Path Planning And Collision Avoidance for Clustered Central Place Foraging

A Thesis Presented to

The Faculty of the Computer Science Department

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

Student Name:
Apurva Bharaswadkar

Advisor:
Dr. Jason Isaacs

December 4, 2019

APPROVED FOR MS IN COMPUTER SCIENCE

Dec. 12, 2019

Advisor: Dr. Jason Isaacs — Date

12-13-19

Dr. Brian Thoms — Date

Dec. 13, 2019

Dr. Vida Vakilian — Date

APPROVED FOR THE UNIVERSITY

12/17/19

Dr. Osman Özturgut — Date

Path Planning And Collision Avoidance for Clustered Central Place Foraging

Title of Item

Swarm Robotics, Central Place Foraging, Path Planning, Collision Avoidance

3 to 5 keywords or phrases to describe the item

APURVA GOPAL BHARASWADKAR

Author(s) Name (Print)

Apurva

Author(s) Signature

12-04-2019

Date

Path Planning And Collision Avoidance for Clustered Central Place Foraging

Apurva Bharaswadkar

December 4, 2019

Abstract

Central place foraging algorithms for multiple robots are gaining attention due to their performance and efficiency in various applications like planetary surveys, mining, object transportation and manipulation. In foraging tasks, multiple robots search for resources and deposit the collected resources to a particular location called “nest” or “home”. If the resources are deposited at a central single collection point, it becomes a central place foraging task. The performance of central place foraging approaches is reduced due to reactive inter-robot collision avoidance. The performance decreases in two cases, first case is when two or more robots collect the resources from the same cluster and go to the central location for deposition and the second case is when the path of one robot going to nest from its search position or vice versa intersects with the path of another robot searching for resources. The approach proposed in this thesis is called Path Planning And Collision Avoidance Algorithm For Clustered Central Place Foraging (PPCA-CCPFA). PPCA-CCPFA concentrates on improving the performance of central place foraging task in terms of reducing the number of inter robot collisions and improving target collection in given time for clustered resource distributions. We compare our approach to the popular Distributed Deterministic Spiral Algorithm (DDSA). The proposed algorithm detects inter robot collision and finds an alternate collision free path for a robot in case 1 and adds a delay time for a robot in case 2. This approach has shown notable increase in the performance of DDSA with a single 8 x 8 resource cluster. This algorithm is tested on a single cluster resource distribution at random locations in the arena for a swarm size of 3 to 15 robots.

Contents

List of Figures

List of Tables

1 Introduction

Swarm Intelligence (SI) [3] is a fundamental discipline that deals with co-ordination of several individual entities to complete a task. The concept focuses on local interactions of the individual entities with each other and with the environment they are in. The individual entities are not under any supervision and are self organized [4]. Swarm Intelligence mainly studies the group behavior and interaction between individual agents for flocks of birds [5, 6], colonies of ants [5, 7] and termites [5, 8], bee hives [5, 9], schools of fish [5, 10], herd of land animals [5, 11], human colonies [5, 12] and proliferation of bacteria [5, 13]. These examples simulate the social behavior and collective intelligence rather than individual structure of the colonies. Over the past two decades, researchers have developed accurate mathematical models and techniques to describe the behavior of social insects to solve business issues and optimize business solutions. It has now become an important aspect in different fields like artificial intelligence, economics, sociology and biology to complete complex tasks through cooperation and division of labor [2].

Swarm Intelligence Systems usually consist of simple interactive individual agents governed by a set of rules. Intelligent behavior is observed as the swarm agents interact locally without global knowledge of the environment. Complex tasks are easily and efficiently completed by swarms that are decentralized. Thus, swarms provide scalable and efficient solutions to complex tasks as compared to an individual agent performing that task [2].

Swarm Intelligent Systems provide robustness and flexibility at lower costs as compared to a single robot.

Swarm robotics is a field inspired from Swarm Intelligence, where multiple robots are coordinated and distributed in a decentralized manner. The collective behavior is observed through the interaction between simple robots and with the environment that they are in. The swarm shows high efficiency, parallelism, scalability and robustness as compared to individual agents for accomplishing a complex task [2]. Robot swarms can perform complex tasks with distributed actions and have higher fault tolerance as any single failure of a certain robot in the group does not affect the performance of the entire swarm. The main drawback of robot swarms is that its performance is decreased due to interference of robots i.e. inter robot collision [14]. Algorithms like path planning, foraging, pattern formation, etc. are used to help robot swarms perform complex tasks.

Foraging algorithms for robot swarms are designed for search or exploration problems and are used as a benchmark for swarm robotics performance evaluation [15]. In the foraging algorithms, the robot swarm searches a particular area for certain resources. When the searched resources are required to be transported to a central location, these algorithms are called central foraging algorithms. Some of the applications for foraging algorithms are planetary exploration and crop harvesting [16, 17]. The performance of foraging algorithms is usually defined by the collection of resources and the duration needed to collect those resources. Though robot swarms are efficient for foraging tasks, the performance of foraging tasks is significantly affected by the swarm size. Previous research [16, 18] has showed that the performance of foraging algorithms have reduced when the size of swarm exceeds a certain number of robots. This decrease in the performance of foraging algorithms is associated with interference of robots in the swarm. The robots either miss detection of the resources or take more time to collect the resource because of inter robot collision.

Path Planning problems are critical for robot swarms in order to increase the performance and duration of the foraging task with the increase in the count of robots in the swarm. Path Planning finds a collision free path for a robot whose initial and target position are known. One approach for path planning is to plan the paths of every robot in swarm independently and then coordinate their paths. Another approach is to sequentially plan the path based on the priority of the robot in the swarm [19].

In this thesis, path planning and collision avoidance is integrated with an existing central place foraging algorithm to improve its performance. Distributed Deterministic Search Algorithm (DDSA) is a central foraging algorithm that generalizes the spiral search pattern for robot swarms of any number of robots [16]. The DDSA has three characteristics: 1) it is simple and deterministic; 2) it collects all the resources closer to the nest i.e. central location first; 3) it achieves complete coverage minimum resampling of the search area in error free case. The DDSA first generates the spiral pattern for each robot in the swarm. The robots search resources in spiral path starting from the central location. A robot goes to the nest to deposit the found resource and returns back to the spiral position once the collected resource is deposited. When robots leave their spiral position and go towards the nest or vice versa, there may be cases where multiple robot paths intersect or are collinear. DDSA uses reactive inter robot collision avoidance wherein robots slightly turn off their course to avoid each other and retry to get on their original course.This increases the time required to collect the resources by robot swarm [16].

This thesis provides a technique to improve the collection of resources in a given duration and to reduce the inter robot collision for the DDSA. The path of a robot that goes to the nest to deposit collected resources or the path of the robot returning to its spiral search position after depositing the resource is checked with other robots. Two robots can collide if one of them is in spiral and another is going to the nest to deposit the collected resource or is coming back from the nest to search position after depositing the collected resource. In this case, the robot farther from the collision point is stopped and the other robot is allowed to pass avoiding the collision. The robot going to the nest to deposit the collected resource traverses a triangular path to reach the original search position in the search spiral. This allows multiple robots to efficiently collect and deposit resources from the cluster reducing the collisions and managing the collection time. This thesis generalizes the path coordination to get a consistent path for every robot in the swarm and can be used for swarms of any number of robots. Checking for inter-robot collision allows the robot swarm to efficiently collect resources in lesser time as compared to the DDSA. The robots need to communicate their current and target locations with their neighbors to perform this collision check. The thesis limits the requirement of robots having global knowledge to local interactions by defining the radius for communicating with other robots and also demonstrates that the performance of DDSA is improved significantly by the planning techniques. The path planning and collision avoidance for cluster resource distribution techniques are tested on various cluster locations in the arena with different swarm size ranging from 3 robots to 15 robots and shows to be an efficient planning and collision avoidance technique .

The main contributions of this thesis are two-fold:

1. Develop a path planning and collision avoidance technique for multi-robot system. This helps reduce the physical interference of robots with coincident paths using spatial delay and intersecting paths adding time delay.

2. Evaluate the performance of the multi-robot foraging system in terms of target collection rate and average collision rate on a single 8 x 8 cluster using DDSA and ARGoS swarm simulator.

The rest of this document is structured as follows: In Chapter 2, the related works are presented. In Chapter 3, the method for multi-robot path planning and collision avoidance for clustered resource distribution problem is proposed. Tests, results, main conclusions and perspectives of future work are shared in Chapter 4.

2 Background

This chapter is organized into different sub-sections that introduce the research domains used in this thesis. The first sub-section gives a brief information and history of Swarm Intelligence and Swarm Robotics. The benchmark problem area studied for swarm robotics is central place foraging algorithm. The second sub-section gives an insight into the problem domain of central place foraging and its use to understand the swarm behavior. The main problem affecting the performance of central place foraging algorithms is interference among the robots. Collision avoidance techniques and path planning are mainly used in order to reduce the collisions and improve the performance of swarms. The third sub-section discusses the path planning techniques and different coordination methods used in swarm robotics. The last sub-section of this chapter concentrates on the researches similar to the approach used in this thesis.

2.1 Swarm Intelligence and Swarm Robotics

Current day applications like remote surgery, web enabled digital appliances, sensor networks and orbiting satellites need complex communication systems that have numerous interacting entities. There is a need for new approaches in network management and controlling these complex systems. One promising approach was Swarm Intelligence. Swarm Intelligence represents an idea that the complex system containing interacting entities can work together with minimal control on these entities and their interactions. Swarm Intelligence is based on the observation of social organisms such as ants or bees. Ants or bees conduct complex tasks like searching for food sources in an organized manner with simple rules and display collective survival. There is still research ongoing on how these small creatures show collective intelligence with low brain power and interactions to exhibit a global purpose [20].

The concept of group of simple agents solving optimization problems on graphs, lattices and networks was present in early days before the term "Swarm Intelligence" was defined [3]. Butrimenko [21] applied this idea to the field of telecommunications, Stefanyuk [22] to coordination of multiple radio stations, Tsetlin [23] to biologically inspired automata in random environment displaying collective behavior through their interaction. Rabin [24] introduced randomized algorithms to solve concurrent coordination choice problem for multiple processes. Swarm Intelligence is a field that has been evolving for 25 years. The term "Swarm Intelligence" was used in context of Cellular Robotics by Gerardo Beni and Jing Wang in 1989 to define intelligent behavior of multiple interacting agents in n-dimensional space to form pattern by interacting with neighbors [3].

With the spread of Swarm Intelligence (SI) term over the years, it has gained a broader meaning and holds any concept of collective behavior [20]. Gerardo Beni in [20] presents broader SI notion as:

The intuitive notion of "swarm intelligence" is that of a "swarm" of agents (biological or artificial) which, without central control, collectively (and *only collectively) carry out (unknowingly, and in a somewhat-random way) tasks normally requiring some form of "intelligence".* However, this definition only partially explains the widespread SI term and does not define all types of collective behavior for simple agents. The characteristics of SI by listing the advantages of "swarms" over centralized systems [2, 20] are as follows:

1. Economical: The swarm components are mass producible, modular, interchangeable and disposable as they are structurally same and simple.

2. Reliable: The performance of swarm is not affected much due to damage of few components as the components are same.

3. Scalable: The swarm can adapt to different population size without a major software/ hardware change.

4. Parallel: Swarm can efficiently perform search tasks with multiple targets distributed in vast area.

5. Energy Efficient: Simple smaller multiple robots save lot energy and the life time of swarm is more as compared to single robot.

There is no widely accepted definition of SI or mathematical model to define it. Many terms with varied definitions have been associated with SI applications: emergent behavior, self-organized behavior and collective intelligence. These lack of definitions and mathematical models cause difficulties in using the SI concept to its full potential. SI concepts can be applied practically when the efficient behavior in social colonies can be characterized by optimized mathematical models combined with structural framework that requires consistent behavior of every swarm entity and a concrete way of defining the problem statement to focus on a particular research area [20].

Different fields of science and technology deal with the idea of Swarm Intelligence: robotics, artificial intelligence, computation, economics, etc. Three broad areas of SI are: scientific interest versus technological interest, standard mathematics versus cellular computational mathematics and synchronous operation versus asynchronous operation [25].

The popular scientific and technological interest in SI was by Beni and Wang with the study of social insects and design of distributed robotics system in 1989 [25]. In 1999 Bonabeau et al. [3] dealt with the scientific and technological interest in parallel. The first pioneering biological study was "double bridge" in 1989 by Goss et al. [26] about the foraging behavior of ants and how they choose the shortest path between their nest and food source. The experiment describes that given two branches: one shorter and other longer, the ants randomly choose the branch to the food source from the nest and while returning back to the nest most of the ants choose the shorter branch. The ants find the best path by laying pheromones - a chemical through which they all communicate and use stigmergy (modification of the environment). This study showed the collective intelligence and self organization of ant colonies. Later Dorigo et al. [27] proposed a distributed problem solving and optimization based on "artificial ants" to solve complex problems using ant colony behavior. They developed three instantiations based on behavior of ants and demonstrated it on traveling salesman problem to solve optimization problems. Many other bio-inspired algorithms were developed to solve swarm optimization problems: group movements of flocks of birds and schools of fish were demonstrated by groups of agents called "boids" [28], algorithms inspired from wasps [29] and termites [8]. Bee algorithms have gained attention since last decade [30, 25].

Swarm Robotics is the application of SI to a group of robots. There has been lot of research in this field since 1980 [31]. In 1986, R.Brooks proposed a robust behavioral architecture with multiple layers for mobile robot control system that did not need central control module [32]. In 1993, Matarić discussed the implementation of swarm robots called "Nerd herd", an approach to understand group behavior through simple interaction between agents [33]. Simultaneously there were different research on hardware level for the robots in swarm robotics. The use of numerous small, cheaper and simple gnat robots in place of a single robot for incorporating parallelism is described in [34] and this approach can be applied for tasks requiring flying, swimming or crawling. Similarly, [35] proposes a new type of autonomous robotic units called "Cellular Robotic Systems" to solve complex tasks collectively using algorithms for distributed robotic systems.

There are various classification methods used for swarm robotics. In 1993, Dudek et al. [36] classified swarm robotics based on five areas: swarm size, communication range, communication topology, communication bandwidth, swarm reconfigurability and swarm unit processing ability. Cao et al. [37] presented classification of cooperative robotics in five areas: group architecture, resource conflicts, origins of cooperation, learning and geometric problems. Luca Iocchi et al. [38] presented the classification of multi-robot systems on the basis of reactive and social behavior. Lynne provided classification based on different research areas [31]. Brambilla et al classified swarm robotics from engineering perspective and real world applications [39].

In 2005, a swarm robotics application called ANTS (autonomic nanotechnology swarm) project by NASA gained media attention [40]. This project used "nanobots" a swarm of autonomous microscopic robots for space exploration tasks. Also in 2005, the European Union sponsored a swarm robotics project exhibiting autonomous self-assembly forming organized structures, obstacle avoidance and transport tasks with group cooperation techniques using new types of robots called "s-bots" was completed. This project was later continued as "Swarmanoid" project. The main objective was to develop distributed robot system of small heterogeneous, autonomous and dynamically connected robots. The project built around 60 robots of three types: eye-bots, hand-bots, and foot-bots. This project was completed in 2011 [41]. The "Kilobot" project demonstrated the largest swarm of 1024 self organized autonomous robots in 2014 [25].

There are various fields of research such as multi-robot systems, multi-agent systems and sensor networks inspired from swarm behavior that are often confused with swarm robotics. Though these research fields use cooperative behavior among multiple entities to accomplish a special task, there are fundamental differences associated with population size, homogeneity, control and application areas [2].

The differences between different systems is illustrated in Table 1.

	Swarm Robotics	Multi-robot system	Sensor network	Multi-agent system
Population size	Large range	Small	Fixed	Small range
Control	Decentralized	Centralized or Remote	Centralized or Remote	Centralized or Hierarchical
Homogeneity	Homogeneous	Mostly Heterogeneous	Homogeneous	Homogeneous or Heterogeneous
Flexibility	High	Low	Low	Medium
Scalability	High	Low	Medium	Medium
Environment	Unknown	Known or unknown	Known	Known
Motion	Yes	Yes	No	Rare
Typical applications	Post-disaster relief Military application Dangerous application	Transportation Sensing Robot football	Surveillance Medical care Environmental protection	Net resources management Distributed control

Table 1: Comparison of Different Systems [2]

The study of swarm robotics requires testing and observing the performance of developed algorithms on large number of robots. As it is difficult to afford the physical robots, computer simulators are used by researchers for their research. Simulators are easy to setup, less expensive and convenient to test the algorithms. Different simulators platforms like Gazebo, Player/Stage, ARGoS and UberSim are popular for swarm robotics research.

Swarm robotics is a fairly a new research field and although many algorithms have been proposed in this field, it is far from practical applications. Lack of benchmark test, wide range of problem definition, less experience in working with swarm robotics and simple algorithms have slowed the progress in this field. Limited computing and sensing do not fully allow the current swarm robotics system to demonstrate collective behavior. Ongoing research aims to enhance mathematical models to take advantage of the swarm potential for everyday applications. Swarm robotics is still confined to research field due to the cost involved in manufacturing of the hardware and difficulties in designing an efficient robot with sensors, actuators and electronic components to execute the cooperative algorithms. However, new developments in the electro-mechanical field and the design of efficient cooperative algorithms using knowledge of biology and swarm intelligence is encouraging the application of swarm intelligence to solve real world problems [2].

2.2 Central Place Foraging

Swarm robotics has many potential applications, however, swarm robotics being scalable and robust have never been used to solve real world problems due to lack of proper models defining swarm behavior. The current focus of research in swarm robotics is acquiring the desired collective behavior and understanding the properties of swarm behavior. Researchers test swarm robotics algorithms for a particular application on a simplified testbed to avoid the complications and problems arising in real world applications. Foraging is one of the popular testbeds for swarm robotics systems [39, 37].

In foraging tasks, robots will search for "prey" or "food" scattered in the environment and bring them back to the "nest" [37]. When the targets or "food" objects are transported to a single collection point, it becomes a central place foraging task. These foraging tasks can be conceptualized to real world complex tasks such as search and rescue tasks. Foraging tasks are also studied to understand the effect of interference on swarm of robots caused due to the competition of space between the robots [42]. The foraging tasks are especially used as testbed for group exploration wherein robots cooperate to explore and navigate through an environment, group transport in which robots cooperate to transport a heavy object that is heavy for a single robot to move and collective decision-making wherein robots interact with each other to produce complex behavior [39, 43].

Foraging is a benchmark problem in robotics for the following reasons [15]:

1. It integrates generic class of problems like navigation, object manipulation and transportation, object identification and exploration.

2. It is a basic problem for the study of robot-robot interaction and cooperation.

3. It can be used to solve many real world applications like planetary exploration, mining and harvesting.

4. Observing and understanding the efficient foraging behavior in social insects can provide inspiration and models for artificial systems.

Finite state machines can be used to model robot behavior using a fixed number of states. Each state defines a particular behavior or action of the robot and the robot can transition through any of the states based on a trigger of some external or internal event. The robot can be only in one state at any given time [15]. Figure 1 referred from [15] shows the four states of basic foraging task [15]:

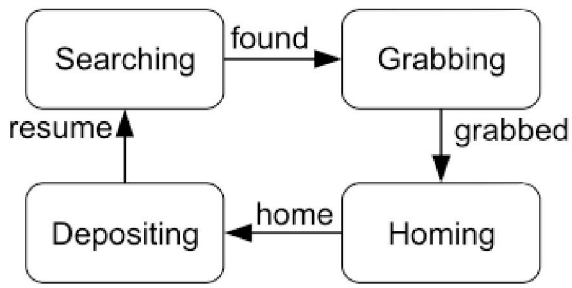

Figure 1: Basic Foraging Finite State Machine

- Searching: The robot moves in the search space to locate targets using sensors in this state. The robots can wander at random or move in some pattern to search for targets. This is the default state for the robots. The robot changes its state to “Grabbing” if it finds the target else it remains in the “Searching” state.

- Grabbing: In this state, the robot grabs the target to transport it to the “nest” or “home” location. The assumption is that the target is small enough to be grabbed by a single robot. Some targets need collective transportation by more than one robot. Once the target is grabbed, robot changes its state to “Homing”.

- Homing: In this state, the robot moves towards the “nest” or “home” location to deposit the collected target. Going to home for the robot involves determining the position of the "nest" relative to its current position, orientation towards the "nest" and navigating to the "nest". When the robot reaches the "nest", it changes its state to "Depositing". Several methods are used by the robot to reach the "nest". For instances, odometry is used to trace the robots path to the "home", following a marked trail or identifying the "home" location using a beacon.

- Depositing: In this state, the robot deposits the collected target to "nest" and changes its state to "Searching" to resume the search task. The robot can directly resume search on deposition of resource or use site fidelity to resume the search operation to find out if there are more resources near that location. Site fidelity is remembering the location of previously found resource.

The behavioral design of the autonomous robots can be achieved by decomposing the control system based on task achieving behaviors. The control system of autonomous robots performing complex task like trying to reach a particular place in minimal time should actively respond to the high priority goals like finding the shortest path along with servicing the low level goals like processing the input of sensor data and providing output to actuators for instance, avoiding a obstacle detected in the robot path by turning in random direction [32, 44].

The primary foraging models are stochastic or deterministic. In stochastic foraging, robots go in a direction for some time and if no resource is found, then a new random direction is chosen. This technique is useful in dispersing the robots with satisfactory area coverage. However, it takes time to search for resources that are far from the "nest" location once the resources closer to the "nest" location are collected. In stochastic models, the parameters of the environment like encounter rate (target detection rate) may change unpredictably. Time and efforts have to be invested in sampling of the environment. Conversely, deterministic models assume that the "forager" has some knowledge about the characteristics of the environment in some cases. Deterministic approaches follow a predetermined pattern to have maximum area coverage in minimal time and avoid resampling the same area [45, 16, 44]. Animals forage their prey according to rules that can be best expressed by probability. Thus, stochastic models resemble nature more closely than the deterministic models. However, there are many deterministic optimal foraging behavior models because they are most of the times simpler than the stochastic models and most of them can be converted to stochastic models easily by representation of random variable by their mean values [45]. The thesis uses a deterministic foraging approach as the DDSA which is also a deterministic foraging model is used as baseline for comparing the performance of this thesis. Deterministic models offer better comparison of the foraging models as they are independent of the hardware or simulator in which they are implemented.

Researches have focused on searching techniques by simplifying the resource recognition and gathering process. When sensors on the robot detect a resource, they have to carry it to the home location by grasping it using a gripper. In some cases, the robot just needs to capture the picture of detected resource [44, 15]. Task partition is used in foraging for activities related to transportation. It is a division of tasks into sub-tasks. There are several advantages of task partitioning such as physical separation of robots which reduces robot interference [46, 47] and competition for new resources [46]. It also allows the sub-tasks to be allotted to workers that are better suited for the task [46, 48]. In foraging, task partitioning is implemented by limiting arena size in which every robot operates. Resources are delivered to the target location with the help of several robots either by handing over the resource or depositing it on the ground [46]. This helps distribute the amount of work with slight surprising delivery time as observed in CPFA and preventing the error prone worker to hold the resource indefinitely [18, 44].

The task partition approach has two limitations. First, it mostly depends on specific setup for which it is designed. For instance, it becomes difficult to partition tasks in the case where you need to capture images of the resource. Secondly, its scope becomes narrow assuming that interference of robots is the only way to partition the task. In the cases where reducing the interference is not important, task partitioning can involve overhead costs as it may require multiple time picking up of resource using grippers [46]. In few approaches, recruitment process is used to notify other workers to pickup the detected object if the current worker cannot pickup the object. One of the early recruitment procedures using a light beacon to attract workers that can pick up the resource to the worker that has found the resource is proposed in [49]. Sugawara and Sano [49] demonstrated that their recruitment strategy makes the robots perform worse when resources are distributed uniformly and performs well in cluster resource distribution. This result is similar to the result of CPFA [18, 16, 44] but is reversed in case of DDSA [16].

Foraging is studied as it provides useful framework for probing into design and implementation issues for multi-robot systems. The foraging task introduces concepts of Parallelism and Robustness. Multiple robots working together simultaneously may complete the task faster and the performance of the group is not affected by failure of a robot. However, team of robots working together introduces problems like interference. The robot-robot interference and collision avoidance affect the performance of the group [42]. The performance in terms of number of targets collected in given time interval of multi robot foraging system does not increase monotonically with the increase in the group size because of collisions between robots [15].

The emphasis of foraging of social insects can be observed on the design of controllers for multi-robot systems. The study of foraging task offers many advantages for collective robot groups:

1. Distributed control mechanism.

2. Scalability: every robot has the same controller irrespective of its size.

3. Flexibility: robots can be added or removed easily without major design

change and hampering the performance of system.

4. local sensing: collective behavior can be achieved in multi-robot systems through local interactions

5. Adaptability: robots through simple learning can operate in uncertain and hostile environment [42, 50].

Multi-robot foraging systems are stochastic non-linear dynamic systems [15]. Therefore, it is challenging to develop mathematical models confidently stating the correctness of developed algorithms. Mathematical models help to analyze the whole parameter space and optimal parameters. However, experiments of multi-foraging using computer simulation or real-robots without any mathematical models limit the analysis of the parameter space, which, in turn makes it difficult to prove its correctness in real world applications [15]. Additionally, foraging algorithms tend to have performance variations, which are dependent on the hardware or simulator on which they are implemented [16]. A key aspect of multi-robot foraging system dynamics is also the interference due to over crowding of the robots and competition for targets [15].

To apply multi-robot foraging algorithms to real world applications there is a strong need of validation for safety and dependability that can be provided by mathematical modeling [15]. Central place foraging algorithms do not have standard guidelines or criteria for comparison and hence, it is difficult to compare the foraging algorithms across systems and prove their effectiveness. The performance of central place foraging algorithms can be affected by the means of distribution of resources [16, 18]. Apart from distribution type, the arrangement of resources also affects the foraging performance. The resource distributions commonly observed in literature are uniform, clustered or power law distributions. Usually for foraging algorithms, clustered or partially clustered distributions are used as naturally occurring resources are clustered at certain spots [16, 51, 50]. The robots are allocated equally to uniform target task, less so for partially clustered and least for clustered cases. The unequal allocation reduces performance of foraging. However, this performance can be increased by recruiting robots to collect targets from clusters. Also, the performance of foraging decreases near the clusters as the collision between robots increase. The placement and number of targets to collect also affect the foraging performance [16].

The performance of swarm foraging is also affected because of the scarcity of resources in the environment with the increase in the swarm size. Therefore, the foraging swarm performs efficiently for optimal swarm size [16, 18, 52, 44]. Liu et al [52] used regrowth of resources to replenish the resource supply and Pini et al [46] added the resource to same location once it is collected by the robot to improve the performance of foraging under complex and varied environmental conditions using effective interactive and emergent behavior.

Several aspects of central place foraging such as sharing of information, cooperation of robots and division of labor have been highly researched and demonstrated. However, to date there has been no forerunner research for real-world applications demonstrating the integration of self-organized cooperative search, object manipulation and transport in unknown or unstructured real-world environments for autonomous multi-robot foraging tasks [15]. The future direction for foraging would be to continue exploring new foraging algorithms which mimic the behavior of social insects and the application of these algorithms on foraging robots to tackle real world problems. This can be achieved by development of standard design and test procedures for analysis and designing of multi-robot foraging systems, standard criteria and quantitative benchmarks to evaluate and compare several foraging approaches and achieving safety, reliability and stability of multi-robot foraging systems [15]. DDSA algorithm proposes to be the baseline of comparison for all the central place foraging algorithms [16].

2.3 Multi-Robot Coordination

Social insect societies are the best example of distributed systems wherein emergent behavior is observed through interactions among individuals [53]. Multiple robots need to be moved over specific regions for exploration, meet at a certain common point or move in synchronization for many applications. These tasks need to be performed with minimal communications among different robots and with limited knowledge of the global state of the system [54]. Some of the most researched areas in multiple robot coordination are: path planning [53, 55], target tracking [53, 56], traffic control [53, 57] and formation generation [53, 58].

Popenoe defined collective behavior as [59]:

" Behaviour that occurs in response to a common influence or stimulus in relatively spontaneous, unpredictable, unstructured and unstable situations."

Collective behavior includes two types of behaviors, Cooperative and Competitive. Cooperative behavior is observed when robots interact with each other to complete a common goal, for instance, in foraging tasks. Competitive behavior is observed when robots compete against each other to fulfill their self-interest for example two player games [60]. Cooperation in multi robot systems can be achieved through communication amongst the robots. Multi robot path and motion planning is a cooperative type of problem. The thesis will focus on cooperative behavior.

When multiple robots arrive or want to use the same resource simultaneously, resource conflicts arise. There are three types of resource conflicts for multi robot systems communication medium, object manipulation, and sharing space. These conflicts can be solved with the help of coordinated robots. Communication can help robots to learn information from other robots in cooperation tasks. In explicit communication, robots need to have access over the communication media to share information with each other. This sharing of resource creates a bandwidth limitation conflict. Ye et al. [61] evaluated the communication strategies for wireless networked robots performing a resource transportation task. The research handled the bandwidth limitation problem by using a network simulator taking into consideration the protocol characteristics and propagation conditions [61]. Rybski et al. [62] experimented on a surveillance task performed by multiple robots and analyzed how limited communication bandwidth affects the performance and accomplishment of task. They built miniature robots called "Scouts" and developed a distributed software system with a novel scheduling mechanism to control them and maximize the use of limited resources [62, 63].

Resource conflict occurs when multiple robots need to manipulate an object together [60]. The most popular topic studied is cooperative"box pushing problem" by Matarić et al. in which two six legged autonomous robots need to transport a box together by interacting and taking turns to perform the task [64]. Kube and Bonabeau [65] demonstrated the formalized model of robotic implementation for cooperative transport in ants wherein ants change their alignment and position in-order to move a large prey towards the nest.

The third type of conflict is space sharing which is studied in terms of motion planning, collision and congestion avoidance [60]. Coordinating the independently planned paths for multiple robots to avoid congestions and deadlock conditions is presented in [66]. Whenever the distance between robots falls below a certain threshold, the robots monitor their trajectories by interacting with each other and may add a delay interval or plan an alternate trajectory to avoid collision. Similarly, [67] describes how robots traveling in opposite directions can avoid congestion by cooperating with each other and alerting the other members of the team about congestion risk. This thesis presents cooperative motion planning in foraging task for a robot swarm.

Coordination can be of two types [38, 68]:

1. Static: Usage of some kind of pre-defined protocol for performing the task. It is also known as offline or deliberative coordination. It can handle complex tasks but the real time handling of the task can be poor. For example, keeping sufficient space between robots, traffic control problems rules such as "stop at intersection" and "keep right".

2. Dynamic: Coordination accomplished during the task using analysis and information through communication. It is also known as reactive or online coordination. Dynamic coordination can have difficulty in handling complex tasks however, it meets real time handling. For example, the robot encountering an obstacle will try to avoid the obstacle without affecting the behavior of other team members.

Coordination of multi-robot systems use communication to share their position, environment state and sensor data with each other. Farinelli et al. [69] classified communication into direct and indirect communication. In direct communication, robots use some kind of hardware device to signal other robot team members. In indirect communication, robots use stigmergy for communication [69]. Cao et al. [37] classified communication into three types (i) Interaction via Environment: The environment itself is the communication medium, (ii)Interaction via Sensing: It refers to local interactions between robots as a result of sensing one another. It is often emulated using infrared or radio and, (iii) Explicit communication: It occurs either by direct or broadcast message usually by ethernet, wireless or other forms.

Decision-making in multi-robot coordinated systems requires an intellective process for selection of alternative scenarios for task accomplishment. Decision making can be centralized or decentralized. In centralized approaches, one robot acts as a coordinator and the accomplishment of the task is centered around that robot [53]. The central coordinator has the information about the environment and shares it with other robots through a communication medium. Centralized approaches are effective for smaller groups of robots and the performance is affected with uncertainties of dynamic environment or communication failures. If the central coordinator fails, then there must be another robot that can replace the coordinator to avoid failure of the whole system [60]. In decentralized approaches, there is no single coordinator. Each robot coordinates its own movements and plans to avoid collisions [53]. Decentralized approaches respond comparatively better to dynamic environments and are robust, flexible and scalable. However, this flexibility can result in suboptimal performance of the decentralized systems.Decentralized approaches can be divided into two types (i) Distributed: In this approach, each robot coordinates its own movements, (ii) Hierarchical: It is a hybrid between centralized and decentralized approach. It has one or more local coordinators which coordinates robots into clusters [60]. GOFER [70] is an example of centralized multi-robot system wherein there is a central scheduling task that has global knowledge of the tasks to be performed and the availability of the robots to perform the task [70, 60].

Similarly, in [71] for the task of object manipulation and transportation, the global motion planner decides when and where to manipulate the object. Luna and Bekris [72] provided an efficient centralized sequential "Push and Swap" path planning method for multiple robots operating on a discrete roadmap. An example of distributed path planning is described in [19]. This paper describes an approach using distributed prioritized planning wherein every robot plans its own path at the same time and then checks for collisions between paths. If collisions are observed, lower priority robots must replan their paths. Another example of decentralized planning is presented in [66] where each robot plans its path independently and then coordinate to avoid collision by altering their paths by communicating with their neighbors.

Motion planning involves producing a continuous obstacle-free path for a robot from start configuration to goal in a configuration space. It is an important topic in robotics research as robots accomplish tasks by moving in real world. Motion planning should consider the obstacles in environment along with inter-robot collisions while planning [73]. In environments with stationary obstacles and moving obstacles, path planning based on geometric configuration of environment returns an optimal path in polynomial time if it exists. However, motion planning with moving obstacles is NP-hard problem and non-solvable for two dimensions [1]. This makes motion planning a very difficult problem. The three major families of motion planning are summarized in Figure 2 [1].

The cell decomposition method decomposes a configuration space into

	Cell decomposition approach	Potential field approach	Roadmap approach	
			Voronoi diagram	Sampling-based method
Strength	optimal paths can be found	efficient and easy to implement	resulting paths tend to maximize clearance	efficient for high-dimensional problems
Weakness	strongly depends on the grid resolution of the world	easy to fall into local minima	inefficient and complex to implement	solutions often sub-optimal
Completeness	complete			probabilistically complete
Performance	computational complexity depends on the number of points			rate of convergence depends on the use of local planner
Main applied field	multi-robot area coverage	multi-robot formation control	multi-robot exploration	industrial manipulators

Figure 2: A Comparison of Classic Multi-robot Motion Planning Approaches [1]

contiguous areas called cells. This method is usually used in multi-robot area coverage problems. The goal is to provide an obstacle free path (sequence of obstacle free cells) from the starting point to the goal point [74, 60]. Guo et al. [1] proposed a D* decentralized path planning algorithm. Another example of a cell decomposition method is presented by Bennewitz et al. [75, 76]. They presented a path planning strategy based on the A* algorithm [77]. This approach is randomized and reorders robots to replan their paths till they have minimum path length.

The Potential Field (PF) approach generates a path by combining attraction towards goal with repulsion from obstacles [74, 60]. This approach is used for multi-robot control formation. The research in [78] provides an obstacle avoidance approach using "artificial potential field" for mobile robots and manipulators. In this approach the problem of collision avoidance that is generally considered to be a high-level planning problem is applied to different control level of manipulators. Another example of a potential field based approach is presented in [79]. The approach presents a potential navigation function that coordinates multiple agents in a particular formation (shape and orientation) avoiding collisions between them. The weakness of PF approaches is that it gets caught in local minima convergence or aimless oscillations in the case where obstacles are very close to each other [74, 80].

In roadmap approaches, motion planning is done through a roadmap which is set of collision- free paths (road network) from the start point to the end point [80]. Voronoi diagram is a roadmap approach which specifies all points equally spaced from the closest obstacle. In other words, Voroni diagram designs roads in a way to be as far as possible from obstacles [80, 60]. Voronoi diagram may not find the shortest path but it gives maximum clearance from obstacles. Bhattacharya et al. [81] proposed an approach using Voronoi diagram to obtain a path that is a close estimation of the shortest path fulfilling the threshold values specified by the user. Another research using Voronoi diagram is presented in [82] where a team of exploring robots is coordinated by using Voronoi diagram by segmentation of the environmental map to minimize the overall exploration time. Another roadmap approach is the probabilistic roadmap (PRM), where motion planning is achieved by randomly generating the collision free configurations and connecting some of them. This approach is widely used for robot arms in manufacturing and engineering fields [60]. A two phase path planning involving a learning and a query phase is presented in [83, 74]. This method based on PRM is used for static workspaces and high dimensional configuration spaces. A probabilistic roadmap is constructed and stored as a graph in learning phase wherein edges correspond to paths and nodes correspond to collision free configurations. In the query phase, start and goal configurations are connected to any two nodes of the generated roadmap and a path joining them is searched. Another roadmap approach is rapidly exploring random tree (RRT) which does motion planning as a tree search problem by constructing an incremental tree of configurations by adding a free space random configuration that is closely connected to already present configuration in the tree [60]. Single query path planning method for high dimensional configuration space is proposed in [84]. The method incrementally builds two randomly exploring trees rooted at start and goal configuration. Each trees explore the space around them and proceed towards each other using greedy heuristic approaches [84, 74].

Both RRT and PRM are sample based methods and are the new age motion planning methods for high dimensional or geometrically complex configurations. The main reason for this is because unlike cell decomposition and potential field methods, the running time of these methods does not grow in proportion with the dimension of the configurations and are easier to implement. Also, sampling methods sometimes fail to find a solution even if one exists [60].

The classical motion planning methods which were widely used in early research now have been replaced by heuristic approaches like genetic algorithm, fuzzy logic, artificial neural networks and wavelets [74, 80]. The reason being that heuristics are useful for dynamic environments and are close to the human way of behavior learning [74]. The existing motion planning that use both heuristic and classical methods have their pros and cons and combining multiple methods together may solve the complex requirements of multi robot systems like flexibility, scalability and reliability [60].

This thesis focuses on developing a collision-free path for robots avoiding the inter-robot collisions. These collisions can be described in terms of intersection points or collinear points. The aim of this thesis is to formulate an alternate path or stop a robot to avoid these intersections. This thesis proposes a decentralized collision avoidance method for multiple robots with coincident or intersecting paths.

2.4 Related Work

The swarm robotics problem domain and the solutions in this research field are inspired from nature. Many swarm robotics applications require multiple robots to detect and collect targets. Spiral search patterns for foraging are studied extensively and have been found to provide desirable performance [17, 16, 85]. They guarantee collection of nearest targets first and they have complete coverage of the area with minimum sampling [16]. This thesis incorporates the Distributed Deterministic Spiral Algorithm (DDSA) [16] for searching the space. We are using DDSA as the base and integrating our path planning and collision avoidance approach to better avoid the inter-robot collisions. The performance of this thesis is compared to the DDSA. The DDSA generalizes single robot square spiral to any number of robots. DDSA is a type of central place foraging algorithm used to test how the foraging performance scales with number of robots. Ryan and Hedrick proposed a square search pattern for fixed-wing unmanned aerial vehicle (UAV) for searching water targets which is similar to DDSA [86]. Similar approaches to the DDSA algorithm are defined in (i) Approach using parallel searching in the plane with fixed number of robots that are independent of the dimension of the plane [87], (ii) Distributed spiral search algorithm for odor localization problem as observed in ants [88, 89], (iii) Search pattern consisting of system of loops of ever increasing size centered about the origin with path integration as observed in Cataglyphis ants [90], (iv) Searching spiral by equally partitioning the environment among multiple robots [91], (v) "proof of concept" for circular distributed spiral search for multiple robots whose movements are coordinated using shared data structure [92] and, (vi) deterministic interlocking spiral starting from common point for multiple agents searching targets in coordination [93].

While designing distributed foraging algorithms for multiple robot systems, interference can be considered a pragmatic tool for evaluating the performance of these algorithms. Interference can be physical or non-physical. Physical interference is when robots compete for space while non-physical interference is when robots compete for sensory resources like sharing radio bandwidth [94]. This thesis concentrates on handling physical interference by using collision avoiding and path planning techniques.

Goldberg and Matarić [94] have evaluated the performance for multi robot systems using interference. They focus on calibrating the arbitrary behavior schemes and controllers based on inter-robot interference. They have presented three cases (i) Homogeneous implementation: multiple robots have similar behavior, are independent and act in parallel. In this case, multiple robots try to drop off the collected resources at the "home" region simultaneously which results in high amount of interference, (ii) Pack arbitration implementation: all robots have similar behavior but don't act independently and in parallel. There is some form of hierarchy implemented such that the robot with higher dominance drops off the resource first and then exits the nest and other lower dominant robots are not allowed to go towards the "home" region till the higher robot leaves the "boundary" region. This makes only one robot to act at a time. Some form of communication is used to decide the dominance among the robots and, (iii) Case Arbitration implementation: not all robots have similar behavior and this differentiates robots into groups and divides the tasks among robots. Each of these groups work independently and in parallel. The goal is to reduce inter-robot interference by properly assigning tasks to robots. It is observed that homogeneous implementation has maximum interference near the "home" region, caste arbitration has distributed but substantial interference and pack arbitration has the least amount of interference. However, homogeneous task completion time is shortest, then caste arbitration and then pack arbitration implementation. Thus, it depends on the task that decides what should be the priority: time or interference reduction. It was observed that caste arbitration is not the satisfactory implementation in either of the case [94].

The mathematical formulation for multi-robot task allocation with deadlines considering the effect of interference is formulated in [95]. The research models interference as linear function and studies how interference affects the performance of task allocation in multiple robots. The optimal solution is obtained by solving the linear integration function. Similarly, [42] presents a mathematical model of homogeneous foraging robots with the goal of understanding the effects of inter-robot collision on their performance. The paper studies two foraging cases: The first case, where homogeneous robots only collect objects and second case where the homogeneous robots find and deposit the object at a predefined "home" location. It is observed that in the first case, the foraging performance improves with the swarm size. However, the performance is sub-linear and interference causes the performance of individual robot to decrease. In the second case, it is found that the performance is maximized for an optimal swarm size but it decreases again with the increase above the optimal swarm size. Again, inter-robot collision causes the individual robot's performance to be monotonically decreasing function of swarm size. The experimental parameters decide the optimal swarm size. This optimal swarm size value is smaller if the robots have a longer maneuver time to avoid the obstacles [42].

Path planning and collision avoidance methods need to be developed such that multi-robot system avoid obstacles and effectively perform the task. Typically, path planning is divided into global planning for searching the configuration space from the start location to the goal location and local planning to avoid static and dynamic obstacles. The paper [96] provides a decentralized navigation algorithm for a workspace shared by humans and robots. The algorithm is based on a velocity obstacle paradigm which is a geometric representation of all the velocities that will result in a collision. This paper uses different cost maps and sampling with different cost factors accounting for humans and robots sharing the same workspace. Proper velocity is selected to avoid collision of a robot with other robots and humans. The paper [97] presents a "step forward approach" for avoiding obstacles using the omni-directional vision systems, automatic control and dynamic programming. The algorithm uses priority to avoid collision such that the lower priority robot reduces velocity or stops to allow the higher priority robot to pass. Motion of robots is predicted using prediction module before making the decision.

To improve the performance of foraging algorithms that are limited by physical interference of robots in multi robot systems, [98] proposes a "bucket brigading" strategy, which needs every robot to focus on specific region of the arena. The robot detecting a resource in the region will transport it to the neighboring sub-region in the direction of "home" location. This reduces overcrowding and physical interference.

Distributed prioritized path planning is used in order to avoid collision of intersecting robots. The robots in a team are assigned priorities based on the obstacle map and robots decide their own static priority as a function of initial path estimation and local information. The robots sequentially compute their path based on the priorities and assign their path to the obstacle map so that the next robot can plan its path around that map. For a robot team of n robots, the path planning of collision free paths is complete [99].

The approach in [19] uses the prioritized planning mentioned above to plan the paths and additionally uses a coordination module based on Artificial Bee Colony (ABC) algorithm to find a collision free path by generating consistent velocity profiles by considering higher priority robots as dynamic obstacles. Each robot predicts the position of other robots based on the information sent in the previous iteration. If the path is collision free then the robot keeps the profile, else generates a new consistent profile and sends it to lower priority robots. The paper demonstrates this method on the "corridor problem" scenario where 3 robots have intersecting paths to reach their goal. The higher priority robots traverses the path without any waiting time, the next higher priority robot updates its path to be consistent with the higher priority robot and the lowest priority robot updates its profile and waits for other two higher priority robots to cross the corridor. The above mentioned solutions don't consider overlapping paths. In [100], the authors propose a collision avoidance technique for coincident and intersecting paths through fictitious points. The introduction of fictitious points ensures that the robots don't pass over the common stretch of the path simultaneously. The nonlinear formulation helps speed restrictions and maintain proper time difference so that the robots don't collide on common paths.

The most similar approach to the work presented in this thesis is in [101]. The work uses "holding pattern" for depositing the detected target to the "nest" location. This is similar to the idea used at the airports to avoid congestion and collision of the airplanes. If the robots collect the resource from the same cluster and are close to each other, they take turns to go to the "home" location instead of all going together. The robots pick up a closest of four points around the "home" location forming larger triangular paths resulting in lesser collisions.

This thesis focuses on physical interference avoidance of the foraging robots. When two robots are likely to collide on intersecting paths, the robot farther from the colliding point is stopped by adding time delay. When multiple robots have collinear paths, a spatial delay is added to every robot depositing the resource at the "nest". Each robot traverses a longer alternate path forming a triangle to avoid the collision on coincident paths. This approach helps to reduce the interference caused at the "home" location when multiple robots simultaneously deposit the resources collected from the cluster.

3 Path Planning And Collision Avoidance for Clustered Central Place Foraging: PPCA-CCPFA

3.1 Problem Statement

The central place foraging task consists of multiple agents searching for desired targets in an unexplored environment and depositing the found targets at a central "home" location. The characteristics of the environment, swarm and resource distributions can be specified by the researcher in order to properly observe particular intended behavior of the foraging research topic. The foraging task requires integration of aspects like object manipulation, communication, localization, path planning and collision avoidance. Any constraint or problem observed in one of these aspects greatly affects the performance of the foraging task. Analysis, understanding and evaluation of foraging tasks can be efficiently carried out by focusing on a single behavior or combination of behaviors. The motivation of this thesis work is to focus on path planning and collision avoidance for the multi-robot central place foraging task. The specifications of the environment in terms of arena shape and size, swarm size and resource distribution will help in developing and evaluating the objective of the thesis work.

Resources are distributed spatially and temporally [102]. Resource distribution can be uniform, clustered or partially clustered. Usually the resources found are clumped in varying sizes. The distribution of the resources affect the performance of multi-robot central place foraging tasks. When the resources are uniformly distributed all the robots benefit. However, if the resources are distributed in clumped patches, robots spend great amount of time avoiding physical interference near the central "home" location and the clumped resource. The effect of interference is proportional to the swarm size. Thus, there is a need of path planning and collision avoidance technique to reduce the inter-robot collision and lower the time expense on the collision avoidance. This thesis focuses on developing a path planning and collision avoidance method for a robot swarm collecting resources from a cluster and depositing them at the central depot as resources are usually found in clumps.

The multi-robot path planning problem is defined as: given n robots with known initial point and goal point and working in the same space, finding path for each robots that is free of obstacles. Consider a workspace $W \in \mathbb{R}^2$. The robot position is given by $C(x, y)$ where x and y are the coordinates and the orientation is given by θ. The state space X is the cartesian product of all the robot configurations. A pair of robots i with orientation θ_i and j with orientation θ_j in collision is defined as X_{obs} such that $X_{obs}^{ij} = C(x_i, y_i) \cap C(x_j, y_j) \neq \emptyset$. Obstacle free state space is given by $X_{free} = X \setminus X_{obs}$. The goal of path planning is to find a sequence of motion from start position $S(x, y)$ to goal position $G(x, y)$ free of inter-robot collision. The sequence of motion consists of rotation and linear motion of the differential drive robot. The linear velocity is given by v and rotational velocity is given by ω.

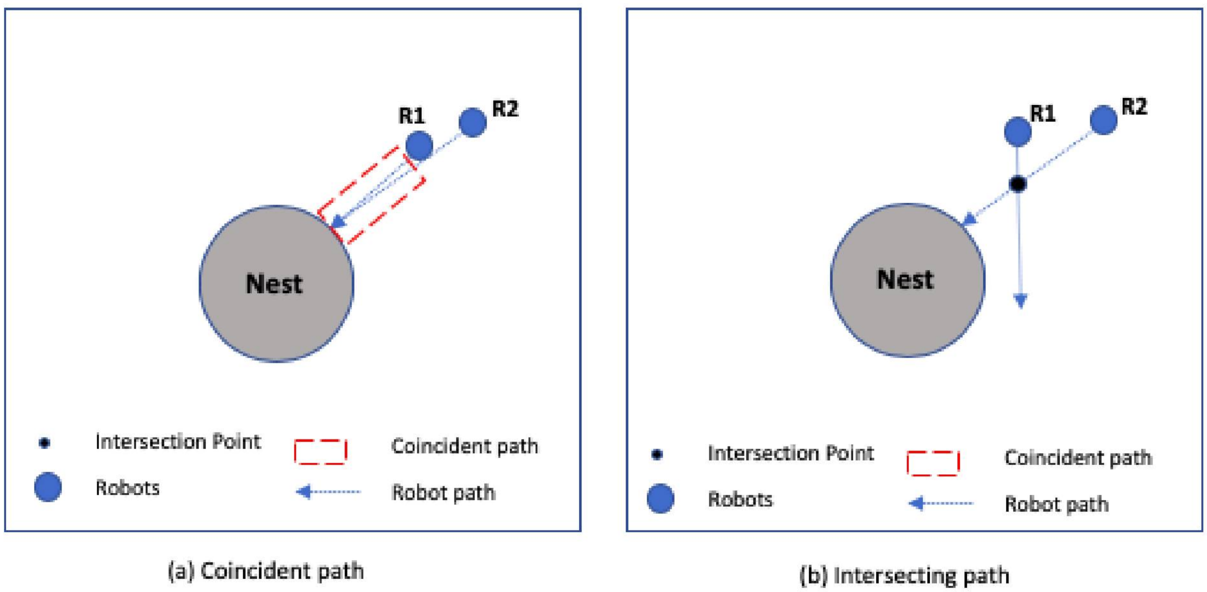

Figure 3: Inter robot collision cases

Given a swarm size n, the robots have to search the space for resources and deposit the collected resources at a central depot. There are chances of collisions when more than one robot travels from the nest location to the search position or vice versa. The colliding robots can be in the same, different or opposite direction. The collisions can be avoided by adjusting the robot speeds, adding a time delay or choosing an alternate path. The possibility of collision exists when: i) robot paths intersect at a particular point called intersection point; ii) robot paths are coincident and there are multiple points of collision. Figure 3 depicts the above mentioned cases.

There can be more than two robots having coincident paths (robot paths very close to each other or overlapping) when they simultaneously collect the resources from cluster and also when they return to their respective search positions from the "nest". In case(a) of Figure 3, collision is avoided by adding spatial delay so that there is a free passage for robots to travel. In case(b) of Figure 3, the path of the robot searching and the robot going to or coming from the "nest" may intersect at a particular point. The collision in this case is avoided by adding time delay.

3.2 DDSA algorithm

The central place foraging algorithm used in the thesis is distributed deterministic spiral algorithm (DDSA) [16]. Path planning and collision avoidance is integrated with the DDSA. The DDSA broadens a square spiral from one robot to any number of robots. The generated spiral path of the robot preserves the determinism and guarantees optimality [16]. There is no standard formulation for comparison of central place foraging algorithms. This makes evaluation of algorithms difficult. The DDSA is proposed as a point of comparison to solve this problem. The performance of systematic search strategies such as the DDSA reduces with the presence of error and with the increase in the size of swarm. The performance of the approach developed in this thesis is compared to the DDSA for a swarm size of 3 to 15 robots.

Figure 4 shows the spiral pattern of DDSA in the ARGoS simulator [16,

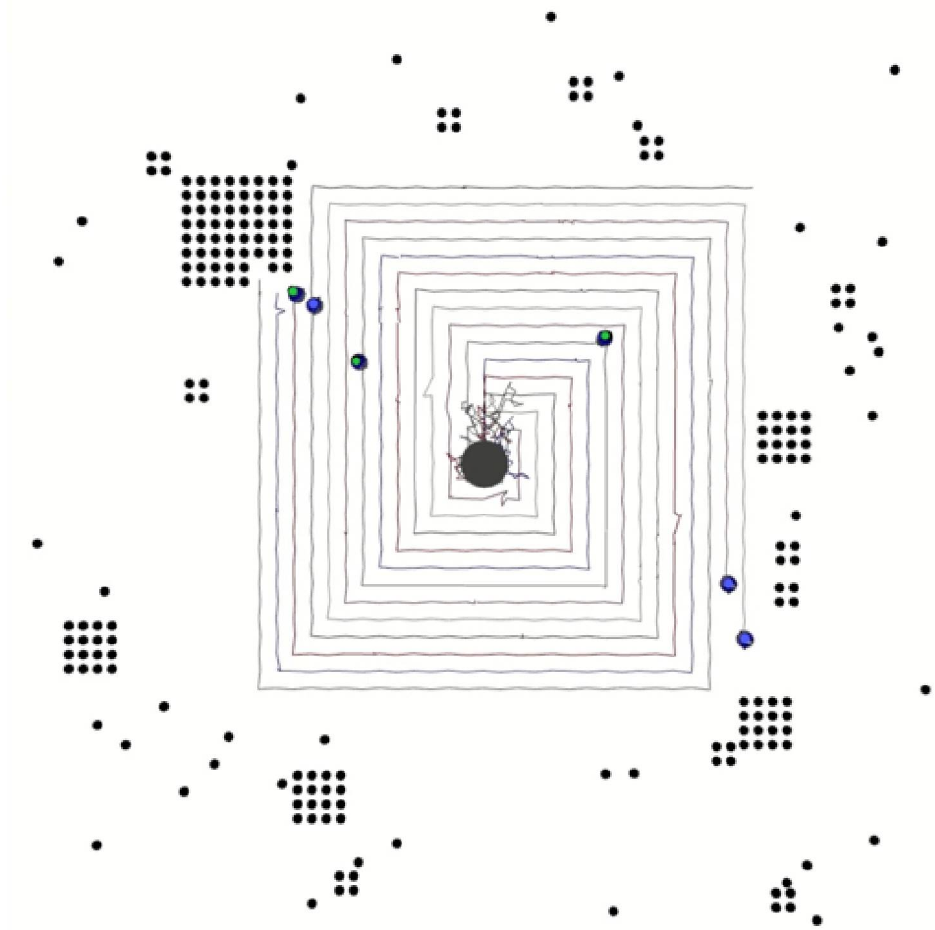

Figure 4: DDSA overhead view

103]. The Figure 4 is referred from [16]. The spiral patterns are the paths that robots traverse while searching the workspace for resources. The robots are shown with blue or green dots. Green robots are carrying the collected targets to nest while blue robots are searching the space for resources. The black dots represent the targets. The target distribution shown is a partially clustered distribution. The spiral pattern starts from the central collection point. The robots' path from the nest to the search position and vice versa are not displayed. The additional details of spiral generation and DDSA can be found in [16].

3.3 Proposed Path planning and Collision avoidance approach

In the case where the robot paths are coincident, the robots may collide along the path. When multiple robots collect resources from the cluster and travel towards the "nest" location a reactive collision method is used to avoid the collision. However, the reactive collision method is not effective and spends lot of time when the robots are traveling in opposite directions. When one robot deposits the resource and tries to return to the search position, its path is coincident or closer with other robots who have collected the resources from the cluster. In this case, a "waypoint" at a certain angle and distance from the nest is added in order to add space between the paths of robots and avoid the collisions near the "nest". Similarly, there is a chance of intersection of robot paths when one robot is searching for resources and another robot is traveling to the "nest" to deposit a resource or vice versa. In this case, the robot that is farther form the collision point is stopped such that it allows another robot to pass without collision.

The efficiency of the added "waypoint" is dependent on the angle between the two vectors i) path from the search position to the nest position;

ii) path from the nest position to the new "waypoint". Also the spatial distance added affects the efficiency of the added "waypoint". The "waypoint" added forms a triangular path along which the robot travels and helps add safety margin between the incoming robots to the "nest" and outgoing robots from the "nest". To select optimal angle and distance parameters of the new waypoint, the performance of the proposed algorithm is checked on different value combinations of waypoint angle and waypoint distance. These combinations are called different waypoint sets. For intersecting robot paths, the time delay interval is calculated using the time, displacement and velocity kinematics.

This thesis is tested on only one cluster of size 8 x 8 and can work for any single cluster scenario in the workspace. Any central place foraging approach can be used for searching the cluster. This thesis uses DDSA [16] as central place foraging algorithm. The path planning and collision avoidance approach is integrated with the DDSA to solve the congestion problem at the "nest" location.

Figure 5(a) displays the reactive collision avoidance approach for coincident paths in DDSA. Figure 5(b) shows the coincident case for PPCA-CCPFA. The robot travels a triangular path before returning back to its search position. The spiral search paths are not shown in the figure. The formulation of the collision avoidance and path planning can be described as follows:

- Coincident paths: The robots search the space for resources in spi-

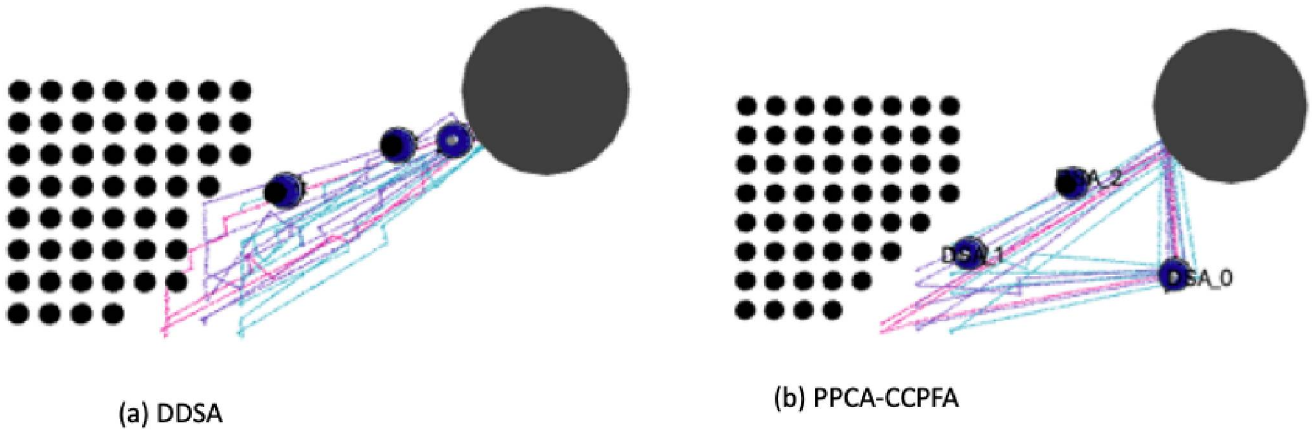

Figure 5: Comparison of Coincident Case in PPCA-CCPFA and DDSA

ral. If the robot encounters a cluster resource, it goes towards the central depot to deposit the resource. While going back to the search position, the robot goes to a "waypoint" calculated at a certain angle and distance from the position of the robot at the central location. The "waypoint" is always to the left of the path of the robot from its "nest" position to the search position. Adding the waypoint to left makes sure that robot does not get in the way of other robots collecting resources from the same cluster. Let $S(x_{si}, y_{si})$ be the search (start) position and $G(x_{gi}, y_{gi})$ be the goal position of the robot i. Let the distance of the waypoint from the nest be d and the angle of rotation be given by θ_i. The rotation angle is calculated with respect to the angle of vector $\overrightarrow{SG} = (x_{gi} - x_{si}, y_{gi} - y_{si})$. The center of the nest is given by the origin. The waypoint coordinates to be calculated $W(x_{wi}, y_{wi})$ is given by:

$$x_{wi} = d\cos\theta_i \tag{1}$$

$$y_{wi} = d\sin\theta_i \tag{2}$$

- Intersecting paths: The intersection of paths is possible for a robot searching in spiral path and another robot going to "nest" to deposit the collected resources or vice versa. The robot farther from the intersection or collision point is stopped. Let $S(x_{si}, y_{si})$ be the start and $G(x_{gi}, y_{gi})$ be the goal position of robot i. Let $S(x_{sj}, y_{sj})$ be the start and $G(x_{gj}, y_{gj})$ be the goal position of robot j. Let d_i be the distance of the robot j from the start point to the intersection point and similarly, let d_j be the distance of the robot j from the start point to the intersection point. The time to reach the collision points of respective robots can be easily calculated with the knowledge of robot linear and rotation velocities. Suppose t_i and t_j be the times to reach intersection points for the robots i and j

  The robot farther from the collision point can be found out by:

$$d_{farther} = \max(d_i, d_j) \tag{3}$$

  The time interval Δt added to stop the robot farther from the collision point is given by:

$$\Delta t = \min(t_i, t_j) \tag{4}$$

Algorithm 1 PPCA-CCPFA

```
▷ Distributed across robots
for all Robots i ← 0 to R do
    if Path Planning and Collision Avoidance Activated then
        if Target Collected or Target Deposited then
            FindNeighbors() ← N
            M ← ComputeCollisionMatrix();
            for all Robots i ← 0 to N do
                for all Robots j ← (i+1) to (N−1) do
                    if M_{i,j} == COINCIDENT then
                        SetWayPoint() ← W
                        W ← CalculateWayPoint();
                        if Robot going away from the nest then
                            Go to waypoint W
                        end if
                    else if M_{i,j} == INTERSECTING then
                        if Robot is farther from Collision Point then
                            Δt ← CalculateStopTime();
                            StopRobot();
                        end if
                    end if
                end for
            end for
        end if
    end if
end for
```

The PPCA-CCPFA approach is presented in Algorithm 1. For a robot swarm of size R, every robot that collects or deposits the target, finds its neighbors N within certain radius. Then, the robot computes an adjacency matrix for its neighbors indicating the values of No collision or Intersecting

Paths or Coincident Paths. Then, every element of the matrix is checked for the values. If the value is Coincident Paths and the robot is going to the nest, waypoint for robot is calculated based on the parameters of waypoint distance and waypoint angle. The waypoint is added only when the robot deposits the collected target. If the matrix element value is Intersecting paths, then a delay time is calculated based on the speed, distance and time to reach the intersection point. If the robot is farther from the intersection point than its neighbor, then a delay is added to the robot to avoid the collision.

The base algorithm for 1 is the DDSA described in [16]. Path planning and collision avoidance is checked every time the robot collects or deposits a resource. PPCA-CCPFA is independently distributed across every robot in the swarm. The robot that collects or deposits the resource first finds its neighbors within its specific radius. The scope of the thesis does not include methods of communication for finding the neighbors. A collision matrix is calculated for the robot which defines the type of collision of the robot with its neighbors No collision, Intersecting paths or Coincident paths. Depending on the type of collision, the path planning and collision avoidance technique is implemented. The approach in this thesis is integrated with the DDSA [16] and hence, the path planning and collision avoidance approach is activated once every robot in the swarm completes its one spiral of searching the workspace. This allows the proposed approach to function properly and efficiently.

The integration of PPCA-CCPFA with DDSA is shown in the Algorithm

Algorithm 2 Integration of PPCA-CCPFA with DDSA

```
▷ Distributed across robots
for all Robots i ← 0 to R do
    ▷ Create a spiral pattern to follow and store it
    for c ← 0 to N Circuits do
        Q.enqueue ( ⟨0, _gD_N( i, c, R) ⟩)
        Q.enqueue ( ⟨_gD_E( i, c, R) , 0⟩)
        Q.enqueue ( ⟨0, −_gD_S( i, c, R) ⟩)
        Q.enqueue ( ⟨−_gD_W( i, c, R) , 0⟩)
    end for
    ▷ Start at collection point and perform spiral
    while ¬Q.empty() do
        if Check if First circuit completed then
            Activate PPCA-CCPFA
        end if
        w ← s + Q.dequeue()
        Move toward w
        if target found at current location s then
            if PPCA-CCPFA Activated then
                PPCA-CCPFA Check
            end if
            Return to collection point with target
            if at collection point then
                Deposit target
                if PPCA-CCPFA Activated then
                    PPCA-CCPFA Check
                end if
                Return to location s
            end if
        end if
    end while
end for
```

2. The base algorithm for the DDSA is referred from [16]. The blue pseudo-code is the part of PPCA-CCPFA . The black pseudo code is part of DDSA and the details can be found in [16].

Reactive obstacle collision avoidance is achieved through proximity sensors. The robots turn either right or left depending on the direction in which other robot is detected. In the Coincident path case of PPCA-CCPFA, when multiple robots collect targets or deposit targets simultaneously, the robot that is farther from the central "home" location is stopped to insert a small delay. Readings of the proximity sensor can be used to check if the current obstacle detected is in the direction of the robot. Depending on the direction, a small delay is inserted. This delay helps maintain some safe distance between robots and avoids congestion.

4 Simulation Results

4.1 Setup

The problem domain of the central place foraging task should work efficiently over different resource distributions in the workspace. In this thesis, the focus is on clustered resource distribution inspired by the way resources are found in the real world. Also the collisions around the "nest" are more as multiple robots collect resources from the cluster simultaneously. To evaluate the proposed PPCA-CCPFA approach, a cluster distribution with single cluster of size 8 x 8. is used The performance of this approach is evaluated by measuring the target collection rate and average collision rate for multiple random locations of the cluster in the workspace.

The simulator used for this thesis is the ARGoS simulator [103]. ARGoS simulator is a multi physics robot simulator. The simulator provides high accuracy (close similarity to real environments), high flexibility (supports heterogeneous robots) and high efficiency (optimized computational resources to provide shortest simulation run time possible). ARGoS can simulate complex environments with a large heterogeneous robot swarm.

In the experiments, 64 resources in the form of single 8 x 8 cluster are randomly placed in the square arena space of $100\ m^2$. All experiments run for 30 minutes. The performance of DDSA [16] and the proposed approach are compared on 10 random locations of clusters in the arena. Update cycle

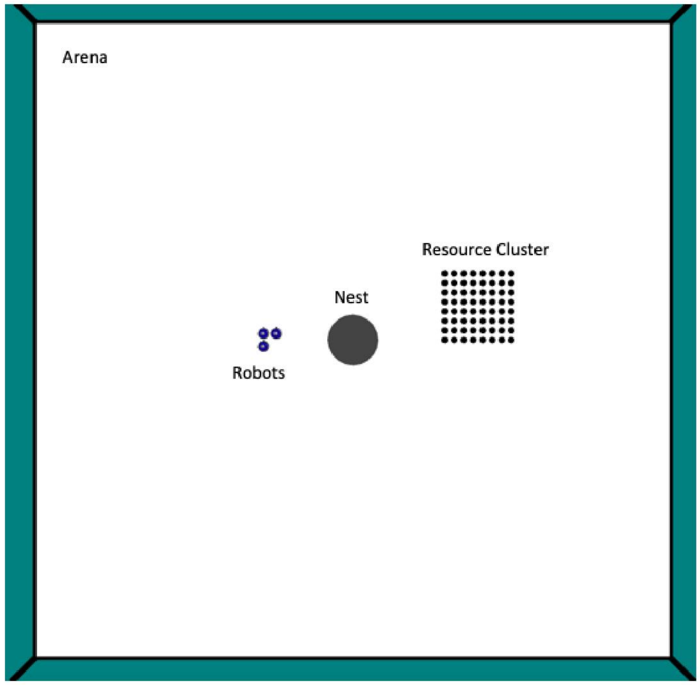

Figure 6: Initial ARGoS configuration of PPCA-CCPFA and DDSA

of 480 per second for 2D physics solver in ARGoS simulator is used for the experiments. The robots simulated have parameters similar to the physical iAnt robots [16, 18]. The simulation setup is similar to the DDSA [16] so that it is easier to compare its performance with that of the thesis. To simulate the robot hardware, the robot has $8\ cm$ radius with a camera facing downward to detect the resources. The resources have a radius of $5\ cm$. The gap between the spirals is $13\ cm$ [16]. The robot has a forward speed of $8\ cms^{-1}$ and a rotation rate of $10\ cms^{-1}$ approximately equal to $1.25\ rads^{-1}$. The "nest" radius is $4\ cm$ and it is assumed that the "nest" is represented by a beacon [16, 18] . The robots move $8\ cms^{-1}$ towards their goal between reorientations [16, 104]. There are no static obstacles in the arena. The parameters for the PPCA-CCPFA approach: location of the cluster, the distance of “waypoint” and angle of the “waypoint” along with the environment and robot parameters mentioned above can be configured in the ARGoS simulator. Figure 6 shows the initial configuration with 3 robots and one 8 x 8 cluster placed at a location. The elements in the Figure 6consists of central gray area represents the nest, single 8 x 8 cluster of resources represented by black dots, white space is the arena with green walls and the three blue dots represent the robots. The performance of each approach is evaluated at 10 different locations of the cluster for swarm size of 3 to 15 robots.

4.2 Results

The performance evaluation is measured using target collection rate and average collision rate for the swarm size of 3 to 15 robots. The performance is also measured in terms of target collection rate and average collision rate per robot for a particular swarm size. The experiments are performed for 16 combinations of “waypoint” distance and “waypoint” angle. The consistent performing values of “waypoint” distance and “waypoint” angle are compared with the performance of DDSA. The table 2 specifies the combination values. The combination values depend on the robot radius and a gap distance that need to be between consecutive robots.

Serial No.	Distance	Angle
Set 1	0.2 m	30 degree
Set 2	0.3 m	30 degree
Set 3	0.5 m	30 degree
Set 4	0.6 m	30 degree
Set 5	0.2 m	40 degree
Set 6	0.3 m	40 degree
Set 7	0.5 m	40 degree
Set 8	0.6 m	40 degree
Set 9	0.2 m	70 degree
Set 10	0.3 m	70 degree
Set 11	0.5 m	70 degree
Set 12	0.6 m	70 degree
Set 13	0.2 m	80 degree
Set 14	0.3 m	80 degree
Set 15	0.5 m	80 degree
Set 16	0.6 m	80 degree

Table 2: “Waypoint” Distance and “Waypoint” Angle for PPCA-CCPFA

The performance of PPCA-CCPFA is evaluated based on the distance and angle of the “waypoint”. The best performing set is chosen by analyzing the graphs and selecting the most consistent performing set with respect to the DDSA. For each angle, the best performing distance is selected. Later the four sets of angle and distance are compared to simplify the consistent performing set. Two parameters considered for performance evaluation: rate of target collection and rate of average collision. The rate of target collection specifies the number of targets collected in 30 minutes for each swarm size. The rate of average collision specifies the average number of collisions encountered by all the robots in 30 minutes altogether for a swarm size.

Figure 7 shows the comparison of DDSA and PPCA-CCPFA with Set values of: Set 1 to Set 8 for Target Collection Rate. Similarly, Figure 8 shows the comparison of DDSA and PPCA-CCPFA with Set values of: Set 9 to Set 16 for Target Collection Rate. The mean values of rate of target collection over 10 cluster locations are plotted in the Figures 7 and 8. The X-axis represents the swarm size and the Y-axis represents the average number of targets collected by the swarm per second.

Figure 9 shows the comparison of DDSA and PPCA-CCPFA with Set values of: Set 1 to Set 8 for Average Collision Rate. Similarly, Figure 10 shows the comparison of DDSA and PPCA-CCPFA with Set values of: Set 9 to Set 16 for Average Collision Rate. The mean values of rate of average collision rate over 10 cluster locations are plotted in the Figures 9 and 10. The X-axis

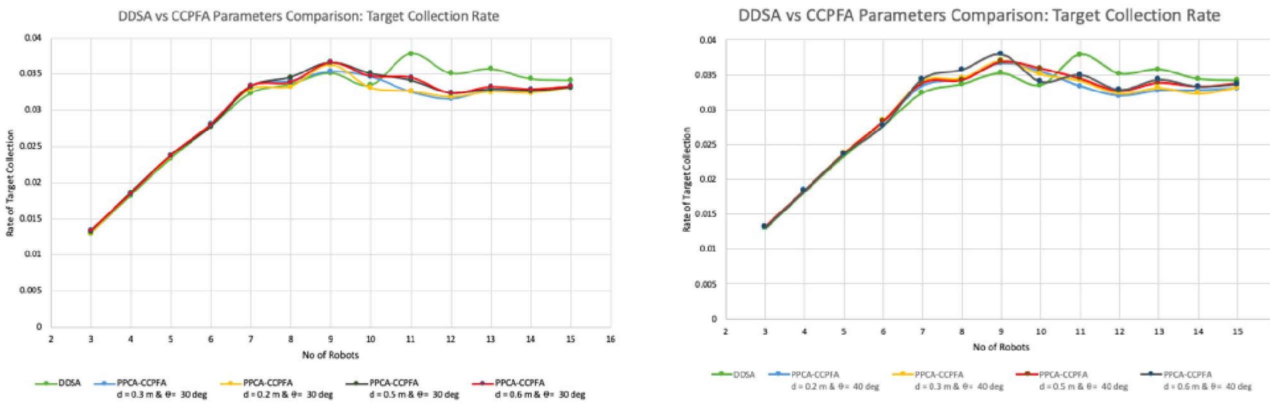

Figure 7: PPCA-CCPFA vs DDSA: Set 1 to Set 8 for Rate of Target Collection

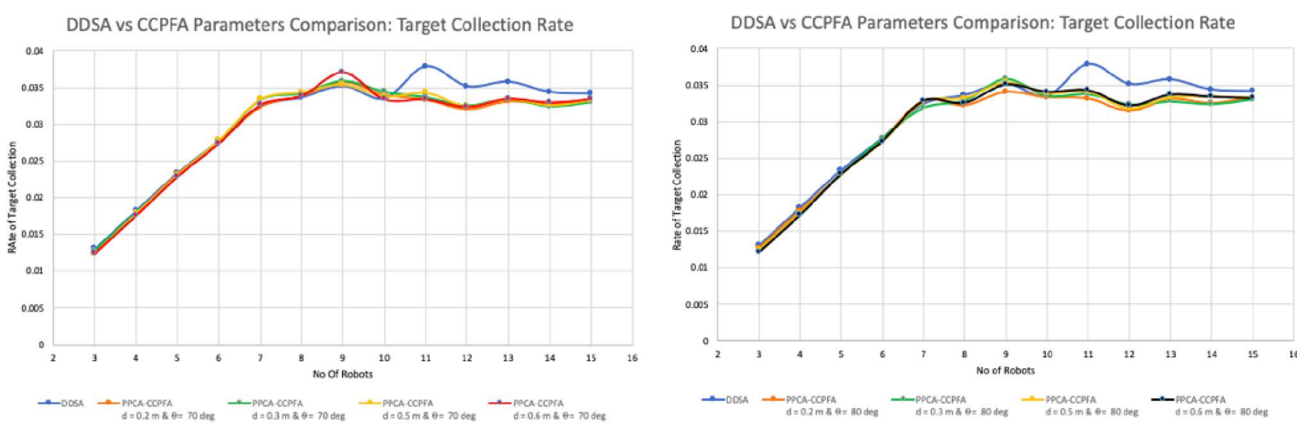

Figure 8: PPCA-CCPFA vs DDSA: Set 8 to Set 16 for Rate of Target Collection

represents the number of robots in the swarm for a particular experiment and the Y-axis represents the average number of collisions encountered by the swarm per second.

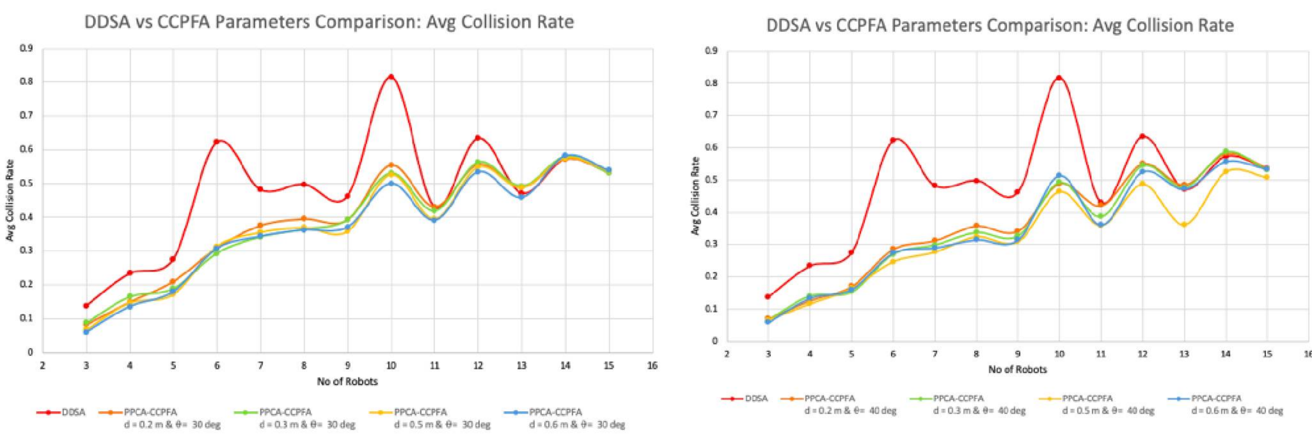

Figure 9: PPCA-CCPFA vs DDSA: Set 1 to Set 8 for Average Collision Rate

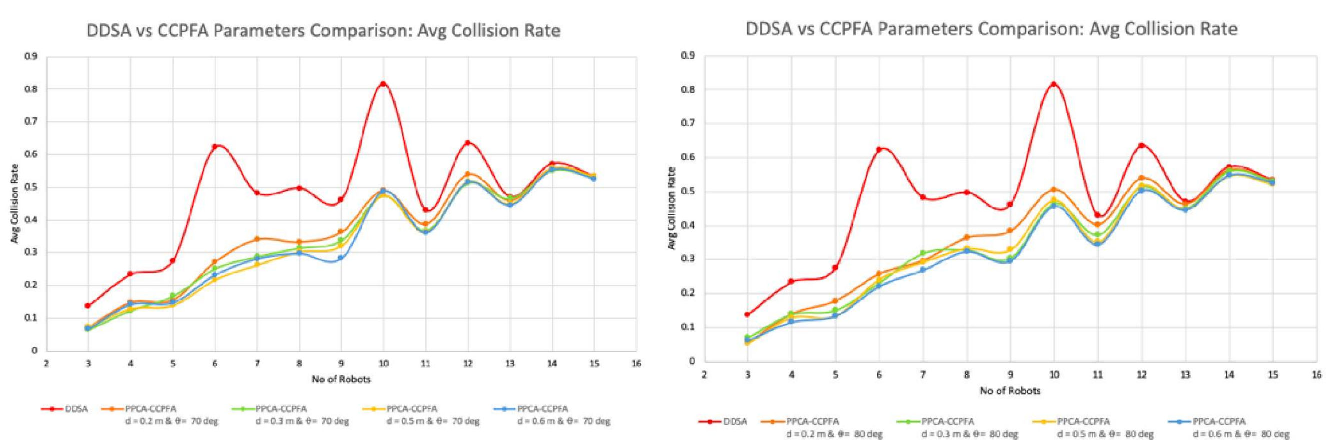

Figure 10: PPCA-CCPFA vs DDSA: Set 8 to Set 16 for Average Collision Rate

The higher the value of the Y-axis, the better is the target collection rate for the swarm in Figure 7 and 8. Lower values of average collision rate mean lesser collisions encountered by the swarm. The algorithm that defines these two characteristics demonstrates better performance. Thus, from the Figures 7, 8, 9 and 10 it can be said that the 4 sets that show a combination of higher target collection rate and lower collision rate are: Set 4, Set 7, Set 11 and Set 16.

The consistent performing sets are selected and compared to the DDSA. Figures 11 and 12 show the comparison of consistent performing sets of PPCA-CCPFA. The selected sets have better target collection rate as well as lower average collision rate than the DDSA for swarm size of 3 to 10.

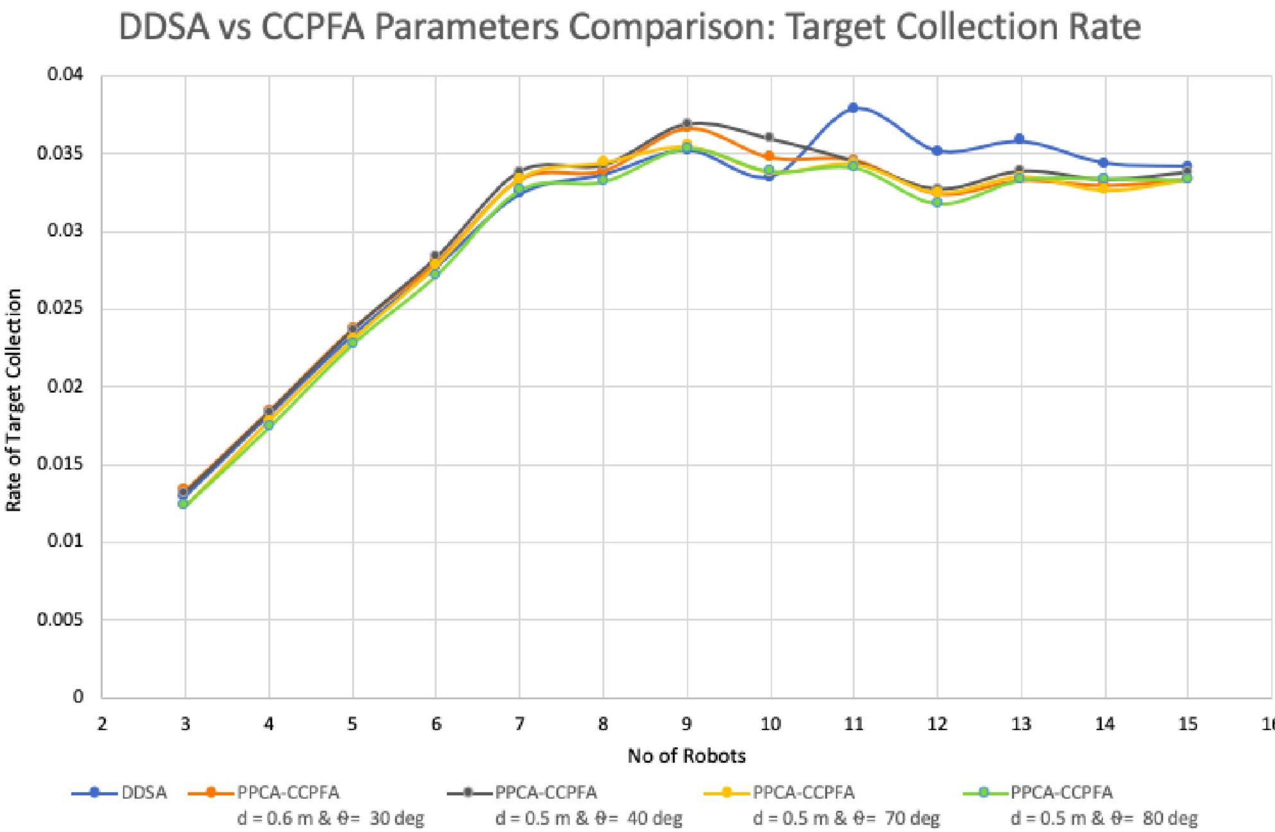

Figure 11: PPCA-CCPFA vs DDSA: Consistent performing sets for Rate of Target Collection

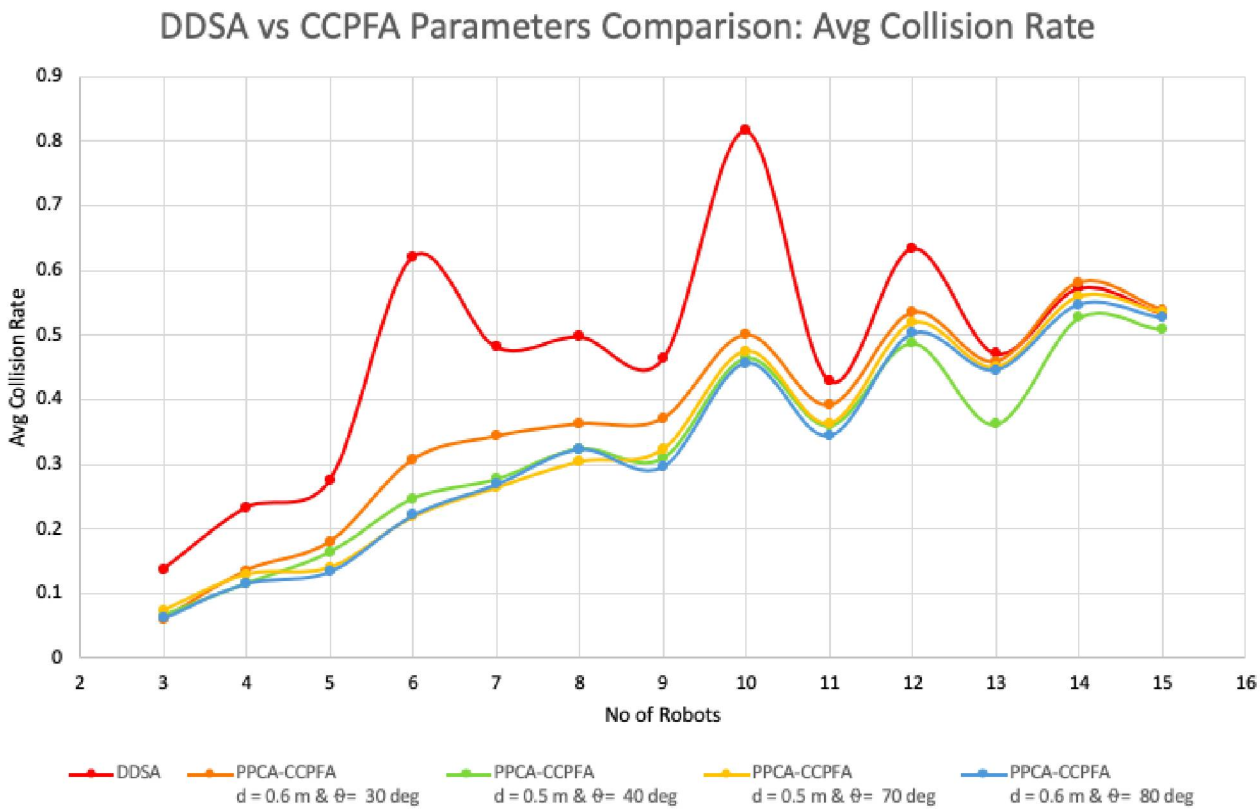

Figure 12: PPCA-CCPFA vs DDSA: Consistent performing sets for Average Collision Rate

The target collection rate with confidence interval of 95% and average collision rate with 95% confidence interval indicating the range of values that are observed 95% of the time when experiments are performed are shown in Figure 13 and Figure 14, respectively. From the figures, it can be observed that PPCA-CCPFA algorithm with parameters "waypoint" distance = $0.5\ m$ and waypoint angle = $40\ degrees$ perform better than the

DDSA for swarm size range of 3 to 10. The performance of PPCA-CCPFA algorithm however, decreases with the increase in the swarm size.

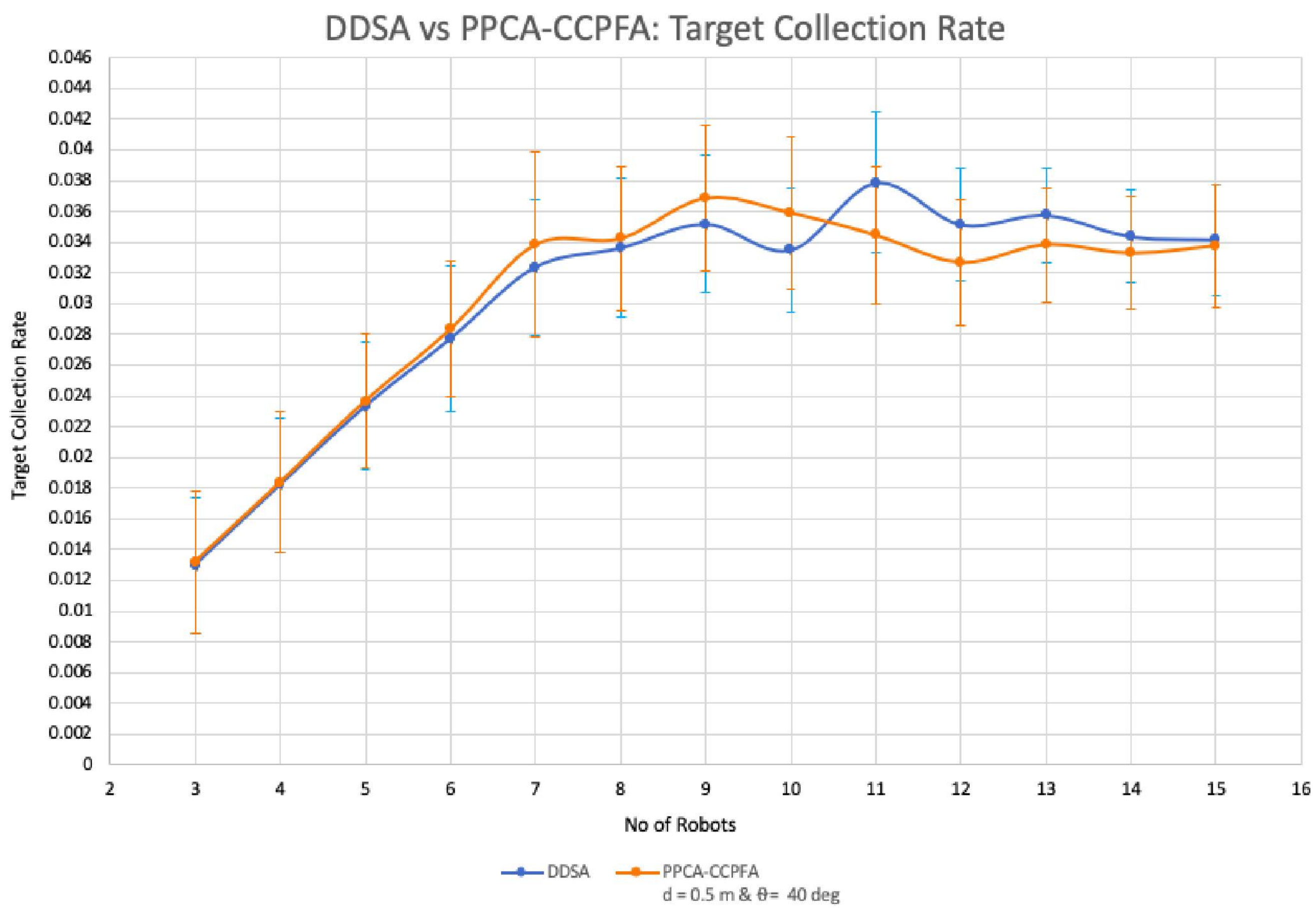

Figure 13: PPCA-CCPFA vs DDSA: $\theta = 40\ deg$ and $d = 0.5\ m$ set for Rate of Target Collection

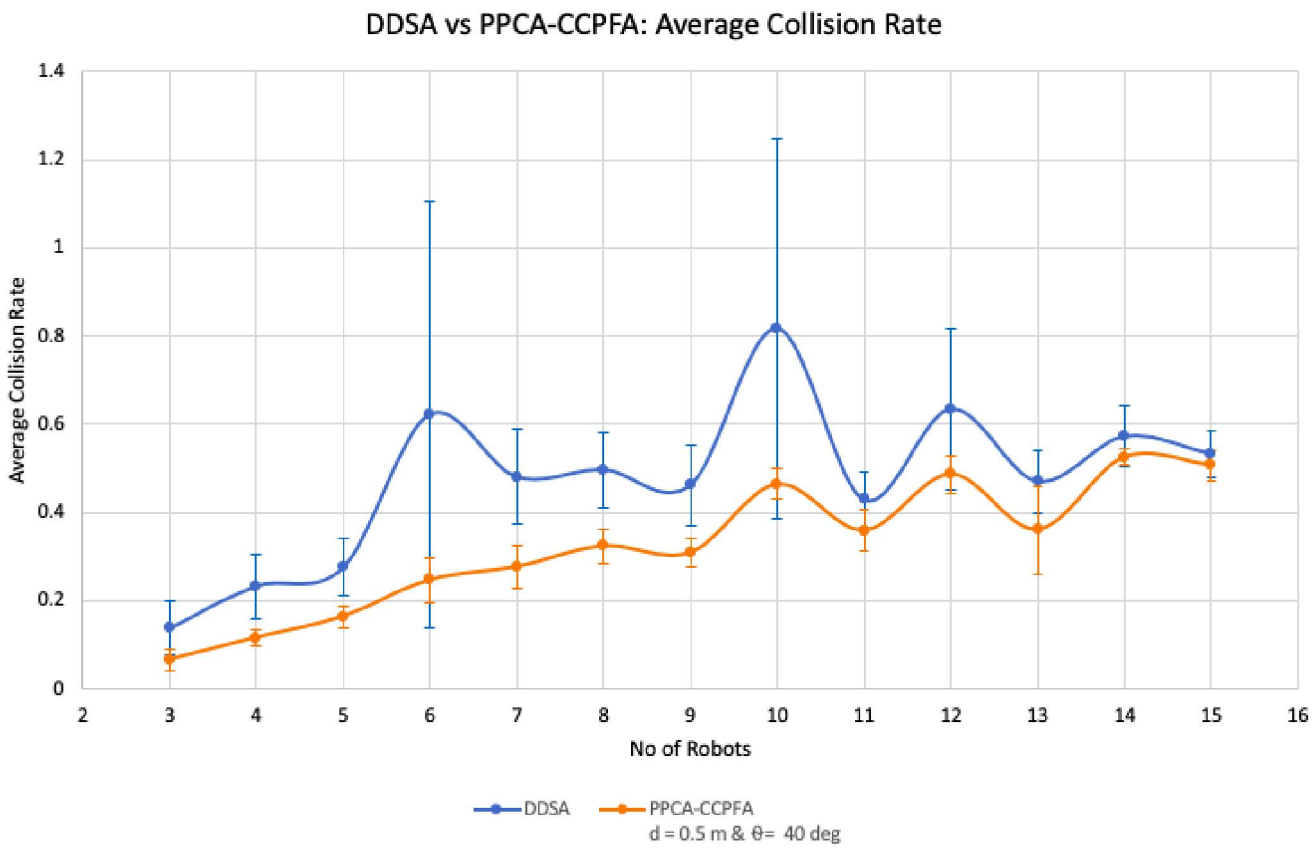

Figure 14: PPCA-CCPFA vs DDSA: $\theta = 40\ deg$ and $d = 0.5\ m$ set for Average Collision Rate

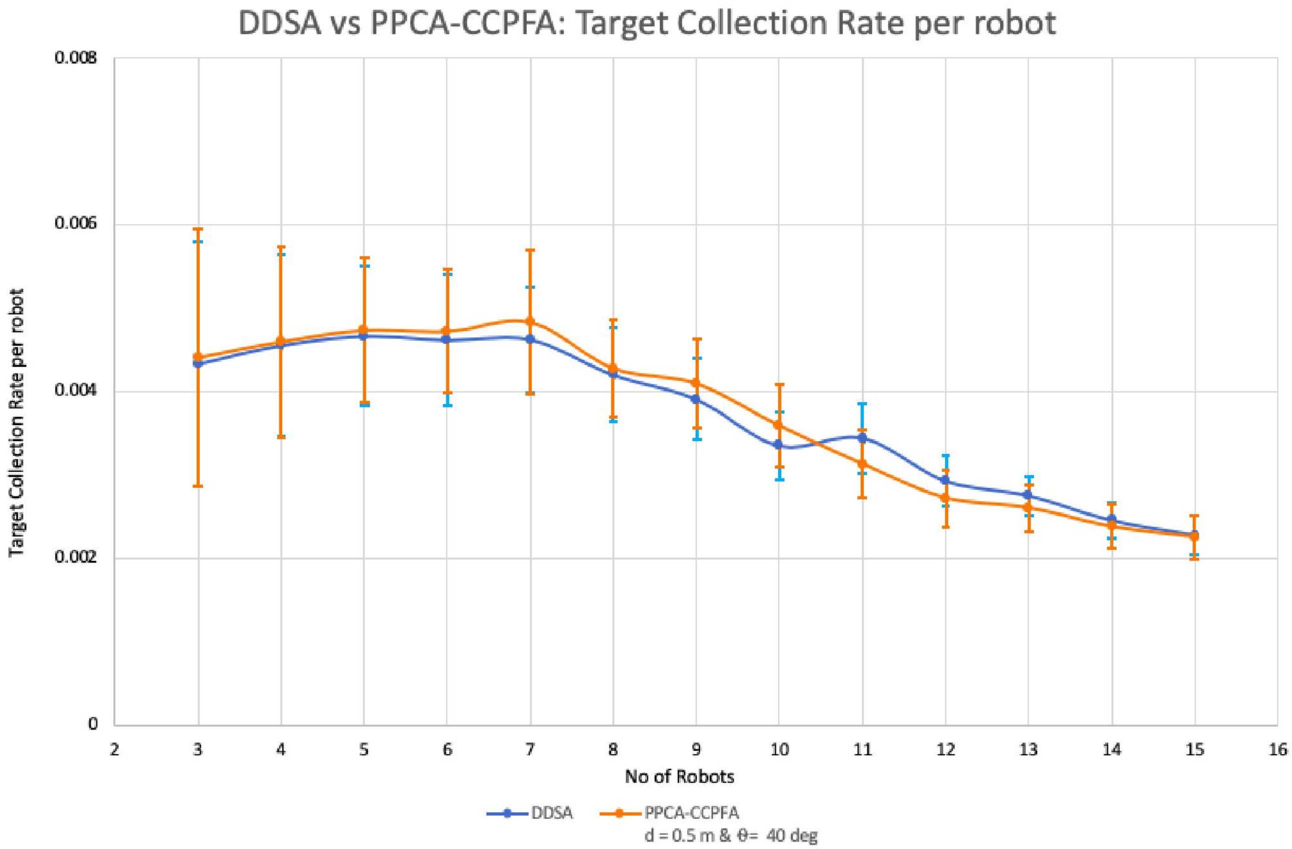

Figure 15: PPCA-CCPFA vs DDSA: $\theta = 40\ deg$ and $d = 0.5\ m$ set for Rate of Target Collection per robot

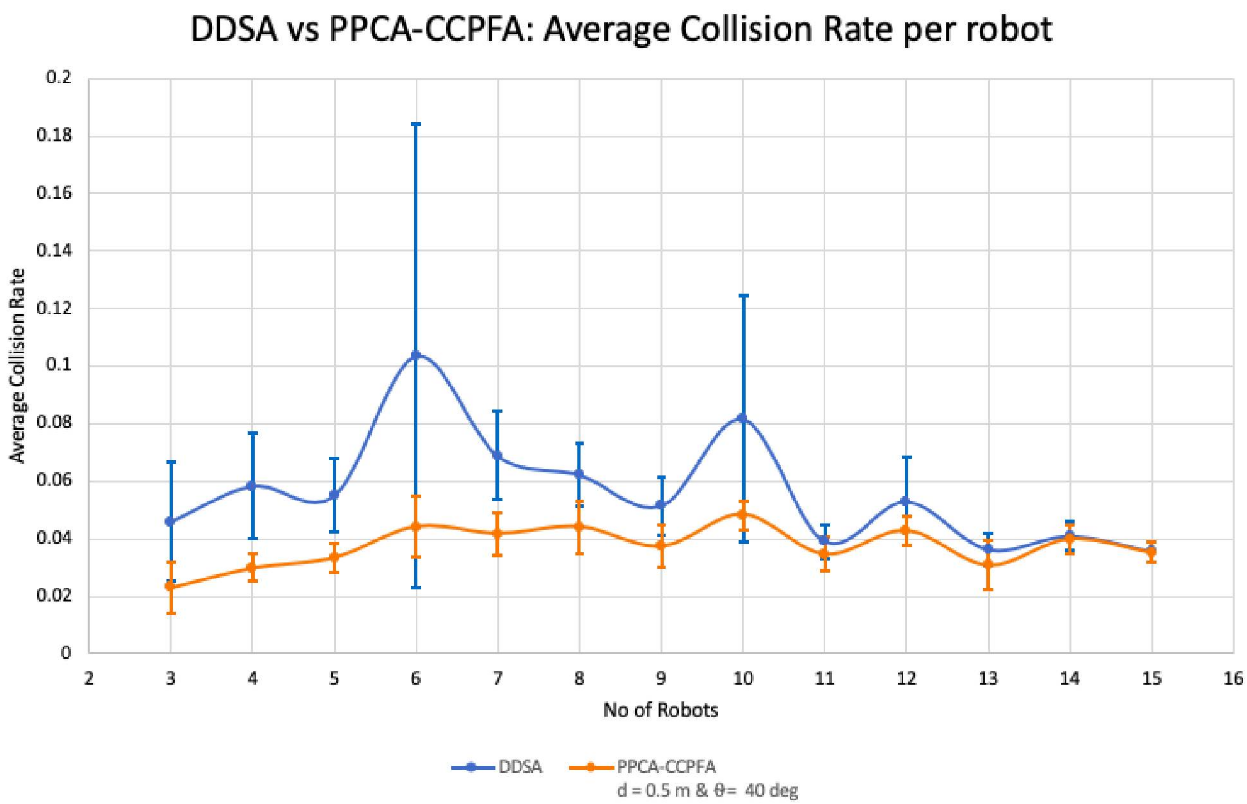

Figure 16: PPCA-CCPFA vs DDSA: $\theta = 40\ deg$ and $d = 0.5\ m$ set for Average Collision Rate per robot

The target collection rate per robot and average collision rate per robot with 95% confidence interval is shown in the Figure 15 and Figure 16 respectively.

4.3 Analysis

The main area of investigation in the thesis is the interference observed in robots for the clustered resource distributions. The number of collisions observed near the cluster and central "home" location increase with the cluster distribution affecting the performance of central place foraging algorithms. In the DDSA and the CPFA algorithms, targets are collected faster in the uniform resource distribution than in the clustered resource distribution. This can be due to the unequal allocation of targets to the robots and increased collisions between the robots [16, 18, 51]. This inspires the focus of this thesis on collision avoidance for clustered resource distributions.

Figure 13 shows the performance comparison for the DDSA and PPCA-CCPFA for the rate of target collection. The thesis approach aims at reducing the number of collisions and improving the resource collection rate. It can be seen from the Figure 13 that the PPCA-CCPFA approach performs slightly better than the DDSA for the swarm size in the range of 6 to 10. The average target collection rate is improved by approximately 3% for the swarm size of 3 to 10. The best performing swarm size of 10 followed by 9 is observed in terms of target collection for PPCA-CCPFA. Whereas, DDSA has best performing swarm size of 11 in terms of target collection rate.

Figure 14 shows the average collision rate for the swarm for DDSA and PPCA-CCPFA. The average reduction in average collision rate for the swarm size 3 to 15 is approximately 33% for the PPCA-CCPFA approach in comparison to the DDSA. The best performing swarm size in terms of average collision rate for the proposed approach is 6 followed by 10 and then 9. Thus, analyzing the graphs in figures 13 and 14, it can be observed that the PPCA-CCPFA performs best for swarm size 9. The swarm size 9 has both good target collection rate and lower average collision rate. Thus, optimal swarm size is 9 for the PPCA-CCPFA and the performance starts decreasing beyond this size.

Figure 15 shows the average target collection rate per robot for a particular swarm size for both the DDSA and PPCA-CCPFA. It is observed in foraging tasks that an increase in the swarm size reduces the performance of each robot in the swarm due to interference between the robots and competition for finite resources. It can be observed that the reduction of inter-robot collisions in the PPCA-CCPFA approach helps the per robot target collection rate by approximately 3% for swarm size 3 to 10. Similarly, figure 16 shows the average collisions observed in the swarm per robot reduced by 29% for swarm size of 3 to 15. These figures help to understand the exploitation vs exploration problem for the central place foraging tasks. It can be observed that collision avoidance and path planning tasks help improve the exploitation of resources per robot in the swarm. Though, the PPCA-CCPFA approach does not fully help target collection efficiency per robot, it helps reduce the average collisions encountered per robot. The approach may help improve the target collection if combined with integration of recruitment and techniques with more intelligent understanding of the environment and communication between the robots [105, 44].

The main reason for low target collection rate improvement can be attributed to the spatial delay that PPCA-CCPFA approach adds to avoid the collisions.This method reduces the number of average collisions but increases the trip time for the robots to reach their search position. The "waypoint" distance in some cases affects the target collection rate. In some cases, interference near the added waypoint is observed if the cluster is located closer to the "nest". Additional computation of waypoint distance with respect to cluster location may help reduce the current observed interference. The PPCA-CCPFA algorithm is activated once every robot completes its one spiral search. Thus, for the swarm size of more than 10, the activation of the algorithm takes place pretty late and this in turn does not help in avoiding earlier collisions as well as adds additional time for target collection for the spatial delay added. The application of the PPCA-CCPFA on real robots may show better target collection rate as compared to the simulator. This is because the reactive collision avoidance on real robots take more time as compared to what is observed in simulators. The reduction in number of collisions in the PPCA-CCPFA may reduce this time and help in better target collection rate for the clustered distributions. Recruitment and allocation of the proper number of robots for clustered resources may help improve the performance of per robot and help in the design of a scale invariant foraging robot system.

5 Conclusion and future work

The focus of the thesis was to pursue performance improvements in the DDSA [16] using path planning and collision avoidance methods for the clustered resource distributions. This thesis presented adding spatial delay for coincident paths and time delay for intersecting paths for the robots collecting resources from the cluster. The robots having coincident paths travel a triangular path by going to a “waypoint” calculated based on parameters: distance and angle. Different combinations of “waypoint” distance and angle were used to observe the performance of the proposed approach. The time delay was calculated based on motion kinematics for the intersecting robots. The concentration was to avoid the congestion observed near the cluster and central depot location when multiple robots collect resources affecting overall performance of the central place foraging algorithm. The proposed approach then was compared to the popular DDSA [16] for performance evaluation with a single 8 x 8 cluster, swarm sizes of 3 to15 and ten random cluster locations.

The results showed that there was reduction of average number of collisions among the robots and increment of target collection rate for the swarm size from 3 to 10. However, the target collection rate decreased with the increase in the swarm size. This is be due to the spatial delay that may have increased the trip time for the robots. It was also observed that the average number of collisions are reduced significantly and are lower than that observed for the DDSA [16] even when the swarm size increases. The "waypoint" used to avoid the congestion was calculated using different combinations of the distance from the nest and angle from the current path of the robot. Trial and error method for these combinations was tested and evaluated. It was found that the distance of $0.5\ m$ from the center of the nest and angle of $40\ degrees$ performed better. The experiments were performed for 30 minutes and the performance of resource collection per unit time and collisions encountered per unit time were used to check the performance of the proposed approach.

Though, the approach reduced the number of collisions for the swarm, the target collection rate didn't improve significantly due to the spatial delay trial and error method. The performance can be improved by computing the angle and distance of the "waypoint" based on the cluster location in the environment. Also, the effect of time cost reduction due to the reduction in average collision can be observed properly on physical robots better than in a simulator. In real robots, the reactive collision avoidance methods consume more time than what it takes in a simulator. To better evaluate the performance of the proposed algorithm, it is activated once every robot of the swarm completes its one search spiral. This is because the robots cause lot of congestion at the central depot location as every robot starts its spiral from the "nest". Thus, the algorithm gets activated comparatively late as compared to smaller swarm sizes in cases of larger swarm sizes and this hampers its performance. Random distribution of robots in the workspace at the start of the experiment may help in this case.

The thesis is a pilot project for using path planning techniques with central place foraging algorithm to improve multi-robot foraging performance. The thesis can be improved by inclusion of proper recruitment and allocation of robots for exploration. By separating the search and collection phase, the recruitment and allocation can be achieved efficiently [105, 44]. This would result in initial searching cost time. However, it would efficiently help in designing of scale invariant swarm foraging algorithm reducing the interference among the robots and increasing the target collection. It is highly worth investigating the causes for performance degradation when swarm size increases above 10. Also, it is worth investigating the performance of the proposed approach on different resource distributions and exploiting the clusters using optimal number of robots which would in turn simplify the path planning techniques.

References

[1] Yi Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 2612–2619 vol.3, May 2002.

[2] Ying Tan and Zhong-yang Zheng. Research Advance in Swarm Robotics. *Def. Technol.*, 9(1):18–39, 2013.

[3] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *From Natural to Artificial Swarm Intelligence*. Oxford University Press, Inc., New York, NY, USA, 1999.

[4] E. Bonabeau and C. Meyer. Swarm intelligence. A whole new way to think about business. *Harv. Bus. Rev.*, 79(5), 2001.

[5] Marco Dorigo. Swarm robotics: The coordination of robots via swarm intelligence principles. In Mike Hinchey, Anastasia Pagnoni, Franz J. Rammig, and Hartmut Schmeck, editors, *Biologically-Inspired Collaborative Computing*, pages 1–1, Boston, MA, 2008. Springer US.

[6] Xian Bing Meng, X. Z. Gao, Lihua Lu, Yu Liu, and Hengzhen Zhang. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. *J. Exp. Theor. Artif. Intell.*, 28(4):673–687, Jul 2016.

[7] Manju Manju and Chander Kant. Ant Colony Optimization: A Swarm Intelligence based Technique. *Int. J. Comput. Appl.*, 73(10):30–33, Jul 2013.

[8] J. Scott Turner. Termites as models of swarm cognition. *Swarm Intell.*, 5(1):19–43, Mar 2011.

[9] Thomas Schmickl, Ronald Thenius, and Karl Crailsheim. Swarm-intelligent foraging in honeybees: benefits and costs of task-partitioning and environmental fluctuations. *Neural Computing and Applications*, 21(2):251–268, Mar 2012.

[10] Fran Sérgio Lobato and Valder Steffen. Fish swarm optimization algorithm applied to engineering system design. *Lat. Am. J. Solids Struct.*, 11(1):143–156, Jan 2014.

[11] Konstantinos V. Katsikopoulos and Andrew J. King. Swarm intelligence in animal groups: When can a collective out-perform an expert? *PLoS One*, 5(11):1–5, Nov 2010.

[12] Louis Rosenberg. Artificial swarm intelligence, a human-in-the-loop approach to A.I. *30th AAAI Conf. Artif. Intell. AAAI 2016*, pages 4381–4382, 2016.

[13] Erin S. Gloag, Lynne Turnbull, and Cynthia B. Whitchurch. Bacterial Stigmergy: An Organising Principle of Multicellular Collective Behaviours of Bacteria. *Scientifica (Cairo).*, 2015:1–8, 2015.

[14] Iñaki Navarro and Fernando Matía. An Introduction to Swarm Robotics. *ISRN Robot.*, 2013:1–10, Sep 2013.

[15] Alan FT Winfield. *Foraging Robots*, pages 3682–3700. Springer New York, New York, NY, 2009.

[16] G. M. Fricke, J. P. Hecker, A. D. Griego, L. T. Tran, and M. E. Moses. A distributed deterministic spiral search algorithm for swarms. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4430–4436, Oct 2016.

[17] Scott Burlington and Gregory Dudek. Spiral Search As An Efficient Mobile Robotic Search Technique. Technical report, 1999.

[18] Joshua P. Hecker and Melanie E. Moses. Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1):43–70, Mar 2015.

[19] M. A. Contreras-Cruz, J. J. Lopez-Perez, and V. Ayala-Ramirez. Distributed path planning for multi-robot teams based on artificial bee colony. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 541–548, June 2017.

[20] Mark Fleischer. Foundations of swarm intelligence: From principles to practice. `https://arxiv.org/pdf/nlin/0502003.pdf`, 2005.

[21] AV Butrimenko. On the search for optimal routes in changing. *Izv. Akad. Nauk SSSR. Ser. Tekhn. Kibern*, 6, 1964.

[22] V.L. Stefanyuk. *On Mutual Assistance in the Collective of Radiostations*. Information Transmission Problems 7, 1971.

[23] M.L. TSetlin and SCITRAN (Firm). *Automaton Theory and Modeling of Biological Systems*. Mathematics in science and engineering. Academic Press, 1973.

[24] Michael O. Rabin. The choice coordination problem. *Acta Informatica*, 17(2):121–134, Jun 1982.

[25] Gerardo Beni. *Swarm Intelligence*, pages 1–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019.

[26] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, Dec 1989.

[27] Alberto Colorni, Marco Dorigo, and Vittorio Maniezzo. Distributed Optimization by Ant Colonies. In *ECAL91 - Eur. Conf. Artif. LIFE*, pages 134–142, Paris, France, 1991. Elsevier Publishing.

[28] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, 1987. ACM.

[29] Pedro Pinto, Thomas A. Runkler, and João M. Sousa. Wasp swarm optimization of logistic systems. In Bernardete Ribeiro, Rudolf F.

Albrecht, Andrej Dobnikar, David W. Pearson, and Nigel C. Steele, editors, *Adaptive and Natural Computing Algorithms*, pages 264–267, Vienna, 2005. Springer Vienna.

[30] Dervis Karaboga and Bahriye Basturk. Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In Patricia Melin, Oscar Castillo, Luis T. Aguilar, Janusz Kacprzyk, and Witold Pedrycz, editors, *Foundations of Fuzzy Logic and Soft Computing*, pages 789–798, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[31] Y. Mohan and S. G. Ponnambalam. An extensive review of research in swarm robotics. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 140–145, Dec 2009.

[32] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, March 1986.

[33] Maja J. Matari. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2):321 – 331, 1995. Moving the Frontiers between Robotics and Biology.

[34] Anita M. Flynn. Gnat robots (and how they will change robotics). 1987.

[35] G. Beni. The concept of cellular robotic system. In *Proceedings IEEE International Symposium on Intelligent Control 1988*, pages 57–62, Aug 1988.

[36] Gregory Dudek, Michael R. M. Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, Dec 1996.

[37] Y. Uny Cao, Andrew B. Kahng, and Alex S. Fukunaga. *Cooperative Mobile Robotics: Antecedents and Directions*, pages 7–27. Springer US, Boston, MA, 1997.

[38] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. Reactivity and deliberation: A survey on multi-robot systems. In *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, pages 9–32, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[39] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, Mar 2013.

[40] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff. Nasa's swarm missions: the challenge of building autonomous software. *IT Professional*, 6(5):47–52, Sep. 2004.

[41] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy,

D. Burnier, A. Campo, A. L. Christensen, A. Decugniere, G. Di Caro, F. Ducatelle, E. Ferrante, A. Forster, J. M. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Retornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stutzle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics Automation Magazine*, 20(4):60–71, Dec 2013.

[42] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, Sep 2002.

[43] Julia T. Ebert, Melvin Gauci, and Radhika Nagpal. Multi-feature collective decision making in robot swarms. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 1711–1719, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[44] Nicholas Dolan-Stern. Central Place Foraging: Separating Search and Collection. Master's thesis, California State University Channel Islands, 2018.

[45] D. W. Stephens and Eric L. Charnov. Optimal foraging: Some simple stochastic models. *Behavioral Ecology and Sociobiology*, 10(4):251–263, Jul 1982.

[46] Giovanni Pini, Arne Brutschy, Carlo Pinciroli, Marco Dorigo, and Mauro Birattari. Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adaptive Behavior*, 21(2):118–136, 2013.

[47] Adam G. Hart and Francis L.W. Ratnieks. Leaf caching in atta leafcutting ants: discrete cache formation through positive feedback. *Animal Behaviour*, 59(3):587 – 591, 2000.

[48] F. L. W. Ratnieks and C. Anderson. Task partitioning in insect societies. *Insectes Sociaux*, 46(2):95–108, May 1999.

[49] Ken Sugawara and Masaki Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D: Nonlinear Phenomena*, 100(3):343 – 354, 1997.

[50] L. Steels. Cooperation between distributed agents through self-organisation. In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pages 8–14 supl, July 1990.

[51] J. P. Hecker, J. C. Carmichael, and M. E. Moses. Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 434–440, Sep. 2015.

[52] Wenguo Liu, Alan F. T. Winfield, Jin Sa, Jie Chen, and Lihua Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305, 2007.

[53] R. Doriya, S. Mishra, and S. Gupta. A brief survey and analysis of multi-robot communication and coordination. In *International Conference on Computing, Communication Automation*, pages 1014–1021, May 2015.

[54] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75–88, Aug 2007.

[55] A. Yamashita, M. Fukuchi, J. Ota, T. Arai, and H. Asama. Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 4, pages 3144–3151 vol.4, April 2000.

[56] Maja J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, Mar 1997.

[57] S. Yuta and S. Premvuti. Consideration on cooperation of multiple autonomous mobile robots-introduction to modest cooperation. In *Fifth*

International Conference on Advanced Robotics 'Robots in Unstructured Environments, pages 545–550 vol.1, June 1991.

[58] T. Arai, H. Ogata, and T. Suzuki. Collision avoidance among multiple robots using virtual impedance. In *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications*, pages 479–485, Sep. 1989.

[59] D. Popenoe. *Sociology*. Prentice Hall, 1993.

[60] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.

[61] Wei Ye, R. T. Vaughan, G. S. Sukhatme, J. Heidemann, D. Estrin, and M. J. Mataric. Evaluating control strategies for wireless-networked robots using an integrated robot and network simulation. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 3, pages 2941–2947 vol.3, May 2001.

[62] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. P. Papanikolopoulos. Performance of a distributed robotic system using shared communications channels. *IEEE Transactions on Robotics and Automation*, 18(5):713–727, Oct 2002.

[63] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos. Effects of limited bandwidth communications channels on the control of multiple robots. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 369–374 vol.1, Oct 2001.

[64] M. J. Mataric, M. Nilsson, and K. T. Simsarin. Cooperative multi-robot box-pushing. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 556–561 vol.3, Aug 1995.

[65] C.Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1):85 – 101, 2000.

[66] M. Jager and B. Nebel. Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 3, pages 1213–1219 vol.3, Oct 2001.

[67] L. S. Marcolino and L. Chaimowicz. Traffic control for a swarm of robots: Avoiding group conflicts. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1949–1954, Oct 2009.

[68] E. Todt, G. Rausch, and R. Suarez. Analysis and classification of multiple robot coordination methods. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 4, pages 3158–3163 vol.4, April 2000.

[69] A. Farinelli, L. Iocchi, and D. Nardi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2015–2028, Oct 2004.

[70] P. Caloud, Wonyun Choi, J. . Latombe, C. Le Pape, and M. Yim. Indoor automation with many mobile robots. In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pages 67–72 vol.1, July 1990.

[71] A. Yamashita, T. Arai, Jun Ota, and H. Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 19(2):223–237, April 2003.

[72] R. Luna and K. E. Bekris. Efficient and complete centralized multi-robot path planning. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3268–3275, Sep. 2011.

[73] Jean-Claude Latombe. *Introduction and Overview*, pages 1–57. Springer US, Boston, MA, 1991.

[74] A. S. H. H. V. Injarapu and S. K. Gawre. A survey of autonomous mobile robot path planning approaches. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pages 624–628, Oct 2017.

[75] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89 – 99, 2002. Ninth International Symposium on Intelligent Robotic Systems.

[76] Lynne E. Parker. *Multiple Mobile Robot Teams, Path Planning and Motion Coordination in*, pages 5783–5800. Springer New York, New York, NY, 2009.

[77] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[78] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, March 1985.

[79] H. G. Tanner and A. Kumar. Towards decentralization of multi-robot navigation functions. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4132–4137, April 2005.

[80] Adham Atyabi and David Powers. Review of classical and heuristic-based navigation and path planning approaches. *International Journal of Advancements in Computing Technology (IJACT)*, 5:1–14, Jan 2013.

[81] P. Bhattacharya and M. L. Gavrilova. Roadmap-based path planning - using the voronoi diagram for a clearance-based shortest path. *IEEE Robotics Automation Magazine*, 15(2):58–66, June 2008.

[82] K. M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1160–1165, Sep. 2008.

[83] L. E. Kavraki, P. Svestka, J. . Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug 1996.

[84] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001 vol.2, April 2000.

[85] Elmar Langetepe. On the optimality of spiral search. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algo-*

rithms, SODA '10, pages 1–12, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[86] A. Ryan and J. K. Hedrick. A mode-switching path planner for uav-assisted search and rescue. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1471–1476, Dec 2005.

[87] Ricardo Baeza-Yates and René Schott. Parallel searching in the plane. *Comput. Geom. Theory Appl.*, 5(3):143–154, October 1995.

[88] A. T. Hayes, A. Martinoli, and R. M. Goodman. Swarm robotic odor localization. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 2, pages 1073–1078 vol.2, Oct 2001.

[89] Ofer Feinerman, Amos Korman, Zvi Lotker, and Jean-Sebastien Sereni. Collaborative search on the plane without communication. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing*, PODC '12, pages 77–86, New York, NY, USA, 2012. ACM.

[90] M. Müller and R. Wehner. The hidden spiral: systematic search and path integration in desert ants, cataglyphis fortis. *Journal of Comparative Physiology A*, 175(5):525–530, Nov 1994.

[91] Z. B. Hao, N. Sang, and H. Lei. Cooperative coverage by multiple robots with contact sensors. In *2008 IEEE Conference on Robotics, Automation and Mechatronics*, pages 543–548, Sep. 2008.

[92] Hendrik Skubch. *Evaluation*, pages 195–233. Springer Fachmedien Wiesbaden, Wiesbaden, 2013.

[93] Alejandro López-Ortiz and Daniela Maftuleac. Optimal distributed searching in the plane withand without uncertainty. In Mohammad Kaykobad and Rossella Petreschi, editors, *WALCOM: Algorithms and Computation*, pages 68–79, Cham, 2016. Springer International Publishing.

[94] Dani Goldberg and Maja J. Matarić. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 637–642. AAAI Press, 1997.

[95] Jose Guerrero, Gabriel Oliver, and Oscar Valero. Multi-robot coalitions formation with deadlines: Complexity analysis and solutions. *PLOS ONE*, 12(1):1–26, January 2017.

[96] Daniel Claes and Karl Tuyls. Multi robot collision avoidance in a shared workspace. *Autonomous Robots*, 42(8):1749–1770, Dec 2018.

[97] C. Cai, C. Yang, Q. Zhu, and Y. Liang. Collision avoidance in multi-robot systems. In *2007 International Conference on Mechatronics and Automation*, pages 2795–2800, Aug 2007.

[98] D. A. Shell and M. J. Mataric. On foraging strategies for large-scale multi-robot systems. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2717–2723, Oct 2006.

[99] P. Velagapudi, K. Sycara, and P. Scerri. Decentralized prioritized planning in large multirobot teams. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4603–4609, Oct 2010.

[100] Marina B. A. Souza, Edimar José de Oliveira, Leonardo Willer de Oliveira, and Antonio Paulo G. M. Moreira. Collision avoidance for multi-robot systems with coincident paths based on fictitious collision points using nonlinear formulation. In Anibal Ollero, Alberto Sanfeliu, Luis Montano, Nuno Lau, and Carlos Cardeira, editors, *ROBOT 2017: Third Iberian Robotics Conference*, pages 60–71, Cham, 2018. Springer International Publishing.

[101] S. Bhattacharya and R. Agrawal. Development of robot swarm algorithms on an extensible framework. In *SoutheastCon 2017*, pages 1–6, March 2017.

[102] Pat Monaghan and Neil B. Metcalfe. Group foraging in wild brown hares: effects of resource distribution and social status. *Animal Behaviour*, 33(3):993 – 999, 1985.

[103] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, Dec 2012.

[104] BCLab-UNM. DDSA-ARGoS github code repository. `https://github.com/BCLab-UNM/DDSA-ARGoS`, 2016.

[105] N. Dolan-Stern, K. Scrivnor, and J. Isaacs. Multimodal central place foraging. In *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pages 72–78, Jan 2018.