Else // it is an excluded robot

Wandering algorithm

Evaluate its individual solution $h[t]$

If $h(x_n[t]) > h_{best}$ // robot has improved

$h_{best} = h(x_n[t])$

Exchange information with teammates about the individual solution $h_n[t]$ and current position $x_n[t]$

Build a vector $H[t]$ containing the individual solution of all N_X robots within the excluded subgroup ($swarmID$=0)

If max$H[t] > H_{best}$

H_{best}=max$H[t]$

If h_{best}=maxN_I $H[t]$ // this is one of the best N_I performing robots of the excluded subgroup

If $N_X \geq N_I$ and $rand(\ ) N_X/N_T > rand(\ )$ // small probability of creating a new subgroup

$swarmID = swarmID\_new$ // include this robot in the new active subgroup

Broadcast the need of $N_I - 1$ robots to any available excluded robot

Else

If receives information about the need for a new robot

$swarmID$=$swarmID\_received$ // include this robot in the active subgroup

N_S=N_S+1

Exchange information with teammates about N_S

If receives information about the need of creating a new subgroup

$swarmID$=$swarmID\_new$ // include this robot in a new active subgroup

N_S=N_I // reset number of robots in the subgroup

N_s^{kill}=0 // reset number of excluded robots

SCs=0 // reset search counter

until stopping criteria (convergence/time)

3.3 Communication Optimization via MR-LEACH

The MANET is utilized as a pervasive communication schema in the RDPSO. Nevertheless, the MANET is an example of brute communication where all nodes broadcast messages to all other nodes in the swarm where some communication is long hops. Long transmission distance and multiple relays contribute to heavy communication traffic and do not encourage efficiency. The communication schema must follow a mission-related design to enhance the communication of the QRDPSO, i.e., based on the behaviour that one should expect from the swarm robotics. In this work, the MR-LEACH schema is selected, and the implementation has the following steps:

1. Validation of the proposed QRDPSO model with simulation using MATLAB
2. The comparison between the QRDPSO model uses a routing protocol for MANET such as the Ad Hoc on-demand Distance Vector (AODV), against the MR-LEACH.
3. Validation of the stability of the QRDPSO with enhanced communication under various conditions.

Briefly, enhancing QRDPSO communication requires a strategy to handle the communication traffic in a more sustainable method. One strategy to avoid brute communication is to look into clustering of the swarm into multiple smaller groups. The next subsection describes this strategy.

3.3.1 Sharing information in the QRDPSO

The QRDPSO ensures connectivity of the network from $m(xi[t])$ a term described in section 3.2.2. Nevertheless, brute communication and long hops create more traffic which can overwhelm the network. Packet data structure shared between robots is defined to send messages (see Figure 3.6). The number of bytes necessary for the main message, *i.e.*, Data byte(s), will depend on the message itself. For instance, if a robot wants to share its position and consider a planar scenario, then two bytes may be enough to represent each axis's coordinates.

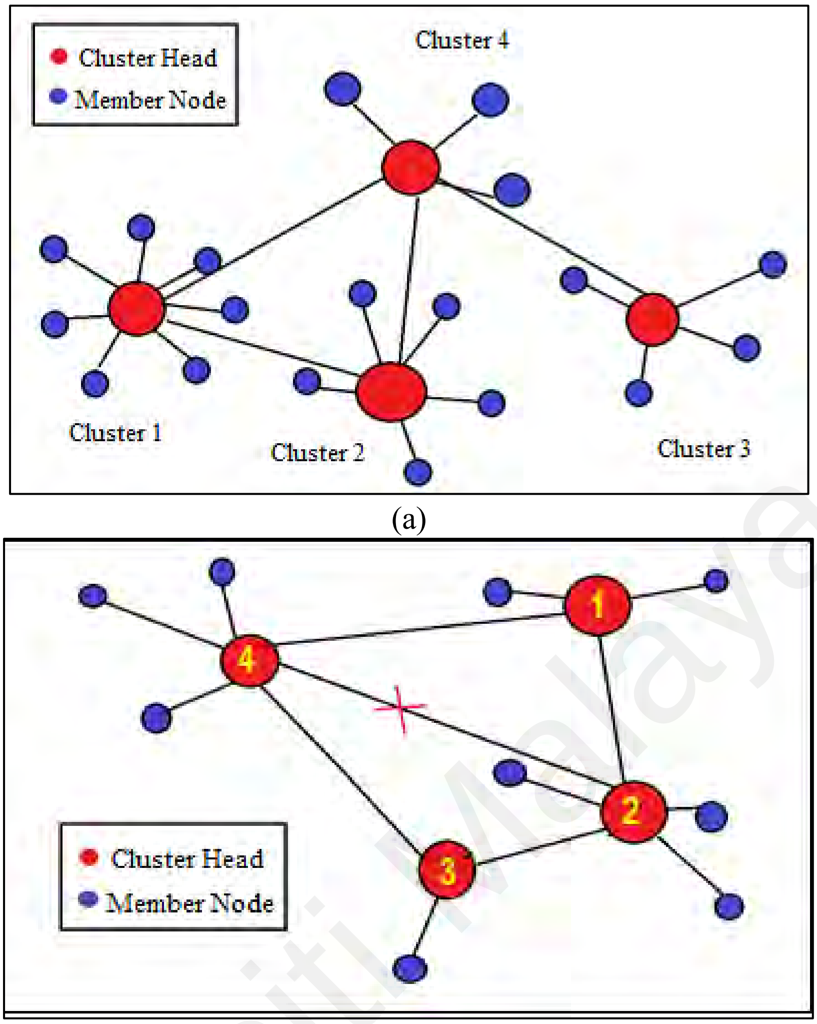
Header bit [0,1]	Data byte(s)
0 Local Broadcast to neighbours	Number of bytes depends on specific data
1 Broadcast to whole subgroup	

Figure 3.6: General communication packet structure for a subgroup of Ns robots

A strategy to cluster the swarm network is introduced to the QRDPSO. The strategy includes labelling each robot with a robot ID and each swarm division with cluster head ID. For the swarm robot to maintain communication, the connectivity between robots (cluster head, non-cluster) using MR-LEACH as a communication protocol, is described as the following steps:

1. Each node randomly decides to become a cluster head (CH), where each node n (robots) chooses a random number between zero and one. If the obtained number is less than a threshold $T(n)$, the node becomes cluster-head,
2. A cluster head broadcasts the advertisement "Hello" message to all the nodes around it,
3. Upon reception of this message each non-cluster node will decide to join a specific CH depending on the smallest distance,
4. CH creates a TDMA (Time Division Multiple Access) based transmission schedule for each node in the cluster. The cluster head allocates the communication time slot for each member node in the cluster based on TDMA cluster heads, and these members can receive transceiver signal only on the given time slot to effective use of power,
5. CH aggregates the data received from nodes inside the cluster and sends it to the CHs to get the best solution between them.

Applying equation (3.4), the link matrix $L = \{l_{i,f}\}$ can be calculated as functions of either distance d_{max} between a cluster head and a non-cluster. Together, they form the adjacency matrix $A = \{a_{if}\}$. Equation (3.5) can be adopted, and the connectivity matrix can be defined similarly to calculate the cluster head hop distance. A connectivity function $m(x_i(t))$ is then defined. A higher $\alpha 2$ will enhance the ability to maintain a network connection. To further understand how the QRDPSO maintains the MR-LEACH connectivity considers the topology in Figure 3.7.



(b)

Figure 3.7: (a) The MR-LEACH connectivity topology, and (b) the clustering hierarchy for MR-LEACH connectivity (bottom)

Figure 3.7(a) shows an illustration of the MR-LEACH schema with multi-hop clustering routing. Smaller swarm divisions are possible as long as a CH is appointed, and all robots are labelled with an ID. However, each pair of nodes must be at least two disjoint routes between the network. The distance means the failure of a single node does not influence the network partition. In this work, a clustering hierarchy fault-tolerant system is proposed for autonomous mobile robots, so at any time CHs can use an exclusive channel to send messages.

As shown in Figure 3.7(b), CH robot 2 is the nearest neighbour to CH robot 1. The nearest neighbour of CH robot 2 is CH robot 3. However, the distance between them is smaller than CH robot 2 and CH robot 4. Finally, the nearest, not previously chosen, a neighbour of CH robot 4 is robot 1 and 3. If a robot fails due to energy depletion, for example, CH 2 fails, robot 1 will be unable to communicate with robot 3, but robot 3 can connect to another channel with robot 4. The bi-connectivity $k = 2$ gets the desired performance in simple exploration like finding a gas leak in a room. A more complex exploration requires a higher $k > 2$ MR-LEACH connectivity. Algorithm 4 is the MR-LEACH algorithm for cluster formation.

<u>Algorithm 4: MR-LEACH Algorithm (Farooq et al. (2010)).</u>

```
     Node = Sensing Node
     S= Set of all Sensing Nodes in the Network
     Neighboring Nodes = Null; // No neighbors discovered
 for ∀ Nodes ∈ S
       Broadcast _ HELLO (nodeID, Energy);  // nodeID = Robot ID
 for ∀ Nodes ∈ S
      begin
         Re cv _ BroadCast _ MSG(nodeID, energy)
         ID= NeighbouringNode.searchNodeID(nodeID)
     if (ID≠ nodeID)
         NeighbouringNode.insert(nodeID, energy)
   end
   for ∀ Nodes ∈ S
       begin
         nodeWithHigestEnergy = neighbouringNodes.getHighestEnergy()
      if (nodeWithHighestEnergy < nodesEnergy)
          BROADCAST _ HEAD _ MSG(nodeID)
      end
```

```
for ∀ Nodes ∈ S
    begin
      Re v _ Head _ MSG(ID)
      Cluster _ Head.insert(ID, ReceivedSignalStrength)
end
 for ∀ Nodes ∈ S
    begin
        Select Cluster Head ; // non-cluster node will decide to join a CH
      depending on the smallest distance .
        Send _ Cluster _ Join _ MSG(ID);
  end
for ∀ ClusterHeads ∈ S
   Re cv _ Join _ MSG(ID)
```

3.3.2 Converging to the Optimal Solution

In section 3.2, $P_{i,n}^{j}$ represents the best positions of the robot. Therefore, robots from the same active sub-group, i.e. not in the socially excluded sub-group, need to share their best cognitive solution $[t]$ and current position $[t]$, to compute the position of the robot that has the best social solution. For instance, if one wishes to find a victim, the best performing robot will have the highest solution. Suppose a robot from the active sub-group was unable to improve. In that case, the information about its position and solution is irrelevant to the group, i.e. the collective behaviour will not change. Therefore, and as a rule of thumb, a robot only needs to share its current solution and position if it can improve its best cognitive solution, *i.e.*, $fn[t+j]>fn[t]$, $j\in\mathbb{N}$. Otherwise, the robots must memorize the best solution of the sub-group and corresponding position. This data needs to be exchanged between all teammates, by broadcasting to the whole sub-group via MR-LEACH, consequently reducing the robots' energy and increasing its lifetime.

Header Bit **1**	**Data Bytes : CH (ID) - Xn (t) - Fn(t)**

Figure 3.8: Communicating the packet structure to a robot to get the best solution

Figure 3.8 represents the packet structure sent from a robot that was able to improve its solution. This communication packet structure allows robots from active sub-groups to cooperatively converge to the solution. The packet is only sent if a robot improves its best cognitive solution. The following sub-section evaluates the MR-LEACH communication using the MR LEACH Simulator.

3.3.3 Experimental Results

Testing the nodes' energy consumption by adaptively increasing the clustering hierarchy and testing nodes' lifetime can be done in three steps. Using the MR-LEACH simulator, the steps are:

1- Facilitate the MR-LEACH to deal with mobile nodes.

- Remove the single hop sink
  - ➢ It needs multi-hop to continue searching for another group (fault tolerance system) in the swarm.
  - ➢ Cooperation between robots and detect the victim in shorter time no need a delay.
  - ➢ Avoid failure when a sink is dead or crash.
- The nodes fixed in the simulator, so it should be change fixed nodes to dynamic.

2- Constitute dynamic neighbouring criteria to tie the mobile nodes with the mobile cluster heads.

- After that, connect each node with neighbour CH, and CH with neighbour CH depend on calculating the distance to each CH and finding the smallest distance.
- In figure 3.9, the experiment shows five dynamic cluster heads, with each cluster head possibly to connect to 25 dynamic nodes (non-cluster):

3- Test the MR-LEACH control with the new evolutions, see Figure 3.10:

- The first test shows which round that the nodes are dead.
- The second test shows the nodes energy consumption when the number of round increases.

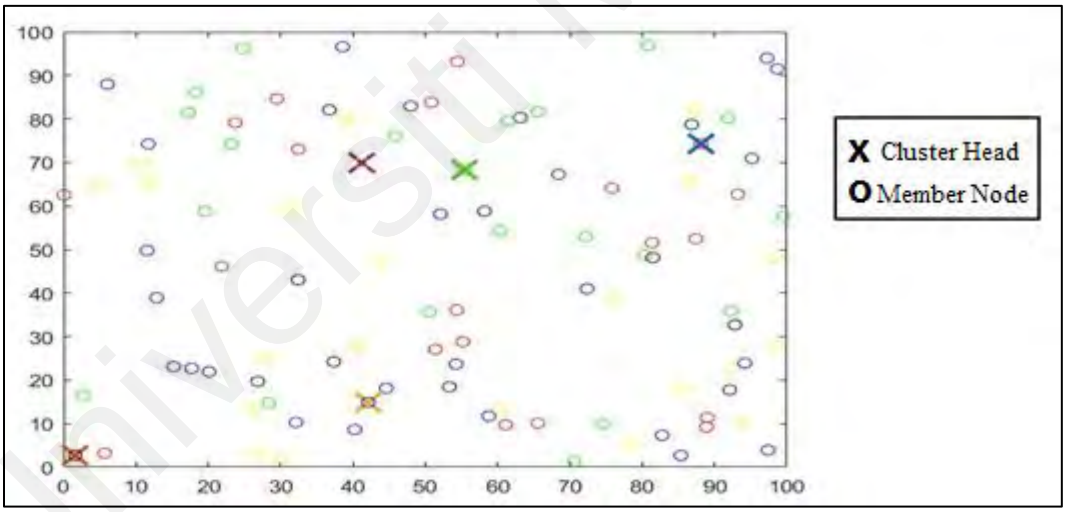


Figure 3.9: MR-LEACH simulator 100 x100 Area

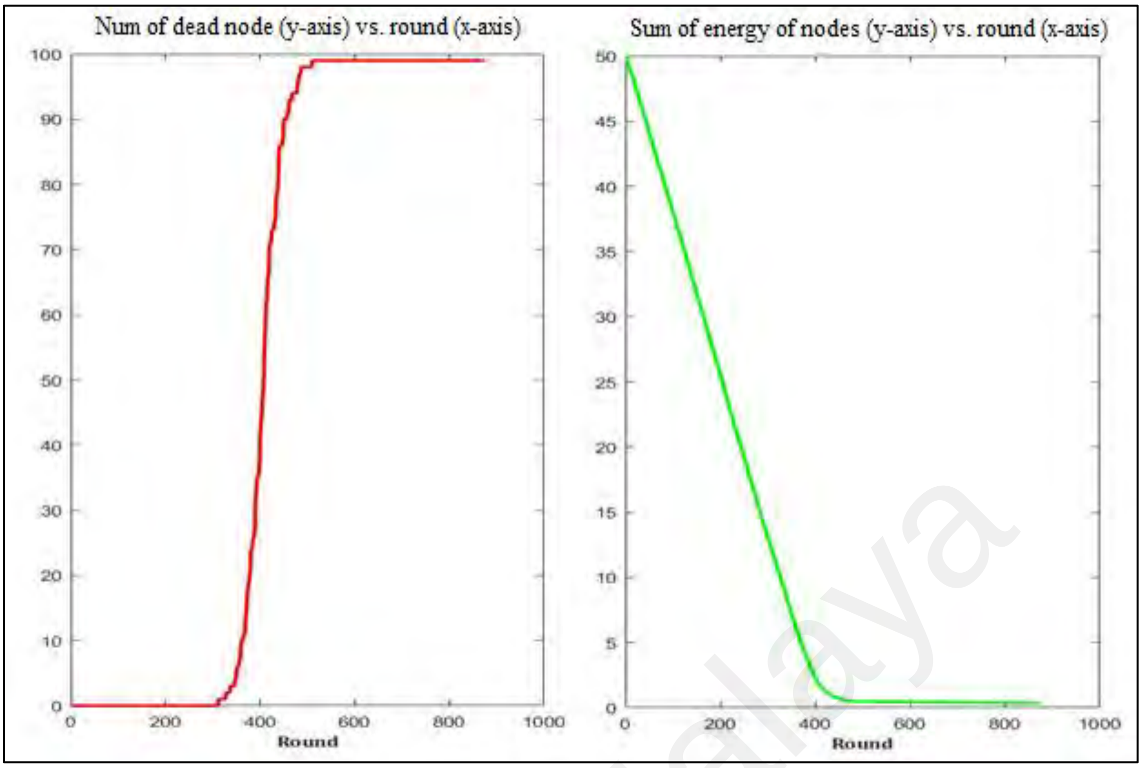


Figure 3.10: MR-LEACH control with new evolutions

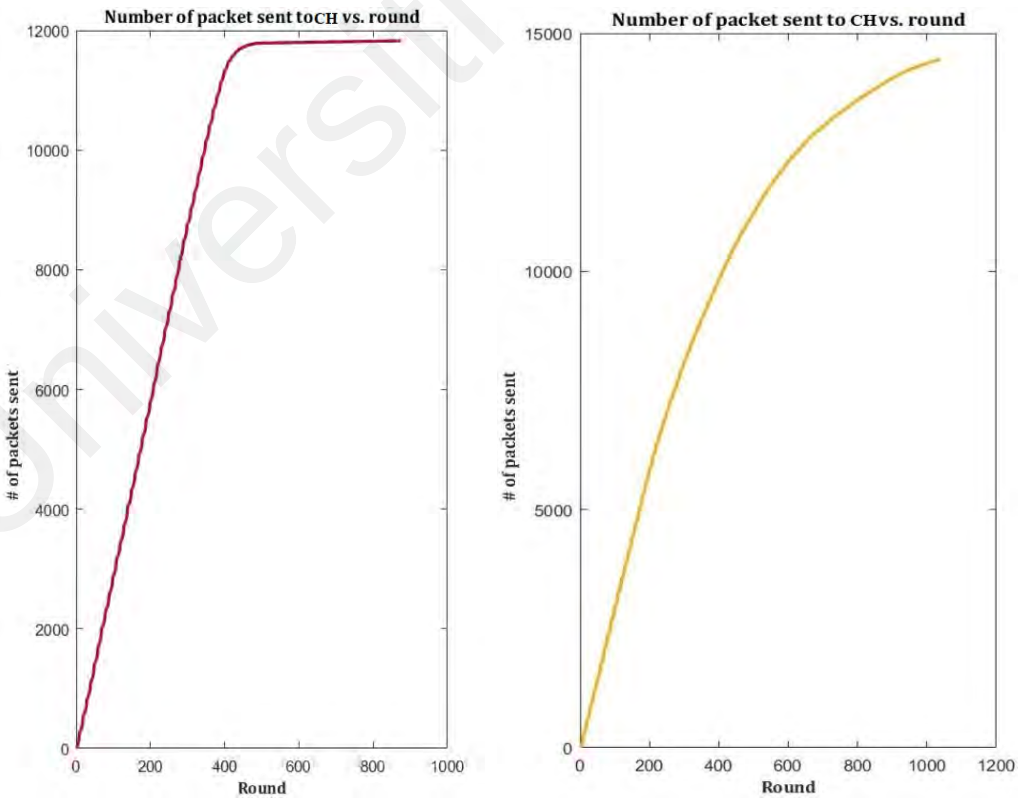


Figure 3.11: The number of messages sent from static CHs (left), and dynamic CHs (right) to respective non-CHs

Figure 3.10 shows the first experiment (left) with three dead nodes reported at around 300. The number of dead nodes keeps increasing until round 450 where all the nodes are reportedly dead. In the second experiment (see right), the nodes energy consumed were decreasing when the number of round increases; at round 0, all the nodes were at 50 indicating full energy. The nodes consumed energy so much that at around 450, all nodes' energy level is depleted (dead). Also, that reduces the communication complexity as information needs to be exchanged between all teammates, i.e., broadcasted to the whole subgroup using multi-hop communication.

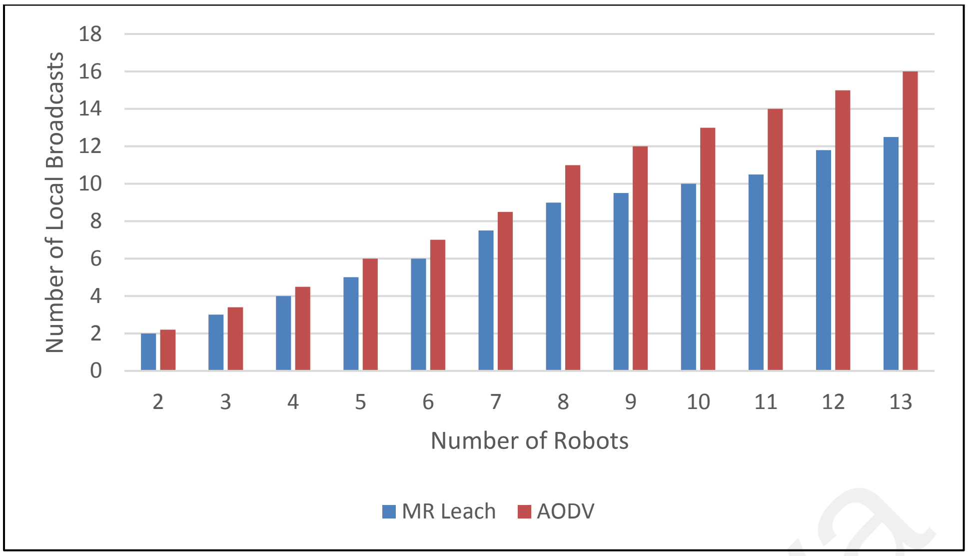
A topology with five dynamic CHs can connect to 25 static non-CHs, and the information shared through the topology is useful to the collective performance. Figure 3.11 (left) shows 3000 packets sent to CH at round 200. The number of packets keeps increasing until round 430 where all the nodes are reportedly dead with 11900 packets. Figure 3.11 (right) shows 3500 packets sent to CH at round 200. The number of packets increases reaching round 1000, which shows messages flooding through the subgroups even though all the nodes are reportedly dead. The continuous messaging shows static non-CHs sends useful and useless information to the CHs. For this reason, dynamic CH routing should control the messaging for the topology so only useful information is shared. As the number of nodes decreases, the probability that a robot has to improve also decreases.

Optimization of the communication procedure between nodes under the MR-LEACH was presented in this chapter. Moreover, the dynamic MR-LEACH was improved, considering nodes motion compare with static nodes. Such improvements were motivated by the need to use large robots without significantly increasing the communication overhead. It is noteworthy that the amount of useful information will vary depending on several conditions (e.g., number of robots, scenario, mission objectives, among others).

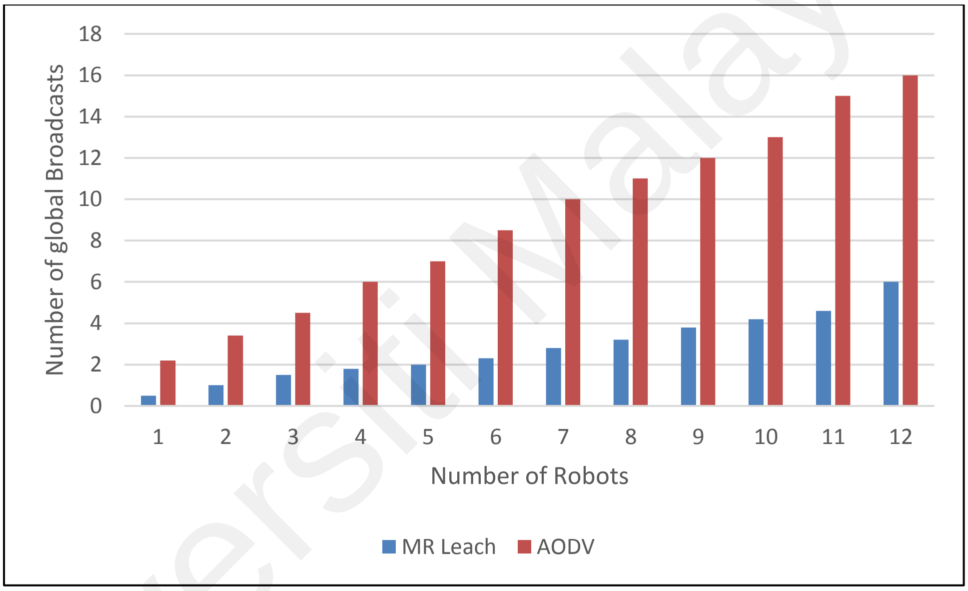
The number of local and global broadcasts is evaluated within each subgroup. For 50 trials, the QRDPSO with MR-LEACH and AODV are tested to understand better how robots develop within the QRDPSO. Figure 3.12 represents, for each subgroup, the total number of local (see Figure 3.12(a)) and global broadcasts (see Figure 3.12(b)). Local broadcasts increase over time, almost proportionally to the number of robots. The pattern is observed in both QRDPSO using AODV and QRDPSO using MR-LEACH.

The critical difference between the two protocols is in the number of global broadcasts between robots belonging to various social statuses. Only the improvement of socially active subgroups depends on such communication. Therefore, as the total number of socially active robots decreases, the number of socially excluded robots increases, and the possibility of success also decreases (i.e. enhancing the current solution). Consequently, this decreases the necessary number of global broadcasts from excluded subgroups.

In general, as one observes, socially active robots using AODV usually have a higher amount of messages flooding through the whole subgroup than MR-LEACH. The CH collected the data obtained from the same cluster nodes and sends it to the CHs. This is interesting because global broadcast is connected to enhancing subgroups that need the population's global consent. As a result, such global broadcasts diminish over time. It appears that this kind of global message is. This kind of global message seems to be significantly less recurrent in socially excluded subgroups. Notice that this dramatically decreases communication difficulty as this data needs to be shared between all teammates, i.e. broadcast using multi-hop communication to the entire subgroup. Consider a setup of 4, 8 and 12 robots with one optimal and one sub-optimal solution in a small scenario.



(a)



(b)

Figure 3.12: The normalized average number of (a) local, and (b) global broadcasts.

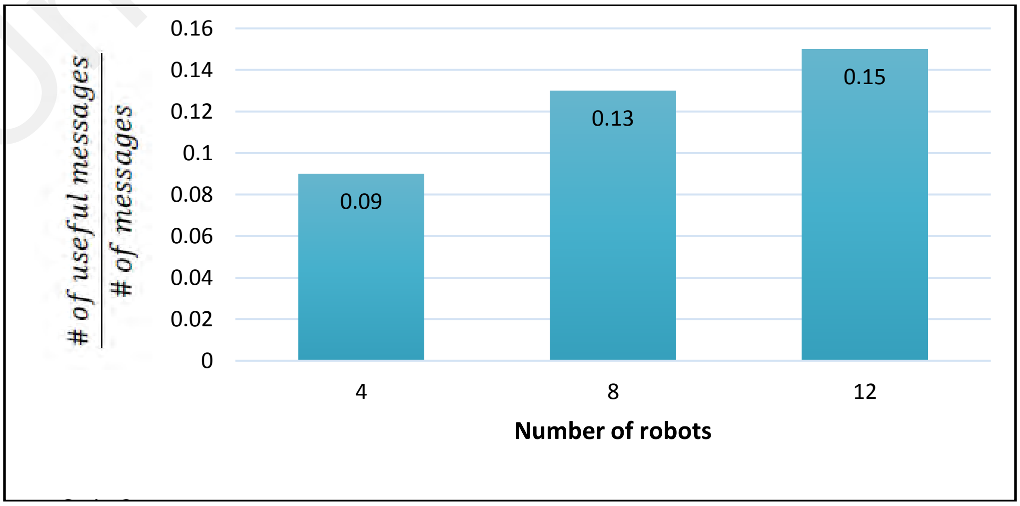


Figure 3.13: The ratio between the number of useful messages and the total number of messages received from the experimental evaluation

As Figure 3.13 depicts, 12 robots represent the most crucial condition tested to the chances that the subgroup has to improve. Even with under 80 trials, it was possible to observe that a robot could only improve in approximately 15% of the iterations. Therefore, only about 15 % of the shared information is useful for collective success. If the number of robots decreases, the possibility of a robot improving also decreases slightly, thereby reducing the amount of useful information slightly.

3.4 Chapter Summary

In this chapter, I presented the methodology to achieve three novel contributions from this work. The following chapter describes the setup and experiment done considering the scope of the search and rescue task and the control parameters proposed to compare the performance of QRDPSO against its predecessor, the RDPSO. Partial to Chapter 4 is the performance of the QRDPSO swarm for exploration compared to its predecessor, the RDPSO. Chapter 4 also discusses the performance of the QRDPSO internal swarm communication. Two protocols are examined; the AODV and MR-LEACH through a series of experiments that determine their interrupts and power consumptions.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Overview

This work focuses on the utility of swarm robotic strategies into real-world operations, such as search and rescue (*SaR*). This chapter presents the results and analysis of the QRDPSO performance for swarm robotic exploration and communication. The chapter begins by introducing the simulation setup followed by evaluating the results of QRDPSO against its predecessor, the RDPSO. The chapter continues with examining the performance of the AODV and MR-LEACH communication protocols to enhance the QRDPSO internal swarm communication.

4.2 Environment setup

For the experiment, the definition of a swarm of robots if given in a MATLAB simulator running on a high-performance workstation Lenovo W530 with Intel iCore7, 2.67GHz processor and 16GB of RAM. The experiment aims to measure the performance of the proposed QRDPSO compared to its predecessor, the RDPSO in terms of a swarm's cooperation in searching optimal solution while performing obstacle avoidance and maintaining robot connectivity. The analysis revolves around the observation of the convergence time and number of robots lost for a search and rescue task.

The following scopes are defined for the experimental setup:

- The obstacles' locations are unknown for the robots and are randomly spread in the environment.
- The shape and the occupied area of each obstacle are random and vary between the obstacles.
- There is only one target in the environment.
- The location of the target is unknown to robots.

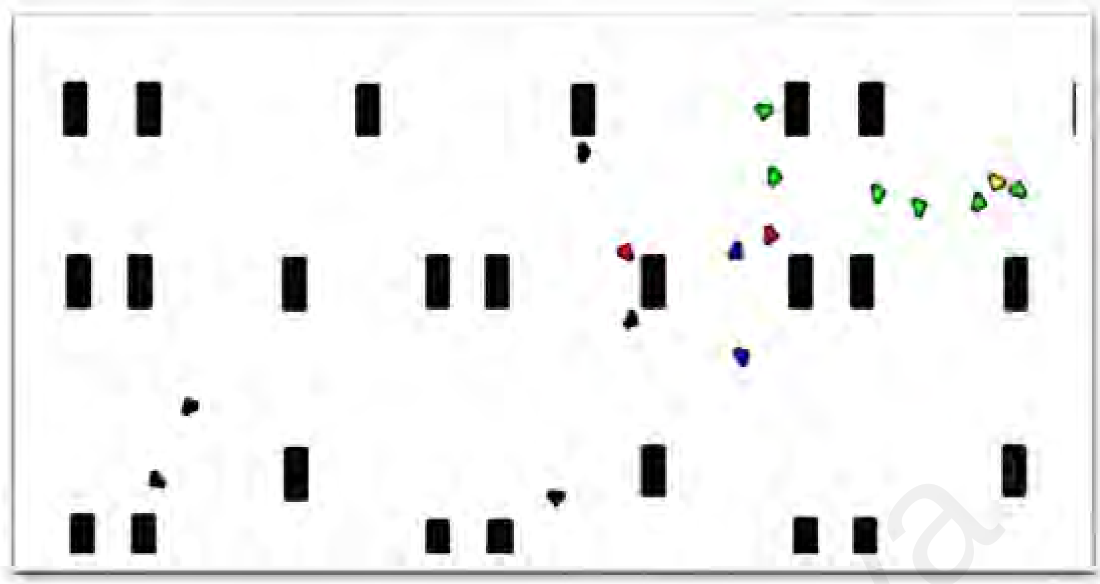


Figure 4.1: A 300x300m environment used in the experiment with random obstacles and a single target (victim)

Figure 4.1 shows the environmental setup used for the experiment. In Figure 4.1, the rectangular blocks represent random obstacles generated. The triangular markers represent robots. The triangular-shaped in yellow represents the victim at a random location unknown to the robots. The green triangles denote robots which successfully located the victim and are proceeding towards it. In blue and red, the robots face some trouble navigating around obstacles with the potential to get stuck in local optima.

In this example, several black triangles are depicted as far away from other robots (on their own at random positions, respectively). These robots have lost communication with the swarm and are moving randomly in the hope to regain the communication range with the swarm. These robots may get back on track towards the victim if they can receive signals from other robots.

4.3 The QRDPSO exploration and communication performance

The proposed QRDPSO algorithm is tested from swarm robot exploration and communication carefully addressing all research questions as described in chapter 1. Table 4.1 revisits the research mapping and highlight the completion of all research objectives:

Table 4.1: The research mapping revisited

Research Questions	Research Objectives	Methodology	Outcome / Contribution
1. How can one improve the maturity and rate of convergence for RDPSO during swarm-robot exploration?	To formulate a searching strategy to reach global best in shorter time in existing RDPSO algorithm	Application of quantum computing theory (QPSO) to extend the RDPSO algorithm	A novel QRDPSO algorithm that improves convergence maturity and rate during swarm-robot exploration over RDPSO algorithm
2. How can one enhance the swarm communication for robot energy conservation and prolonged lifetime during exploration?	To reduce the energy consumption of sensor (robot) nodes by adaptively increasing the clustering hierarchy	Application of MR-LEACH schema that partition the (robot) network into different layers of clusters	A coordinated swarm movement strategy which conserves the robot's energy and extends the robot's lifetime during exploration
3. How does the MR-LEACH schema perform in avoiding local optima and finding global best in comparison to benchmark protocol such as MANET?	To test the performance of QRDPSO swarm with MR-LEACH schema in avoiding local optima and finding global best	Application of search and rescue problems in testing the performance of MR-LEACH in comparison to benchmark protocol such as MANET	The MR-LEACH schema performs better time over benchmark protocol such as MANET in avoiding local optima and finding global best (victims)

4.3.1 The QRDPSO swarm exploration performance

For robot exploration, the convergence time of the swarm remains the most significant investigation. This subsection describes the performance of QRDPSO over RDPSO in a search and rescue mission in terms of two critical factors. First, the time needed by each algorithm in finding the victims, and second, the number of robots lost. Figure 4.2 shows that when increasing the number of 5, 10, 15 and 20 robots decreases the time needed to find the optimal solution over maintaining the Ad-Hoc On-Demand Distance Vector Routing (AODV) connectivity and obstacles avoidance for both QRDPSO and RDPSO.

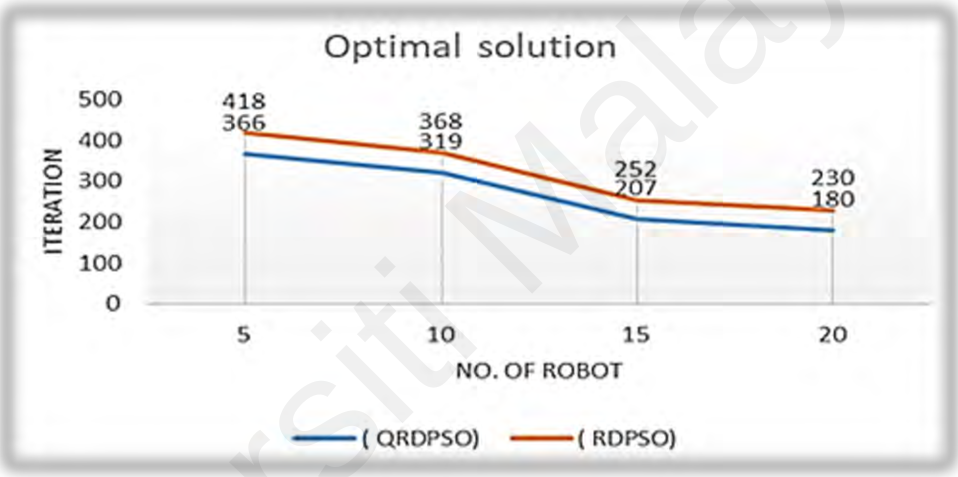


Figure 4.2: Comparison of the time required by QRDPSO over RDPSO in finding the optimal solution

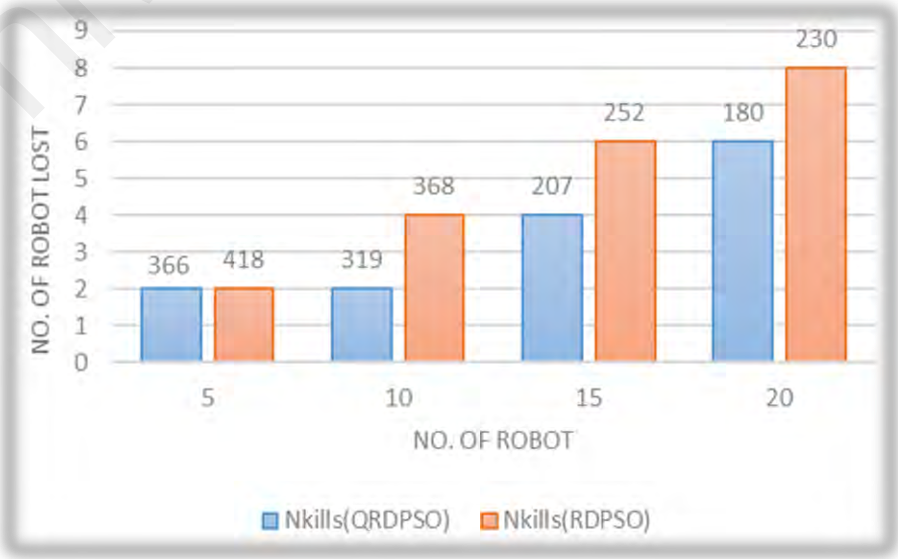


Figure 4.3: The number of robots lost after completing the mission

When 20 robots are deployed to search the victim (optimal solution) in Figure 4.2, the QRDPSO requires 180 iterations, and the RDPSO needs 230 iterations to reach a victim. These results show that robots with lower energy consumption can rescue the victim faster. Another significant attribute to robot exploration is to avoid robot loss.

Figure 4.3 shows the number of robot loss after completing a search and rescue mission. When the number of robots is increased, the number of robot loss increases, which decreases the overall time needed to find the optimal solution. These robots are out of communication range, so there are no messages between robots, leading to no objective function improvement. However, it is observed that these lost robots may move on to other regions and resume their task towards an optimal solution.

The ratio between the area covered by the robots and the total area of the scenario compares both RDPSO and QRDPSO. Figure 4.4 depicts the ratio of the covered area for each different team size. The charts' vertical lines represent the range retrieved from the 20 trials of each different team size. The chart shows that QRDPSO provides broader coverage immediately after the initial deployment. Furthermore, the differences in the covered area of both algorithms are more apparent with larger populations because of the high number of intersection in the robots' sensed areas, when these are deployed using a random deployment.

Figure 4.5 show the outcome when running the QRDPSO to investigate robot loss. The simulation runtime takes about 10mins to capture the swarm progression from initialization until convergence. The black triangle represents a sample of robot loss in the experiment. Figure 4.6 repeats a similar experiment with the QRDPSO predecessor, the RDPSO, for comparison.

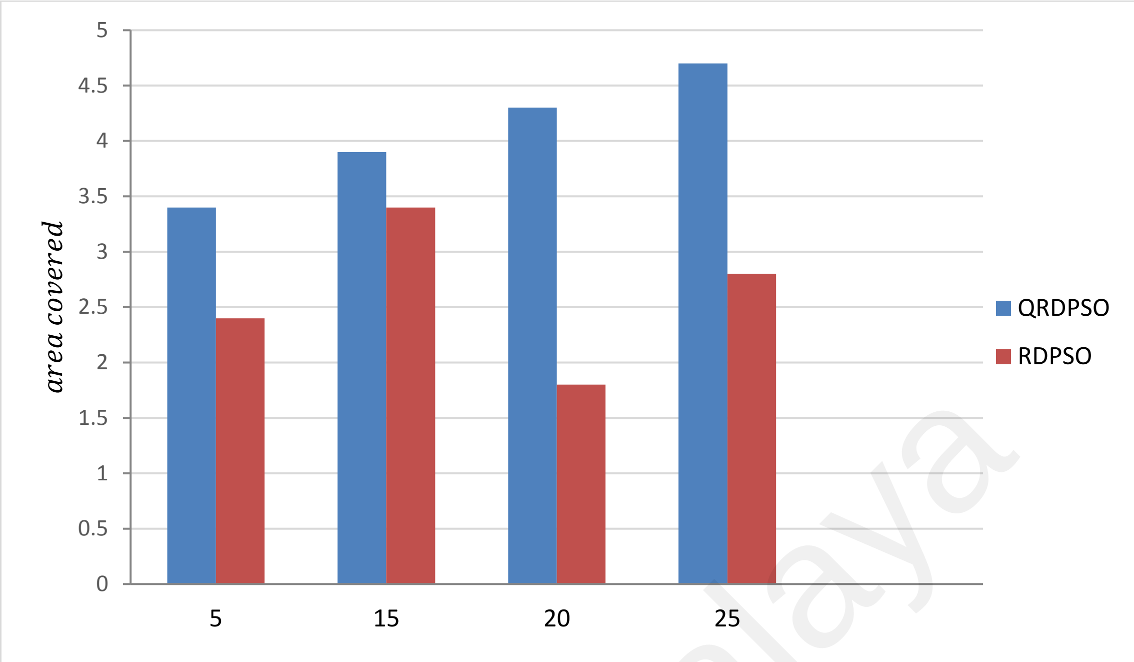
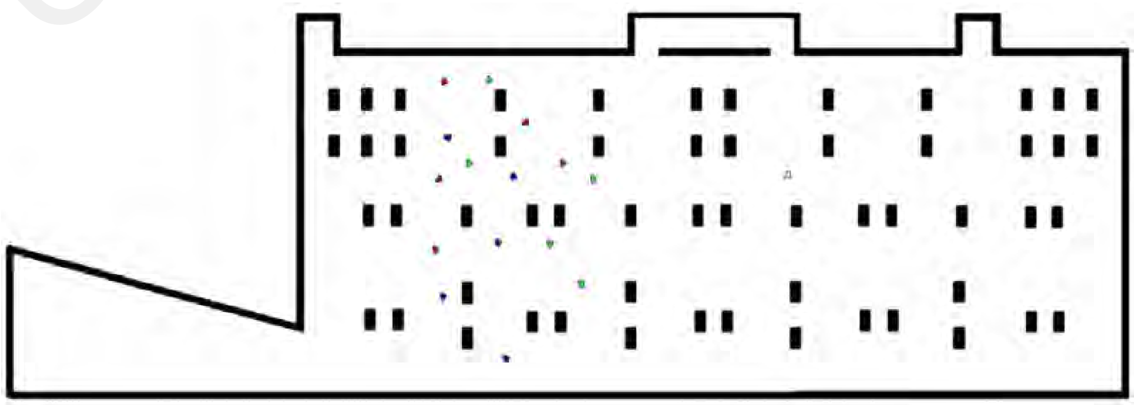
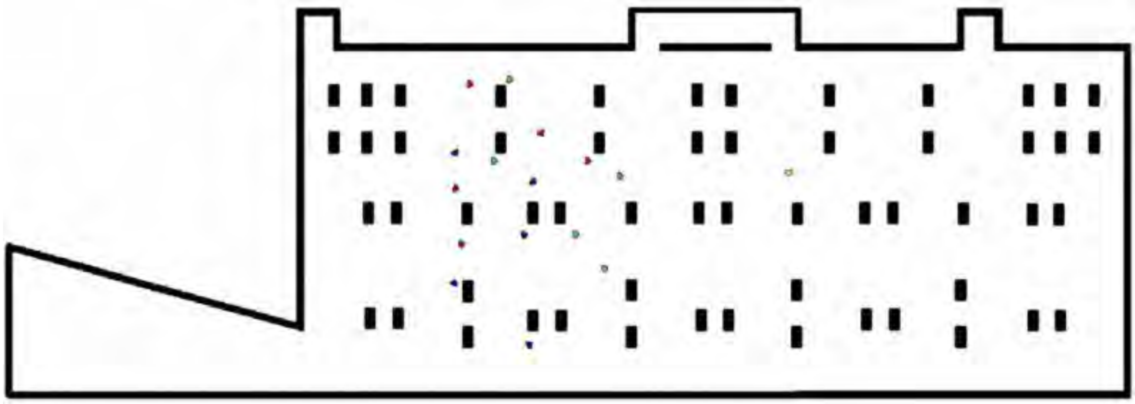


Figure 4.4: Ratio of covered area for a population of 5, 15, 20 and 25 robots grouped into three subgroups

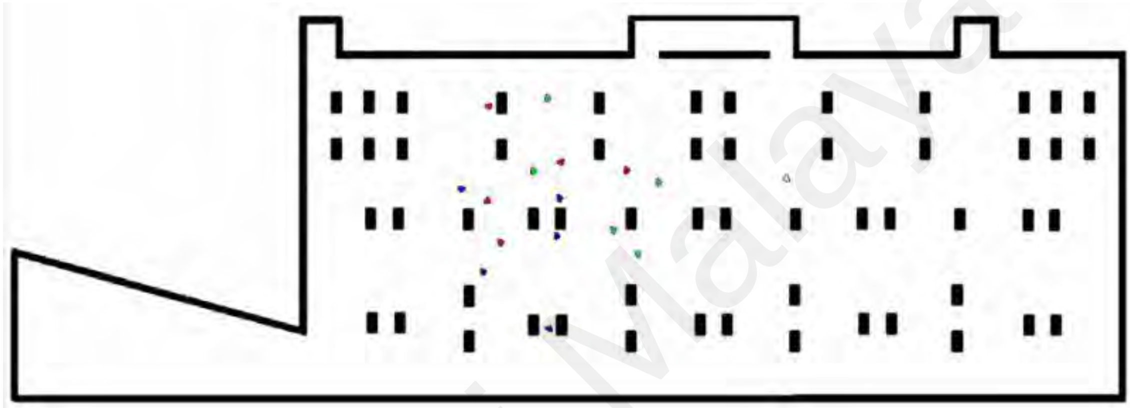
(a) The QRDPSO at t=0 (initialization)



(b) The QRDPSO at t = 1 min

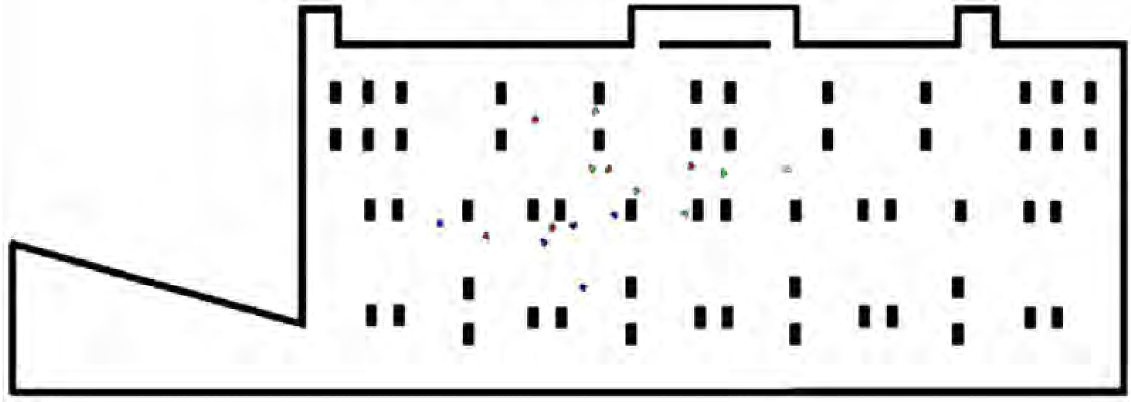


(c) The QRDPSO at t = 2 min

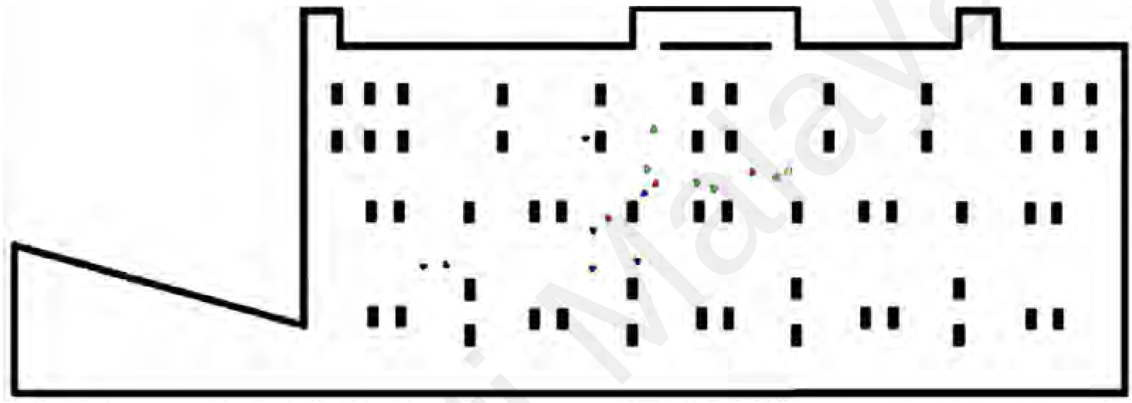


(d) The QRDPSO at t = 4 min

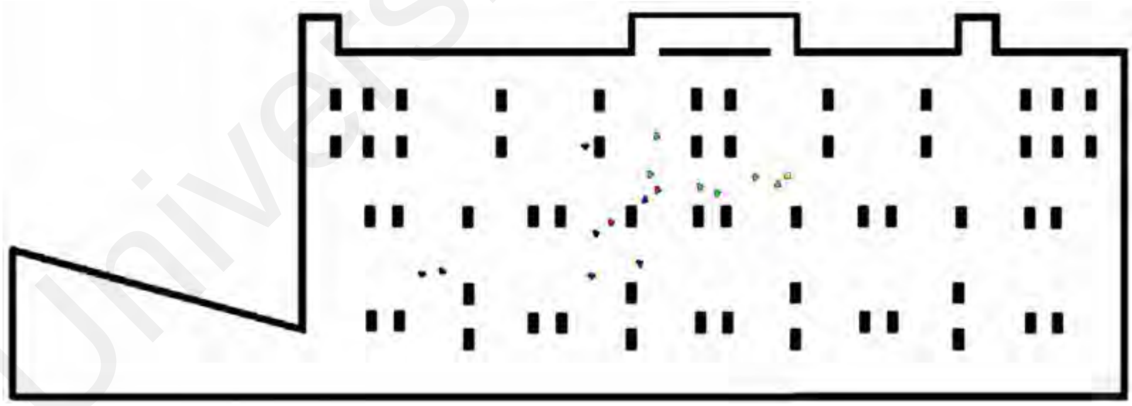
(e) The QRDPSO at t = 6 min



(f) The QRDPSO at t = 7 min

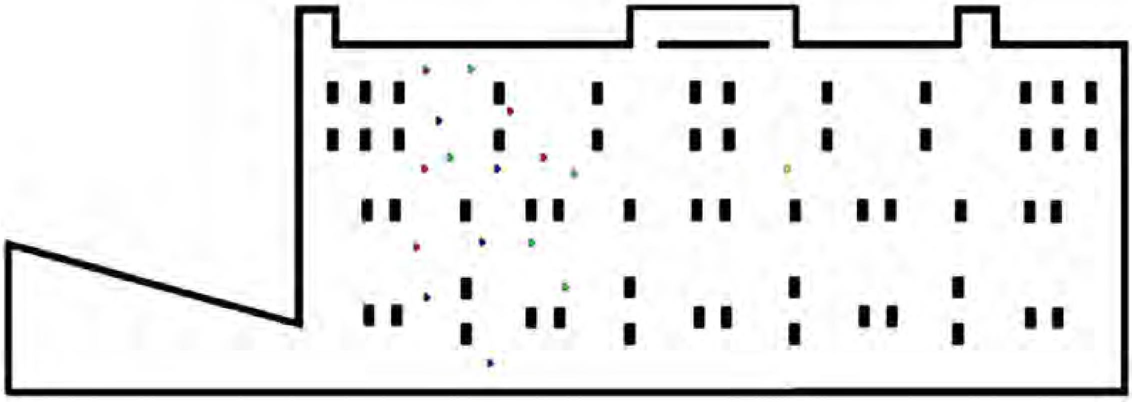


(g) The QRDPSO at t = 9 min

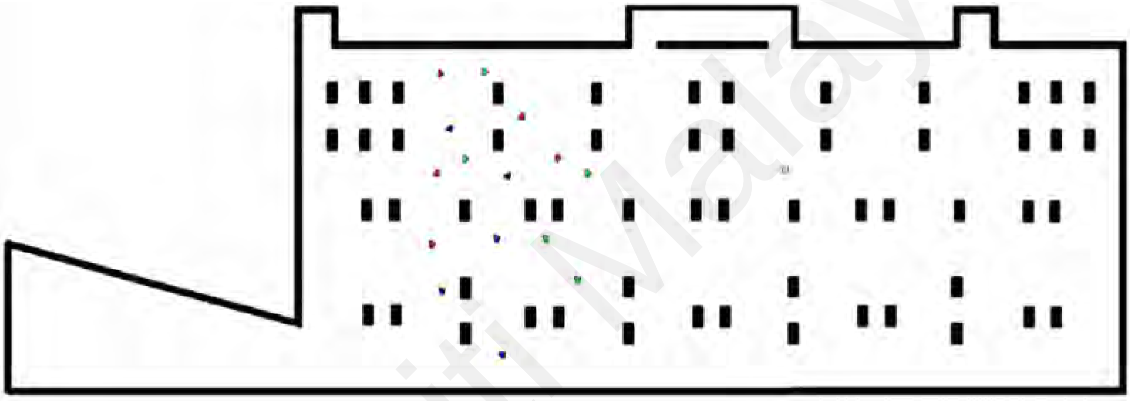


(h) The QRDPSO at t = 10 min (3 robot loss)

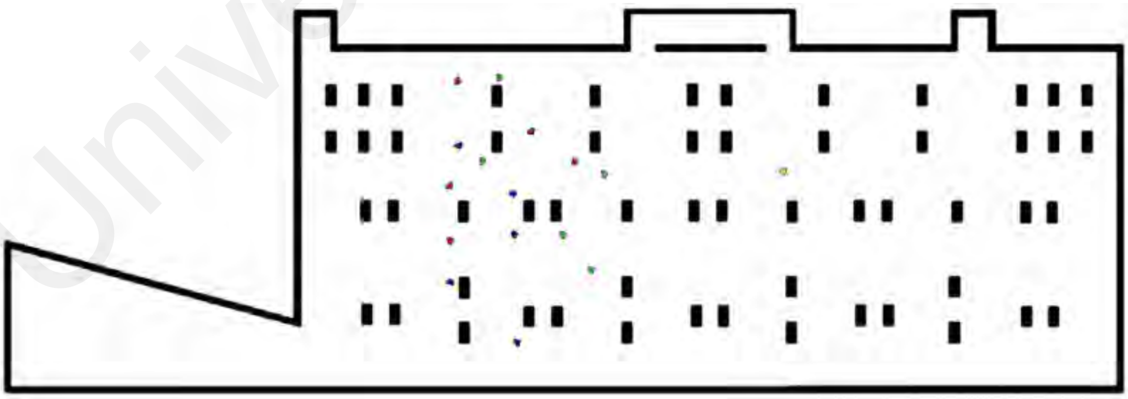
Figure 4.5: The QRDPSO simulation at initialization and several intervals



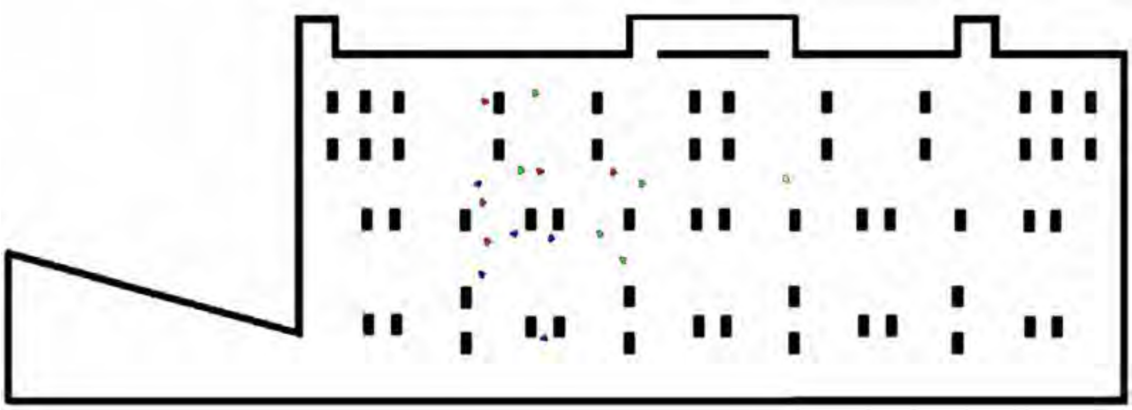
(a) The RDPSO at t=0 (initialization)



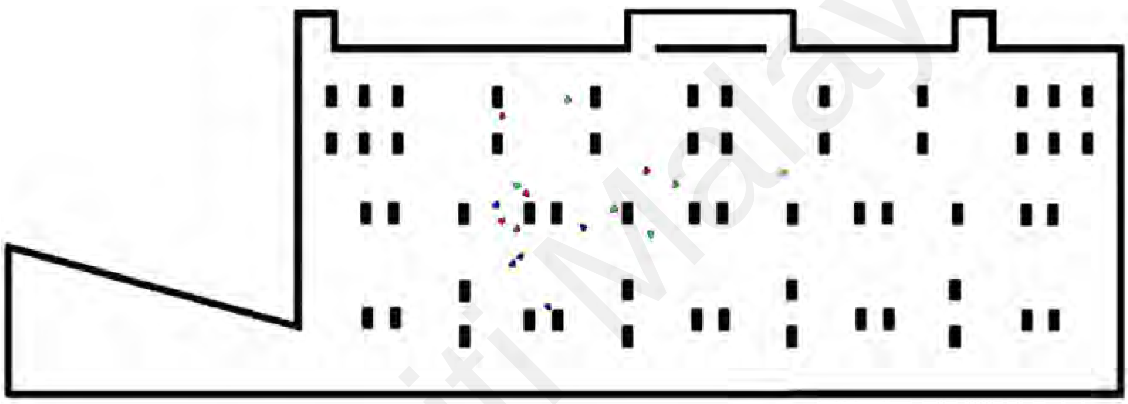
(b) The RDPSO at t=1 min



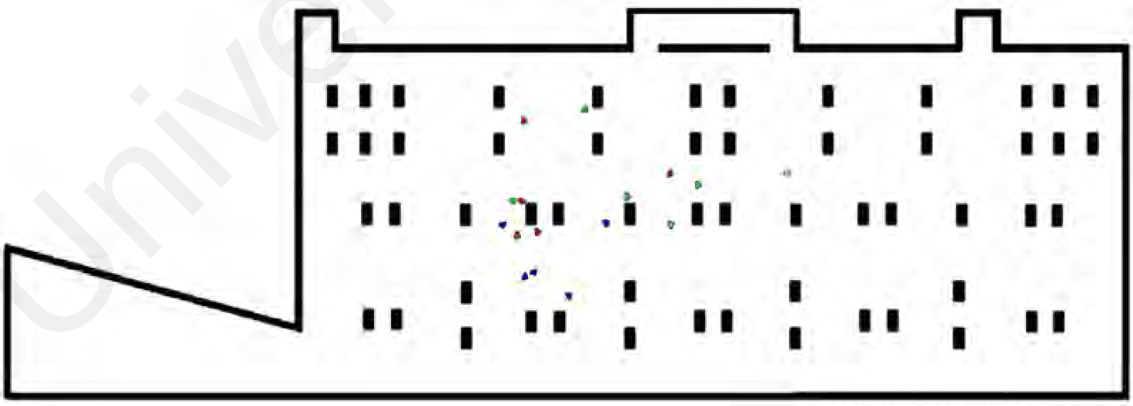
(c) The RDPSO at t=2 min



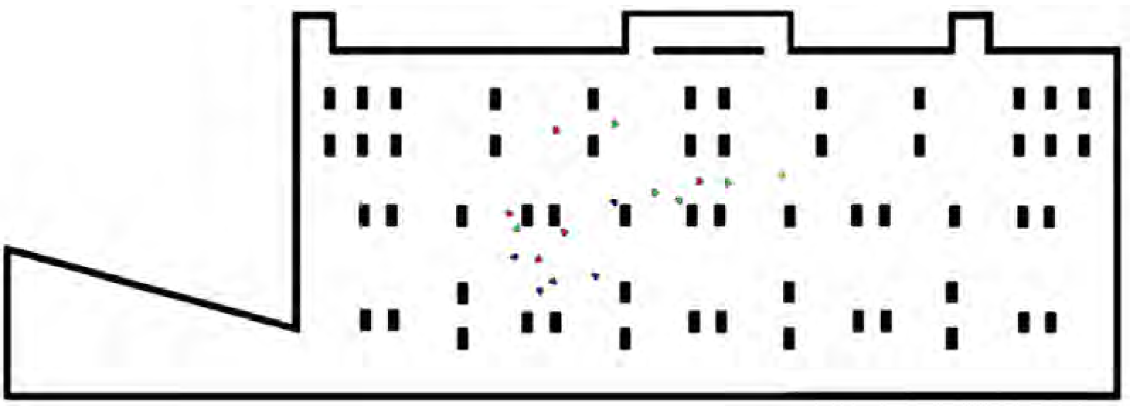
(d) The RDPSO at t=4 min



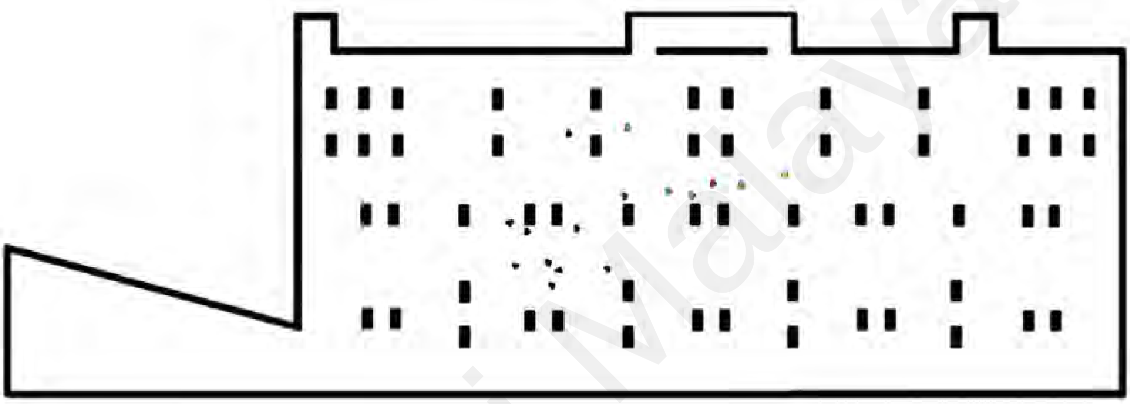
(e) The RDPSO at t=6 min



(f) The RDPSO at t=7 min



(g) The RDPSO at t=9 min



(h) The RDPSO at t=10 min (8 robots are loss)

Figure 4.5: The RDPSO simulation at initialization and several intervals

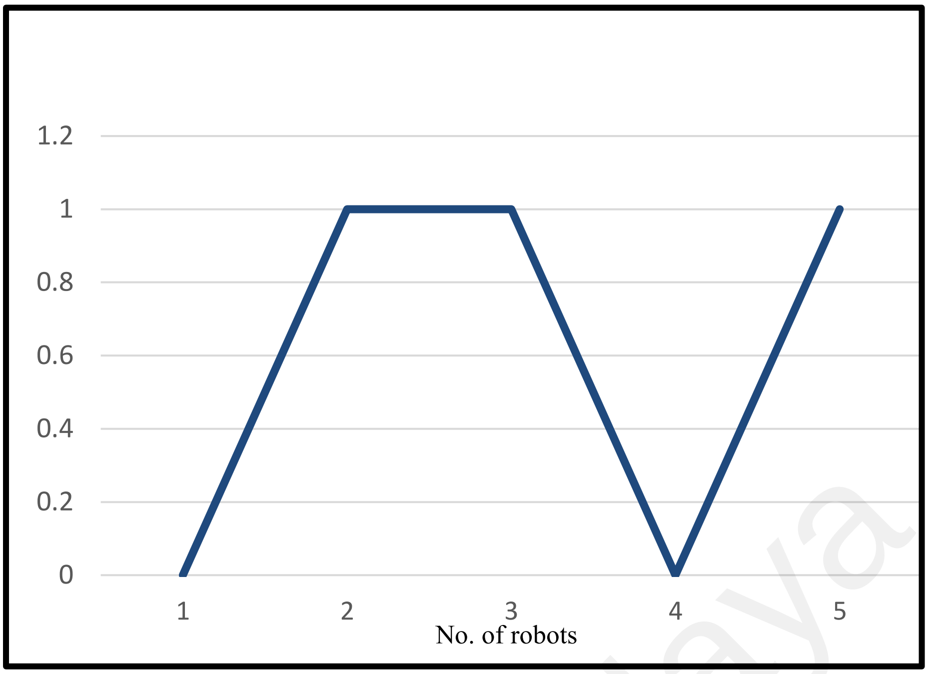
The QRDPSO exploration is superior to its predecessor, the RDPSO in terms of robot loss. After 10 minutes of simulation in the same experimental setup, the RDPSO started to lose robots and eventually ends with eight robot loss (see Figure 4.5 (h)). On the other hand, the QRDPSO shows lesser communication disruption and can connect to all robots in the swarm for most of the time. The QRDPSO ends with three robot loss at 10 minutes. Each robot is an asset to a swarm. They are significant for the exploration task because they increase the odds at improving searching chances. This study shows improving the RDPSO with quantum behaviour boost the searching performance of the swarm robots.

4.3.2 The QRDPSO communication performance

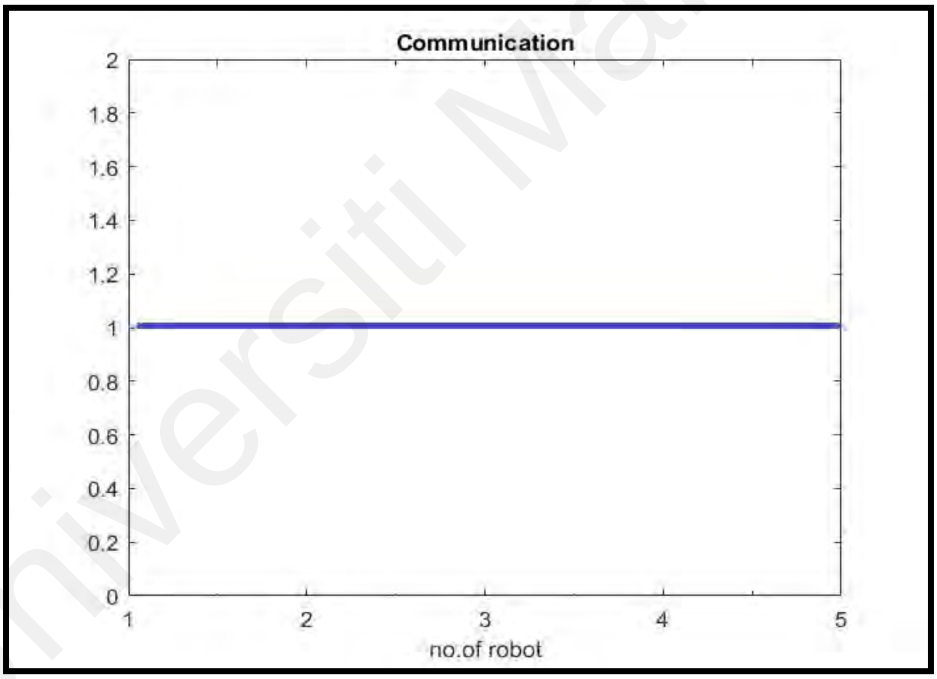
This research explores two communication protocols for QRDPSO. The first protocol under examination in this subsection is the AODV communication protocol, and the second one is the MR-LEACH proposed by Farooq et al. (2010). In the AODV protocol, the communication between robots shows interrupts. Figure 4.6(a) shows robot no. 1 receives no communication with robot no.4 but connects with robots no. 2, 3 and 5. In comparison, Figure 4.6(b) shows the communication in QRDPSO using MR-LEACH shows no interrupt between robots, so all robots maintain fully connected.

Quality of inter-robot communication can influence swarm convergence. Figure 4.7 shows that increasing the number of populations of robots to 5, 10, 15 and 20 decreases the time needed to find the optimal solution over maintaining the AODV and the MR-LEACH connectivity as well as the obstacles avoidance for the QRDPSO. In QRDPSO with AODV, five robots can reach the optimal solution in 366 iterations, and the QRDPSO with MR-LEACH needs 375 iterations, but the RDPSO requires 418 iterations to reach a victim. For ten robots, the QRDPSO with AODV can reach the optimal solution in 319 iterations, and QRDPSO with MR-LEACH needs 329 iterations, but in RDPSO, 368 iterations are required to reach a victim.

Fifteen robots are deployed in QRDPSO with AODV search to find the victim, it needs 207 iterations, and QRDPSO with MR-LEACH needs 220 iterations, but the RDPSO requires 252 iterations. Finally, when 20 robots are deployed to search the victim (optimal solution), the QRDPSO with AODV requires 180 iterations and QRDPSO with MR-LEACH needs 202 iterations. Still, RDPSO needs 230 iterations to reach a victim. These results show that robots with lower energy consumption can rescue the victim faster.



(a)



(b)

Figure 4.6: The QRDPSO communication with (a) AODV and using (b) MR-LEACH

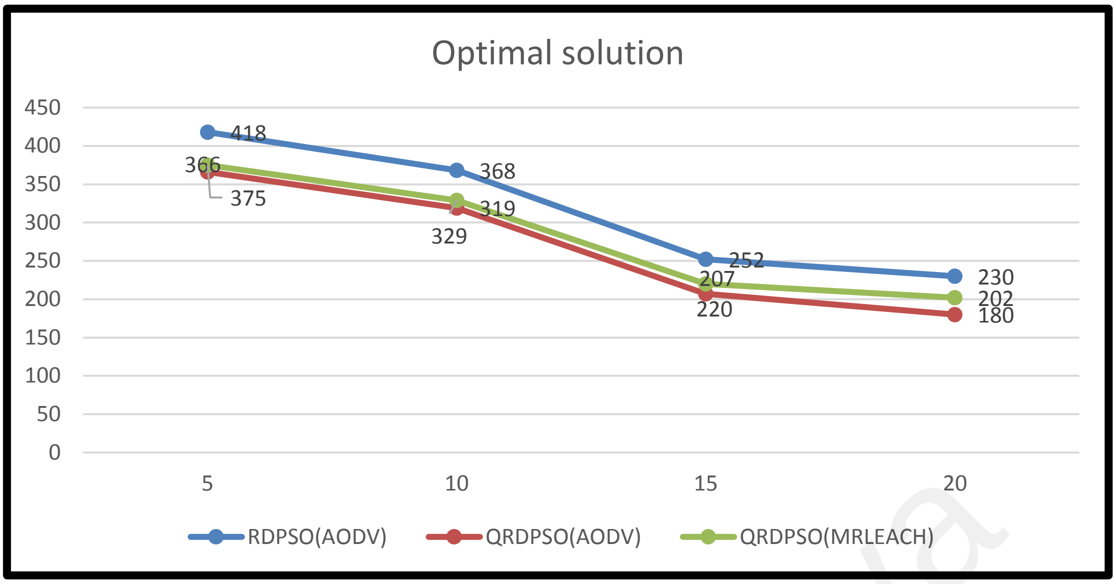


Figure 4.7: Comparison between QRDPSO using AODV and MR-LEACH

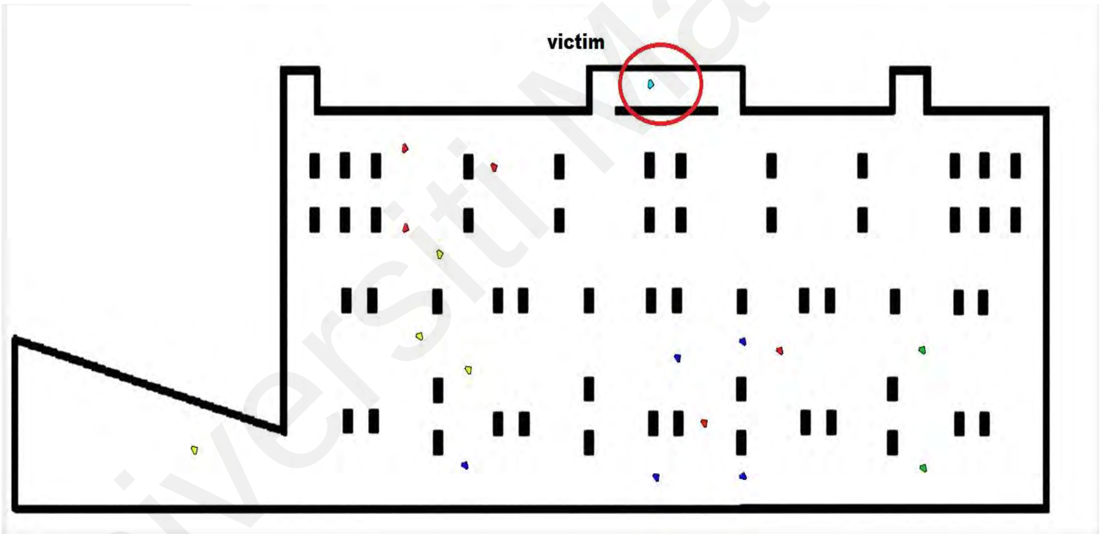


Figure 4.8: A 300x300m environment used in the experiment to compare QRDPSO(AODV) & QRDPSO (MR-LEACH)

Figure 4.8 shows the environmental setup used to compare the QRDPSO with AODV and the QRDPSO with MR-LEACH to investigate the number of robot kills. The triangular markers represent robots. The triangular-shaped in the red circle represents the victim at a random location unknown to the robots. A rectangular block represents a random obstacle hiding the victim.

In Figure 4.9, the environment which the robots are running the QRDPSO using AODV, green triangles denote robots which successfully located the victim and is proceeding towards it. One robot (marked by the blue circle) faces some trouble navigating and getting stuck in local optima. In this example, several black triangles are scattered far away from other robots (black circle). These robots lost communication with the swarm and are moving randomly in the hope to regain the communication range with the swarm. These robots may get back on track towards the victim if they can receive signals from other robots.

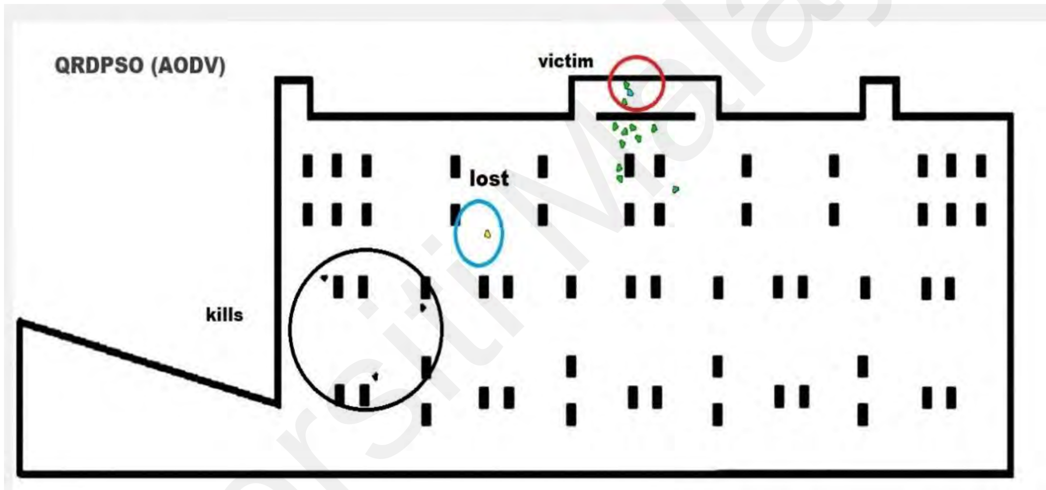


Figure 4.9: A QRDPSO algorithm using (AODV) protocol

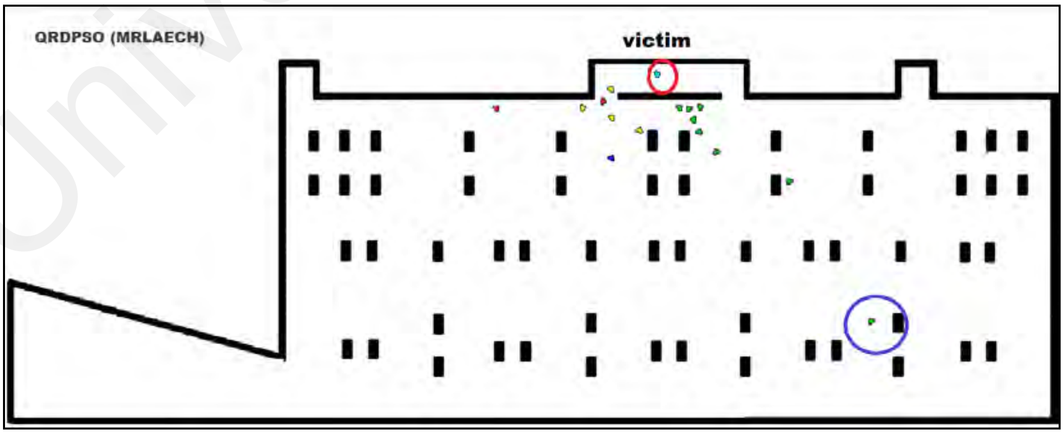


Figure 4.10: A QRDPSO algorithm using (MR-LEACH)

In Figure 4.10, when the robots use QRDPSO with MR-LEACH, it is observed that the communication is not lost with the swarm, and the robots successfully locate the victim and able to avoid getting stuck in local optima. Even the robot marked in the blue circle is not lost. The robot is separated but receives information to update its position and join the rest of the swarm.

Figure 4.11 shows a comparison between the QRDPSO running AODV and the QRDPSO with MR-LEACH. It is observed that increasing the number of populations of robots to 5, 10, 15 and 20 increases the number of robots lost when QRDPSO with AODV is used. When the MR-LEACH is used, the QRDPSO swarm can find the optimal solution without significant robot loss.

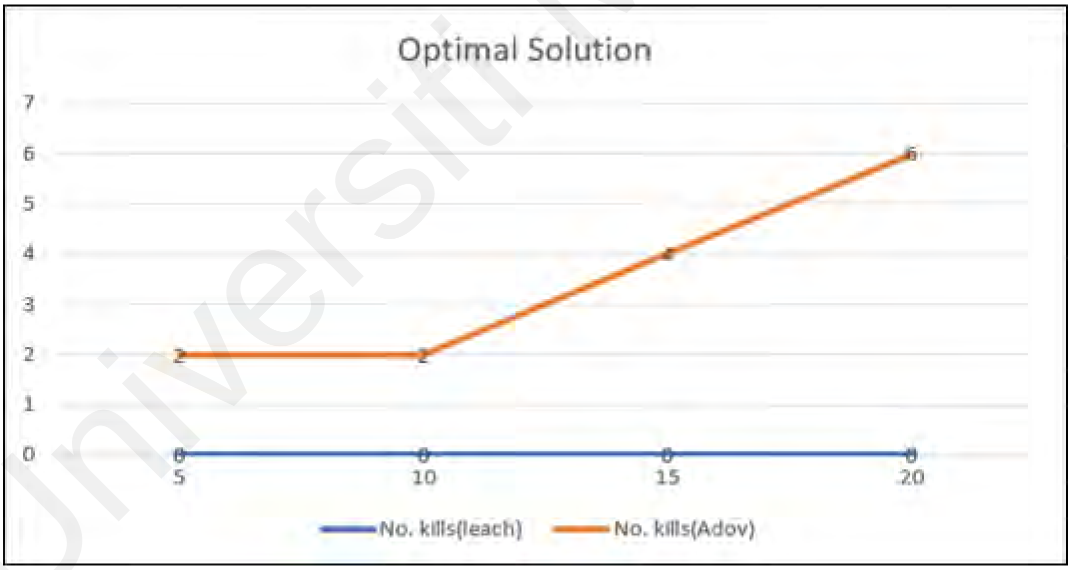
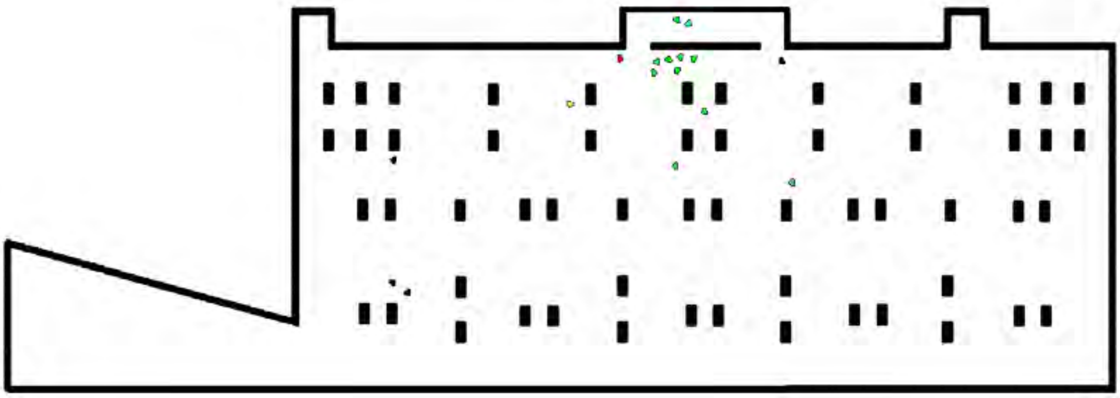


Figure 4.11: Several robot kills when the QRDPSO applies AODV and the MR-Leach protocol

Another attribute which is investigated in the experiment is channel availability for fault tolerance. Autonomous mobile robots have difficulty in carrying out the complex mission in a dynamic environment. In this study, a fault-tolerant system is designed for autonomous mobile robots using channel availability. The CHs can exchange messages between each other by using any channel. The QRDPSO robots were connected meaning they may continue sending messages without interruption even when one or more clustering hierarchy components fail.

The robot's communication performance is tested and analysed using two network protocols, the AODV and the MR-LEACH. For the QRDPSO, the performance of the swarm communication is optimum (not interrupted). However, the QRDPSO with AODV shows interrupts and communication break downs. Figure 4.12 shows an experiment's progression where the simulation runtime is about 5 minutes for QRDPSO with AODV. The same experimental setup is performed for QRDPSO with MR-LEACH in Figure 4.13.

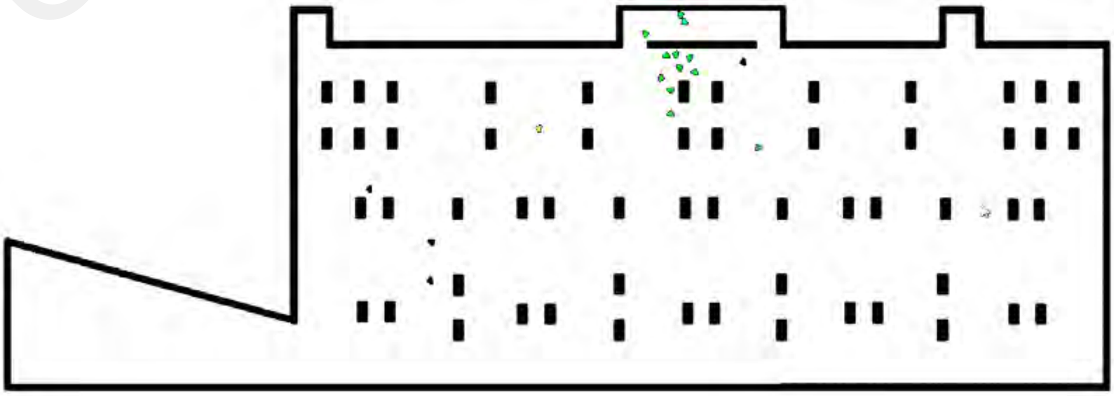
(a) The QRDPSO (AODV) at t=0 min (initialization)



(b) The QRDPSO (AODV) at t=1 min (initialization)

(c) The QRDPSO (AODV) at t=2 min

(d) The QRDPSO (AODV) at t=3 min

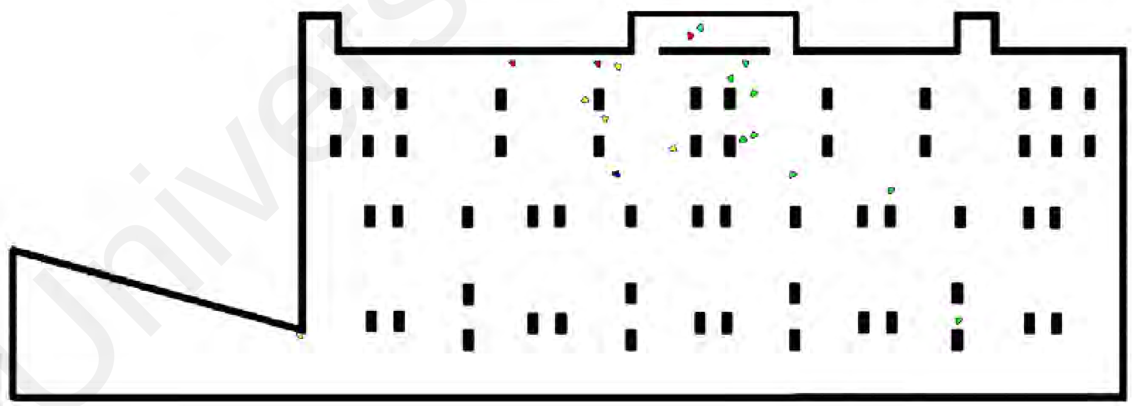


(e) The QRDPSO (AODV) at t=4 min

(f) The RDPSO (AODV) at t=5 min (4 robots lost communication)

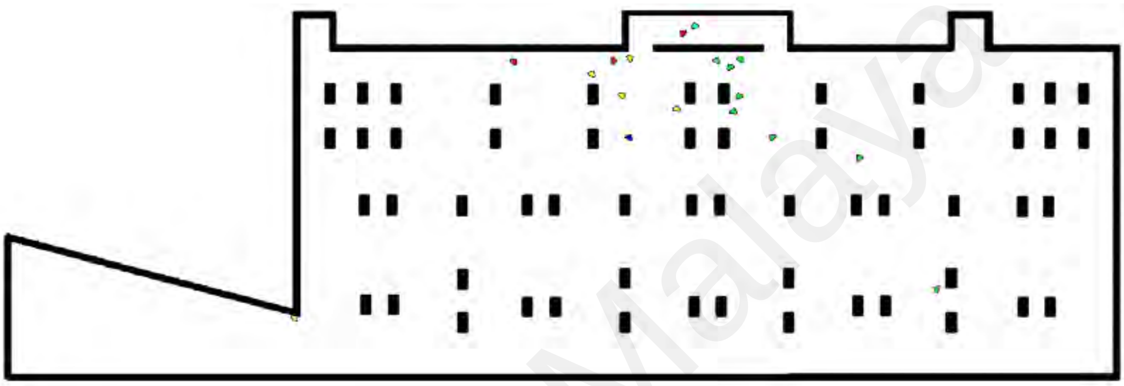
Figure 4.12: The QRDPSO swarm with the AODV communication protocol

(a) The QRDPSO (MR-LEACH) at t=0 min



(b) The QRDPSO (MR-LEACH) at t=1 min

(c) The QRDPSO (MR-LEACH) at t=2 min



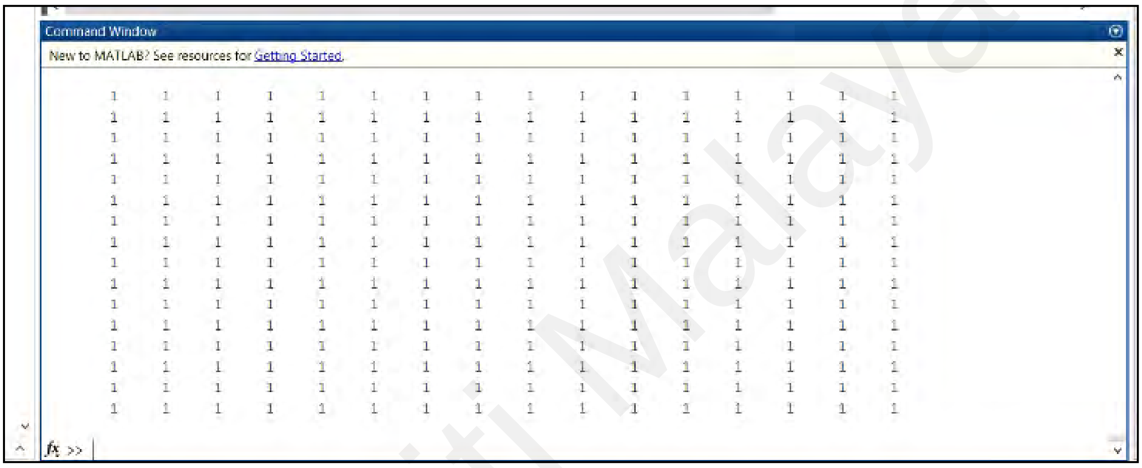
(d) The QRDPSO (MR-LEACH) at t=3 min

(e) The QRDPSO (MR-LEACH) at t=4 min

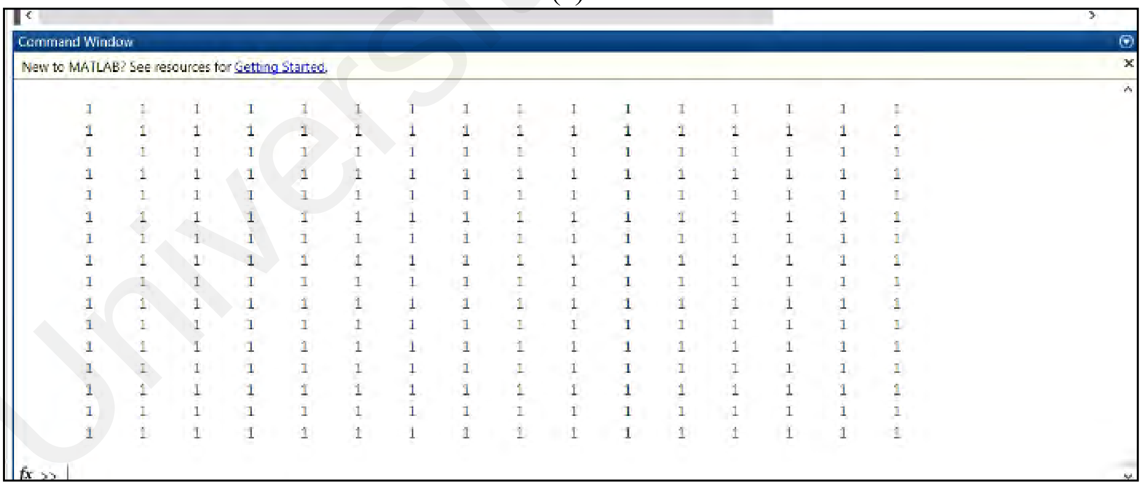
(f) The QRDPSO (MR-LEACH) at t=5 min (no robot loss detected

Figure 4.13: The QRDPSO swarm with the MR-LEACH communication protocol

The simulator indicates the messaging happening inside the QRDPSO swarm through publishing the adjacency matrix. The adjacency matrix denotes 0 when an interruption occurs. Otherwise, the matrix publishes 1 to indicate connectivity. When MR-LEACH is running, the QRDPSO adjacency matrix is fully connected. Figure 4.14 shows two examples when MR-LEACH is running at t=1 and t=4.



(a)



(b)

Figure 4.14: The QRDPSO (MR-LEACH) adjacency matrix at (a) t=1, and (b) t=4 showing full connectivity among the swarm robots

4.4 Energy consumption

The energy consumption of nodes should be minimized to increase the nodes' lifetime. Figure 4.16 shows the different energy consumed by 20 nodes concerning the simulation time between two protocols, the AODV and the MR LEACH. Based on the results obtained, it is shown that the QRDPSO running MR-LEACH consumes less energy than the QRDPSO running AODV. The result also showed that the power consumption kept increasing in both protocols when the simulation time increases.

This behaviour means the MR-LEACH can increase the lifetime for the nodes more than the AODV. Figure 4.16 shows the difference in the energy consumed by a node concerning the simulation time, showing that MR-LEACH consumes less than AODV. And the power consumption is increasing in both protocols when the simulation time increases. So, the MR leach increase lifetime for the nodes more than AODV. For Example, in 900 iteration the AODV consumed energy 63 % and MR LEACH 48%. So, the MR LEACH increase lifetime for the nodes more than AODV.

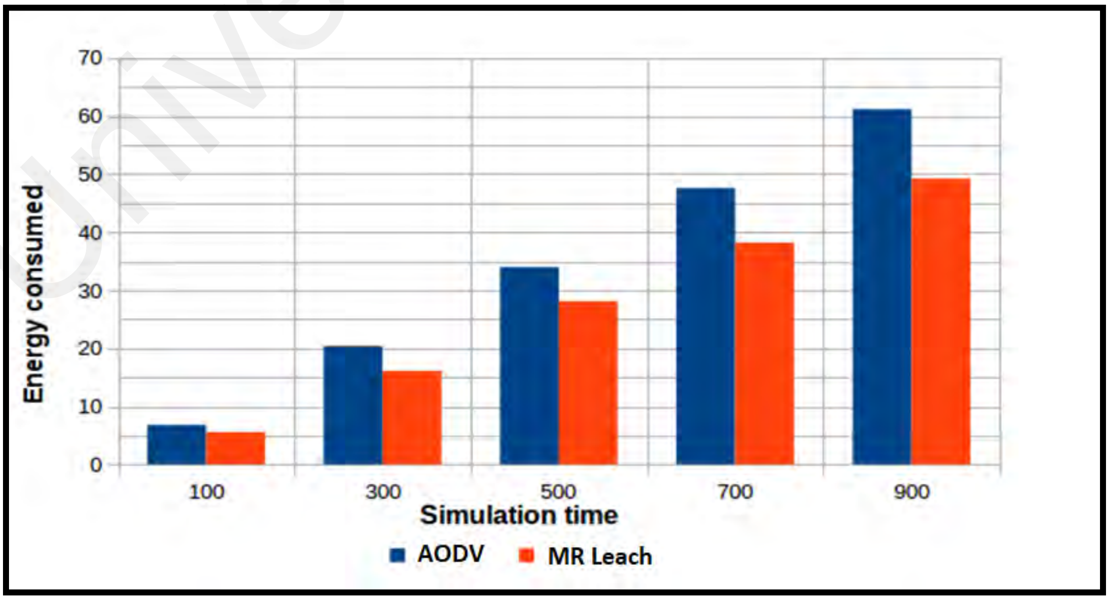


Figure 4.15: Comparison Energy Consumption between AODV & MR- LEACH

4.5 Discussion

This chapter presented a comparative study of the QRDPSO and the RDPSO algorithm's predecessor on a MATLAB simulator. Following similar experimental setup for both algorithms, the experiment showed that the QRDPSO model has a linear convergence of the whole population (robots) when reaching the global best solution and showing fewer robots lost than the RDPSO. Also, the QRDPSO performs better in both speed and energy consumption in comparison to the RDPSO. When twenty robots are deployed to search the victim (optimal solution), the RDPSO requires 20 min: 23 sec and QRDPSO used 10 mins: 01 sec to reach a victim. The capacity to keep resources helps the QRDPSO swarm to rescue the victim faster than RDPSO.

Communication is vital for the swarm to maintain cooperation. It can be reported that there is an improvement in terms of connectivity among individual robots in the QRDPSO swarm over the RDPSO. Nevertheless, there is still much to explore regarding enhancing the QRDPSO swarm communication for robot energy conservation and prolonged lifetime during search and rescue exploration. In this work, adopting a multi-hop clustering routing protocol using the MR-LEACH minimizes the entire sensor nodes (robots). MR LEACH consumed energy 48%, and ADOV consumed energy 63 % in 900 iterations. So, the MR LEACH increase lifetime for the nodes more than AODV.

Interestingly, the performance analysis shows that even though the QRDPSO with MR-LEACH performs well compared to the QRDPSO with AODV in terms of the increased lifetime of robots and reduce interrupt communication, the QRDPSO with AODV can reach the optimal solution faster than QRDPSO with MR Leach.

4.6 Chapter summary

This chapter shows the experimental setup and discusses the performance of the QRDPSO on two main aspects; exploration and communication. For exploration, the experimental setup and simulation compare the performance of the QRDPSO to its predecessor, the RDPSO. The result shows that the QRDPSO has a superior attribute, i.e. significant reduction of robot loss during swarm exploration.

For communication, the experimental setup and simulation dive into protocols to enhance the QRDPSO internal messaging. Two communication protocols have been evaluated; the AODV and the MR-LEACH. Results show that the AODV is more robust in speed but consumes too much energy and at times cause communication interruption. The MR-LEACH has a better pace and power management, which promotes swarm endurance. Swarm endurance is vital; thus, the MR-LEACH is advantageous in supporting exploration tasks. The next chapter concludes the thesis.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

This thesis begins with a notion: the quantum behaving particle agents in the QPSO show ideal searching capabilities, avoidance of premature convergence and faster convergence speed. Hypothetically, its adoption to practical swarm robotics algorithms such as the RDPSO can boost the searching capabilities, help robots avoid local optimal solutions and converge to reach the optimal global solution within a shorter time. Further investigation shows the notion *is* valid, with some conditions, and thus constitutes the most significant contribution of this work: discovering an *ideal* and *practical* quantum behaving algorithm for swarm robotics exploration and communication, in short, the QRDPSO.

The QRDPSO algorithm's success is attributed to the foundation of the QPSO equation. The particle movement in the QPSO is not based on velocity parameters but defined as wave (i.e. quantum-like), consequently offering stability and convergence speed. However, the QPSO lacks two properties limiting it from making waves onto practical robotics swarm applications. Namely, a communication function, because unlike particles, robots are supposed to communicate with other robots. Therefore, an obstacle avoidance function is required so a robot can use its sensors to navigate safely in its environment.

Having an obstacle avoidance function is where conceptually, the QRDPSO resemblance to the QPSO begins and ends. It begins by taking the QPSO form but reforms it by adding the two new functions using standard deviations. Even though the two new functions are available in RDPSO, it is noteworthy to claim that the two new functions' parameterisations, communication, and the obstacle avoidance functions used in this work are nowhere similar to the RDPSO's in practice. An innovative approach to impose a cost function or objective function to minimize the distance between the robot and victim to approach zero have been introduced in this work.

From there on, an in-depth analysis of distributed approach and the Darwinian paradigm shows why such combination is distinct for a real-time swarm robotics algorithm like the RDPSO. In practice, the RDPSO algorithm's insights over the likes of other PSO-based algorithms are threefold. One, the RDPSO is the only algorithm scalable to large populations of robots. Two, the RDPSO is traffic-savvy as it proposed reducing the amount of mandatory knowledge exchange among one robot to another. Three, the RDPSO can show higher speed convergence with better accuracy.

To capitalize on the strengths of the RDPSO for practical swarm robotics exercise, four methods forwarded by the RDPSO are adopted in the implementation of the QRDPSO as follows:

1. Inclusion of a *punish* and *reward* mechanism to mimic the deletion and creation of robots, keeping the Darwinian principles of *survival-of-the-fittest* and extending the swarms ability in avoiding sub-optimal solutions,
2. Inclusion of an obstacle avoidance algorithm to avoid collisions in static and dynamic environments, particularly unavoidable for practical use cases such as the SaR missions,
3. Inclusion of an enforcing multi-hop network connectivity algorithm to ensure that the network communication protocol remains connected throughout the mission with no interruptions between the robots, and

4. Inclusion of architecture to construct a planar deployment of robots using a network communication protocol to distribute the swarm robots as much as possible at the same time.

Where the RDPSO used MANET as the network communication protocol, the QRDPSO algorithm is completed with the MR-LEACH schema to control the robotics swarms' communication traffic. It is interesting to observe that with the MR-LEACH as the communication protocol, the interconnectivity in the QRDPSO remains uninterrupted. Full connectivity is mainly due to the MR-LEACH schema set as sink-less, which allows the robots to communicate directly to other robots without reporting individually to the base station. Such setup cuts transmission time, so messaging between the robots becomes instant, and the communication traffic is much less hectic, appealing to achieve the SaR missions' goals. In the implementation of the MR-LEACH schema, each node (or robot) is redefined as dynamic and not a static, mimicking the active behaviour of robots in the swarm. Choosing a distributed, multi-hop strategy over single-hop also significantly improves fault tolerance in the QRDPSO.

The MR-LEACH protocols the robots more flexible movement that can cover broad areas by using clustering hierarchy design. The robot's movement is freer than AODV and no interruption between robots. The information exchange between robots without interrupt can avoid local optima and find global best (victims) compared with AODV. A communication schema such as MR-LEACH optimizes the QRDPSO swarm for search and rescue mission. Most significantly, the QRDPSO swarm covers a wide area, without falling into local optima and promoting fast global best search.

In conclusion, this work shows how the QPSO, a quantum-based particle behaving algorithm is extended with a distributed Darwinian approach to producing a novel PSO derivation for swarm robotics application, coined the QRDPSO. This thesis showcased in length the details on the formulation and specific parameterization conditioning for the QRDPSO algorithm to overcome communication constraints and avoid robots from getting trapped in local solution in search and rescue simulations.

5.2 Future work

It is without reservation that some questions are raised from this work which provides motivations for future work. In particular, the RDPSO has shown success in simulation and after much investigation, found some success recently in real-time. How would the QRDPSO fare in the real world? Can the QRDPSO swarm robots cooperate and perform search and rescue in hostile environments? Would the MR-LEACH schema sustain the integrity and no-interruption policy of inter-robot communication? How far or large is the search space before the QRDPSO swarm robots lost communication? Further research needs to be done to understand the nature of the real world.

From a practical standpoint, the new QRDPSO with MR-LEACH could be useful as an approach for robot exploration and communication. A practical QRDPSO could lead to a new paradigm for swarm robotics. However, a more robust algorithm is due before performing tests in the real world and different environments, such as multiple targets.

Finally, the approach developed here is not restricted, and the QRDPSO algorithm proposed, their methods, tools and insights can, and should, be applied to other swarm robotic algorithms and applications.

REFERENCES

Abbasi, A., Hossain, L., Hamra, J., & Owen, C. (2010). Social Networks Perspective of Firefighters' Adaptive Behaviour and Coordination among them. *2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing*, (pp. 819-824). Hangzhou, China.

Abd-El-Wahed, W. F., Mousa, A. A., & El-Shorbagy, M. A. (2011). Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. *Journal of Computational and Applied Mathematics, 235*(5), 1446-1453.

Abdulla, A. E., Nishiyama, H., Yang, J., Ansari, N., & Kato, N. (2012). HYMN: A novel hybrid multi-hop routing algorithm to improve the longevity of WSNs. *IEEE Transactions on Wireless Communications*, *11*(7), 2531-2541.

Abichandani, P., Benson, H. Y., & Kam, M. (2011). Constraints, Decentralized Multi-Vehicle Path Coordination under Communication. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, (pp. 2306-2313). San Francisco, CA, USA.

Afsar, M. M., & Tayarani-N, M. H. (2014). Clustering in sensor networks: A literature survey. Journal of Network and Computer Applications, 46, 198-226.

Ahmad, N., & Agah, A. (2014). Plan and intent recognition in a multi-agent system for collective box pushing. *Journal of Intelligent Systems*, *23*(1), 95-108.

Akat, S. B., Gazi, V., & Marques, L. (2010). Asynchronous particle swarm optimization-based search with a multi-robot system: simulation and implementation on a real robotic system. *Turkish Journal of Electrical Engineering & Computer Sciences*, *18*(5), 749-764.

Al-khawaldah, M., Younes, T. M., Al-Adwan, I., Nisirat, M., & Alshamasin, M. (2014).

Automated multi-robot search for a stationary target. *International Journal of Control Science and Engineering*, *4*(1), 9-15.

Alejo, D., Cobano, J. A., Heredia, G., & Ollero, A. (2014). Collision-free 4D trajectory planning in Unmanned Aerial Vehicles for assembly and structure construction. *Journal of Intelligent & Robotic Systems*, *73*(1-4), 783-795.

AnandRao, D. T., & Parthiban, L. (2018). A survey on Energy Efficient Routing Protocols. *International Journal of Pure and Applied Mathematics*, *118*(18), 3109-3113.

Andriessen, J. E. (2012). *Working with groupware: understanding and evaluating collaboration technology*. Springer Science & Business Media.

Aras, R., Dutech, A., & Charpillet, F. (2004). Stigmergy in Multi Agent Reinforcement Learning. *Loria, Inst. Nat. de Recherche en Inf. et Autom.*, (pp. 468-469). Nancy, France.

Asimov, I. (1982). *The Complete Robots: I, Robot.* Voyager.

Ayash, M., Mikki, M., & Kangbin, Y. (2012). Improved AODV Routing Protocol to Cope with High Overhead in High Mobility MANETs. *6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, (pp. 244-251). Palermo, Italy.

Balch, T., & Hybinette, M. (2000). Social Potentials for Scalable Multi-Robot Formations. *Proceedings of the 2000 IEEE International Conference on Robotics, ICRA.* San Francisco, CA, USA.

Bang, A., & Ramteke, P. (2013). MANET: History, challenges and applications. *International Journal of Application or Innovation in Engineering & Management*, 2, 251.

Baroudi, U. (2017). Robot-assisted maintenance of wireless sensor networks using wireless energy transfer. *IEEE Sensors Journal*, 17(14), 4661-4671.

Bernal, M. A., Cortés, W. S., & Leguizamón, M. Á. (2017). MANET: Advantages, challenges and applications for education. *Sistemas & Telemática*, 15(43), 45-63.

Blum, C., Chiong, R., Clerc, M., De Jong, K., Michalewicz, Z., Neri, F., & Weise, T. (2012). Evolutionary optimization. In *Variants of evolutionary algorithms for real-world applications* (pp. 1-29). Springer, Berlin, Heidelberg.

Cai, Y., Chen, Z., Li, J., Li, Q., & Min, H. (2013a). An adaptive particle swarm optimization algorithm for distributed search and collective cleanup in complex environment. *International Journal of Distributed Sensor Networks*, *9*(12), 560579.

Cai, Y., Chen, Z., & Min, H. (2013b). Improving particle swarm optimization algorithm for distributed sensing and search. In *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 373-379). IEEE.

Cai, Y., & Yang, S. X. (2016). A PSO-based approach with fuzzy obstacle avoidance for cooperative multi-robots in unknown environments. *International Journal of Computational Intelligence and Applications*, *15*(01), 1650001.

Casteigts, J. Albert, S. Chaumette, A. Nayak, I. Stojmenovic, Biconnecting a network of mobile robots using virtual angular forces, in: *IEEE 72nd Vehicular Technology Conference Fall, VTC 2010-Fall, Ottawa, ON*, 2010.

Chen, C.-H., & Chen, Y.-P. (2011). Convergence Time Analysis of Particle Swarm Optimization Based on Particle Interaction. *Advances in Artificial Intelligence, 2011*(204750), 1-7.

Chen, X., Ghadirzadeh, A., Folkesson, J., Björkman, M., & Jensfelt, P. (2018, October). Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3110-3116). IEEE.

Chen, C. H., Wu, W. X., & Chen, C. Y. (2013). RETRACTED: Ant-inspired collective problem-solving systems. *Journal of Vibration and Control*, *19*(16), 2481-2490.

Clayton, N. R., & Abbass, H. (2019, June). Machine teaching in hierarchical genetic reinforcement learning: Curriculum design of reward functions for swarm shepherding. In *2019 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1259-1266). IEEE.

Cleghorn, C. W., & Engelbrecht, A. P. (2015). Particle swarm variants: standardized convergence analysis. Swarm Intelligence, 9(2-3), 177-203.

Cornejo, A., & Lynch, N. (2010). Fault-Tolerance Through k-Connectivity. *IEEE International Conference on Robotics and Automation (ICRA2010), Workshop on Network Science and System Issues in Multi-Robot Autonomy.* Anchorage, Alaska.

Collinsm, T. J., & Shen, W. M. (2017, April). Particle swarm optimization for high-DOF inverse kinematics. In *2017 3rd international conference on control, automation and robotics (ICCAR)* (pp. 1-6). IEEE.

Couceiro, M. S., Clemente, F. M., & Martins, F. M. (2013). Analysis of football player's motion in view of fractional calculus. *Central European Journal of Physics, 11*(3), 1-15.

Couceiro, M. S., Fernandes, A., Rocha, R. P., & Ferreira, N. M. (2013 (In Press)). Understanding the Communication Complexity of the Robotic Darwinian PSO. *Robotica Journal of Cambridge*.

Couceiro, M. S., Ferreira, N. M., & Machado, J. T. (2011). Fractional Order Darwinian Particle Swarm Optimization. *3th Symposium on Fractional Signals and Systems, FSS'2011*, (pp. 127-136). Coimbra, Portugal.

Couceiro, M. S., Ferreira, N. M., & Machado, J. T. (2012). Hybrid adaptive control of a dragonfly model. *Communications in Nonlinear Science and Numerical Simulation, 17*(2), 893-903.

Couceiro, M. S., Figueiredo, C. M., Ferreira, N. M., & Machado, J. T. (2010). Electric Vehicle Drive System with Adaptive PID Control. *29th IASTED International Conference on Modelling, Identification and Control (MIC2010).* Innsbruck, Austria.

Couceiro, M. S., Figueiredo, C. M., Fonseca Ferreira, N. M., & Tenreiro Machado, J. A. (2009). Biological Inspired Flying Robot. *Proceedings of IDETC/CIE ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.* San Diego, USA.

Couceiro, M. S., Figueiredo, C. M., Luz, M. A., & Delorme, M. J. (2014 (In Press)). Zombie Infection Warning System Based on Fuzzy Decision-Making. In R. Smith, *Mathematical Modelling of Zombies.* Ottawa, Canada: University of Ottawa Press.

Couceiro, M. S., Figueiredo, C. M., Luz, M. A., & Delorme, M. J. (2014). Zombie Infection Warning System Based on Fuzzy Decision-Making. In R. Smith, *Mathematical Modelling of Zombies.* Ottawa, Canada: University of Ottawa Press.

Couceiro, M. S., Figueiredo, C. M., Luz, J. A., Ferreira, N. M., & Rocha, R. P. (2011). A Low-Cost Educational Platform for Swarm Robotics. *International Journal of Robots, Education and Art, 2*(1), 1-15.

Couceiro, M. S., Figueiredo, C. M., Portugal, D., Rocha, R. P., & Ferreira, N. M. (2012). Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Platforms. *IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS2012), (pp. 4614-4619). Vilamoura, Algarve, Portugal.

Couceiro, M. S., Luz, J. M., Figueiredo, C. M., & Ferreira, N. M. (2012). Modeling and Control of Biologically Inspired Flying Robots. *Robotica Journal of Cambridge, 30*(1), 107-121.

Couceiro, M. S., Luz, J. M., Figueiredo, C. M., Ferreira, N. M., & Dias, G. (2010). Parameter Estimation for a Mathematical Model of the Golf Putting. In V. M. Marques, C. S. Pereira, & A. Madureira (Ed.), *Proceedings of WACI-Workshop Applications of Computational Intelligence* (pp. 1-8). Coimbra, Portugal: ISEC - IPC.

Couceiro, M. S., Machado, J. T., Rocha, R. P., & Ferreira, N. M. (2012). A Fuzzified Systematic Adjustment of the Robotic Darwinian PSO. *Robotics and Autonomous Systems, 60*(12), 1625-1639.

Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2012). Introducing the Fractional Order Robotic Darwinian PSO. *Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (ICNPAA2012)*, (pp. 242-251). Vienna, Austria.

Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2013a). Mechanism and Convergence Analysis of a Multi-Robot Swarm. *Journal of Intelligent & Robotic Systems*.

Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2013b). On a Macroscopic Model of an Evolutionary Swarm Robotics Exploration Approach. *IEEE Transactions on Evolutionary Computation*.

Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2014). Mechanism and convergence analysis of a multi-robot swarm approach based on natural selection. *Journal of Intelligent & Robotic Systems*, *76*(2), 353-381.

Couceiro, M. S., Mendes, R., Fonseca Ferreira, N. M., & Tenreiro Machado, J. A. (2009). Control Optimization of a Robotic Bird. *European Workshop on Movement Science (EWOMS'09).* Lisbon, Portugal.

Couceiro, M. S., Portugal, D., Gonçalves, N., Rocha, R., Luz, J. M., Figueiredo, C. M., & Dias, G. (2013). A Methodology for Detection and Estimation in the Analysis of the Golf Putting. *Pattern Analysis and Applications, 16*(3), 459-474.

Couceiro, M. S., Portugal, D., & Rocha, R. P. (2013). A Collective Robotic Architecture in Search and Rescue Scenarios. *Proceedings of the 28th Symposium On Applied Computing (SAC2013)*, (pp. 64-69). Coimbra, Portugal.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011, November). A novel multi-robot exploration approach based on particle swarm optimization algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* (pp. 327-332). IEEE.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011a). A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms. *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, (pp. 327-332). Kyoto, Japan.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011b). Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms. *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR2011)*, (pp. 284-289). Kyoto, Japan.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2013a). A PSO Multi-Robot Exploration Approach Over Unreliable MANETs. *Advanced Robotics, 27*(16), 1-14.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2013b). Fault-Tolerance Assessment of a Darwinian Swarm Exploration Algorithm under Communication Constraints.

Proceedings of IEEE International Conference on Robotics and Automation (ICRA2013), (pp. 2000-2005). Karlsruhe, Germany.

Couceiro, M. S., Rocha, R. P., Ferreira, N. M., & Vargas, P. A. (2013). Darwinian Robotic Swarms for Exploration with Minimal Communication. *IEEE Congress on Evolutionary Computation (CEC'2013), Special session on Evolutionary Robotics, 2013.* Cancún, México, USA

Couceiro, M. S., Rocha, R. P., Ferreira, N. M., & Vargas, P. A. (2013, June). Darwinian robotic swarms for exploration with minimal communication. In *2013 IEEE Congress on Evolutionary Computation* (pp. 127-134). IEEE.

Couceiro, M. S., Rocha, R. P., Figueiredo, C. M., Luz, J. M. A., & Ferreira, N. F. (2012, October). Multi-robot foraging based on Darwin's survival of the fittest. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 801-806). IEEE.

Couceiro, M. S., Rocha, R. P., Figueiredo, C. M., Luz, J. A., & Ferreira, N. M. (2012). Multi-Robot Foraging based on Darwin's Survival of the Fittest. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012)*, (pp. 801-806). Vilamoura, Algarve, Portugal.

Couceiro, M. S., Rocha, R. P., & Martins, F. M. (2015, May). Towards a predictive model of an evolutionary swarm robotics algorithm. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2090-2096). IEEE.

Couceiro, M. S., Vargas, P. A., Rocha, R. P., & Ferreira, N. M. (2014). Benchmark of Swarm Robotic Distributed Techniques in a Search Task. *Robotics and Autonomous Systems, 62*(2), 200-213.

Dadgar, M., Couceiro, M. S., & Hamzeh, A. (2017, October). RDPSO diversity enhancement based on repulsion between similar ions for robotic target searching. In *2017 Artificial Intelligence and Signal Processing Conference (AISP)* (pp. 275-280). IEEE.

Dadgar, M., Jafari, S., & Hamzeh, A. (2016). A PSO-based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing*, *177*, 62-74.

Dai, X., Jiang, L., & Zhao, Y. (2016). Cooperative exploration based on supervisory control of multi-robot systems. *Applied Intelligence*, *45*(1), 18-29.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. H. Milford; Oxford University Press.

Dean, M. (1913). Book of Proverbs. *Catholic Encyclopedia. Adapted from Holman Bible Handbook on Proverbs*.

Deng, W., Zhao, H., Yang, X., Xiong, J., Sun, M., & Li, B. (2017). Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment. *Applied Soft Computing*, *59*, 288-302.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., Foster, A. (2011). *Swarmanoid: a novel concept for the sudy of heterogenous robotic swarms.* Bruxelles, Belgium: IRIDIA.

de Sá, A. O., Nedjah, N., & de Macedo Mourelle, L. (2014, June). Genetic and backtracking search optimization algorithms applied to localization problems. In *International Conference on Computational Science and Its Applications* (pp. 738-746). Springer, Cham.

de Sá, A. O., Nedjah, N., & de Macedo Mourelle, L. (2016). Distributed efficient localization in swarm robotics using min–max and particle swarm optimization. *Expert Systems with Applications*, *50*, 55-65.

Djaneye-Boundjou, O., Xu, X., & Ordóñez, R. (2016, July). Automated particle swarm optimization based PID tuning for control of robotic arm. In *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)* (pp. 164-169). IEEE.

Ducatelle, F., Di Caro, G. A., & Gambardella, L. M. (2010, July). Cooperative self-organization in a heterogeneous swarm robotic system. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation* (pp. 87-94).

Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942-1948). Citeseer.

Elfwing, S., & Seymour, B. (2017, September). Parallel reward and punishment control in humans and robots: Safe reinforcement learning using the MaxPain algorithm. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (pp. 140-147). IEEE.

Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware - Some Issues and Experiences. *Communications of the ACM, 34*(1), 38-58.

Engelbrecht, A. P. (2007). *Computational intelligence: an introduction*. John Wiley & Sons.

Farooq, M. O., Dogar, A. B., & Shah, G. A. (2010, July). MR-LEACH: multi-hop routing with low energy adaptive clustering hierarchy. In *2010 Fourth International Conference on Sensor Technologies and Applications* (pp. 262-268). IEEE.

Floreano, D., & Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies.* Cambridge, MA: MIT Press.

Francis, S. L., Anavatti, S. G., & Garratt, M. (2013, August). Real time cooperative path planning for multi-autonomous vehicles. In *2013 International Conference on*

Advances in Computing, Communications and Informatics (ICACCI) (pp. 1053-1057). IEEE.

Gasser, R., & Huhns, M. N. (2014). *Distributed artificial intelligence* (Vol. 2). Morgan Kaufmann.

Ghamisi, P., Couceiro, M. S., Benediktsson, J. A., & Ferreira, N. M. (2012). An Efficient Method for Segmentation of Images Based on Fractional Calculus and Natural Selection. *Expert Systems with Applications, Elsevier, 39*(16), 12407-12417.

Gong, B., Li, L., Wang, S., & Zhou, X. (2008, August). Multihop routing protocol with unequal clustering for wireless sensor networks. In *2008 ISECS international colloquium on computing, communication, control, and management* (Vol. 2, pp. 552-556). IEEE.

Goodrich, M. A., Pendleton, B., Kerman, S., & Sujit, P. B. (2012). What Types of Interactions do Bio-Inspired Robot Swarms and Flocks Afford a Human? *Proceedings of Robotics, Science and Systems (RSS)*, (pp. 105-112). Sydney, Australia.

Groß, R., O'Grady, R., Christensen, A. L., & Dorigo, M. (2011). The Swarm-bot experience: Strength and mobility through physical cooperation. In S. Kernbach, & S. Kernbach (Ed.), *Handbook of Collective Robotics* (pp. 49-76). Pan Stanford Publishing.

Grotuss, J. (2011). *The developmental effect of the presence of a recipient in a modified dictator game*. Florida Atlantic University.

Guo, J., Wang, J., & Cui, G. (2012). Online path planning for UAV navigation based on quantum particle swarm optimization. In *Advanced Technology in Teaching-Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009)* (pp. 291-302). Springer, Berlin, Heidelberg

Guo, J., Wang, X., & Zheng, X. (2010, July). Trajectory planning of redundant robot manipulators using QPSO algorithm. In *2010 8th World Congress on Intelligent Control and Automation* (pp. 403-407). IEEE.

Han, Z., Wu, J., Zhang, J., Liu, L., & Tian, K. (2014). A general self-organized tree-based energy-balance routing protocol for wireless sensor network. IEEE Transactions on Nuclear Science, 61(2), 732-740.

Helen, D. & Arivazhagan, D. (2014). Applications, advantages and challenges of ad hoc networks. *Journal of Academia and Industrial Research*, 2(8), 453-456.

Hereford, J. M., & Siebold, M. A. (2010). Bio-Inspired Search Strategies for Robot Swarms. In E. M. Martin, *Swarm Robotics, From Biology to Robotics* (pp. 1-27).

Herrero, S. (2018). *Bear attacks: their causes and avoidance*. Rowman & Littlefield.

Hollinger, G., & Singh, S. (2010). Multi-Robot Coordination with Periodic Connectivity. *IEEE International Conference on Robotics and Automation, ICRA2010*, (pp. 4457-4462). Anchorage, Alaska, USA.

Huang, Z. (2016). Improved quantum particle swarm optimization for mangroves classification. *Journal of Sensors, 2016*.

Huber, M. J., & Durfee, E. (1995). Deciding when to commit to action during observation-based coordination. *Proceedings of the First International Conference on Multi-Agent Systems*, (pp. 163–170).

Islam, M. R., Tajmiruzzaman, M., Muftee, M. M. H., & Hossain, M. S. (2014). Autonomous robot path planning using particle swarm optimization in dynamic environment with mobile obstacles & multiple target. In *International conference on mechanical, industrial and energy engineering* (pp. 1-6).

Izadi, D., Abawajy, J. H., Ghanavati, S., & Herawan, T. (2015). A data fusion method in wireless sensor networks. *Sensors*, *15*(2), 2964-2979.

Julian, B. J., Angermann, M., Schwager, M., & Rus, D. (2012). Distributed robotic sensor networks: An information-theoretic approach. *International Journal of Robotics Research (IJRR), 31*(10), 1134-1154.

Kadry, S. (Ed.). (2018). *Stochastic Methods for Estimation and Problem Solving in Engineering*. IGI Global.

Kaur, H., Sahni, V., & Bala, M. (2013). A survey of reactive, proactive and hybrid routing protocols in MANET: *A review. International Journal of Computer Science and Information Technologies*, 4(3), 498-500.

Kar, A. K. (2016). Bio inspired computing–a review of algorithms and scope of applications. *Expert Systems with Applications*, *59*, 20-32.

Keijsers, M., Kazmi, H., Eyssel, F., & Bartneck, C. (2019). Teaching Robots a Lesson: Determinants of Robot Punishment. International Journal of Social Robotics, 1-14.

Kernbach, S., Häbe, D., Kernbach, O., Thenius, R., Radspieler, G., Kimura, T., & Schmickl, T. (2013). Adaptive collective decision-making in limited robot swarms without communication. *The International Journal of Robotics Research, 32*(1), 35-55.

Keshavdas, S., & Kruijff, G. J. (2012). Functional Mapping for Human-Robot Collaborative Exploration. *Proc. of the 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2012)*, (pp. 1-6). College Station, TX, USA.

Kim, Y. D., Kang, J. H., Sun, D. H., Moon, J. I., Ryuh, Y. S., & An, J. (2010). Design and Implementation of User-Friendly Remote Controllers for Rescue Robots Used at Fire Sites. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, (pp. 377-382). Taipei, Taiwan, China.

Kim, K. T., Lyu, C. H., Moon, S. S., & Youn, H. Y. (2010, April). Tree-based clustering (TBC) for energy efficient wireless sensor networks. In 2010 IEEE 24th international conference on advanced information networking and applications workshops (pp. 680-685). IEEE.

Kobayashi, T., Aotani, T., Guadarrama-Olvera, J. R., Dean-Leon, E., & Cheng, G. (2019, August). Reward-Punishment Actor-Critic Algorithm Applying to Robotic Non-grasping Manipulation. In 2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob) (pp. 37-42). IEEE.

Kuchwa-Dube, C., & Pedro, J. O. (2018, October). Altitude and attitude tracking of a quadrotor-based aerial manipulator using super twisting sliding mode control. In *Proceedings of the 6th International Conference on Control, Mechatronics and Automation* (pp. 65-69).

Kulkarni, R. V., & Venayagamoorthy, G. K. (2010). Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews*), 40(6), 663-675.

Kumar, A. S., Manikutty, G., Bhavani, R. R., & Couceiro, M. S. (2017, September). Search and rescue operations using robotic darwinian particle swarm optimization. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1839-1843). IEEE.

Kuswadi, S., Adji, S. I., Sigit, R., Tamara, M. N., & Nuh, M. (2017, October). Disaster swarm robot development: On going project. In *2017 International Conference on Electrical Engineering and Informatics (ICELTICs)* (pp. 45-50). IEEE.

Lee, S., Choe, H., Park, B., Song, Y., & Kim, C. K. (2011). LUCA: An energy-efficient unequal clustering algorithm using location information for wireless sensor networks. *Wireless Personal Communications*, 56(4), 715-731.

Leung, K. Y., Barfoot, T. D., & Liu, H. H. (2012). Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach. *Journal of Intelligent and Robotic Systems, 66*(3), 321-342.

Li, B., Li, D., Zhang, Z., Yang, S., & Wang, F. (2015). Slope stability analysis based on quantum-behaved particle swarm optimization and least squares support vector machine. *Applied Mathematical Modelling*, *39*(17), 5253-5264.

Li, R. F, Tokgo, M.C., Han, C.H., & Hu, L. (2014). Trajectory Planning for Mobile Robot Navigation with Modified Method in Random Obstacle Environments. *International Conference on Advanced Computer Science and Engineering (ACSE).*

Li, A. Q., Cipolleschi, R., Giusto, M., & Amigoni, F. (2016). A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings. *Autonomous Robots, 40(4)*, 581-597.

Lin, H., Wang, L., & Kong, R. (2015). Energy efficient clustering protocol for large-scale sensor networks. IEEE Sensors Journal, 15(12), 7150-7160.

Liu, W., & Winfield, A. F. (2010). Modeling and Optimization of Adaptive Foraging in Swarm Robotic Systems. *The International Journal of Robotics Research, IJRR, 29*(14), 1743-1760.

Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An improved particle swarm optimization for feature selection. Journal of Bionic Engineering, 8(2), 191-200.

Macwan, A., Nejat, G., & Benhabib, B. (2011). Optimal Deployment of Robotic Teams for Autonomous Wilderness Search and Rescue. *Proceedings in IEEE/RSJ*

International Conference on Intelligent Robots and Systems, IROS11, (pp. 4544-4549). San Francisco, CA, USA.

Mahapatra, R. P., & Yadav, R. K. (2015). Descendant of LEACH based routing protocols in wireless sensor networks. *Procedia Computer Science*, *57*, 1005-1014.

Mardini, W., Yassein, M. B., Khamayseh, Y., & Ghaleb, B. A. (2014). Rotated hybrid, energy-efficient and distributed (R-HEED) clustering protocol in WSN. wseas transactions on communications, 13, 275-290.

Martins, S. C. S. (2013). *Modeling of plasmid dynamics and persistence in a community of hosts* (Doctoral dissertation, University of Birmingham).

Masehian, E., & Sedighizadeh, D. (2013). An improved particle swarm optimization method for motion planning of multiple robots. In *Distributed autonomous robotic systems* (pp. 175-188). Springer, Berlin, Heidelberg.

Maslow, A. H. (1943). A theory of human motivation. *Psychological review*, *50*(4), 370.

Meng K,Wang HG, Dong Z,Wong KP (2010) Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *Power Systems IEEE* Transactions on 25, 215–222.

Mi, Z., Yang, Y., & Liu, G. (2011a). Coverage Enhancement of Mobile Multi-agent Networks while Preserving Global Connectivity. *IEEE International Conference on Robotics and Automation, ICRA2011*, (pp. 5381-5386). Shanghai, China.

Mi, Z., Yang, Y., & Liu, G. (2011b). HERO: A Hybrid Connectivity Restoration Framework for Mobile Multi-Agent Networks. *IEEE International Conference on Robotics and Automation, ICRA2011*, (pp. 1702-1707). Shanghai, China.

Mishra, S., Sahoo, S., Thakkar, H., & Chakrabarty, A. (2012). Cluster Tree Power Aware Routing Protocol for WSN (CTPA). IJTEL, 1, 10-13.

Mukhija, P., Sawhney, R., & Krishna, K. M. (2010, May). Multi robotic exploration with communication requirement to a fixed base station. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1* (pp. 1515-1516).

Nakisa, B., Rastgoo, M. N., Nasrudin, M. F., & Nazri, M. Z. A. (2015). A multi-swarm particle swarm optimization with local search on multi-robot search system. *Journal of Theoretical and Applied Information Technology*, *71*(1), 129-136.

Nicholson, T. (2008). *Symposium on Social Inclusion Down Under* (pp. 1-5). Melbourne, Australia: Brotherhood of St Laurence.

Pei, Y., Mutka, M. W., & Xi, N. (2010). Coordinated Multi-Robot Real-Time Exploration With Connectivity and Bandwidth Awareness. *IEEE International Conference on Robotics and Automation (ICRA2010)*, (pp. 5460-5465). Anchorage, Alaska, USA.

Pini, G., Brutschy, A., Scheidler, A., Dorigo, M., & Birattari, M. (2012). *Task partitioning in a robot swarm: retrieving objects by transferring them directly between sequential sub-tasks.* Université Libre de Bruxelles. Bruxelles, Belgium: Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA).

Pires, E. J., Machado, J. A., Cunha, P. B., & Mendes, L. (2010). Particle swarm optimization with fractional-order velocity. *Journal on Nonlinear Dynamics, 61*(1-2), 295-301.

Potkonjak, V. (2012). Robot Handwriting: Why and How?. In *Interdisciplinary Applications of Kinematics* (pp. 19-35). Springer, Dordrecht.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *14th Annual Conference on Computer Graphics and Interactive Techniques*, *21*, pp. 25-34. New York, USA.

Rocha, R. P., Portugal, D., Couceiro, M., Araújo, F., Menezes, P., & Lobo, J. (2013, October). The CHOPIN project: Cooperation between Human and rObotic teams in catastroPhic INcidents. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (pp. 1-4). IEEE.

Sabattini, L., Chopra, N., & Secchi, C. (2011a). On decentralized connectivity maintenance for mobile robotic systems. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, (pp. 988-993). Orlando, Florida, USA.

Sabattini, L., Chopra, N., & Secchi, C. (2011b). Distributed Control of Multi–Robot Systems with Global Connectivity Maintenance. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, (pp. 2321-2326). San Francisco, CA, USA.

Saikishan, D., & Prasanna, K. (2010). *Multiple robot co-ordination using particle swarm optimisation and bacteria foraging algorithm.* BTech thesis, Department of Mechanical Engineering, National Institute of Technology, Rourkela.

Salem, A. O. A., & Shudifat, N. (2019). Enhanced LEACH protocol for increasing a lifetime of WSNs. *Personal and Ubiquitous Computing*, *23*(5-6), 901-907.

Sànchez, N. D. G., Vargas, P. A., & Couceiro, M. S. (2018, July). A Darwinian Swarm Robotics Strategy Applied to Underwater Exploration. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-6). IEEE.

Scutella, R., Wilkins, R., & Horn, M. (2009). *Measuring Poverty and Social Exclusion in Australia: A Proposed Multidimensional Framework for Identifying Socio-*

Economic Disadvantage (Vol. 4). Victoria, Australia: Melbourne Institute Working Paper Series.

Shannon, C. E. (1949). *The Mathematical Theory of Communication.* University of Illinois Press.

Shaw, I. S. (1998). *Fuzzy Control of Industrial Systems: Theory and Applications.* Kluwer Aca-demic Publishers.

Shi, Z., Zhang, X., Tu, J., & Yang, Z. (2013). An Efficient and Improved Particle Swarm Optimization Algorithm for Swarm Robots System. In *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013* (pp. 329-337). Springer, Berlin, Heidelberg.

ShubhamSingh, S. P., Chowdhury, J., Das, J. K., Agarwal, D., Surjan, D., & Nayak, K. (2018). Umbilical Cord Acid-Base Analysis: A Fuzzy Diagnosis Technique. *International Journal of Pure and Applied Mathematics*, *118*(18), 3397-3405.

Singh, S. K., Singh, M. P., & Singh, D. K. (2010). Routing protocols in wireless sensor networks–*A survey. International Journal of Computer Science & Engineering Survey (IJCSES)*, 1(2), 63-83.

Smith, A. J., & Hollinger, G. A. (2018). Distributed inference-based multi-robot exploration. *Autonomous Robots, 42(8), 1651-1668*.

Song, B., Wang, Z., & Zou, L. (2017). On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. *Cognitive Computation*, 9(1), 5-17.

Spanogianopoulos, S., Zhang, Q., & Spurgeon, S. (2017). Fast formation of swarm of UAVs in congested urban environment. *IFAC-PapersOnLine*, *50*(1), 8031-8036.

Staranowicz, A., & Mariottini, G. L. (2011). A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '11)*, (pp. 1-8). New York, USA.

Steinberg, M. L. (2011). Biologically-inspired approaches for self-organization, adaptation, and collaboration of heterogeneous autonomous systems. *Proceedings of SPIE Volume on Defense Transformation and Net-Centric Systems*, *8062*, pp. 1-13. Orlando, Florida, USA.

Stirling, T., Roberts, J., Zufferey, J.-C., & Floreano, D. (2012). Indoor Navigation with a Swarm of Flying Robots. *IEEE International Conference on Robotics and Automation (ICRA2012)*, (pp. 4641-4647). Saint Paul, MN, USA.

Suarez, J., & Murphy, R. (2011). A Survey of Animal Foraging for Directed, Persistent Search by Rescue Robotics. *Proceedings of the 2011 IEEE International Symposium on Safety,Security and Rescue Robotics*, (pp. 314-320). Kyoto, Japan.

Sun, J., Lai, C. H., & Wu, X. J. (2016). *Particle swarm optimisation: classical and quantum perspectives*. Crc Press.

Sun, J., Xu, W., & Feng, B. (2004, December). A global search strategy of quantum-behaved particle swarm optimization. In *IEEE Conference on Cybernetics and Intelligent Systems, 2004.* (Vol. 1, pp. 111-116). IEEE.

Tan, Y. K. (2017). Energy harvesting autonomous sensor systems: design, analysis, and practical implementation. *CRC Press*.

Tang, B., Zhanxia, Z., & Luo, J. (2017). A convergence-guaranteed particle swarm optimization method for mobile robot global path planning. *Assembly Automation*.

Tang, Q., & Eberhard, P. (2011a). Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics. *Mechanics based design of structures and machines*, *39*(2), 179-193.

Tang, Q., & Eberhard, P. (2011b). A PSO-based algorithm designed for a swarm of mobile robots. *Structural and Multidisciplinary Optimization*, *44*(4), 483.

Tardioli, D., Mosteo, A. R., Riazuelo, L., Villarroel, J. L., & Montano, L. (2010). Enforcing Network Connectivity in Robot Team Missions. *The International Journal of Robotics Research*, 29(4), 460-480.

Tarmizi, W. F. W., Elamvazuthi, I., Perumal, N., Nurhanim, K., Khan, M. A., Parasuraman, S., & Nandedkar, A. V. (2016, September). A Particle Swarm Optimization-PID controller of a DC Servomotor for Multi-Fingered Robot Hand. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)* (pp. 1-6). IEEE.

Tillett, J., Rao, T. M., Sahin, F., Rao, R., & Brockport, S. (2005). Darwinian Particle Swarm Optimization. In B. Prasad (Ed.), *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, (pp. 1474-1487). Pune, India.

Tokgo, M., Li, R., Kim, C., Hu, L., & Kim, M. (2014a). A method for trajectory planning of mobile robot in random obstacles environment. In *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics* (Vol. 2, pp. 51-54). IEEE.

Tokgo, M., & Li, R. (2014b). Estimation method for path planning parameter based on a modified QPSO algorithm. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications* (pp. 261-269). Springer, Cham.

Tsai, W. (2002). Social structure of "coopetition" within a multiunit organization: Coordination, competition, and intraorganizational knowledge sharing. *Organization science*, *13*(2), 179-190.

Turner, R. M., Rode, S., & Gagne, D. (2013, October). Toward distributed context-mediated behavior for multiagent systems. In International and Interdisciplinary Conference on Modeling and Using Context (pp. 222-234). Springer, Berlin, Heidelberg.

Van Den Bergh, F. (2001). An Analysis of Particle Swarm Optimizers (PSO). *Pretoria, University of Pretoria*, 78-85.

Van Steenkiste, T., Van Der Herten, J., Deschrijver, D., & Dhaene, T. (2019). ALBATROS: adaptive line-based sampling trajectories for sequential measurements. *Engineering with Computers*, *35*(2), 537-550.

Vardy, A. (2012, May). Accelerated patch sorting by a robotic swarm. In *2012 Ninth Conference on Computer and Robot Vision* (pp. 314-321). IEEE.

Vijayvargiya, K. G., & Shrivastava, V. (2012). An amend implementation on leach protocol based on energy hierarchy. International Journal of Current Engineering and Technology, 2(4), 427-431.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing*, *22*(2), 387-408.

Wang, D., Wang, H., & Liu, L. (2016). Unknown environment exploration of multi-robot system with the FORDPSO. Swarm and Evolutionary Computation, 26, 157-174.

Wang, Y., Wei, Z., Shao, M., & Zhang, G. (2013). A new method to calibrate robot visual measurement system. *Advances in Mechanical Engineering*, *5*, 548509.

Wang, Z., Qin, L., & Yang, W. (2015). A self–organising cooperative hunting by robotic swarm based on particle swarm optimisation localisation. *International Journal of Bio-Inspired Computation*, *7*(1), 68-73.

Xue, T., Li, R., Tokgo, M., Ri, J., & Han, G. (2017). Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm. *Soft Computing*, *21*(9), 2421-2437.

Yuan, X., Wang, P., Yuan, Y., Huang, Y., & Zhang, X. (2015). A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem. *Energy Conversion and Management*, *100*, 1-9.

Yuea, J., Zhang, W., Xiao, W., Tang, D., & Tang, J. (2012). Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks. *Procedia Engineering*, *29*, 2009-2015.

Yueqiang, Y., & Zhiwei, T. (2014, January). The study of path planning of welding manipulator based on improved qpso. In *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation* (pp. 774-777). IEEE.

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control, 8*(3), 338-353.

Zhong, X., & Zhou, Y. (2012). Maintaining wireless communication coverage among multiple mobile robots using fuzzy neural network. *IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, (pp. 35-41). Suzhou, China.

Zieliński, C., & Kornuta, T. (2014). Specification of tasks in terms of object-level relations for a two-handed robot. In *Recent Advances in Automation, Robotics and Measuring Techniques* (pp. 543-552). Springer, Cham.

Zuo, T., Min, H., Tang, Q., & Tao, Q. (2018). A Robot SLAM Improved by Quantum-Behaved Particles Swarm Optimization. *Mathematical Problems in Engineering*, *2018*.