



Towards a Swarm Intelligence Benchmark Framework

Submitted by

Jabez Leong Kit

Thesis Advisor

Prof. Roland Bouffanais & Prof. Soh Gim Song

Engineering Product Development Pillar

2022

PhD Thesis Examination Committee

TEC Chair:	Prof. Tan U-Xuan
Main Advisor (External):	Prof. Roland Bouffanais
Co-advisor(s):	Prof. Soh Gim Song
Internal TEC member 1:	Prof. Foong Shaohui
Internal TEC member 2:	Prof. Yuen Chau

Declaration

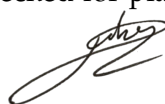
I, Jabez Leong Kit, declare that this thesis titled, “Towards a Swarm Intelligence Benchmark Framework” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

I hereby confirm the following:

- I hereby confirm that the thesis work is original and has not been submitted to any
- I hereby grant SUTD the permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created in accordance with Policy on Intellectual Property, clause 4.2.2.
- I have fulfilled all requirements as prescribed by the University and provided 1 copy of my thesis in PDF.
- I have attached all publications and award list related to the thesis (e.g. journal, conference report and patent).
- The thesis does not contain patentable or confidential information.
- I certify that the thesis has been checked for plagiarism via turnitin/ithenticate. The score is 28%.

Name and signature: Jabez Leong Kit



Date: 10th Jan 2022

Abstract

Natural swarms have inspired the development of artificial swarming systems capable of solving a wealth of problems across numerous areas, including robotics, optimization, finance, organizational study, and beyond. The intriguing collective behaviors of eusocial animals—insect colonies, flocks of birds, schools of fish—have left researchers wondering “What governs the swarm? Who provides the instructions to direct and control the swarm?”. A key answer is that these systems exhibit a unique form of intelligence, a collective intelligence dubbed swarm intelligence.

Swarm Intelligence, a relatively new term, was first coined by Beni and Wang in 1993, while they were working on cellular robotic systems. Different groups of researchers like biologists, roboticists, computer scientists, and physicists have since built upon this definition, from the perspective of their own discipline. This has led to multiple non-unified definitions of the concept of swarm intelligence, which can have an adverse effect on the advancement of swarm intelligence and swarming systems. For instance, apparently simple questions like “Does this system possess swarm intelligence?”, “How (swarm) intelligent is this system?”, and “Is system A more (swarm) intelligent than system B?” are absolutely nontrivial. This points to the lack of a form of quantification of swarm intelligence. On the swarming system design front, recent works have also suggested that there is a need for standardization in its design methodology. A benchmark problem would seem to be a reasonable and much needed solution to enable the comparison of swarm intelligence across various systems, such that further improvements to the system can be made.

This thesis aims at: (i) revisiting the concept of swarm intelligence, (ii) proposing a quantification of swarm intelligence, (iii) developing a benchmark framework, and (iv) performing a quantitative study of the transferability of swarm intelligence from one problem to another.

Publications

1. Kit, J. L., Mateo, D., Bouffanais, R. (2018, November). A decentralized mobile computing network for multi-robot systems operations. In 2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON) (pp. 309-314). IEEE.
2. Kit, J. L., Dharmawan, A. G., Mateo, D., Foong, S., Soh, G. S., Bouffanais, R., Wood, K. L. (2019, August). Decentralized multi-floor exploration by a swarm of miniature robots teaming with wall-climbing units. In 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS) (pp. 195-201). IEEE.
3. Kwa, H. L., Kit, J. L., Bouffanais, R. (2020, May). Optimal swarm strategy for dynamic target search and tracking. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (pp. 672-680).
4. Kwa, H. L., Kit, J. L., Bouffanais, R. (2021). Tracking Multiple Fast Targets With Swarms: Interplay Between Social Interaction and Agent Memory. arXiv preprint arXiv:2108.07122.
5. Kwa, H. L., Kit, J. L., Bouffanais, R. Tailoring Exploration and Exploitation in Multi-Agent Systems with Short-Term Memory and Limited Social Interaction.
6. Kwa, H. L., Leong Kit, J., Bouffanais, R. Balancing Collective Exploration and Exploitation in Multi-Agent and Multi-Robot Systems: A Review. *Frontiers in Robotics and AI*, 438.

Acknowledgements

Firstly, I thank my advisor, Prof. Roland Bouffanais, for being an inspiration to me. He has been my mentor since my undergraduate days in SUTD, who have given me countless advice and guidance throughout the past 10 years. His perspective has always inspired new thoughts and ideas in me. With him, I have learned to be bold in pursuing ambitious ideas, be persistent and pivot throughout the research journey, and to see the value of my own work.

Secondly, I thank my colleagues Kwa Hian Lee, Dr. Nikolaj Horsevad, Dr. David Matteo, Dr. Maryam Kouzehgar, Dr. Sreetej Lakkam, Dr. Kuan Yoke Kong, Francesco Vallegra, Alvee. It was enjoyable to share a laboratory (3.306) with the amazing individuals whom have inspired my work, shared knowledge with me, and given feedback to improve my work.

Thirdly, I thank my wonderful family. My lovely wife, Ann, have been my biggest support throughout my PhD journey. Her continuous love and support has kept me going and even more so when things did not turn out well. She has scarified to take care of our two little ones, Zachary and Shekinah, so well so that I can pursue my studies.

Fourthly, I thank my parents. They have instilled in me life values that have helped me through this journey. Conversations with them have always been encouraging and they have never failed to light me up.

Finally, I want thank God for His providence and bringing me through this exciting journey. He has never failed to comfort and love on me through difficult times. I am grateful and thankful for this journey that He has planned for me. I also thank my church to their continuous prayer and support.

"For I know the plans I have for you," declares the Lord, "plans to prosper you and not to harm you, plans to give you hope and a future." Jeremiah 29:11

Contents

PhD Thesis Examination Committee	i
Declaration	ii
Abstract	iii
Publications	iv
Acknowledgements	v
1 Introduction	1
1.1 Swarming systems	1
1.1.1 Biological swarms	2
Ant colonies	2
Bee colonies	3
Flocking Birds and Schooling Fish	4
1.2 Swarm Intelligence	5
1.2.1 Artificial Swarm and Swarm Intelligence Applications	7
1.3 Research Questions	8
1.4 Research Contributions	9
2 Quantifying Swarm Intelligence	10
2.1 Introduction	10
2.2 Swarm Intelligence Definitions	11
2.3 Swarm Intelligence, Flexibility and Adaptability	13
2.4 Exploration-Exploitation balance in Swarm Intelligence	16
2.5 Swarm Intelligence and its environment	17
2.6 Conclusion and Potential outlook	18
3 Experimental Investigation: Decentralized multi-floor exploration by a swarm of miniature robots teaming with wall-climbing units	20
3.1 Introduction	20
3.2 System Architecture	23
3.2.1 Individual robotic unit: O-climb and O-map	23
3.2.2 Decentralized Swarming Technology	24
3.3 Collective Mapping Approach	26
3.3.1 Map Representation	26
3.3.2 Communication Network	27
3.3.3 Exploration Strategy	28
3.4 Experimental Results & Discussion	28

3.4.1	Scalability experiments	28
3.4.2	Robustness and recovery experiments	29
3.4.3	Flexibility experiments	31
3.4.4	Multi-floor exploration	32
3.5	Conclusions	32
4	Balancing Collective Exploration and Exploitation in Multi-Agent and Multi-Robot Systems: A Review	34
4.1	Introduction	34
4.2	Exploration and Exploitation	36
4.2.1	Characterizing Exploration and Exploitation	36
4.2.2	Measuring Exploration and Exploitation	38
	Spatial Distribution Based Metrics	38
	Probability Based Metrics	40
	Other Metrics	40
4.3	Agent Response Methods	41
4.3.1	Changes in State	42
	Self-Determined State Changes	43
	Assigned State Changes	46
4.3.2	Small Response Changes	49
	Attraction-Repulsion Dynamics	49
	Random Walks	52
	Time Intervals	52
4.3.3	Area and Task Assignment	53
4.4	Information Dissemination Methods	54
4.4.1	Changes in Network Topology	55
	Number of Neighbors	55
	Communication Range	57
	Forced Network Changes	58
4.4.2	Stigmergy	59
4.5	Conclusion	60
5	Towards a benchmark framework that probes the flexibility of swarming systems.	72
5.1	Introduction	72
5.1.1	What is swarm intelligence?	74
5.1.2	Quantifying swarm intelligence	75
5.1.3	Our research question	77
5.1.4	Our contributions	77
5.2	Existing Benchmark Problems	78
5.3	Design Methodology of SI Benchmark Framework	79
5.3.1	Environment	80
5.3.2	Agents	84
5.3.3	Communication network	85
5.4	Strategies-Swarming Algorithms	85
5.4.1	Artificial Bee Colony	85
5.4.2	Particle Swarm Optimization	87

5.4.3	K-Nearest Neighbor with repulsion (KNN-R)	88
5.5	Design of Metrics	90
5.6	Benchmark Framework: Collective Mapping Simulation	90
5.6.1	Results and Discussion	91
	Effect of workload distribution on the flexibility of the system . .	91
	Effect of dynamic100 environment on the flexibility of the system.	94
	Effect of various dynamic environmental change (early to late) .	95
5.7	Conclusion	98
6	Qualitative study on the Transferability of Swarm Intelligence	103
6.1	Introduction	103
6.2	Strategy: PSO adaptive repulsion	104
6.3	Problem 1: Collective Mapping	107
6.4	Problem 2: Multiple fast tracking	109
6.5	Results and discussion	110
6.6	Conclusion	111
7	Conclusions	114

List of Figures

3.1	O-climb (left) and O-map (right).	24
3.2	The electronic suite of hardware components of the ORION platform. Modules on the bottom row—PCB, LiDAR, and wireless charger module—were designed and custom-built for ORION.	25
3.3	Multi-block diagram of the developed software framework for ORION.	26
3.4	Scalability experiments for five different MRS sizes: $N = 1, 2, 4, 6$ and 8 . Each point—mean value and the associated standard deviation—is obtained from the results of five repeated experiments with same N value and identical initial conditions. The insert shows the scaling of two characteristic times with N for these collective mapping operations by means of Methods 1 and 2.	29
3.5	Robustness experiments: each of the five experiments are shown with different colored dots. The mean performance and its associated standard deviation is shown in black. The red (resp. blue) dashed lines correspond to the instant at which two units are removed (resp. added) of the MRS. The insert shows the rate of change of the number of explored cells C_e	30
3.6	Flexibility experiments: each of the five experiments are shown with different colored dots. The mean performance and its associated standard deviation is shown in black. The red dashed lines correspond to the instant at which the floor area is expanded by 50%. The insert shows the rate of change of the number of explored cells C_e	31
3.7	Snapshots at successive time instants (t_0, t_1, t_2, t_3) of the mapped areas by 12 units across two floors (F_1 and F_2).	32
4.1	Flowchart detailing how an agent deterministically decides on its response when provoked by environmental stimuli.	42
4.2	Inter-agent interaction forces generated by various repulsion and attraction schemes. Positive values indicate an attractive force and negative values indicate a repulsive force.	50
4.3	(Left) A standard communication model with an agent in communication with its 10 nearest neighbors. (Right) A forced switching communication model where $k_s = 5$ of the nearest neighbors are substituted with a more distant set of neighbors within the black circle with a probability of p . [1]	59
5.1	Simulation environment true maps: 1, 2, 3 pillars and scattered workload distribution	82
5.2	Static environment; comparing 1p, 2p, 3p, scattered	93

5.3	Swarm Intelligence Index (SII) of systems when change is injected @100 iteration	95
5.4	Dynamic environment change @ iteration 100; comparing 1p, 2p, 3p, scattered	100
5.5	Swarm Intelligence Index (SII) of systems in 2 pillars environments . . .	101
5.6	Slow to fast environmental changes; comparing static, dynamic1000, dynamic600, dynamic100	102
6.1	(Left) Agents using adaptive repulsion being blocked by the search-space boundaries and neighboring agents; (Right) agents remaining in place after losing track of the target with the low exploratory behavior in the presence of low constant repulsion.	106
6.2	PSO adaptive repulsion collective mapping performance, varying $k = [0,1,3,5,9,15]$ (of pillars = 2, dynamic change @iteration100, sensor_range = 0.5m, [min rep, max rep] = [3m, 12m]	111
6.3	Performance vs. k	112
6.4	A concept of swarming system characteristic map	112

List of Tables

4.1	List of works using spatial distribution metrics to quantify MAS exploration and exploitation.	64
4.2	List of works using probability based metrics to quantify MAS exploration exploitation dynamics.	65
4.3	List of works using their own developed methods or metrics to quantify the level exploration and exploitation carried out by an MAS.	65
4.4	List of works using self-determined state change strategies to influence a system's exploration and exploitation dynamics.	66
4.5	List of works using assigned agent responses to influence a system's exploration and exploitation dynamics.	67
4.6	List of works using small response changes to influence the level of exploration and exploitation of an MAS.	68
4.7	List of works using agent area and task assignments to influence the level of exploration and exploitation of an MAS.	69
4.8	List of works using different information dissemination strategies to influence the level of exploration and exploitation of an MAS or MRS. . . .	70
4.9	List of works using stigmergy to influence the level of exploration and exploitation carried out by an MAS.	71
5.1	Benchmark Simulation Parameters	83
5.2	Simulation parameters: Effect of workload distribution	91
5.3	Simulation parameters: Effect of dynamic100 environment	94
5.4	Simulation parameters: Effect of (slow to fast) dynamic change	96

*To my lovely wife Ann,
and our amazing children: Zachary and Shekinah*

Chapter 1

Introduction

1.1 Swarming systems

Social insects are intriguing and have fascinated many naturalists, biologists, social scientists, and have been a source of inspiration for roboticists, and even poets. “What is it that governs here? What is it that issues orders, foresees the future, elaborates plans, and preserves equilibrium?,” wrote Maeterlinck [2]. Every agent seems to have a different agenda, and yet the colony, as a whole, exhibit some levels of organization. Such self-organized collective behaviors observed in insect colonies and social animal groups, emerge from simple and repeated local interactions among entities [3]. If one takes a micro-perspective, these entities—a.k.a agents such as ants, schooling fish or flocking birds—are indeed complex in the way they process sensory inputs, communicate (directly or indirectly) with fellow agents, and form local decisions that drive their actions. Yet, this agent complexity cannot in itself explain the complexity of the workings of the group. As a swarm, the individual agents are governed by a set of local rules that make use of locally gathered information to determine their actions in order to achieve a global objective. In such a complex system, it is often rightfully said that the whole—i.e., the emergent behavior—is more than the sum of its parts.

Swarming systems display a number of sought-after features like: (i) decentralization—i.e. the absence of a leader that controls the entire swarm’s behavior, instead it emerges from self-organization, (ii) local perception leading to global knowledge by means of effective distributed information sharing, (iii) robustness—i.e. the ability for the system to continue functioning despite sub-system failures, (iv) scalability, (v) adaptivity—i.e. the ability to learn/change behavior to respond to new operating conditions, and (vi) flexibility—i.e., the capacity to solve problems/perform tasks in a changing environment.

Decentralization and local perception suggest that the system does not have a single point of failure or potential information transfer congestion, which are the key ingredients for a robust and scalable system. Moreover, when considering agents that are simple enough, an economical argument can be made: it costs lesser resources—e.g. financial or computational resources—to produce these agents, and ultimately the system. It is often stated that a ‘high’ number of agents (N) is required to make a swarm. However, it is important to note that the absolute size of the swarm N is not deterministic of how scalable the system is. From our work, we have shown that N has to

be considered within its operating domain and operating conditions. For example, in the scalability experiment detailed in Chapter 3, it is reported that collective performance deteriorates as N increases, any more agents than 8 will not largely improve the system's performance. Finally, flexibility and adaptivity arise from a combination of perception, information transfer, local decisions based on local rules that allow swarming systems to deal with dynamic environments and changing goals. It can be argued, that flexibility and adaptivity are the central features of swarming systems. Features like robustness and scalability can still be achieved if a system that is not swarming. In other words, without flexibility and adaptivity, there is no swarm.

1.1.1 Biological swarms

In the different collective organizations that exist in nature, repeated local interactions is key for collective behaviors to emerge. In nature, there are numerous methods of interaction that exist, and they can be broadly classified into two groups: direct interactions and indirect one. Direct interactions consist of physical contacts, direct visual observations of agents, etc. Indirect interactions, key to stigmergy—i.e. altered environments which signal to other agents to change their behaviors, for example pheromones deposited by ants to lead others to the various food sources. Self-regulation, foraging, flocking, and schooling are some examples of such collective animal behaviors. In what follows, we rapidly examine a few commonly studied natural swarming systems.

Ant colonies

Ant colonies are eusocial communal—i.e. the highest level of organization of sociality. These colonies performs cooperative brood care, multiple adult generations within, and a division of labor into reproductive and non-reproductive groups. Since hatching, each ant seems to know its role in this complex community that it is living with. Although there is a "leader" queen ant, it does not provide any instructions to the newly hatched ants that they can follow for the rest of their lifetime. Neither does the colony provide an orientation to the young ants. In such a heterogeneous swarm, the individual ants execute distinct functions within the colony. This can differ across different species of ants, but generally the colony will be described as the following.

During the early stages of forming the colony, the queen flies around to look for its mate. Once they mate, the queen will shed off its wings and starts forming a colony. It uses eggs, fat, and wing muscles to survive the first phase until the earliest workers come into existence. From there the queen will live in a chamber deep within the colony where it will perform its key function of laying eggs and mating with the males to continue expanding the colony.

The worker ants perform the role of maintaining and helping the colony to survive. There are multiple tasks for the worker ants within and outside of the colony. The workers that are inside the colony most likely will stay their throughout their lifetime. From within, the workers run a complex waste management system that keeps

the colony extremely clean and free of mold. They will dedicate an area for dumping waste—i.e. food waste, faeces, and even dead ants. They will also care for the brood by feeding and cleaning them. The worker ants stomach is a shared property, sometime named as the “social stomach”, where they collect and store food and can regurgitate food to the rest of the colony back in the nest. It is interesting to know that different classes of brood will be fed differently by means of a brood recognition mechanism [4]. For example, the worker ant brood will be fed lesser than a queen ant brood. On the outside, the worker ants traverse the environment and forage for food. The mechanism of which the workers gather food for its colony is being widely studied as Ant Colony Optimization (ACO), mimicking the process of depositing pheromone and for others to maneuver back and forth between food sources and the colony.

Note that a colony’s workload and its division of labor does not stay static. Some studies have shown that the dynamic division of labor within an ant colony depends extensively on its size. For instance, during an early stage of the colony, more worker ants might be focusing on expanding the physical nest than to care for the broods. This can then be an environmental cue for the queen to decide the fertilization process, yielding different ants—e.g. fertilized eggs will hatch a female worker ants to maintain the colony, while non-fertilized eggs will hatch to be male ants with a single purpose of mating with the queen [5]. Such a division of labor is an apparent display of swarm flexibility. When the worker ant is discharged of its duty of caring for the brood, it can move on to perform other tasks like cleaning. Sometimes, this discharge is due to the age of the worker ant, the older the ant is the further its role will be further away from the queen. Other worker ants can also join the foraging sub-colony when the situation requires it, e.g., when the colony needs more food.

Bee colonies

The organizational structure of a bee colony is very similar to that of ant colonies. The queen, male drones and female workers have similar functions, except for some behaviors that are different. The queen bee defines its colony by producing a unique scent that every bees within its colony will pick up. This scent becomes the bees’ hive identity, distinguishing itself from the other neighboring hives. The queen bee like the queen ant, lays unfertilized eggs. Depending on the dynamics of the colony, the queen bee can regulate the ‘output’ of its fertilizing process—i.e. controlling the number of male drones, female workers, new queens—for the benefit of its colony. Before the queen bee is able to fertilize, it needs to make its maiden mating flight, and it will be its last. During this flight, it will fly far away from its hive looking for a gathering of male drones [6]. The purpose of a far flight is to prevent inbreeding which over time can weaken the colony [7]. During this far-away flight, the queen will mate multiple times and store enough sperm for the rest of its lifetime. Bees do show high levels of self-regulation or self-organization—e.g. during a time when the colony is over-populated, the old queen bee with half the bees population will leave the hive in swarm and search for a suitable location to build a new nest.

Similar to worker ants, the worker bees also perform multiple tasks in maintaining and ensuring the survival of the colony. Some of the worker bees will care for the brood,

feed the brood by mixing their unique concoction of pollen, nectar, honey and some of their saliva to create pellets of “bee bread”. As the bee bread ferments, it becomes easier for the brood to digest. While some feed, the other worker bees goes out to forage for various food sources. The worker bees perform two foraging work concurrently, to collect from two distinct nutrient sources in the form of pollen (nitrogen) and nectar (energy). Studies have shown that these two foraging activities depend on different factors. In [8], they found that nectar foraging does not depend on the honey storage levels within colonies. Individual foragers did not change their foraging behavior in response to the colony’s honey storage level. However pollen foraging activity is extremely sensitive to the colony state. Pollen foraging behavior is modulated by two stimuli: (1) the presence of brood and young larvae, and (2) the quantity of stored pollen [9].

The complex inner mechanisms of bee colonies foraging have inspired numerous works around the “Artificial Bee Colony” (ABC) optimization. This mechanism involves worker bees performing different sub-tasks to achieve a collective behavior. Generally there are three division of foraging worker bees involved: the scouts, employed and onlookers. Initially, the scouts will perform random flights around its nest to look for potential food sources. When a scout finds a food source, it will become employed to this particular food source. In other words, it will always return to this food source until it decides to quit [10]. Each time this employed bee returns to the hive with food, besides passing the food to the relay bees—bringing the food deeper into the hive— it transmits information to other colony members by means of a waggle dance. As part of this dance, it encodes location, predatory factor and profitability of forage sites. This decoded information will either encourage or discourage onlooker bees to follow the employed bee to the designated food source to bring back more food. Interestingly, in [10] it is mentioned that there are other means of social information—i.e. tremble dance (food storer activation) and the stop signal (forager inactivation)—that may affect the foraging activity. These bees not only function as gatherers of food for their colonies, but also as sensory units gathering information for the colony.

When the quality and amount of food source is depleted, an employed bee may “re-sign” and become a scout bee to traverse the environment and search for new food sources. Bees have one of the best navigation systems in the natural world. It is able to constantly traverse to its flower destination even when the environment changes. Currently, there are multiple models proposed mimicking the bee navigation strategy—e.g. some studies suggested that the bees use a mechanism called egocentric navigation which is based on direction responses and actions independent of environmental cues [11, 12].

Flocking Birds and Schooling Fish

In Figs. 1.1a and 1.1b, swarming systems like a school of fish milling in the ocean and a birds flocking display awe-inspiring collective behaviors. Such collective movements like flocking and schooling, observed in birds or fish respectively share numerous universal characters, and often are defensive in nature. Although the complete set of functions associated with such swarming behaviors is still being discussed, the defensive

mechanism of swarming is clearly established: it ensures that no individual stand out to their predators, becoming an easy target. As a swarm, they can benefit from its visually larger size to fend off potential predators.



(A) A school of Barracuda swimming.
Credit: iStock



(B) A flock of birds in the United Kingdom.
Credit: Unsplash/James Wainscoat

These large groups of animals move cohesively through space as they maintain distance—i.e. repulsion and attraction—from each other while also aligning their direction of travel. It has only recently been confirmed that there is no single central entity that instructs the other agents about the speed and direction of travel—i.e. the dynamics of the agent. Instead, these agents adjust their speed and direction of travel solely based on their observations of a few neighboring agents—i.e. purely based on locally acquired information.

1.2 Swarm Intelligence

We have briefly introduced the intriguing and awe-inspiring capabilities of biological swarms that exhibit unique collective behaviors. Over the past three decades, the intelligence of these collective systems has been studied, formalized and discussed. These collective behaviors exhibited, without a central command, have suggested that there may be some form of intelligence in these systems. In [13], Beni et al. first attempted to define the intelligence of cellular robotic systems and coined the term “Swarm Intelligence”. The term caught the attention of the community studying swarming systems. Thereafter, many researchers have also attempted to define swarm intelligence in their own way. These definitions vary depending on the field of study and according to different perspectives of naturalists, biologists, physicists, computer scientists, engineers and roboticists. Although there are a few elements that transcend disciplines and perspectives when it comes to defining swarm intelligence, there are also a number of issues that can be misleading and hamper future works on swarming systems or swarm intelligence.

In these definitions, we have seen the use of “swarm intelligence” interchangeably with “swarm/collective behaviours”. Firstly, intelligence does not equate to behaviors. Behaviors are exhibited and can be observed, quantified, and evaluated. On the other hand, Intelligence is the intellectual capability, which is marked by complex cognitive

feats and high levels of motivation and self-awareness [14]. Ironically, we need to observe and evaluate the behaviors in order to attempt to quantify intelligence. Drawing an analogy from human intelligence, for example the concept of Intelligence Quotient (IQ) test is to put candidates through the test to measure their reasoning ability. The questions were designed to probe the candidates' intelligence; putting the candidates in an environment that requires them to use their intelligence. Likewise, we can only say that swarm intelligence is approximately swarm behaviors given the right environmental conditions illustrated in eq.1.1.

$$\therefore \text{SwarmIntelligence} \approx \text{SwarmBehavior, given the right environment} \quad (1.1)$$

In another trend, we have seen that Swarm intelligence is often portrayed as collective behavior that is decentralized and self-organized. The effect of this is eminent as we observed some swarm research work revolving around consensus study and order in swarm. The concept of "self-organization" emerges in the 1600s, starting with René Descartes, a philosopher studying ancient atomism. This proposed idea takes its place in physics, chemistry and biology. Self-organization is defined as a process where some form of overall order arises from local interactions between parts of an initially disordered system. It is true that the order we see emerging from natural swarming systems are fascinating and conspicuous, but does it speak of their collective intelligence? If we solely measure order as an indication of the level of swarm intelligence, we have to be mindful that we can be blind-sided. One reason being, order can be implemented with a top-down or a bottom-up mechanism—i.e. centralized and decentralized system can both be ordered but achieved in a different way. In reality, often systems combine both types of ordering mechanism. However, even between the two extreme types of global ordering it is not trivial to discriminate between them [15]. This will not facilitate the idea of defining and quantifying swarm intelligence allowing for comparisons among systems. Instead, in [15], Cavagna et al. has suggested observing another signature of swarming systems, collective response, to determine the systems' self-organization—i.e. a feature of Swarm Intelligence. This argument will be expounded in Chapter 2.

The lack of unified definition of swarm intelligence can potentially create confusion and slow the research efforts in swarming studies. This problem is eminent when we attempt to answer questions like "Does this system possess swarm intelligence?", "How (swarm) intelligence is this system?", and "Is system A more (swarm) intelligent than system B?". These questions are often left unanswered. The current trend that we have observed is often that we apply swarming principles, algorithms, etc. to solve a problem, and we say that "Yes, it works. Ant Colony algorithm solved our path planning problem." It is not trivial to determine how well it worked. In order to propel swarm studies forward, we reckon the need to properly understand and quantify swarm intelligence. There is a need for comparison among swarming systems, for the community to confidently say that one system out-performs another system. This new understanding will allow the community to identify areas of improvement to the current swarming systems. In Chapter 2, the idea of swarm intelligence definition and quantification will be expounded.

1.2.1 Artificial Swarm and Swarm Intelligence Applications

The first work that surfaced as an artificial swarm is a work by Reynolds C. named as the *Boids* [16], which is a simulation developed to study the flocking behaviors of birds. This model showed the emergence behavior from the interaction of the individual agents following a set of simple rules. These rules are (1) separation, (2) alignment, and (3) cohesion. This work has contributed massively to the understanding of swarm emergence behavior and from here other extension models like *Vicsek* model [17] is developed.

In the same year, in 1995, Kennedy et. al. [18] developed the *Particle Swarm Optimization (PSO)*, also intended for the study of simulating social behavior of movement of a flock of birds or a school of fishes. Soon, PSO has found its utility in the area of optimization. This work then drives various works on PSO variants, attempting to improve the performance of this meta-heuristic optimization approach and it has continued to inspire the development of swarm-based optimization algorithm like the *Ant Colony Optimization (ACO)*, *Artificial Bee Colony (ABC)*, and more. These approaches have gradually become popular because of its simplicity in the implementation of these meta-heuristic algorithms, even though it can only provides sub-optimal performance, it is extremely useful when the problems we faced are getting more complex.

These knowledge in Swarm and Swarm intelligence eventually found its way to the world of robotics—i.e. Swarm Robotics. ACO and PSO were the first two swarm-based optimization that researchers attempted to apply into real world applications, in fields ranging from simulation and prediction of crowd behavior [19] to telecommunications [20]. However, the initial development of swarm robotics was faced with challenges of applying swarm algorithms to existing robotic platforms. It requires more than simply applying swarm algorithms, there are many considerations on the robotic part of the research that requires the community to rethink about traditional robotic activities like perception, control, localization, and the design of the multi-robot platform.

The development of swarm robotics started in the early 2000s, where the works focused on using robotic swarm to mimic the known behaviors observed in social insects like foraging, clustering, etc. There are works that also replicate the dynamics of observed in natural systems—e.g. clustering in cockroaches [21], which leads to a mixed biological-robotic society [22]. Robotic swarm does contribute to the study of some natural phenomenon observed like the shortest path between a nest and a food source.

There were a few projects that contributed to Swarm research by bringing international research groups together to further the development of swarm. Swarm-bots project [23] (2001-2005) is an initiative to bring the international community together to investigate cooperation in a swarm of robots. This project featured a swarm of up to 20 self-assembling robots that can connect with one another to form different structures. It was used to study swarm behaviors such as collective transport, area coverage and object search [24], [25]. Swarmanoid project (2006–2010) is an extension work from Swarm-bots project to explore the concept of heterogeneous robotic swarms that consists of flying, climbing, and ground robots that can collaborate to carry out a search

and retrieval task [26]. In 2019, Kit et. al. [27] also presented a multi-robot system (ground-based and climbing) to perform a task of collective mapping across multiple floors in an urban setting.

In the last decade, robotic hardware have seen improvement that can benefit the development of robotic swarms, for example improved processors, sensors, and wireless network technologies. This led to a huge step forward for robotic swarms as researchers can now develop smaller, yet powerful, robots. Kilobots is one of the leading work that have seen the physical number of robots in the system soar to as high as 1000 units working together. It demonstrated the large robot swarm designed to perform shape formation [28] and later being used as an open platform to allow other researchers to demonstrate and study properties of swarms. Swarm robotics were not only limited to ground platforms, there were also works that considered working condition of aquatic surface [29, 30], underwater [31], as well as flying [32, 33].

Beyond the horizon of optimization and robotic swarms, Swarm Intelligence has also found its utility in business organizational study, in areas of human resource management and knowledge transfer. In [34], there is a case study of how Jim Donehey, when he was CIO of Capital One, attempts to move away from a traditional micro-managing style. He made use of the concept of implementing simple rules to govern his IT department. He also made use of gaming chips as a form of reinforcement of the rules to give employees reward if they followed the rules. After a year, Donehey claims that these rules unified his group, and also empowered his staff to make decisions on their own and work with little top-down management. The result: the attrition rate is below 4%, compared with 20% for the IT industry as a whole.

It is also interesting to learn that in organizational learning, the concept of self-organization—i.e. exploration and exploitation—comes into play. As the organization desire to progress, the area of learning is important. Gaining knowledge can help to improve the work of its employees, improve current products, and also to bring about new innovations. In order for the employees to improve, they can learn from one another within the organization and also learn from an external source. When learning from within the organization—i.e. from senior employees, other colleagues, or other departments—it is said to be exploiting information that exists within the organization. While learning from external sources—i.e. learning through workshops by other organizations, conferences, etc.—it is said to be similar to exploring new information. Therefore the organization are actually performing a balance of exploration and exploitation of information—i.e. knowledge.

1.3 Research Questions

Although SI or swarm research have progressed to a point of where SI is applied to solve problems in our physical world, a formal swarming system design methodology has yet to exist. This is evident with recent works focusing on creating a formal design methodology [35, 36, 37] such that a standardize design guideline can be achieved to facilitate researchers and engineers in the development of SI solutions.

On this swarming system design and development front, there are two research gaps that we have identified. Firstly, as discussed before, there is a lack of quantification of SI—i.e. how can we measure SI?—to measure the intelligence(swarm) of a swarming system. This element is important as comparisons can aid in the improvement of the SI through its design and development phases. The multiple perspectives of SI gave rise to the multiple SI definitions, which presented themselves as non-unified definitions of SI, could have indirectly contributed to this lack. Furthermore, from these definitions, quantifying SI may not be as straight forward as the definitions proposed did not take quantification into its scope.

Paired with the the quantification of SI, a benchmark tool can further facilitate the development of SI strategies, which is also desired [37]. Benchmark problems have existed in areas like machine learning and optimization. The 'toy' problems provided a standardize reference for users to evaluate their solutions. Therefore a properly designed benchmark framework tool can be beneficial in many ways, (i) standardize method of SI comparison, (ii) measure, analyze and compare SI, (iii) it can expose hidden areas of algorithm, which can only be probed by specific environment, to evaluate its performance, and (iv) a fast prototyping method to tune and evaluate algorithms even before applying to a physical system.

These research gaps and opportunities led me to the following research question(s) which governs the work within this thesis.

RQ: How can Swarm Intelligence be quantified and compared among systems?

SRQ1: What is Swarm Intelligence?

SRQ2: How can Swarm Intelligence be quantified?

SRQ3: How can we develop a benchmark framework to evaluate and compare Swarm Intelligence of different systems?

1.4 Research Contributions

From the above research questions, the following are the contributions of this thesis work.

1. Quantification of Swarm Intelligence
2. Application of SI in multi-robot system
3. Benchmark framework to compare and analyze flexibility of swarming system.
4. Qualitative study on the transferability of Swarm Intelligence

Chapter 2

Quantifying Swarm Intelligence

2.1 Introduction

Bio-mimicry has been a significant framework for us to find solutions to the many problems that we faced today. The natural world has inspired solutions like gecko-feet for climbing, velcro, high speed train, venturi effect of termite mounds, multi-robot systems etc. A part of bio-mimicry solutions were inspired by natural swarming systems also commonly dubbed as the complex problem solvers—e.g. optimization, multi-robot control mechanisms, swarm robotics, etc. In recent years, the intriguing capability to solve complex problems without a central command—i.e. fully decentralized systems—has garnered research interest from different area of studies from engineering, science, business, mathematics, computer science, etc. This capability displayed by the natural swarming systems has been said to be "intelligent". In the literature, there are a few perspectives regarding the intelligence of swarming systems—i.e. the different definitions of swarm intelligence. Coupled with the lack of unification on this subject, with various definitions, it becomes unclear if a particular system truly possess swarm intelligence. This suggests that SI of a particular system can only be observed (not measured) and it can be subjective, debated or unclear.

Beyond the unclear definition of SI, there are also some unanswered questions which calls for more research attention. If we say that a swarming system is (swarm) intelligent, will there be a time where it display less (swarm) intelligence than the other? If there is, then in which situation(s) will it display a higher level of (swarm) intelligence? In this case, if we take SI—i.e. a swarm solution—out of context, the solution might not be as (swarm) intelligent. Therefore suggesting that there is a 'boundary' for which SI of a particular system exists, and this boundary is what we want to investigate. This is important to allow researchers and engineers to know when to apply swarm algorithm solutions to their problems to yield the benefits of SI, as there will be times when swarm solutions might not out-perform a centralized multi-agent or multi-robot system. For example, by applying a swarm robotic platform to map a room might show that swarming algorithm/principle works for a mapping operation and it might also perform better than a single robot platform in terms of a faster mapping process, but we wonder if the application of swarm intelligence is necessary. In other words, is swarming systems necessary in all situations that we see?

We would like to suggest that this might not be the case. On the contrary, in many operations, non-swarming systems or even a centralized system might suffice. For example, an application of swarm robotics in an automated warehouse is not necessary, given that the warehouse is mostly a static environment and have pre-planned (optimized) procedures to save cost, a centralized multi-robot system works perfectly to solve the logistical problem—e.g. the application of amazon robots in their warehouses. In this scenario, a swarm robotic solution might be a case of over-engineering. It can even be detrimental as the development process of swarm robotic solution might take longer given the nature of the unpredictability of swarms' collective emergence behaviors.

Not being able to understand swarm intelligence on a deeper level, will cause us to apply swarming algorithm/principle to artificial swarm "blindly" and not yielding the full potential of SI. In the application of SI to artificial swarms, researchers and engineers work with different sets of constraints and environment from nature, thus a "plug-and-play" application, might cause the solutions to be not "optimized" for the various tasks that the artificial swarm performs. Therefore artificial swarm design suited for the scenario is crucial, and in my opinion, not possible without a proper quantity or metric to describe the SI of a proposed solution. Currently, in the literature, there are not much work done on metric to measure SI, therefore less conclusive comparison can be reached. Noting that the operational performance of a system does not necessarily equates to SI of the system. Without comparison, it is not possible to have a methodological approach to improving swarming strategies.

The clear need for further understanding and quantification of SI motivated this research work. As we proceed with this chapter, we will re-visit SI in the current literature, presenting 1) SI Definitions, 2) SI Applications, 3) SI definition proposal, 4) Potential outlook in the following sections.

2.2 Swarm Intelligence Definitions

Over the years, multiple SI definitions were formed mainly from two broad perspectives—i.e. bio-inspired and algorithm-inspired—of the research community. The pioneering definition being Beni et. al.'s [13], Definition 2.2.1 ("Swarm Intelligence"):

Systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to unpredictably (#) produce specific (=not in a statistical sense) ordered patterns of matter in the external environment.

Unpredictable is meant as globally 'intractable' or 'externally not-representable'.

Interestingly, this first work that uses the term SI and defining it, is a work on cellular robotics. The above definition captured a few important essence of self-organization. Firstly, it mentioned "systems of non-intelligent robots". This reflects the often simple units (from a macro perspective) we find in nature—e.g. an ant, fish, bird, bee, etc.—existing in a colony or a group which display an intelligence far beyond the individual's. In complexity science, this is described as "the whole is more than the sum

of its parts". Secondly, it mentioned "the ability to unpredictably produce specific ordered patterns of matter in the external environment.". This is in fact the emergence of the collective behavior (self-organization) which can often be observed as "ordered patterns"—e.g. a flock of birds migrating, or a school of fish evading predators.

One camp of SI perspective that branches off from Beni et. al.'s SI definition were the perspective of naturalists or biologists, who works with insect colonies, fishes, birds—i.e. social animals and insects. They provided a definition of SI from a perspective of their long years of observation and study of these animals [38, 39, 34, 40, 41, 42]. In these work, self-organization, that is often observed in nature, is often described to be related or bring about SI. It includes common features involving 1) groups of simple agent and 2) global emergent or self-organization. In which three definitions also included 3) inter-agents local interactions. This definition encompass feature 1 and 2 that are similar to that of Beni et. al.'s definition, with an extension of feature 3 which is not captured in Beni et. al.'s. The three features sum up SI suggested in [43].

These three features mentioned can be observed in nature. For example when a sub-colony of ant foraging for food, "intelligence" can be observed as the ants identify a food source and are able to move back and forth from the food source to its colony, and when it gets depleted, the ants are able to adapt and move on to identify other food sources. This level of food foraging self-organization emerges from repeated local interactions amongst the simple agents—i.e. the use of pheromone trails which is a unique form of communication via external environment; called stigmergy. Without the central command, nature seems to be effective and efficient in their processes like foraging, navigation, task allocation, brood care, and more.

The second camp of SI perspective from the development of SI algorithms. As SI research progresses, for the purpose of further studying of social insects and animals, it has motivated the initial development of SI algorithms. Quickly, SI algorithms find its utility in solving optimization problem which is expected as natural systems perform optimization in food foraging, navigation, internal maintenance of hives, etc. As the works in swarm optimization expands, algorithm-inspired SI definitions [44, 45, 46, 47] surfaced mainly describing SI as bio-inspired computational methods (or algorithms) to solve optimization problems. Some would say that swarming systems are basically a collective information processing unit. In this case, SI governs the process at which information is acquired, processed, making collective decision, and acting on it. In [48], a collective operation and computation framework is proposed, in which a part is "local computation" that each agent performs using only information from a certain local neighborhood. This emphasize the importance of local communication and computation—i.e. local information processing and decision making with (local) information e.g. from local neighborhood.

Although there are similarities in the definitions, the perspectives are still vastly different and within each perspective the definitions also hold differences. This presents a long-standing problem in swarm intelligence study, that is the issue of non-unifying SI definition [43, 49], which can also be inferred from earlier work that calls for quantification of SI [38]. The lack of a unifying definition, hence theoretical foundations, would pose significant problems to quantifying SI and the future development of SI systems.

Also, the numerous definitions of swarm intelligence proposed are non-quantifiable—i.e. qualitative definitions. This implies that SI comparison is still largely implausible.

We could not make statements like "Swarming system *A* is more (swarm) intelligent than swarming system *B*". The lack of quantification of SI will not allow SI comparison among system, noting that there is a difference between system's performance and system's SI. In many of the works, operational performance of the systems are often measured for comparison—e.g. machine learning, optimization research, etc. This provide a way for the basis of improvement and further development of the technology. For example, in one of the most use case of SI, optimization, in a particular task like foraging, we can compare the performance of the use of Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) algorithm. If we use a general metric of "total amount of food gathered at the end of the simulation", most likely one algorithm will outperform the other. However, operational performance does not necessarily correlates to SI—i.e. the system can perform well and still not displaying SI.

The confusion that stems from the vastly different definitions of SI, also entails missed opportunities as well. Now, it is not known the "right" conditions where SI is the most effective. Without a clear understanding of what SI is and how and why it arises, it is very difficult to envision how to take advantage of its true potential. Often, this confusion trickles down to research that freely use the term SI when large numbers of entities are applied. Not having the method to compare SI will cause an eminent effect of artificial swarming systems not being able to improve and be more intelligent, which can stifle the advancement of future swarm studies, real life applications and commercialization of this technology. Therefore this work is important to pave a way for a quantifiable swarm intelligence.

2.3 Swarm Intelligence, Flexibility and Adaptability

After studying the various SI definitions, it is evident that self-organization (SO) or emergent behavior is a central and important feature that describes SI, also mentioned in [43]. SO being the central feature of SI arises from the interplay of four lower-level components: 1) positive feedback, 2) negative feedback, 3) amplification of fluctuations, and 4) multiple interactions [38]. It gave rise to the many behaviors we observed in nature.

This brings us to another question, "What is SO or emergent behavior?". Still these terms are ambiguous as there are many interpretations. Serra et al. [50] describes the concept of SO generally as "highly organized behaviour even in the absence of a pre-ordained design." They go on to further describe examples such as the resonance phenomenon in lasers, and in cellular automata where "unexpected and complex behaviours can be considered as self-organized."

As a fully decentralized system, without central command, the sub-systems are able to communicate and perform its function which leads to the emergence of collective behaviors. Often, the noticeable effect of SO is the order that we observe in nature.

As mentioned before the migration of birds, schooling of fishes, etc. One example of quantifying order is the measure of the direction of travel of the birds compared to the global direction of travel to find the order of the system [51]. If the system is in order or is coherent—i.e. the difference between all the agents' direction and the global system's direction is low—then it suggests a high level of order, while a high difference will suggest a low level of order.

However, the challenge in measuring order as an indication of SO—i.e. SI—is that order is not an outcome of the collective behavior, caused by SO, that is unique to SO. The complication arises as order can be derived from both the centralized and decentralized paradigm. It is not trivial to distinguish between a centralized order and a self-organized order—i.e. a top-down centralized order vs bottom-up decentralized order. Furthermore, as pointed out by Cavagna in [52], systems' hierarchical and distributed control are often a combination of centralized and self-organizing(decentralized) control mechanism. There is an interesting work by Alessandro A. et al. [53] that documented the discovery of a display of collective behavior (SI)—i.e. the emergence of global order or pattern—without collective order in wild swarms of midges. Despite the lack of order, the swarms displayed high level of correlations, totally incompatible with models of non-interacting particles. This work challenges the long-standing understanding of self-organization order.

In the literature, there exists multiple definitions of SO. The common definition of SO is often described on both the micro and macro levels [49, 40, 38]. The micro level representing the local interactions among simple agents and the macro level representing the overall order and pattern that emerges. SO is a mechanism that results in a global pattern to emerge by means of low-level interactions.

There is another self-organization perspective by Prigogine et al. [54] that described self-organization in the chemical reactions and thermodynamic contexts. "His notion is quite enlightening and emphasizes the element of fluctuations in far-from-equilibrium system states. Non-linear system changes, fluctuations as he refers to them, due to either random events or chaotic dynamics, are then amplified by positive feedback mechanisms. This results in structures that emerge spontaneously which often are presented as examples of self-organization." From a similar perspective, Cavagna et al. [52] proposed a method in distinguishing and evaluating self-organization paradigm from centralized paradigm that is not the collective behavior of order that we observed of swarming system, instead collective response can be the alternative measure of SO. Collective response, aka flexibility, is a trademark of self-organizing order and it represents how well a system reacts to a dynamic environment—i.e. the system's response to perturbations. If the system reacts and adapt to the environment changes, that would provide a way to suggest the existence of a self-organizing feature of the system. Both Prigogine et. al. and Cavagna et. al. perceived self-organization as the ability to adapt(structures that emerge spontaneously) to a changing environment(external perturbations).

This collective response that Cavagna et. al. [52] mentioned is the flexibility of the system. From the literature, flexibility also have various meanings. Firstly, some define flexibility as the ability for the system to perform as well in different environments or

scenarios by maintaining the behaviors of the individuals of the system [55]. Different environments do trigger a different level of collective response from the same system, in order maintain its performance, the system will be required to alter or tune its behavior to adapt to a wide range of environmental conditions—i.e. adaptability—as SI is environment dependent. We reckon that it is not possible for a non-tune-able system to maintain its performance in the different environment while maintaining the behaviors of the individual agents.

Secondly, the flexibility definition that we use in our work, is the ability of a system to adapt to a changing environment—i.e. how the system collectively respond to external perturbations [56, 52, 41, 38, 57, 27]. Fukada et al. [56] used a similar flexibility definition however he evaluated it from the perspective of flexibility being a trait of robustness. In general, like scalability, flexibility and robustness are individual feature of a swarming systems [38]. Studies in swarming systems have paid lots of attention to features like scalability and robustness. However, over the years, the key feature of flexibility has not gain as much attention. One of a reason could be that it is not an easy feature to probe and measure.

In practice, flexibility can be an observation of the system's performance right after a change has occurred. The derivative of the performance of the system after the change—i.e. early dynamics—can describe its effect on the system. In our work [27], we have discovered a visible slowing down of the system due to a change, we called it the *recovery period*.

The system's ability to be flexible or collectively respond to a changing environment requires effective information processes. Some would describe a swarming system simply as an information processing unit—i.e. acquiring information, to processing information, to collective decision making, and finally to acting on it. One interesting perspective to take, in regards to the information process, is the concept of exploration and exploitation. In a system, the action of acquiring information can be seen as exploration, while processing and acting on these information can be seen as exploitation. When the environment is static, the system can settle in an ideal exploration-exploitation balance that acquires enough new information and acting them such that they are efficient. However, it becomes interesting when the environment changes. This can be an analogue to kicking the rope when an acrobat is on it, and we can then observe how the system (the acrobat) regain its stability, achieving a new exploration-exploitation balance. The system's flexibility can be described as the ability to restore an exploration-exploitation balance suited to the current environmental conditions.

In the future, the development of SI should seek strategies with tunable exploration-exploitation balance for the system to adapt to different problems—e.g., in the analogy of rope balancing, the system can balance well on a different rope. This tunability—i.e. adaptability—should bring forth a new wave of adaptive SI strategies.

2.4 Exploration-Exploitation balance in Swarm Intelligence

The concept of exploration and exploitation (EE) is essential when studying adaptive processes like SI, where in some strategies, behaviors (exploration-exploitation balance) can be tuned to suit a particular environment or problem. EE trade-off happens often in many systems that are around us. It can be observed in areas like psychology [58, 59], research [60], organizational learning [61], financial investment [62], machine learning [63, 64, 65], eusocial natural systems [66], and more. In general, it poses a dilemma in which a choice has to be made to either 1) choose an optimal decision, assuming the current knowledge is reliable 2) choose a sub-optimal decision to gather new information, assuming that the current knowledge is inaccurate. For example, a scientist is required to perform experiment or simulation to produce scientific contributions while having to study other works in the same field. However, if the scientist spend too much time on studying other works, it will reduce his or her own quality of research results. If the scientist wants to maximize his/her contribution, how much time should he/she spend on studying other works vs. his/her own research?

In the earlier days of EE dilemma discussion, large amount of work focuses on organizational learning and organizational adaptation research. Since March's pioneering article [61] that uses the terms "exploration" and "exploitation" in organizational research, it garnered attention to the existence of this duality. It explained that the effect of lack in either exploration or exploitation can be detrimental to the organization. For example, if an adaptive system that engages purely in exploration can incur high experimental cost without gaining much of its benefit—i.e. creates many undeveloped new ideas and display little distinctive competence. On the other hand, one that engages purely in exploitation will likely find themselves "trap" in a sub-optimal pseudo equilibrium. Therefore March suggested that there is a need to balance EE for the system's survival and prosperity. However, the balance of EE is not trivial. The returns of these two modes are not immediately comparable and therefore the difficulty in choosing the right level of these modes for an optimal system performance. For example, the returns of exploration is not predictable and delayed as it is an "investment" to hopefully gather new information(returns) in uncertain environment.

In the literature, there are two proposed relations between EE. One of such proposed relations by March [61] suggests that EE can be viewed as two ends of a continuum. This is supported by one of his assumptions that EE uses the same pool of finite resources—i.e. when exploration uses more of the resources, naturally exploitation will have to be reduced. However, this perspective can differ if (some of) the resources are infinite like information and knowledge [67]. At times, systems have access not only to the resources they own but also to resources in their external environments (or provided by an external source) [68]. As such, the access to external resources could ease the finite resources constraint imposed on the system before. This led to Katila et. al. [69] conceptualization of EE where relevant resources do not suffer from scarcity, and therefore these variables are proposed to be orthogonal—i.e. statistically not correlated.

The concept of EE have been used in adaptive process research, and it can be applicable to our work on SI, since adaptability (and flexibility) is one of SI features. SI is highly beneficial to overcome 1) environmental change and 2) initial conditions affecting the

performance of the system—e.g. the impact of agents clustering in an area before commencing on a search(exploration) mission. Even in a static environment, SI can be beneficial to overcome different initial conditions. When the system is under such circumstances where uncertainty exists, it needs to adapt and recover to continue performing its intended function—e.g. foraging, navigation, or search and rescue mission. The two different modes of EE serves specific purposes in information gathering. Exploration allows system to search for new information such that it can recover or gain new environmental insights including escaping local minima/maxima. Exploration exists in multiple forms, from intentional exploration—e.g. algorithm that plans for exploring unknown spaces—to unintentional exploration—e.g. repulsion from other agents, error, etc.. In [38], it is mentioned that SO relies on amplification of fluctuations (random walks, errors, random task-switching, etc.) and these randomness is crucial for the discovery of new solutions. On the other hand, exploitation make use of available information to further ‘polish’ the existing knowledge.

Therefore for a true SI to exist, it requires adaptability and flexibility as core features, and we argue that the system needs to possess both exploration and exploitation in order to deal with various environments or change in environmental states. From the perspective of SI, EE has to be mutually inclusive—i.e. co-existence of exploration (to gather new information) and exploitation (utilizing known information)—such that there exists a “best performing” balance for a specific environment or situation.

2.5 Swarm Intelligence and its environment

In order to probe a system to exude SI, it needs to be operating in an environment that is challenging and difficult enough such that SI can be pushed to its limit, requiring a right balance of EE. In Chapter 5, these level of difficulties translate to a constant environmental uncertainty of the locations of occupied cells, usually clustered into pillar(s), for the system of agents to discover. Also, the number of these pillars of workload remain unknown to the system throughout the entire operation. The operation increased in complexity when the changes of locations of the pillars occur, which require the system to increase its exploratory action to combat the instant lost of environmental information. In [70], it pointed out that environmental factors does affect the exploration-exploitation behavior, in the case of the above simulation, 1) depletion and replenishment of resources, 2) structure of the environment, 3) stability and predictability of the environment motivate the system to find its exploration-exploitation balance.

In a warehouse, like Amazon’s or Alibaba’s, where logistical multi-robot system are implemented to transport packages to various destinations within the warehouse, operates in an extremely ordered and planned environment. This style of operation is driven by the need to reduce time and cost factors from procurement to delivery, to get the packages to their customers as soon as possible with the lowest cost. In such an environment, the routes in which the robots take in the warehouse have already been optimized and the robots will only be required to follow instructions given by the centralized warehouse system. In this environment, the nature of SI’s unpredictability,

due to emergence, might offer a sub-optimal performance or can be even detrimental to the performance of the system. Furthermore, this unpredictability can incur a higher cost during development phase. This example shows that it is not necessary to call for the application of SI strategy even in a multi-robot system solution. Therefore the environment and the problem plays a key role in determining the use of SI, and the effectiveness of SI. This also rise a question of whether some SI solutions are over-engineered for the problem that they are solving.

From our perspective, the display of "intelligence" goes beyond the system—i.e. SI cannot be quantified with the system alone, without the environment (context). SI can be observed when the SI system is in a 'best' suited environment, but when the environment changes it could keep SI of the system muted. The new environment can be said to be less complex or not "difficult" enough so as to require the system's SI while operating from within. The system then exercise much less "intelligence" and is seen as such. More often than not, SI is often described in the absence of the "environment" in which the agents operate in. We would like to propose that:

Swarm Intelligence is not an intrinsic property of a system. It is a property of the system evolving in its environment

2.6 Conclusion and Potential outlook

In this chapter, we have discussed the various SI definitions from different field of studies, the relationship among SI, flexibility, and adaptability, the exploration-exploitation dilemma present in SI, and finally discussing about SI and its environment.

We have then proposed to measure flexibility as an indication of the level of SI, as it is a unique outcome of collective behaviors arising from SO—i.e. SI, that cannot be easily replicated by non-SI systems. As we have mentioned, there can be SI and non-SI solutions to a problem, and SI performance is not necessarily dominant. This suggests that in the near future, there is a need to compare performances of SI and non-SI solutions and we need to be aware of comparisons beyond the overall performance of the system—e.g. total number of cells discovered in Chapter 5. Therefore with a better understanding of the areas of performance that SI can bring, there is a need for a quantity, like flexibility, that can differentiate SI from non-SI solutions.

By observing the early dynamics of the system after a change, it can describe the *recovery period* of the system. How long does it take to rediscover the new change? As compared to the other systems, how quickly can I find the new change and continue with my objective? The analysis of the system's flexibility can be supplementing with the concept of exploration-exploitation behavior of the system to further provide a clearer explanation for the system's actions.

In this area of SI research, there is a need for unification in terms of the understanding of SI—i.e. SI definition and SI quantification and metrics. The development of a SI benchmark framework can bring about this unification, for algorithms to be quantify and compared to other algorithms. In order "to concretely support the research community, benchmarks are invaluable tools to measure the progress in a quantitative way

and can be used to challenge the researchers on tasks that grow in complexity.” [37]. This draw similarity to OpenAI(a reinforcement learning problem library), where there are multiple environments for algorithms’ performance to be tested—e.g. through a standard cart-pole problem.

Chapter 3

Experimental Investigation: Decentralized multi-floor exploration by a swarm of miniature robots teaming with wall-climbing units

This chapter is taken whole from [27] by Jabez L Kit, Audelia G Dharmawan, David Mateo, Shaohui Foong, Gim Song Soh, Roland Bouffanais, Kristin L Wood.

Before designing the SI benchmark framework, it is important to experience the workings of a SI solution operating in an environment to understand the various components needed to simulate an SI operations [27]. Here in this work, the flexibility of the system is evaluated in a controlled environment, and the recovery period of the system can be observed.

In this chapter, SI have been applied to a problem of collective mapping, coordinating a swarm of miniature robots spread across multiple floors. Physical experiments have been conducted to evaluate the scalability, robustness and flexibility of the system—i.e. features of a swarming system, eventually demonstrating its ability to collectively explore and map multiple floors in an urban setting.

3.1 Introduction

Autonomous robots are good contestants for intelligent surveillance and reconnaissance (ISR) operations in remote or hazardous environments preventing direct human intervention. One central challenge in ISR operations, however, is the ability to perform effective exploration of dynamic environments. Single-robot autonomous exploration is unquestionably ill-suited for such tasks. Coordinated multi-robot exploration represents, theoretically at least, a viable alternative [71].

Biological multi-agent systems—e.g. bird flocks, schools of fish, ant colonies—are capable of performing a wide range of collective behaviors in a fully decentralized manner [72]. These swarming systems present valuable insights into the development of decentralized, scalable and fault-tolerant multi-robot systems (MRS) that are required to operate in dynamic environments. Swarm-based designs of MRS therefore appear to be a promising strategy for ISR operations in dynamic and unknown environments. However, as highlighted in [73], the transition from a robot-centric design to a system-centric one requires to consider critical elements beyond the electromechanical aspects at the robotic unit level.

Decentralized MRS provide the greatest flexibility in both design and operations, while being afforded with some of the highest levels of fault tolerance. However, these very desirable features come at the cost of an overall increase in the system's complexity. Specifically, for a decentralized MRS to effectively operate in unstructured dynamic environments, the robotic system requires: (i) a distributed communication channel to share state variables and sensed data through an interaction network of a particular topology [74], possibly time-varying [75], (ii) a collective decentralized computing framework processing in real-time, data shared among units [76], and (iii) motion planning or collaborative control strategies, which are key to the effective coordination and division of labor among units having possibly different capabilities [71]. In practice, the divisions between these three key elements are not as clear-cut as it seems. On the contrary, MRS network architecture, distributed computing, and collective motion planning are profoundly intertwined, which is one of the reasons behind our incomplete understanding of biological swarming [72].

Once a communication channel is autonomously established without any external supporting infrastructure, the MRS requires situational awareness as it starts charting its unknown surrounding environment. This classically takes the form of a map, which has to be constructed and updated while a coordinated exploration strategy drives the system based on the current map information and new sensor data harvested by the various mobile units. Among the many different types of maps, the most common and intuitive are occupancy-grid maps (OGM), which are particularly befitting to MRS operations [77]. Other types of maps include feature-based maps (e.g. line maps) and topological maps, which are significantly more computationally and memory efficient than OGM, yet require advanced sensor-data processing to generate an accurate and reliable map [73].

As its name implies, OGM reduces a two-dimensional (2D) surface area into a grid of cells, which are characterized by several possible states—obstacle, unexplored, explored—identified through a probabilistic treatment of the noisy and uncertain sensor data. Range-sensors are particularly well-suited to the generation of OGM. For our purposes, given that we consider a swarm of miniature robots mapping dynamic environments, OGM constitute the optimal solution for three fundamental reasons. First, the smallness of our individual units requires effective obstacle sensing that can be miniaturized accordingly. Ultrasonic sensors have achieved small footprints, but their range is too short and are hardly as accurate as optical sensors. As is detailed below, our custom-made miniature LiDAR apparatus provides an effective-range sensing capability [78]. Second, OGM being based on probabilistic estimations, they are intrinsically

more tolerant to the presence of temporal changes in localized features, and thus lend themselves naturally to the mapping of dynamic environments. Third, the map building/updating process is almost identical when considering a single robot or a MRS, thereby conserving decentralization of the system's operations without compromising its scalability. The key challenge with OGM is its relatively high computational requirements when maintaining large grids [73]. However, this identified limitation can largely be alleviated when considering a decentralized computing framework as well as a distributed storage of the map. With a truly decentralized MRS—swarm robotics ones being one such particular type, individual units need only store and process a small subset of the sensed data and OGM. Effectively, not a single unit possesses the overall map, which should not be required for path-planning purposes.

The network architecture in MRS can either be centralized or decentralized. In the former, computation and control are performed by a single central entity, while in the latter, they are performed locally by the robots with minimal communication among the modules. The decentralized architecture is thus less prone to being affected by a single point of failure [73]. For this reason, and owing to recent technological developments, decentralized MRS architectures are getting prominence. For example in [79], a decentralized algorithm was used to localize a flock of robotic sensor networks. Environmental monitoring tasks were performed by a decentralized swarm of robots in [80, 81, 82], including with heterogeneous swarms [83]. Decentralized exploration and mapping of unknown indoor entity was demonstrated with a pair of autonomous quadcopters in [84].

Despite the number of works on decentralized MRS with certain capabilities, it was found that existing MRS/swarm systems lack supporting experiments and performance data [85], for example on how well they behave under varying number of robots (scalability), against the failure of the individuals (robustness), as well as in response to unknown and dynamic environments (flexibility), which are the key properties of swarm robotics [72]. It was also observed in [73] and [86] that existing MRS were mostly applied to static environments and very few works have been done in the presence of dynamic circumstances. In terms of exploring unknown domains, it was noted in [87] that while mapping of individual floors of an entity has been considered, simultaneous mapping of multi-floor territory has not been much addressed. This capability can add value to the overall mapping task.

The ISR of modern urban environments, whose dense population navigate vertically as much as horizontally, requires of multistory exploration and monitoring [88]. There is no doubt that both tasks are extremely challenging, and particular so for fully autonomous and decentralized MRS. Beyond the dynamic nature of urban environments, there are three broad types of challenges associated with their autonomous exploration and surveillance: (1) intermittent accessibility to some spaces due to closed doors, (2) floor-to-floor transitions through staircases and/or elevators, and (3) ability to establish a communication channel in order to achieve inter-agent information transfer across wireless-hindering physical obstacles such as walls and floors. Attempts have been made to address these challenges separately: e.g. autonomous multi-floor exploration by a single robot [89, 90], semi-autonomous exploration of multi-floor buildings with

a legged robot [91], navigation for service robots in the elevator environment [92], autonomous multi-floor navigation by micro-aerial vehicles [93]. However, to the best of our knowledge, the autonomous exploration and monitoring of an unknown indoor space consisting of multiple floors with dynamic features by a fully decentralized MRS remains an open challenge.

In this paper, a decentralized and heterogeneous system of custom-built miniature robots for autonomous exploration and mapping is developed in-house and evaluated to address those challenges facing current MRS. The performance of this swarming system in terms of its scalability, robustness, and flexibility is extensively tested and quantitatively evaluated. The system is also then tested to simultaneously map two different storeys of considerable size within a campus environment, during normal operating hours and in the presence of students and staff. For the multi-floor mapping experiments, the large gap between the rooms/floors and the concrete walls/ceiling poses an operational challenge in terms of maintaining distributed communications required for our MRS to operate with robots on different floors. Intermediate communication relays located between the floors are necessary to ensure that a communication link is maintained between the swarms evolving on each floors. Indeed, having humans entering the area to physically place static nodes on the walls—especially in the case of hazardous environments—is neither desirable nor practical. This issue is overcome by having a heterogeneous MRS consisting of swarms of land robots teaming with multiple wall-climbing units. The wall-climbing units and ground-mapping ones share the same core architecture in terms of hardware and software; both variants have different sensory suites and mobility apparatuses.

3.2 System Architecture

3.2.1 Individual robotic unit: O-climb and O-map

Our MRS system is called ORION, and multiple generations and variants of ORION base units have been developed [78, 94, 95, 96, 97]. Although this paper focuses on ORION's system-level design, features, and performance, we nonetheless provide in this section critical unit-level information and details about the two key variants: the ground-mapping (O-map) and the wall-climbing (O-climb) units, shown in Fig. 3.1. Specifically, we report on the novel design of ORION's units as the main modules for the MRS and its unified system architecture. The hardware architecture of ORION can be mainly categorized into two parts: the chassis and the reconfigurable wheels. The chassis houses two DC motors and the electronics required for ISR as shown in Fig. 3.2 (see [94] for more details on the individual components). Sensors can be added or removed easily from the electronic architecture depending on the specific needs of a range of ISR tasks. The electronics stack is divided into two independent systems controlled by two software layers: the high-level and the low-level layers as shown in Fig. 3.3.

The high-level layer implements decentralized autonomous behaviors and performs the computationally-intensive tasks associated with them, such as computer vision,

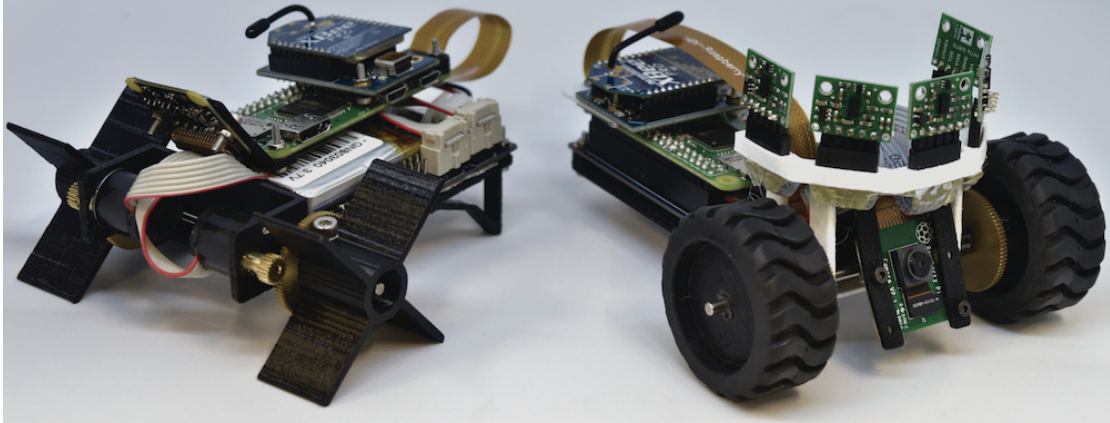


FIGURE 3.1: O-climb (left) and O-map (right).

map-building, and path-planning. The low-level layer manages the platform sensors, actuators and power for real-time dynamic control, obstacle avoidance, and waypoint navigation. The two system layers communicate through a UART serial connection. The low-level layer constantly sends the sensors' data and robot's state to the high-level one for distributed autonomy processing. The high level, in turn, returns instructions at the low level for robot task execution.

With a common electronics stack on the chassis and reconfigurable wheels, the two design variants developed for the ORION MRS, each has its own set of unique capabilities: (i) O-map: a ground robot equipped with mapping function, live-video feed and distributed autonomy capabilities [78], and (ii) O-climb: a climbing robot equipped with live video feed and inter-floor communication capabilities [94, 95, 97], capable of robust internal and external transitions [96]. As shown in Fig. 3.1, O-map units (20 such individual units are operational) are equipped with rubber wheels and ball caster for ease of mobility on various terrains, while O-climb units (4 such individual units are in service) have special wheel-legs with compliant adhesive tape and a tail for robust climbing [95, 97, 96, 98]. The custom-built Light Detection and Ranging (LiDAR) sensor mounted on O-map [78] is omitted in the O-climb variant as it does not perform mapping and to reduce its weight for ease of climbing. This shows the high level of modularity of the ORION's platform design for heterogeneous MRS purposes.

3.2.2 Decentralized Swarming Technology

While decentralization denies a MRS the advantages of a large core computational and communication hub, it offers three highly sought-after features. First, scalability: the system is capable of cooperatively operating under a wide range of system sizes. Second, robustness: the MRS architecture allows for fault-tolerant operation when faced with sudden changes in its interaction network topology and the potential loss of multiple units. Third, flexibility: with appropriate cooperative control strategies, the MRS is in principle capable of operating under unknown and dynamic circumstances. For effective decentralized MRS operations, the robots should be able to communicate in a distributed fashion establishing a dynamic (i.e. switching) communication network

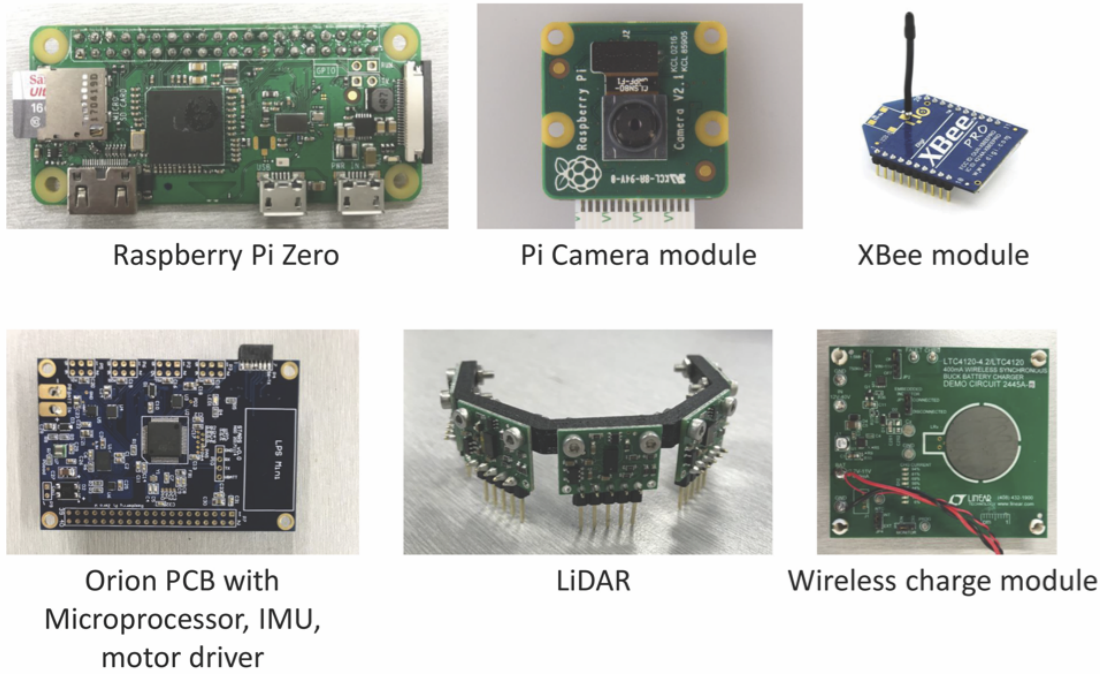


FIGURE 3.2: The electronic suite of hardware components of the ORION platform. Modules on the bottom row—PCB, LiDAR, and wireless charger module—were designed and custom-built for ORION.

where nodes can be added or removed without compromising the MRS operations. To grant ORION these features, the robots are equipped with a swarm-enabling unit (SEU) based on the specifications detailed in [99]. This SEU is platform-agnostic and has successfully been used in a swarm of land robots with varying interaction network topologies [75] and in a heterogeneous swarm of surface vehicles performing a range of cooperative control strategies [83, 82].

Specifically, this is achieved with off-the-shelf, low-power and low-bandwidth wireless XBee-PRO modules capable of creating a distributed mesh network. The communication channel reconfigures itself automatically as the robots move and enter or leave each other's communication range. The performance of this distributed communication approach has been assessed with up to 40 robotic units with distances up to 160 meters and is reported in [82]. This communication network is used by the robots to continuously broadcast their states—their estimated positions, sensor readings and current navigation waypoint—at a rate of 5 Hz. The maximum expected communication range is about 310 m in line of sight and the modules are capable of relaying messages through multiple hops in the network.

The decentralized swarming design principles and the cooperative control strategies are such that the successful collective operation of this system does not require a reliable global communication network between all the robots. The motion of each platform at a given time is determined solely by its own state and the current state of the neighboring robots. Therefore, only short-range and intermittent local communications are required.

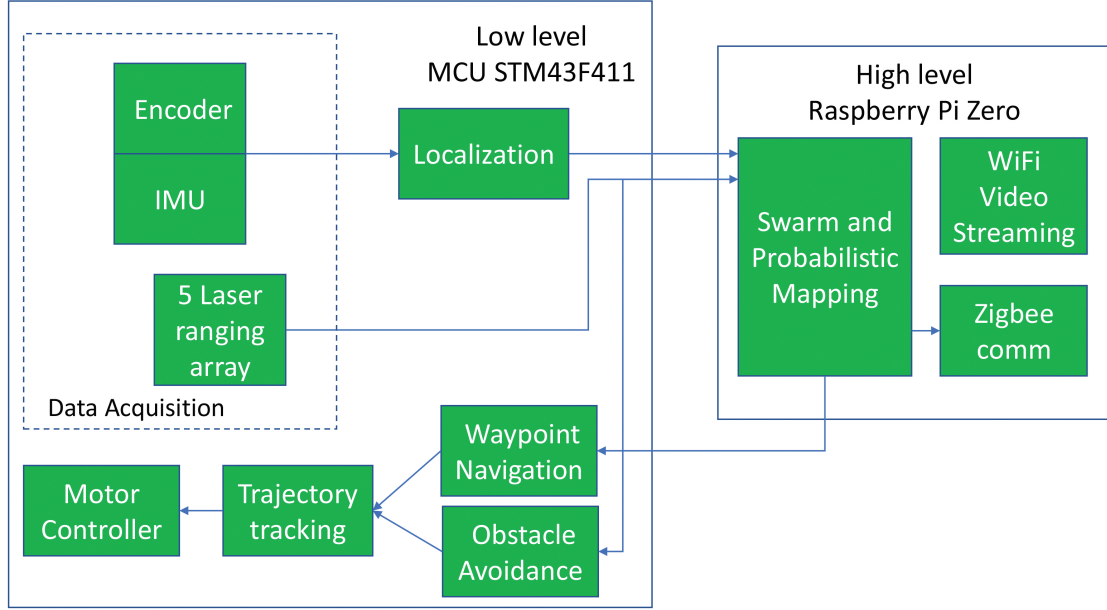


FIGURE 3.3: Multi-block diagram of the developed software framework for ORION.

3.3 Collective Mapping Approach

The decentralized ORION system of ground robots is designed to perform collective mapping operation in the absence of a central command. This bars the possibility of system-level algorithms of coordination, global path-planning, or workload distribution among the individual units. Instead, the collective mapping operation arises as emergent behavior from unit-level algorithms fed with information from neighboring agents.

Since our focus is on cooperative-control strategies for collective mapping, we do not consider the full problem of Cooperative Simultaneous Localization and Mapping (Co-SLAM). We assume that all individual units know their initial poses in a common, global reference frame, and that a localization based on the unit's Inertia Measurement Unit (IMU) and the wheel encoder provides a sufficiently accurate positioning for the duration of the mapping task [78]. If higher accuracy is required, probabilistic localization algorithms such as unscented Kálmán filter can be implemented as reported in [99].

3.3.1 Map Representation

ORION senses its surroundings by means of LiDAR readings that provide a point cloud of nearby obstacles. This point cloud is then incorporated to a 2D OGM representation of the environment (see [78] for more details).

The 2D environment is divided into a regular grid of cells with an associated state representing the posterior probability of occupancy. Each square cell has a size of $1/15$ m and for simplicity all of them start with a prior occupancy probability of 0.5. With each LiDAR sensing, the probability of all the cells in the line-of-sight associated with the sensing are updated according to a Bayesian approach [100]. This classical OGM approach naturally lends itself to the mapping of dynamic environments since the state of a given cell can vary over time following the displacement of obstacles or the opening of a door.

Similar to the approach reported in [76], we consider Markovian processes whereby the robots only share their current LiDARs data—i.e. locally sensed information, not their entire local map. Formally, the local (individual) map m of robot i at time t is obtained by computing the posterior probability $p(m|S_t^i)$ for a collection of sensory data S_t^i —its own sensed data and the sensed data from its connected neighbors identified through the interaction network adjacency a_{ij} matrix—such that

$$S_t^i = \bigcup_{t'=1}^t \{s_j(t'); j|a_{ij}(t') = 1\}, \quad (3.1)$$

with $s_j(t')$ the sensed data of agent j at instant t' , $a_{ii}(t) = 1, \forall t$, and $a_{ij}(t') = 1$ if unit i is connected with agent j at instant t' , and $a_{ij} = 0$ otherwise. It is worth stressing that this approach was designed to be compatible with temporal interaction networks, which have recently been found to be necessary to achieve effective collective responses by MRS [75].

3.3.2 Communication Network

The collective mapping operation requires the robots to exchange some key information with nearby neighbors such as the robot's pose, and LiDAR sensings. As mentioned earlier, the onboard XBee modules form a wireless ad-hoc network, whose dynamic topology can be tuned to achieve optimal collective performance [75]. Given the physical indoor environment that ORION operates in—i.e. no line-of-sight, presence of walls, low battery, latency and low-bandwidth constraints—it is expected that not all units are always connected.

In a system with units A , B and C , if information from A could not be sent to C , this does not suggest that C will not have access to that information. Now, if B gets access to the information from A and includes it for its decision-making process, then the decision made by B , when observed by C , will have information from A “embedded” in it. This indirect propagation of behavioral information through the network is crucial for the collective mapping operation as it dictates how information will flow within the system. The work in [75] reported the optimal network topologies for collective behaviors subjected to local perturbations, which, in the present framework, can be associated with moving obstacles/features.

3.3.3 Exploration Strategy

The swarm dynamics is based on a collective and distributed Frontier-Based Exploration strategy [101]. To decide where to move next, a robot computes at each instant a field of "preference potential" $V(\vec{r})$ representing a non-normalized probability of choosing a point $\vec{r}$ as the next waypoint. The field V is meant to prioritize the exploration of points that are: (i) near the frontier of the explored area, (ii) near the robot, and (iii) far away from the other units. To account for this three factors, we have defined the field as

$$V_i(\vec{r}) = V_F(\vec{r}) \times \frac{1}{\min(\|\vec{r} - \vec{r}_i\|, R_0)} \times \prod_{j \sim i} \|\vec{r} - \vec{r}_j\|^2, \quad (3.2)$$

where $R_0 > 0$ is an arbitrary cut-off distance, V_F is a term characterizing the frontier between the explored free space and the unexplored space, and $j \sim i$ represents the set of neighbors j of unit i . This frontier is obtained by applying a classical Roberts cross operator to the Bayesian map. The waypoint is then selected as the point that maximizes $V(\vec{r})$. This collective mapping approach is formalized in Algorithm 1.

Algorithm 1: Collective Mapping algorithm

```

CollectiveMapping( $r, R, s, map$ )  $\leftarrow$  UpdateOccupancyGrid( $map, s$ )
 $neighbors \leftarrow$  ConnectedNeighbors( $r, R$ ) for  $n$  in  $neighbors$  do _
 $map \leftarrow$  UpdateOccupancyGrid( $map, s_n$ )
 $waypoint \leftarrow$  FrontierExploration( $map, r, R$ ) return  $map, waypoint$ 

```

3.4 Experimental Results & Discussion

Here, we present the results of multiple experiments characterizing and quantifying some key features of our decentralized MRS ORION.

3.4.1 Scalability experiments

The first series of tests assess how the performance of the collective mapping operations scales with the MRS size N . An unknown, irregular floor space (see video [102]) of approximately 60 m² with 5 irregular obstacles in it has been used for all scalability tests. The corresponding surface area is more than 4000 times the footprint of a single unit (12 × 12 cm). Clearly, increasing the number of robots beyond 6 yields only marginal improvement to the system's performance. Given the size of the surface area used, operating beyond $N = 8$ would inevitably lead to drops in performance owing to the overcrowding of the arena. However, this does not mean that our system is not capable of operating with $N > 8$ units, quite the contrary (see Sec. 3.4.4).

The results, presented in Fig. 3.4, reveal a rapid initial increase in the number of explored cells C_e that subsequently reaches a plateau for large N . The initial fast increase in C_e is to be expected the whole space is unknown, hence all cells encountered by the

units are unexplored. At a later stage, the coordination among robots imposes them to travel farther in search of new unexplored cells. We post-processed these data to estimate the characteristic time-scale of collective mapping in the early stages (Method 1) and at a later stage when $C_e = 8,000$. Specifically, Method 1 assumes that C_e progresses according to $C_e(t) = C_e^\infty(1 - \exp(-t/\tau))$ and estimates the time-scale of operation τ by means of linear regressions of $\log(C_e/C_e^\infty)$ for each value of N . Method 2 gives the time it takes to explore the first 8,000 cells. Interestingly, both time-scales exhibit an almost identical linear scaling (in a log-log plot) with the system size N as shown in the insert of Fig. 3.4.

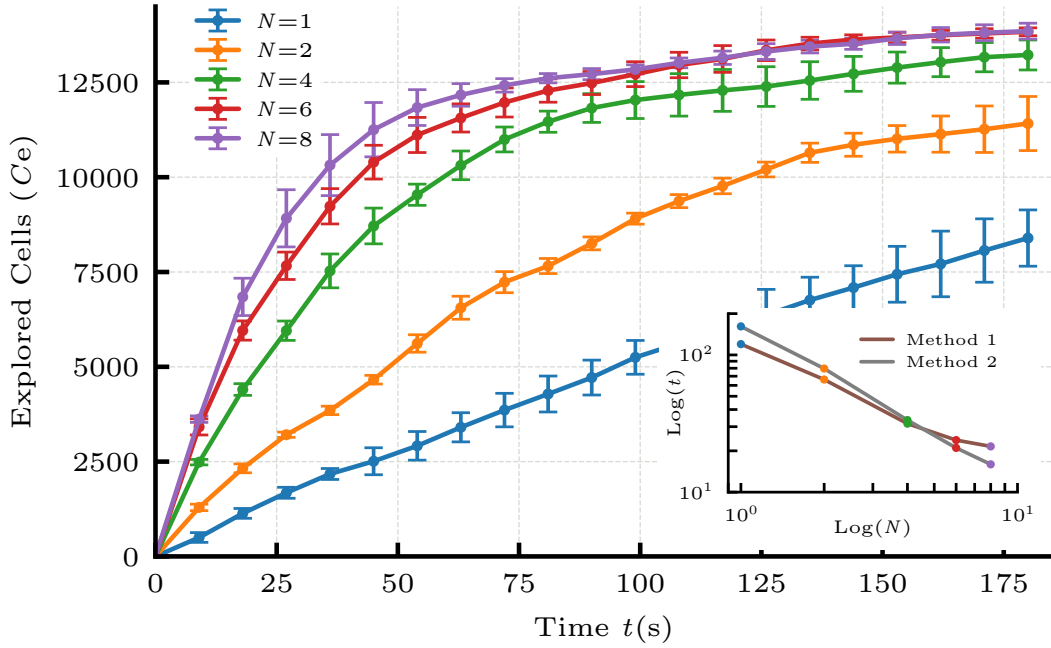


FIGURE 3.4: Scalability experiments for five different MRS sizes: $N = 1, 2, 4, 6$ and 8 . Each point—mean value and the associated standard deviation—is obtained from the results of five repeated experiments with same N value and identical initial conditions. The insert shows the scaling of two characteristic times with N for these collective mapping operations by means of Methods 1 and 2.

3.4.2 Robustness and recovery experiments

As a next step, the fault-tolerance of ORION is assessed by forcing the removal of 2 units out of a total of $N = 4$ in the early stages of the collective mapping process of the same floor area as in the scalability experiments. Specifically, at $t = 20$ s, two units are manually switched off thereby leaving the two remaining units operating by themselves. At $t = 120$ s, two “new” units are injected into the pen and instantly start swarming with the other two units. Beyond the characterization of robustness,

this series of experiments allow us to analyze the dynamics of the emergent recovery process.

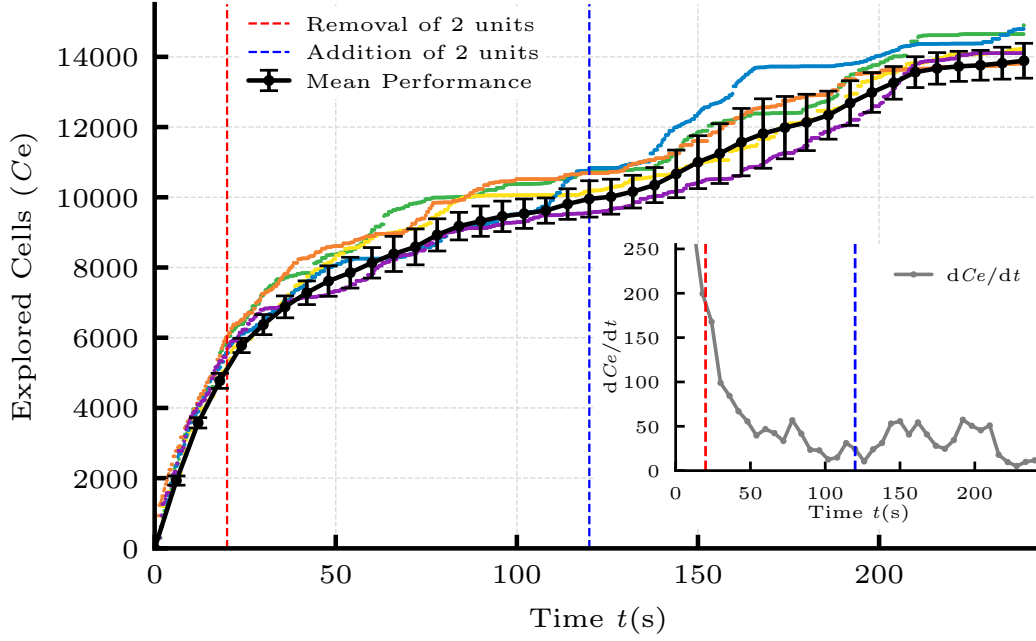


FIGURE 3.5: Robustness experiments: each of the five experiments are shown with different colored dots. The mean performance and its associated standard deviation is shown in black. The red (resp. blue) dashed lines correspond to the instant at which two units are removed (resp. added) of the MRS. The insert shows the rate of change of the number of explored cells C_e .

Figure 3.5 presents the results of five successive experiments (colored dots) as well as the mean performance (with standard deviation in black color). When the removal of the two units takes place, there is a noticeable drop in performance as the number of explored cells slows down. Hundred seconds later, when two new units are injected, the rate at which C_e increases also increases but only after a 20-30 seconds delay (the rate of change of C_e with time is shown in the insert of Fig. 3.5). This delay can be justified by the time needed by the newly formed system to achieve an effective coordinated behavior. The two present units may see their direction of travel affected by the LiDAR information from the two added units. However, passed this delay, the 4-unit MRS exhibits a clear improvement in its performance as attested by the increase in dC_e/dt shown in the insert of Fig. 3.5.

In real-world scenarios, the removal of units occurs with hardware breakdowns, low battery levels, and most commonly with out-of-range communications. Our previous work [82] provides a statistical study on communication success based on distance apart for varying sizes of the MRS.

3.4.3 Flexibility experiments

Here, we consider the collective response of the system to a drastic increase in the area to explore. Specifically, a rectangular (3×9 m) 27 m^2 space is divided into three square areas of almost identical surface area of 9 m^2 (see video [102]). Initially, two robots are placed in the rightmost area, and another two robots are started from the leftmost area. The central area is empty and inaccessible owing to the presence of two walls separating it from the leftmost and rightmost areas. At $t = 0$, each pair of robots starts mapping its exclusive area. But after 30 s, the two walls are removed, thereby opening the two areas mapped by the pairs of O-map to a central void.

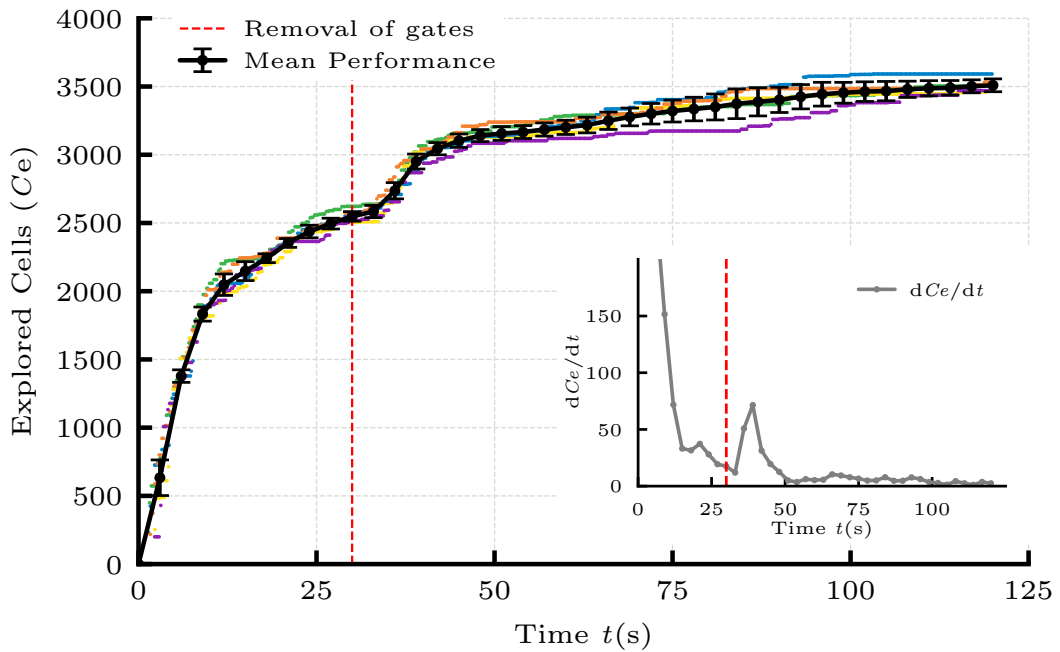


FIGURE 3.6: Flexibility experiments: each of the five experiments are shown with different colored dots. The mean performance and its associated standard deviation is shown in black. The red dashed lines correspond to the instant at which the floor area is expanded by 50%. The insert shows the rate of change of the number of explored cells C_e .

The results for five distinct experiments are reported in Fig. 3.6, showing a high consistency in performance. Before the removal of the two walls, the performance is analogous to the one obtained for our scalability experiments. However, once the walls are removed and the area to explore expanded, a change in performance can clearly be detected after a few seconds delay. Again, this observed delay can be associated with the time it takes to the MRS to adjust its coordinated behavior to changing circumstances. The sharp increase in dC_e/dt shown in the insert of Fig. 3.6 after the removal of the walls is a good indicator. It is therefore very similar to the collective behavior observed during the recovery experiment discussed previously.

3.4.4 Multi-floor exploration

As a last step, we present some qualitative results of a collective exploration over two floors. These results were obtained on our university campus grounds during operating hours. The areas mapped constitute unstructured and dynamic environments, and no external/additional supporting infrastructure was necessary to run the experiments. Eight O-map units were initially positioned on one floor (F_1), and four others on the floor above (F_2). Two O-climb units were placed on a vertical wall and ascended during the experiments, thereby expanding the distributed communication network between floors. The layouts of the surface areas to be mapped on F_1 and F_2 are shown in light gray in Fig. 3.7: they consist of open spaces, elevator lobbies and office rooms with open doors. The overall collective exploration task lasted about 3 to 4 minutes depending on the dynamic obstacles encountered. Snippets of the live operations are shown in the final segment of the video [102].

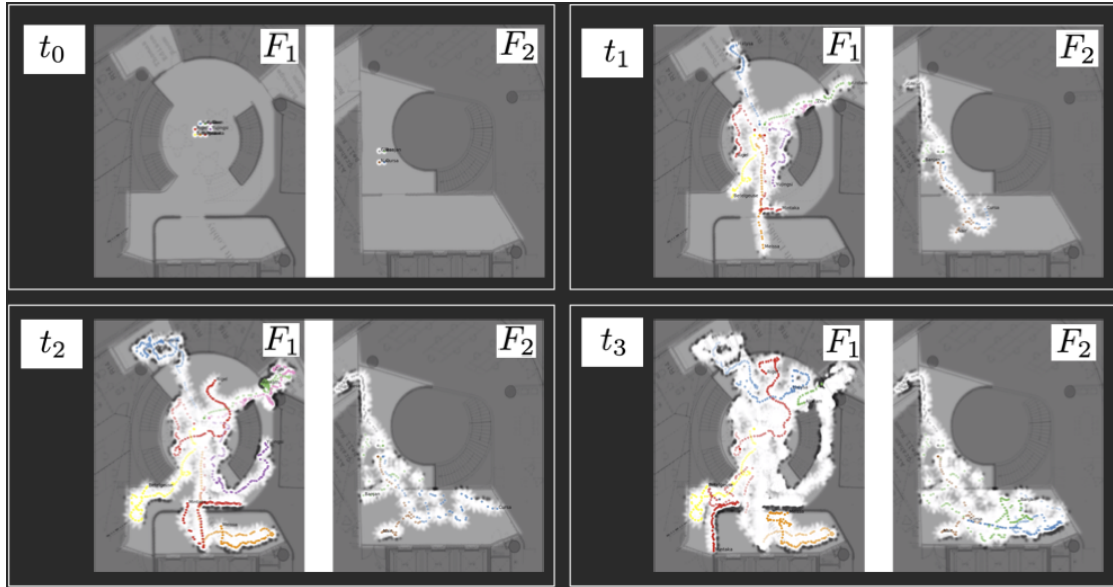


FIGURE 3.7: Snapshots at successive time instants (t_0, t_1, t_2, t_3) of the mapped areas by 12 units across two floors (F_1 and F_2).

3.5 Conclusions

In this paper, we presented the analysis, design, and development of a decentralized multi-robot system, ORION, capable of performing scalable and fault-tolerant explorations of unknown multi-floor indoor environments. ORION operates based on a swarm robotic subsystem that is collectively mapping possibly dynamic environments made up of multiple floors. The wall-climbing units achieve wall-adhesion by means of dry-adhesive whegs and share the same basic robotic architecture as the mapping variant. They serve to expand the distributed communication network between the swarming ground robots that are mapping two floors.

ORION exhibits consistent scalability, robustness and flexibility when collectively mapping unknown and unstructured floor spaces with several irregular obstacles present. It is also capable of recovery by subsequently injecting two other robots while it is collectively operating. Our results also reveal a delayed collective response in two particular cases: (1) when the MRS is rapidly expanded by the addition of several units, and (2) when the MRS responds to abrupt changes in the topology of the environment being mapped.

Chapter 4

Balancing Collective Exploration and Exploitation in Multi-Agent and Multi-Robot Systems: A Review

This chapter is taken whole from [103] by Kwa, Hian Lee and Leong Kit, Jabez and Bouffanais, Roland.

In this chapter, the concept of exploration and exploitation closely related to adaptability and flexibility of SI is studied in the area of multi-agent and multi-robot system. In this review, techniques from other works that attempts to overcome the exploration and exploitation dilemma are discussed. We have also found the lack in metrics available to measure flexibility—i.e. SI.

4.1 Introduction

In recent years, there has been an increasing interest in using multi-agent systems (MAS) to carry out a wide array of complex tasks. In using such collaborative platforms, a large and challenging task can be broken down into smaller tasks that can be tackled by agents with the ability to change their actions based on their local environment [104]. This provides MAS with very high levels of flexibility—i.e., adaptability to changing circumstances—thereby making them an attractive solution to problems in fields ranging from traffic management [105], to machine learning [106], and robotics [107, 108].

A well-known challenge faced by all MAS is the exploration–exploitation dilemma. This dilemma arises owing to the fact that gathering new information (i.e., exploration) and making use of currently available information (i.e., exploitation) tend to be two mutually exclusive activities [109]. Should a system be setup in favor of exploration, it would be able to gather large amounts of knowledge without being fully able to benefit from this knowledge. This may increase the time required for the MAS to accomplish its task or prevent it from achieving its goal altogether [110]. On the other hand, should a system be setup to favor exploitation, the system may be unable to adapt to changes in the environment or may find itself trapped in a local optimum [111, 109, 112]. Despite this apparent trade-off between the two activities, both sets of actions can be performed

simultaneously [113]. As such, one can expect that to maximize the performance of such a system, there ought to be some sort of balance—most likely a dynamic one—between the amount of exploration and exploitation carried out.

Within the field of robotics, MAS have been studied for their applications within Multi-Robot Systems (MRS) [114, 115, 116]. The use of MRS, which can either be controlled in a centralized or decentralized fashion [117], has been demonstrated in a wide variety of tasks, including area mapping [118, 27], area characterization [119, 120], collective construction [121], collective decision-making [122], collective transport [123], perimeter defense or geofencing [124, 125], as well as target search and tracking [126, 127, 128]. The attractiveness of MRS in such tasks stem from three key features: (1) flexibility—the ability for the to adapt quickly to rapidly changing environments, (2) robustness—the ability to cope with component failures within the system, and (3) scalability—the ability to carry out tasks in systems comprised of different number of agents [129].

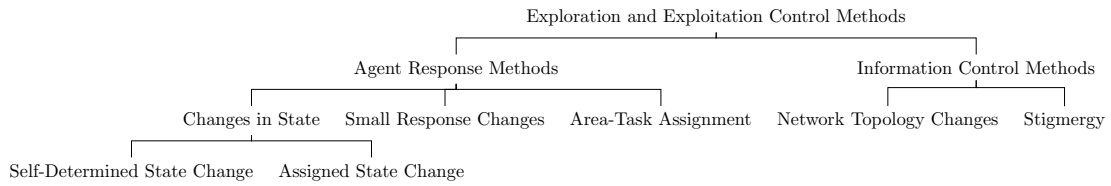
For implementation of swarm strategies in MRS, agent behaviors are largely developed in simulation before being tested in physical systems. However, doing so inevitably leads to the so-called *reality gap*, in which there is a difference between the simulation models and the actual robots used [130, 131]. This could manifest as differences between the actual robot behaviors and those predicted by simulations due to various factors such as variations in robot kinematics, communications noise, and other environmental factors affecting the robot's movements. Given the potentially large number of agents involved, this reality gap may be further widened when modeling MRS. While this indeed poses a unique problem for MRS practitioners, in this paper, we only focus on the behavioral aspects of how robotic agents balance their efforts in exploration and exploitation.

At this point, it should be noted that there is a marked difference in the strategies that are employed for use in static and dynamic tasks. While there is still the need to balance exploration and exploitation when operating in static environments, this balance tilts heavily in favor of exploitation, especially during the latter stages of the task. For example, in the search for a static target, [132] performed an initial random search before transitioning to a Particle Swarm Optimization (PSO) algorithm to facilitate the convergence of agents around the target's position. Similarly, [133] found that biasing the swarm towards exploitation during the latter phases of a static target search task greatly reduces the time taken to accomplish the given mission. Tasks in slow-evolving environments can also be treated as a quasi-static problem and be completed in the same manner. [134] were able to demonstrate a simulated swarm using a comparable algorithm to search for and track a slow-moving target. Due to the focus on exploitation during the latter stages of the tasks, research on MAS carrying out static or quasi-static tasks tend to focus on convergence time and the accuracy of the overall system [135, 119, 120].

Given this critical distinction between static, quasi-static, and fast-evolving tasks, we propose the following definition for the latter: a task occurring in an environment that evolves at a rate at which a single agent is unable to keep up. Examples include tracking a target that can move faster than the agents [136, 128], and dynamic features that

lead to a notable evolution in the optimum agent allocation of a task assignment problem [137]. For a system to effectively carry out its assigned task in a fast-moving dynamic environment, there must be some form of adjustment of the balance between exploratory and exploitative actions throughout the duration of the task. Should an MAS bias its actions towards exploitation over time, such exploitative activity based on outdated information will inevitably result in poorer system performances [138, 139, 140]. Despite this, there is still merit in studying the methods used to control the level of exploration and exploitation carried out in MAS operating in static and quasi-static environments. This is because such systems still are susceptible to being trapped in local optima during the initial stages of the task [141, 135, 142] and variations of strategies employed in static environments have been modified and applied for use in fast-evolving environments [143].

Given the notable importance of the exploration vs. exploitation balance in MAS, we present here a survey of the methods and techniques used to control this balance. First, we explore the concept of exploration and exploitation and the common metrics that practitioners use to quantify the level of exploration and exploitation carried out by their systems. Next, we go through the methods used to control the exploration and exploitation balance. An overview of the types of methods and strategies explored within this review can be found in Figure 4.1. In the conclusion, we present possible directions for future research as well as summarizing remarks.



4.2 Exploration and Exploitation

4.2.1 Characterizing Exploration and Exploitation

Exploration and exploitation is a trans-disciplinary concept used in a variety of fields and contexts. As such, there is no unique definition that can be used to define these two behaviors. Nevertheless, these exploratory and exploitative actions can be characterized by certain observable features. To this end, [112] identifies three key distinct features that distinguish between the two: (i) behavioral patterns, (ii) the uncertainties associated with an actor's choices, and (iii) the expected outcomes as a result of an agent's actions. It is important to note at this point that while these three features classify the actions of a single agent, a single agent's actions do not characterize the exploration–exploitation dynamics (EED) of the entire MAS. It is only through observing the actions of all or a proportion of a system's agents that one is able to accurately characterize the exploration–exploitation balance of an MAS.

An agent's behavioral patterns are one of the most common definitions of exploration and exploitation used as the movements it carries out is a physical quantity that can be directly observed and measured. Many MAS practitioners use the agents' movements and observed positions to determine the system's exploration-exploitation balance (see Sec. 4.2.2). For example, in a target search context, when agents are close together or traveling towards the same region of the area of operations, it can be assumed that they are exploiting a common source of information [138, 144]. Conversely, if agents are moving away from each other or in a random fashion, they can be assumed to be carrying out exploration.

The less tangible aspect of an agent's behavioral patterns are its intentions that drive the observed actions. While not as obvious as an agent's motions, these can still be easily determined by querying the state of an individual agent in an engineered MAS. In their review, [113] posited that exploration and exploitation are associated with a certain level of learning. Therefore it can be said that an agent is carrying out exploitation if it uses previously acquired knowledge and carrying out exploration if it is not doing so [145, 146]. Again, this is another common method of defining exploration and exploitation in MAS, where certain practitioners have developed dedicated behaviors that facilitate exploration and exploitation separately (see Sec. 4.3.1).

Besides observing an agent's behavioral patterns, the amount of uncertainty associated with each action can also be used to classify exploration and exploitation. With this characteristic, exploratory actions are associated with those that have high levels of uncertainty, while exploitative ones are associated with low levels of uncertainty. In MAS operating in unknown environments, these uncertainties are usually presented in the form of a map that updates itself as the task progresses [147, 148, 149]. Using such maps allows an agent to move towards regions of different levels of uncertainty based on its intentions to carry out either exploration or exploitation. As such, an agent that moves towards or gathers information in an area with high uncertainty is deemed to be carrying out exploration, while an agent doing the same in an region of low uncertainty can be said to be carrying out exploitation.

Finally, the type of action an agent performs can be classified based on the expected results of the said action. This is typically exemplified by observe-or-bet tasks. In such a task, an agent is given a chance to observe the environment and gain information while not receiving any reward (exploration). Alternatively, the agent may "bet" on an option and receive its associated reward (exploitation), although the amount of reward received remains hidden from the agent [150, 151]. While these tasks offer a very clear distinction between exploration and exploitation, such contrasts are not as explicit when observing real systems; agents tend to know the quantity of a reward and the action that was performed that led to the reward. These additional pieces of knowledge allow an agent to obtain more information about its surrounding environment and allow agents to simultaneously perform exploration and exploitation [145].

As highlighted by [112], these three characteristics need not be used to define exploration and exploitation in isolation. Multiple characteristics can be used to determine the type of activity being carried out by an agent. For example, when performing an area mapping task, [133] proposed using an MRS that built an occupancy grid map

that increased in resolution as the task progressed. If a robot chose to move into a previously unvisited cell (high uncertainty, learning expected), it was determined to be exploring. If the robot chose to move to a cell with a known object of interest or update its map of the local area with a higher resolution (low uncertainty, reward expected), it was determined to be exploiting.

4.2.2 Measuring Exploration and Exploitation

After selecting their definitions of exploration and exploitation, practitioners next turn their attention to measuring the level and quantifying the balance between the two activities. In the field of computational optimization, a typical strategy in finding a global optimum is through the use of an MAS consisting of many candidate solutions searching the solution space. In this use case, the quantification and monitoring of the system's EED is important as doing so permits it to autonomously detect if the solutions have been trapped in a local optimum or have stagnated. Doing so grants the system the ability to self-modify specified parameters, allowing it to "jump out" and continue with its optimization task [152, 153]. Similarly in MAS operating in dynamic environments, these measurements also allow a system to monitor and prevent solution diversity loss, allowing systems to track moving optima over time [154, 141, 138]. At this point, it should be noted that a system's diversity is analogous to the amount of exploration it carries out. These metrics mostly fall within the categories of spatial distribution based metrics or probability based metrics, although there are also other metrics that have been used that do not fall within these two categories.

While most exploration and exploitation metrics are measured globally, i.e., a system-wide property that allows one to quantify a system's EED as a whole, some of these metrics can also be measured locally by the individual agents, thereby allowing them to adjust their actions appropriately. The use of globally and locally computed metrics both have their associated advantages and disadvantages. While the use of a system-wide EED metric would allow for a more accurate quantification of the overall balance between exploration and exploitation, it also entails that this balance be controlled in a centralized fashion. This would result in the loss of a certain degree of system scalability, flexibility, and robustness. Conversely, if a locally computed metric is used, the agents would be able to use this metric during their operations, allowing them to influence the behavior of the system through various feedback or feedforward loops. While such agent-based metrics may not be able to quantify a system's EED as accurately as a global metric, it allows the system to maintain a level of decentralization and hence, scalability, flexibility, and robustness.

Spatial Distribution Based Metrics

Similar to classifying an agent's actions based on its behavior and movement patterns, the use of metrics based on the agents' spatial distribution is very popular as it is a physical quantity that can be easily observed, measured, and calculated. As previously

mentioned, to quantify the level of exploration and exploitation in computational MAS, the solution diversity of a system is usually calculated. This solution diversity is a measure of the spatial distribution of all candidate solutions within the feasible region. [155] based their diversity metric on the average distance of each solution from the mean. [156] adapted this metric for use in their study of swarm-based metaheuristic algorithms by using the median solution instead of the mean as it reflected the population center of the candidate solutions more accurately. In a similar fashion, [157] calculated a *population standard deviation* to measure the solution spread. In summary, all these metrics are essentially based on low-order indicators of the statistical distribution.

Comparable metrics are in use to measure the amount of exploration and exploitation carried out by agents within MRS. As seen with the metrics used in computational optimization, measuring the spatial distribution of agents is also used to quantify the level of exploration and exploitation. In their target search task, [158, 159] measured the diversity of individual robots within their MRS by calculating the amount of overlap between a robot and its neighbors' search areas. Agent sensor area overlap was also used by [160] who calculated the size of area covered by at least two agents as a measure of the amount of effective exploration performed by their swarm in a surveillance task. This metric is similar to the one used by [161, 162, 163] who used a k -coverage metric that calculated the amount of time a target spent being tracked by at least k agents.

[164] measured the level of exploration and exploitation by tracking three metrics: (1) the minimum distance between agent pairs, (2) the sum of the distances between detected targets and their closest agents, and (3) the distance moved by each agent over the course of one time-step. While often not explicitly stated, the distance between a robot and its neighbors is the most common measure of diversity as most control methods tend to focus on preventing excessive aggregation or spatial distribution [165, 166, 167, 168, 169, 134, 170].

The lack of distance measurements does not preclude the use of spatial distribution metrics. As a proxy for the amount of exploitation performed by each robot, [132] calculated a communications-based aggregation degree from the number of neighbors within visual range, with higher numbers of agents within visual range suggesting a higher amount of aggregation. This eliminates the need to share positional information between agents, allowing an agent to reduce the number of radio transmissions it needs to make. In an adaptation to the BEECLUST algorithm, which was first proposed by [171, 172], [173] determined the level of local aggregation, and therefore the level of exploitation of individual agents, by measuring the time interval between two consecutive neighboring robot encounters.

The aggregation degree may not necessarily need to be calculated with the use of spatial information. In performing a search for multiple targets using several sub-swarms comprised of leaders and followers, [174] calculate the aggregation of the sub-swarms by comparing the fitness value of each sub-swarm leader to the mean fitness value of all the leaders within the swarm. This is similar to the aggregation degree proposed by [175], who calculated the ratio between the agent with the lowest fitness value and the

agent with the highest fitness value. A list of works using spatial metrics to quantify an MAS' EED can be found in Table 4.1.

Probability Based Metrics

While less popular than the use of spatial distance metrics, the probability of actions carried out by individual agents can also be used to quantify the level of exploration and exploitation involved. The most common use of probability based metrics can be found in the application of Lévy walks and flights. Lévy walks are a class of random walks where the distance of the walk, l , is taken from a heavy-tailed probability distribution function, $P(l)$. This probability distribution function is given by the equation:

$$P(l) \sim l^{-\mu}, \quad (4.1)$$

where μ is the Lévy parameter controlling the shape of the probability distribution function. Lower values of μ close to unity lead to a higher probability of longer walk lengths between direction changes, which is associated with higher levels of exploration. Conversely, higher values of μ cause an agent to preferentially carry out shorter walks between changes of direction, characteristic of higher levels of exploitation [176, 177]. The use of the Lévy parameter has been used as a method of quantifying and controlling the level of exploration and exploitation of various MRS applied in different tasks, ranging from foraging [178, 179, 146, 180], to target search [143, 181, 182], to area mapping [183, 184, 185].

In addition to the use of Lévy distributions, action probabilities can be used to set the level of exploration and exploitation carried out by an MRS. To search for a target, [133] used action probabilities to bias an agent in favor of carrying out either exploratory or exploitative actions. In a similar fashion, [186, 146, 179] uses the probability of an agent using information in its memory as means of determining and controlling the level of exploration and exploitation of the system. In their best-of- n problem, [187, 188] used an opinion switching probability threshold as a proxy to measure and control the exploration and exploitation dynamics of their swarming system. With this, higher switching probabilities are associated with exploration, while lower probabilities are associated with exploitation. These probability thresholds are also popular in the related field of task allocation to obtain the optimal distribution of agents among all the tasks presented to the MRS. In such scenarios, the use of probability thresholds allows an agent to decide whether it should stay and continue with its current task (exploitation) or move on and attempt to perform another task within the environment (exploration) [189, 190, 137, 191]. A list of works using such probability based metrics can be found in Table 4.2 for different tasks.

Other Metrics

In addition to spatial and probability based metrics, other metrics have also been used to quantify the level of exploration and exploitation carried out by MRS. These metrics

are detailed and listed in Table 4.3. To solve their Best-of- n problem, [187, 188, 192] employed contrarian agents that do not change their opinion of the best site. This use of stubborn agents are a method of measuring and guaranteeing a minimum level of exploration carried out by the system.

In their collective decision-making task, [193] attempted to measure and maximize the diversity of their MRS by counting the total number of unique messages being broadcast, as well as the proportion of agents broadcasting these messages within the system. Using this metric, a small number of unique messages or a small proportion of agents broadcasting a message indicated low levels of diversity.

To measure the exploration and exploitation dynamics of an MRS tracking a moving target, [128, 194] proposed a metric based on an agent's direction of travel and its bearing to the target. This allowed for an individual agent's contribution, and hence the system's overall exploration and exploitation dynamic to be quantified. [195, 144] later identified that a metric based on an agent's position relative to a target was unable to function when used in a scenario where multiple targets are to be tracked, such as situations defined under the Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) framework. This led to the introduction of an *engagement ratio*, thereby allowing quantification of a swarm's EED in the presence of multiple targets by finding the proportion of agents actively moving towards a target.

In their CMOMMT task, [196] introduced a metric known as the *tracking fairness* that measures the monitoring effort of the MRS across all targets. Should a target receive more attention from the MRS's agents, the system would have a high tracking fairness score, indicating over-monitoring of a target or too much exploitation being carried out by the system. Conversely, a low fairness score indicates that the agents are monitoring all the targets evenly and that there is a good balance between exploration and exploitation.

To develop a tracking strategy for the same CMOMMT problem using deep reinforcement learning [197] used an occupancy grid map (OGM) to characterize the environment. A tracking fairness based metric was then used, comparable to the one proposed by [196], as well as an exploration metric based on the latest observation time of all grid cells in the OGM.

4.3 Agent Response Methods

Once a system's level of exploration and exploitation is able to be quantified, the balance between the two activities can then be controlled. The most common method of adjusting this balance performed by an MAS is by changing an agent's actions in response to locally measured variations in the environment or information originating from the agent's neighbors. This method of controlling exploration and exploitation can elicit two types of response from an agent: (1) trigger a discrete change in state of an agent, causing it to perform a predefined set of actions associated to either exploration or exploitation, or (2) cause a small change in an agent's actions, resulting in a small

change in bias of the agent's actions towards either end of the exploration or exploitation spectrum.

4.3.1 Changes in State

When deploying changes in agent states, a large change in an agent's behavior is effected in response to a given stimulus observed by either an agent, a subset of agents or the entire MAS itself. This gives agents the ability to switch from an exploratory to an exploitative behavior, and vice-versa allows for them to better adapt to their local environment, with the aim to improve the overall performance of the system. These strategies tend to base their exploration strategies on random walks [198], followed by information exploitation driven by either a metaheuristic algorithm or direct travel towards a point of interest should one be detected. A flowchart of such changes in agent response can be seen in Figure 4.1.

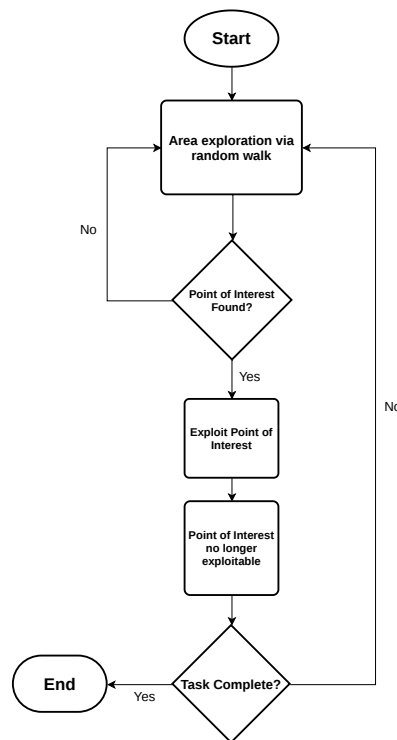


FIGURE 4.1: Flowchart detailing how an agent deterministically decides on its response when provoked by environmental stimuli.

These changes in response can either be triggered by an agent itself as a reaction to its local environment or a coordinating agent can trigger this response change in a certain subset of the MAS. The latter set of strategies usually involves a centralized controller

or locally designated leader to determine which agents are to change their response patterns. As can be expected, each set of strategies comes with its own advantages and disadvantages. For instance, while using decentralized strategies, an MAS is able to retain its flexibility, robustness, and scalability. However, keeping these properties of the system comes at the expense of more precise control over the overall system's EED. Using a set of strategies involving a centralized controller, an MAS user is able to exert greater and more accurate control over the system's EED. However, a trade-off must be found, which inevitably impacts one of the system's three key characteristics of flexibility, scalability, and robustness.

Self-Determined State Changes

In decentralized systems, changes in an agent's state stem from its own reaction to changes in its local environment. Based on this information, an agent decides for itself if it will carry out exploratory or exploitative actions. When determining their own state in response to their environment, an agent normally takes into account local observations carried out by itself as well as information originating from its neighbors. This allows an agent to react quickly to any changes in its local environment, allowing the MAS to retain its system flexibility. Agents are also not reliant on a leader or central controller assigning it an appropriate response, allowing the system to maintain its scalability and robustness. However, given this decentralized control strategy, the MAS will have less control over the overall amount of exploration and exploitation carried out. A list of such strategies and associated tasks can be found in Table 4.4.

Dedicated Exploration and Exploitation Strategies In various systems, agents may switch between pre-programmed exploratory and exploitative behaviors. To capture multiple static targets, the approach proposed by [203] initialized their agents with a random search behavior to explore the environment. In most works involving simulations, a target signal is modeled by using an objective value function, representing the type of information to be exploited by the MAS (e.g., radio signal intensity, chemical plume concentration, etc.). When a target signal was encountered, agents would then employ a PSO algorithm, allowing the system to exploit the target signal and capture the target. After capturing the target, the agents would revert to their original state and resume their random environment search.

Similar changes in state and strategy have been utilized in searching for and tracking dynamic targets. In the tracking of multiple dynamic objects of interest under the CMOMMT framework, [161, 162] programmed their agents to initially explore the area randomly. However when an object of interest was detected, it would transition into a tracking mode, change its response, and ultimately alter its velocity to follow that point of interest. The authors were able to demonstrate that this mixed strategy of random motion and following a target yielded better tracking performances than an exploration-only strategy in which the agents only move around randomly.

To track a fast moving target, [128, 194, 195, 144] initially used inter-agent repulsion to promote area exploration. When a target was detected, an agent and its neighbors would activate a PSO-based tracking behavior, allowing the target's information to be exploited, facilitating target tracking. This behavior would be disabled once the target is no longer detected. The ability to dynamically select between the use of exploratory and exploitative strategies appropriately was demonstrated to be key in facilitating the tracking of the target as it prevented over-exploitation of the target and over-exploration of the search space.

Changes in state may also be triggered by signals originating from neighboring agents. In their work, [205, 206] explored the use of so-called "help calls" broadcast from an agent when a target is predicted to be leaving its observation range. Other robots that receive this help call that are not involved in tracking a target move towards the robot requesting help to assist with the tracking task. The use of help calls is also explored by [161, 162] who studied their applications in conjunction with various response and communication models. [163] expanded on this work, incorporating a *willingness to interact* parameter that determined the propensity of an agent to aid in the tracking of a target.

Agents are also able to form dynamic groups dedicated to exploiting information stemming from a point of interest. The formation of groups is more commonly done together by means of a central controller or through a locally designated leader. However, these dynamic groups can still be formed using decentralized decision-making. [204] proposed a strategy to search for multiple targets in which agents are initialized in the random-walk search mode. When a target is found, a dynamic group comprised of a maximum of 10 agents is formed. Agents in this group exploit the target's emitted signal through the use of a PSO algorithm. At each time-step, agents broadcast their measured target signal strengths and individually decide if they should join or leave the group.

The use of changes in agent response is not limited to applications within the domain of target search and tracking. In their foraging task, [201] split their foraging task into two separate phases. Initially, agents follow a search phase guided by a random walk. After a specified amount of resources has been gathered, agents are assigned to travel to specific regions to exploit high rewarding resource clusters. During this exploitation phase, the agents travel to their allocated locations via a directed random walk. This path becomes straighter, and hence more directed over time, allowing the agents to reduce the amount of exploration carried out.

In a similar foraging task, [179, 146] stored the location of resource patches in a limited agent-based memory fashion. The authors used a truncation probability that determined if an agent's exploration via Lévy walks are to be cut short. This allowed agents to randomly transition between informed movements and random searches. By increasing the truncation probability, agents of a system could be made to use information from its memory more often, leading to a system-level EED that favors exploitation. [195, 144] also made use of memory in their target tracking task. However, instead of changing the probability of an agent accessing information from its memory, the system's EED was tuned by changing the length of an agent's memory. A system can be

made to favor exploitation with the use of longer memory lengths and exploration with the use of shorter memory lengths.

Artificial Immune System Strategies The ability to change between distinct behaviors is also one of the cornerstones of *Artificial Immune System* (AIS) strategies. First proposed by [209], AIS was later defined by [210] as an “adaptive system, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving.” Such strategies are commonly used in computational tasks such as computer security, pattern recognition, and optimization [211, 212], however they have also seen recent applications in various MRS [213]. In this family of strategies, all agents are given a repository of behaviors or rules. These behaviors are associated with state values that are influenced by various stimuli in the environment, such as the presence of tasks or targets, as well as by the state values of encountered agents. The behavior with the highest state variable at a given time-step is the one chosen to be carried out.

Initial adaptations of AIS for use in MRS were mostly limited to the domain of task search and allocation. In the initial work done by [207, 208], the repository of behaviors consisted of four strategies: (1) aggregation, (2) random search, (3) dispersion, and (4) homing to carry out a target tracking task. When a robot encounters another robot, each robot re-evaluates its behavioral state values. Subsequently, the behavior with the largest state value is the one chosen to be activated. Using a similar strategy, [214] also demonstrated the AIS’s potential in situations where is a large influx in quantity of tasks to be performed by the system. The same strategy was applied by [199, 200] in a dynamic shepherding scenario, as well as by [164] in a dynamic target tracking task.

Lévy Parameter Changes Instead of using vastly different movement policies to effect changes in the response of agents, agents using Lévy walks can vary their Lévy parameter. As explained in Section 4.2.2, changing the Lévy parameter μ can be used to influence the overall EED of an MAS; higher Lévy parameter values are used to shorten the walk lengths of agents, prioritizing exploitation, while conversely lower μ values increase walk lengths and hence favor exploration. As such, several MAS make use of changing their Lévy parameter to influence its EED instead of using dedicated exploration and exploitation algorithms.

For a system carrying out a destructive foraging task, [202, 178] used a time-based system paired with a “satisfaction” parameter to determine the transition from the exploitation mode to the exploration mode. When exploring, the MAS used a small Lévy parameter to generate long-step sizes. This was changed to favor smaller steps when a target was found to encourage exploitation. While in the exploitation mode, an agent’s satisfaction parameter value was increased for each time-step a target was encountered and decreased when no target was met. An agent would only transition from the exploitation to exploration when both the satisfaction value reached zero and the minimum time limit for the exploitation mode was reached.

To determine the location of forest fires, [181] proposed using a temperature threshold to determine an agent's state. In their MAS, all agents are initialized in a searching state, using a low μ value to direct area exploration. Should an agent detect a temperature higher than a specified threshold value, it would transition to a more exploitation-focused strategy by employing either a directional-driven Brownian search or regular Brownian search strategy, drawing from a Lévy probability distribution function with a higher μ value.

Assigned State Changes

While self-determined changes in state are mostly found in fully decentralized systems, assigned state changes can be found in fully centralized systems (i.e., systems where the movement of all agents is dictated by a central controlled) and hierarchical systems (i.e., systems with designated 'leader' agents determining the actions of a small proportion of agents). Using assigned state changes, when a stimulus is found, only a specific subset of the system comprised of a fixed number or predefined subset of agents is assigned to exploit the new stimulus information. This is in contrast to the responses seen in the previous section in which agents are able to decide for themselves if they should switch behaviors. Using such behaviors allows for greater and more precise control over an MAS' EED. However due to the more centralized nature of such strategies, an MAS must sacrifice a certain degree of flexibility, scalability, and robustness. Using this class of strategies, the assignment of roles and responses to agents can either be done dynamically as the task progresses, or during the initialization process. A list of works assigning agent response patterns can be found in Table 4.5.

Dynamic Agent Assignment While using assigned state changes does indeed reduce a system's flexibility, a certain degree of flexibility can still be retained by having agents assigned to tasks while a mission is being carried out. [236] demonstrated a dynamic agent assignment strategy in a 2D environment. In their target search scenario, agents were initialized with a random walk to explore the environment. When a target is found, a leader agent is designated to coordinate a dynamic alliance comprised of 4 agents. While these agents carry out the target pursuit task, the rest of the agent continue with exploring the environment. [237] later built on this work and demonstrated this type of strategy for finding targets in an underwater 3D environment. The agents employed initially used a Lévy flight-based cruise phase to search for one of multiple targets. When a target is found, a sub-swarm is formed using a centralized self-organizing map (SOM), which are also used in transfer learning neural networks [240]. Only members of this sub-swarm carry out exploitation of the target's signal through the use of the Dolphin Swarm Algorithm (DSA) while the rest of the agents continue exploring the search area. In both strategies, after successfully capturing the target, the pursuing sub-swarm's agents disband and continue with the search process by reverting to the cruise phase. Important to this strategy is the dynamic nature of the sub-swarm's membership, meaning that agents in the sub-swarm may be substituted should it stray too far from the target.

In their work done on tracking and monitoring of multiple dynamic targets with UAVs in a 3D environment, [239] employed a strategy where all agents carried out a patrol phase based on the Lévy flight. When a target is detected, the agent closest to the target is designated as leader. The leader selects the next two agents that are nearest to the target's location to form a sub-swarm comprising of 3 agents. Only members of this sub-swarm would transition to a PSO-based tracking mode to exploit this information and pursue the target until the target leaves the surveillance area. The leader is also able to substitute agents within the sub-swarm with other agents should the original agents move too far away from the target.

Similar techniques that assign agents to targets have also been demonstrated by [238] in aquatic surface operations and by [234, 235] in underwater operations. Instead of using a random walk, the authors generate a potential field fitness function based on unexplored areas of the environment and the location of obstacles within the operating area. Exploration is carried out through the use of a PSO algorithm that makes use of this potential field by driving agents towards unexplored regions of the environment. When a target is found, the four agents nearest to the target enter a tracking mode and are assigned to pursue that target until it is caught by using the PSO algorithm on the target's emitted signal. Upon a successful target capture, these agents revert to the search mode to look for more targets. In another study done on search and rescue strategies by [230], searcher strategies are assigned to an agent based on the agent's distance to a potential target and the ratio of searchers to targets.

The size of the sub-swarms need not be fixed. In [226, 227, 228, 229], work is done on a Robotic Darwinian PSO (RDPSO), adapted for use with ground robots from the original Darwinian PSO by [241]. [215] later expanded on this work with applications using underwater robots carrying out exploration in a 3D environment. This strategy balances the swarm's EED through social exclusion or inclusion of agents and is done by splitting the MRS into multiple sub-swarms and independent "wanderers" that search for the solution individually. All of these sub-swarms execute the same target tracking strategy independently and there is no direct information transfer between different sub-swarms. However, there are some system-level rules applied during each iteration that serve to keep the membership of the sub-swarms evolving, thereby facilitating indirect information transfer.

The system level rules that govern this social exclusion or inclusion of the RDPSO algorithm is the "punish-reward" rule, as well as the sub-swarm creation and deletion mechanisms. If a sub-swarm improves its best fitness value, it will add the best performing wanderer from the excluded agents into its sub-swarm membership. However if a sub-swarm has not improved on its fitness value, it will exclude its worst performing agent, which becomes an independent "wanderer" itself. If after a specified period of time, a sub-swarm has been rewarded more times than it has been punished, it is allowed to generate a new sub-swarm. This new sub-swarm will be composed of best performing agents among the wanderers. However, if the sub-swarm has been punished more times than it has been rewarded, its population will be reduced, and upon reaching the minimum number of agents, disbanded and all member agents will be socially excluded. These two mechanisms are key to the dynamic nature of the system's sub-swarms and provide a method of indirect information transfer between the groups.

The exploration carried out by the independent wanderers and the indirect information transfer facilitated by the dynamic properties of the sub-swarms' membership guarantees a minimum amount of exploration carried out by the overall system.

In numerical optimization, where it is easier to control the movement of individual agents, it may be decided that sub-swarms may not be needed. As such, a common strategy observed in this field is to trigger the re-initialization of particles within the search space [220, 221, 157]. Using these strategies, a central controller will command the agents to reset their positions to a random location within the search space after a predefined time period or when a large change is detected in the environment. The strategy was adapted for use in robotics by [231] for leak source detection. In their proposed strategy, the average fitness value of each agent was calculated and a subset of agents that had a fitness value less than the system mean were instructed to restart their search process from a different location.

Preassigned Roles Besides dynamic response assignment, agents can be assigned a response pattern during their initialization. Doing so allows for greater control over the system's behavior and exploration and exploitation balance in exchange for less system flexibility. In numerical optimization, [222, 154, 223] proposed a PSO swarm inspired by the atomic model, consisting of agents clustered close to each other dedicated to solution exploitation and agents surrounding the central cluster dedicated to exploration.

To solve the *Traveling Salesman Problem* (TSP), [224, 225] proposed using an Ant Colony System in conjunction with two sets of agents associated with two different levels of pheromone sensitivity. Agents more sensitive to pheromones are more likely to be influenced by environmental stimuli and are therefore able to exploit information stemming from marked search areas. Conversely, less sensitive agents are more likely to be independent explorers, allowing the discovery of potential search regions. The use of Ant Colony System that requires agents to modify and exploit their operating environment will be elaborated upon further in Section 4.4.2.

This use of dedicated agents has also seen applications in the particular class of dynamic best-of- n problems. In this scenario, an MAS is tasked with coming to a consensus on which option out of n possible options offers the best conditions to satisfy the needs of the system [122]. [187, 188, 192] introduced dedicated "stubborn" agents to their system. These agents did not change their opinion of a given site while other agents were allowed to change their opinion based on interactions with other agents using the voter model. These stubborn agents guaranteed a minimum amount of exploration carried out by the MAS and allowed the system to respond more quickly when there were sudden large changes in site quality.

Similar dedicated agents have also been used in various MRS. In the subCULTron project, a heterogeneous swarm of robots was developed and built for the monitoring of an underwater environment [216, 217, 218, 219]. This system relied on a more mobile set of agents, the "aFish", for exploration of areas within the environment, and a less mobile set of agents with higher sensor capabilities, the "aMussel", to exploit the local environment and collect data over long periods of time.

Similar heterogeneous MRS have been used for tracking of mobile targets in hybrid camera systems consisting of mobile and static cameras [232, 233]. However, in contrast to the subCULTron MRS, static cameras serve as the dedicated exploratory agents, identifying targets within the coverage area. The mobile cameras are then assigned to pursue the detected targets by a central controller.

4.3.2 Small Response Changes

In contrast to the large changes in state that result in drastic changes in an agent's behavior and response, an MAS' agents may choose effect smaller changes in its response patterns. While these minor adjustments in an agent's behavior may not result in such extreme changes in a system's overall EED, they allow for more precise adjustments of the level of exploration and exploitation carried out by an MAS. A list of such strategies can be seen in Table 4.6.

Attraction-Repulsion Dynamics

Effecting changes to an agent's attraction and repulsion strength with respect to its neighbors or specified points in the environment is one of the more common small response changes used. Increasing an agent's repulsion strength has the effect of augmenting the amount of exploration carried out by that agent. Conversely, increasing an agent's level of attraction favors more exploitative actions. Done across an entire MAS, this allows for regulation of a system's EED. When applied within an MRS context, the inter-agent repulsion also generates a collision avoidance behavior as a byproduct of this mechanism. Most repulsion vectors are calculated as follows:

$$V_i = \frac{a}{\|x_{i,t}\|^d} \frac{x_{i,t}}{\|x_{i,t}\|}, \quad (4.2)$$

where a is the set repulsion parameter, d is a constant, and $x_{i,j}$ is the vector from an agent i to a point of repulsion j . This point of repulsion is usually a neighboring agent and using an equation similar that seen in Equation 4.2 results in higher repulsion strengths being generated when two agents are close together. Hence, when using such repulsion equations, agents are strongly repelled from each other when they are immediately next to each other (see Fig. 4.2), discouraging excessive exploitation of the environment.

In a swarming MRS developed to find the source of a dynamic odor plume, [255], inspired by the Charged PSO algorithm developed by [222], increased the strength of an inter-agent repulsion parameter when robots came within a predefined distance of each other to avoid high levels of clustering within too small an area. It should be noted that in the experiments done by [255], the size of the environment and the number of robots in operation, and hence the global average swarm density, are known *a priori*. This allowed for the authors to tailor the repulsion parameters and optimize the robot's search behavior accordingly [260]. A similar repulsion scheme was adapted by [134] for use in their dynamic target tracking swarm. Similarly, to prevent excessive agent clustering

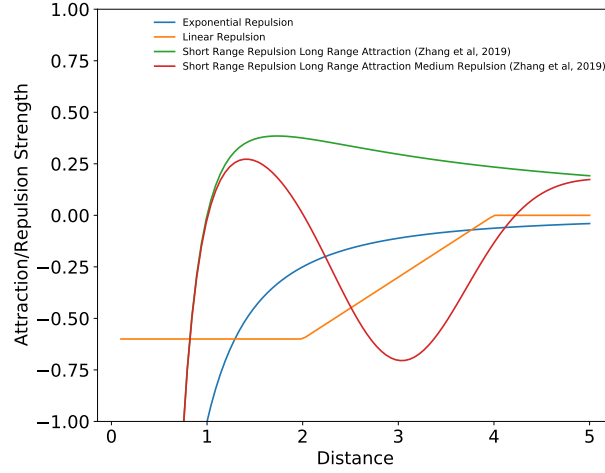


FIGURE 4.2: Inter-agent interaction forces generated by various repulsion and attraction schemes. Positive values indicate an attractive force and negative values indicate a repulsive force.

while carrying out a target tracking task, [128, 194, 195, 144] employed an adaptive repulsion algorithm that gradually increased the strength of inter-agent repulsion while agents were exploring the area. This repulsion strength was also gradually decreased while agents were attempting to move towards a target. In these works, a repulsion equation similar to that seen in Equation 4.2 was used.

In addition to inter-agent repulsion to prevent excessive aggregation when tracking multiple mobile targets, [258] also varied an agent's attraction strength to a target based on the distance between an agent and the target, with targets further away triggering a weaker attraction response. Furthermore, the authors also implemented agent-target repulsion when an agent moved too close to a target to prevent excessive exploitation. Developing this strategy further, [259, 205, 206] reduced the weight of the point of attraction generated by targets already being tracked, preventing the overlap of robot sensing areas, further reducing the over-exploitation of a target. In these works, a linear repulsion function was used, as seen in Fig. 4.2, instead of the exponential function that is normally used.

In a target search task, [256] ignored inter-agent repulsion and focused only on the use of attraction by combining the Lévy walk together with a Firefly Optimization algorithm to generate a velocity vector for each agent at each time-step. An agent's velocity is influenced by the attractiveness value of other agents within its communications range, encouraging exploitation, and a random point within the environment obtained from a Lévy distribution, encouraging exploration. Using the proposed strategy, an agent's attractiveness is based on a function on the amount of time elapsed since it last encountered a target. An agent that sees a neighbor with a high level of attractiveness will bias its movements more towards its neighbor than towards the point of exploration.

Besides the inter-agent repulsion behavior more commonly utilized, [257] also implemented an inter-agent short range repulsion and long range attraction (SRLA) component within their strategy based on inter-agent distances, effectively preventing excessive exploration for their k -capture game carried out by an MAS. In addition to this, they developed a Short Range Repulsion Long Range Attraction Medium Range Repulsion (SRLAMR) scheme where the inter-agent force transitioned between repulsion and attraction depending on the inter-agent distance. These two attraction and repulsion schemes were derived from the work done by [261, 262] and can be seen in Fig. 4.2. Comparing these schemes with the exponential repulsion schemes, the authors found that using the exponential repulsion scheme resulted in too much exploration, preventing targets from being captured, while the SRLA and SRLAMR schemes promoted agent clustering, facilitating the capture of targets. However, the SRLAMR scheme outperformed the SRLA scheme due to the former resulting in the dynamic formation and dissolution of agent clusters over time. These clusters were also distributed across the search space, demonstrating a balance between exploration and exploitation, allowing the system to capture fast-moving targets. By combining and adjusting the strength of separate inter-agent attraction and repulsion fields, [170] has demonstrated that a stable equilibrium position can be attained where agents maintain a fixed relative position to each other even though the entire swarm may be in motion. This allowed for the overall surface of an MRS, and hence the overall EED of the swarm, to be controlled even though the entire system may be moving.

Inter-agent attraction and repulsion is also commonly used in MAS tasked with area coverage and monitoring, which is a task that requires the prioritization of exploration over exploitation. [242] demonstrated how agents of an MRS can be distributed and deployed across a search space through the use of an exponential repulsion scheme. In addition to inter-agent repulsion, robots were also repelled by obstacles in the environment, allowing robots in their MRS to distribute themselves evenly across complex environment shapes.

This area coverage task was expanded on by [243, 244] who demonstrated the use of an physical MRS system for a dynamic area coverage problem: i.e., covering an area that changed shape over time. To achieve this, they augmented inter-agent repulsion together with a potential field gradient attracting agents outside the designated monitoring area towards the area perimeter. This was done to limit unnecessary exploration and encourage exploitation in the correct areas.

Comparable attraction-repulsion dynamics were also demonstrated by [245]. In their area coverage strategy inter-agent repulsion was used to prevent coverage area overlap, therefore stopping over-exploitation of the environment. This was supplemented with an agent-based “selfishness” term that directs the agents towards less visited areas, thereby promoting system exploration.

Random Walks

In environments where there is a lack of information, strategies based on attraction-repulsion dynamics may not be viable due to the lack of stimuli to drive agent activity. To operate in such information sparse environments, random walks, such as Lévy walks, are a popular strategy to promote area exploration. However, the use of Lévy walks may result in inefficient searching due to possible inter-robot collisions and coincidental clustering of robots. As such, [182] developed a random walk in which agents determine their walk sizes based on a local swarm density estimation. This allows an agent to carry out longer random walks—and hence carry out exploration when in a less crowded areas and shorter walks—while allowing the agents to carry out more exploitation when in areas with higher agent densities. The use of an adaptive step size is a common class of strategies in computational optimization algorithms [246, 247, 248, 249]. In such strategies the step size is varied based on the path taken by the solution; the step size is decreased should the solution stay around the same region of the search space for several iterations, enabling better exploitation, and increased should the solution move in the same direction for several consecutive time-steps, facilitating exploration.

Similarly, [180] also varied the Lévy parameter, μ , of individual agents using a parameter based on the amount of time elapsed since an agent's last target encounter. Using this strategy, an agent's Lévy parameter value continuously decreases towards unity while an agent does not encounter any targets. Doing so allowed for an agent to gradually prioritize exploration if it does not encounter a target for long periods of time. In addition to adjusting an agent's Lévy parameter based on the number of resource encounters within a given time-frame. Similar to this, [254] also explored varying a parameter that determines an agent's propensity of prematurely interrupting a Lévy walk should the agent experience a decreasing number of resource encounters. By increasing the chances of interruption, the agent is more likely to change its search path and explore the search area.

During their exploration phase while carrying out a foraging task, [201] used an informed random walk that is initially undirected and localized to look for patches of resources. As an agent gathers information about resource locations, these walks become more directed as agents tend to move directly towards higher yielding patches. This allows agents to exploit high rewarding resource patches while retaining some ability to explore the environment for better locations.

Time Intervals

The use of environment sampling intervals as well as communication time intervals is also a method that can be used to control the level of exploration and exploitation carried out by an MAS. This is the primary method used by the BEECLUST algorithm first proposed by [171, 172, 250]. This algorithm was formulated after observing bees tending to cluster in regions of higher temperature and in regions where clusters are already present. When implemented in an MRS, agents attempted to find an area illuminated by the source with the highest light intensity where the light intensity of

the sources varied over time. Using this algorithm, agents measure their local environment's light intensity only when they collide with other agents and proceed to wait in position for a period of time. This waiting time is determined by each agent individually and increases with the measured light intensity. This leads to a positive feedback loop where longer wait times lead to more aggregation in areas with higher light intensity. This phenomenon is known as Motility-Induced Phase Separation (MIPS) [263] and eventually results in higher levels of exploitation carried out by the system in areas of high agent density. Algorithms making use of MIPS were later studied by [252] for use in underwater environments and demonstrated in physical robot platforms by [253].

While effective, the positive feedback loop caused by MIPS may cause the system to over-exploit a specific area and result in lower levels of responsiveness in the presence of a highly dynamic light source. As such, [173] modified algorithm to account for an agent's local swarm density when determining its wait time. This modification allowed an agent to move off earlier when located in an area of high swarm density, preventing over-exploitation and improving the system's ability to respond and adapt to a highly dynamic environment.

In their area exploration task using agents with very low communication ranges, [118] varied the time interval at which robots regrouped to trade map information at a predefined position. Using this strategy, shorter time intervals forced agents to regroup more often, favoring exploitation. Conversely, longer intervals allowed agents to move further before regrouping, allowing for more exploration. The simulation results obtained suggested that shorter time intervals and higher levels of exploitation performed better in smaller environments while longer time intervals and more exploration yielded better results in larger environments.

Sampling time intervals were also studied by [120] in their area characterization and collective decision-making task. The authors developed an algorithm that caused agents to sample their environment and communicate with their immediate neighbors only after specific time intervals. In changing these observation time intervals, it was shown that shorter intervals result in agents taking more measurements, increasing their rate of exploitation, and being able to accurately characterize their local area. However, the overall system displays lower decision accuracy when characterizing the entire area. As with the work done by [118], this is the result of over exploitation done by the agents due to multiple measurements being taken of the same region of the environment, leading to a high level of spatial correlation between the robots' location and the MRS consensus. In contrast, longer observation intervals lead to less samples being taken. However, since the robots sample over a larger distribution of tiles, a higher level accuracy obtained by the system when attempting to characterize the whole environment.

4.3.3 Area and Task Assignment

In addition to the previously mentioned methods, agents can also be assigned to a specific area or task to manage the overall EED of an MAS. Doing so usually requires

an estimation of where points of interest are expected to be and the location of agents at any given point in time. Also, doing so allows for a belief map of to be generated that maps both the estimated demand for and supply of agents. Moving agents from areas of high supply and low demand allows for more exploration to be carried out by an MAS and prevents the over-exploitation of certain areas. These maps can be generated by individual agents themselves based on local measurements or be done on a global level and transmitted down to the agents. A list of such strategies is summarized in Table 4.7.

Using independently maintained agent supply and demand estimation maps in their target tracking task, [266] derived a utility distribution to drive robots towards areas of the search space in which they would provide the highest amount of utility to the entire system. Similarly, [264] in their patrolling task used a central controller to partition a search area into multiple sub-regions that were assigned to individual agents. Doing this allowed areas that saw more target visits to be partitioned into smaller sub-regions, thus facilitating the assignment of more agents and higher levels of exploitation in that specific area. Belief maps were also used by [190, 191] in a dynamic task assignment scenario in conjunction with task transition probabilities. Using their strategy, agents would increase their probability of performing tasks with higher demand but low agent supply.

In situations where communications are denied in the operating environment or when belief maps are not used, agents can be activated probabilistically to carry out a task based on their level of experience. Such strategies are usually carried out through the use of probability thresholds, with less experienced agents being more likely to be deployed to carry out their own exploration of the environment. [265] utilized this type strategy in a resource foraging scenario where agents with less map information had a higher probability of being deployed in the environment to gather resources, giving them the chance to build up their personal maps of the operating environment. While this type of strategy does not yield the best performance, it allows the system to maintain a level of robustness by having a large pool of experienced agents. This in turn enables the system to continue with its task in the event where its most experienced agents are lost.

4.4 Information Dissemination Methods

The second broad category of approaches towards controlling the exploration and exploitation dynamics of an MAS is based on adapting the way information is disseminated across the system. This system-wide information transfer can be achieved by means of direct or indirect communication strategies. Indirect methods are often observed in nature with agents altering the environment to convey information to fellow agents: this process is known as stigmergy. For example, some ant colonies use pheromones to guide worker ants to various food sources [267, 268]. Direct methods can be observed in colony of bees when the dancer bee dances to signal the onlooking bees on the quality of the food sources found. With engineered systems, radio transmissions offer a direct means of communication that is the most common among MRS

for dissemination of information. These methods are generally used in tandem with an agent response method that controls the amount of information an agent has access to. By doing so, an agent's actions and responses can be influenced indirectly and can have large effects on the system's collective response to the environment [3]. A list of main strategies that influence an MAS EED through changing the level of communications between agents can be found in Table 4.8.

4.4.1 Changes in Network Topology

MAS that use direct communication methods can control the information flow within the system by varying several possible communication network parameters, such as number of neighbors (for individual agents), agent's communication range, and various techniques like sub-swarm assignment and forced network change. These are further elaborated in the following sub-sections.

Number of Neighbors

One of the network parameters that can affect the dissemination of information is the number of direct (or immediate) neighbors that each agent possesses. It is worth stressing that the concept of "neighbor" shall not be limited to classical definition according to a metric or Euclidean distance. Instead, this concept should be considered in its most general form, which is conveniently captured by the network theoretic concept of node degree [281]. In general, most of the studies consider the number of neighbors as a system-level parameter—i.e., all the agents i have the same number of neighbors $k_i = k$ —although it is absolutely possible to envisage a heterogeneous system, in which the k_i values are different. When considering this parameter k , it is crucial to distinguish between directed and undirected networks. Specifically, with a directed network, the information flows across the system in one direction, i.e., if agent A is a neighbor of agent B then the information flows from A to B , but not necessarily the other way round. Directed networks make the relationship between agents asymmetric in terms of information flow. On the other hand, with undirected networks, that relationship is symmetric, meaning that the information flows in both directions when two agents are connected.

Through these repeated local information exchanges, an agent is able to form local decisions based on the information gathered by itself and its immediate neighbors—still in the network sense. For instance, using the PSO algorithm, changing the number of neighbors an agent communicates with is one of the key methods for controlling the level of exploration and exploitation carried out by the system [273, 274, 275, 276, 128, 194, 195, 144]. By increasing k , information about the global "best" solution can be propagated more swiftly around the system, thereby encouraging exploitation. Conversely, by reducing the degree k , information propagation is slowed down. This prevents agents from over-exploiting the same information, thereby increasing system exploration.

To prevent agents from being trapped in local optima, [174] developed a robotic PSO strategy using a synthesized dynamic neighborhood. In this framework, an agent expands its neighborhood should it be determined to be trapped in a local optimum. By using such expanding neighborhoods, trapped agents have a chance to find and follow a "leader" with a better fitness value, thereby progressing towards the global optimum. This concept was further explored in [272, 27, 280, 128, 194, 195, 144], where various k -nearest neighbor systems were evaluated in multiple targets tracking and multi-robot mapping operations. It was observed that system performance tends to improve as k increases to an optimal value [272, 280, 128, 194, 195, 144]. Beyond this optimum, increasing k yields detrimental effects to the system performance due to the MAS' tendency to over-exploit information at higher levels of connectivity. This optimal level of connectivity was also found to be dependant on the rate at which the environment evolves; within a fast changing dynamic environment, a system with lower k performs better due to the slower spread of outdated information across the system, thereby promoting exploratory behavior.

Such observations have also been made when observing MAS in static environments. [148] observed an optimal k when attempting to obtain multi-agent consensus in a noisy static environment and reduce the error between this consensus and the ground truth. The simulations also demonstrate a well-known result in network science where efficient and well connected networks result in better short-term results to the detriment of long-term performance while more diverse and less connected networks result in poorer short-term results and better performances in the long-term [282, 283, 284]. When attempting to reach a consensus on the best direction of movement, [270] demonstrated that there was an ideal *attention limit*, the number of stimuli an agent can process at any given point in time, at which the accuracy of the system consensus was maximized. This occurs because a large attention span results in strongly connected flocks that tend to converge and exploit a single source of information very quickly while a small attention span leaves the system vulnerable to small variations in an individual's movement. Due to the rapid convergence on a single source of information—i.e., over-exploitation, it is very difficult for informed individuals to inject new sources of information to the system and promote further exploration, resulting in low system accuracy. However, when agents have a very low attention limit, the system becomes susceptible to noise and variations in the movement of individual agents, causing the system to explore many different potential sources of information, also leading to the degradation of the system's performance [272].

[27] also used this method of control by changing the number of neighbors in the network to perform collective mapping operations. In this work, the individual robotic units get access to sensor information based on the k -nearest-neighbor network. This information, in the form of occupancy grid maps, is used by the agents to form localized decisions about the next waypoint. This allows for a fully decentralized control of the agents to move towards unexplored spaces via frontier exploration [285].

Communication Range

For the operation of MRS, the maximum communications range between two robots is of significant interest as it is one of the key factors that determines the maximum number of neighbors in an agent's communications network. As such, many studies carefully focus their investigations on the limitations associated with the communication range.

[278] studied the swarming system behavioral plasticity at the collective level in dynamic environments with a class of foraging problems. The nature of the changes occurring in the environment is associated with the quality of food sources in the environment—i.e., at each food patch quality change interval, the deposited quality changes. Various strategies have been explored in these dynamic environments. Some strategies are based on the idea of exploration-exploitation by tuning two parameters: the probability of switching to an exploring agent and the communication range. It is interesting to note that in static environments, a “maximum” information sharing rate is beneficial when the system makes collective decisions. However, when the swarm operates in a dynamic environment, information transfer needs to be carefully controlled to achieve a balance between exploration and exploitation. This balance is essential for the system to achieve high levels of adaptivity to find new and possibly better quality foraging sites, instead of constantly exploiting the same foraging site(s).

Along the same vein, [269] worked on finding and maintaining a balance between peer agreement (exploitation) and exploration of new solutions. Their simulated swarm is modeled after locusts in a two-dimensional space and their movement is restricted to either clockwise or counter-clockwise. The agents use information available to them (e.g., neighbors and its own information) and taking into account system noise to locally decide whether to move in a clockwise or counter-clockwise direction. This work analyzed the relation between social feedback and noise, and collective coherence. The method aims to minimize the number neighborhood connections while maximizing the system's collective coherence and response to the sudden addition or removal of agents (i.e., changing swarm sizes). Classically, the global agreement is an emergent behavior resulting from the sum of local coordination efforts. However, if the neighborhood is small, the agents are more willing to adapt their actions to the local environment, a result also demonstrated by [271, 277] in a movement consensus task and in a dynamic optimum area aggregation task respectively. However, this over-willingness to adapt leaves an agent vulnerable to random fluctuations of its neighbors' output—i.e., the swarm becomes undecided, with a low degree of coherence [286, 287]. This is similar to using too little neighbors in an agent's communications network as explained in the previous section. Therefore the strategy deployed controls the swarm by changing the size of its neighborhood, and hence its level of exploration and exploitation.

[279] proposed a Non-Reciprocal-Interactive-Based (NRIB) model based on the idea of “exploration-exploitation.” Essentially, a non-reciprocal interaction network is directed. Initially, the homogeneous agents are spatially distributed in the field. Once an agent enters an area of tasks, its workload increases, while the area tasks decreases at a constant rate. As the neighbors detect this increase in workload within its network, they will approach it and execute it cooperatively, effectively exploiting this new source of

information. When the tasks are completed, the workload of the robots will naturally decrease and they will start repelling one another, exploring the area and seeking out other tasks.

Similar to the work done by [269], the improved NRIB model includes a key parameter to better direct the system's exploitative activities: the radius of communication range. The intention is for the agents to look for other agents that need help outside of their initial neighborhood. In case where there are too many agents with a high workload, it could create confusion for the agent to determine its next waypoint. Therefore, reducing the radius of communication range can help the agents find the nearest neighbor with the highest workload, allowing exploitative actions to be better directed. An automatic tuning mechanism is implemented to adjust the radius of communication range based on the workload of the neighborhood.

Forced Network Changes

Should a fixed level of connectivity be desired, changes can be made to the network's structure by changing an agent's communications neighbors. This allows an agent to maintain the same communication range or number of neighbors, while allowing for more information diversity in the agent's network. [161] dealt with online multi-object k -coverage. The strategy is inspired by an ant foraging method that consists in both negative and positive feedback loops. When two cameras are connected to each other, this connection is considered as a link and that will be reinforced as long as the cameras continue to "see" the same object. The strategy also allows for the weakening of this connection as all link strengths decay over time. This decay opens the possibility for it to be used in dynamic environments as the system can unlearn the links, preventing the over-exploitation of outdated information sources.

In the work by [161], various network structures are used, namely the k -nearest, k -furthest, and k -random networks. The k -nearest and k -furthest networks are based on Euclidean distance, while the k -random network is based on a set of randomly chosen neighbors across the entire system, allowing the communication to gradually expand to "all" cameras. Using topologies such as the k -furthest or k -random networks essentially allowed the agents to explore a larger area by permitting the gathering of information from neighboring agents further away. This increased amount of exploration was studied while using various response models (e.g., moving to assist in the tracking of the closest target, moving to assist in the tracking of the target with the lowest number of pursuing agents, etc.). Using wider networks paired with the response model increased the level of exploration of an agent as well as allowed the agents to select the most appropriate source of information to exploit that will benefit the system the most.

[1] demonstrated a "Forced-Switching" network model for their distributed consensus task, where an agent switches its k_s nearest neighbors to a more distant set of neighbors with a given probability, p (see Fig. 4.3). This allows agents to obtain a wider, yet still local, information set. Similar to the k -furthest and k -random network structures used in [161], this network topology essentially allows an agent to explore the environment

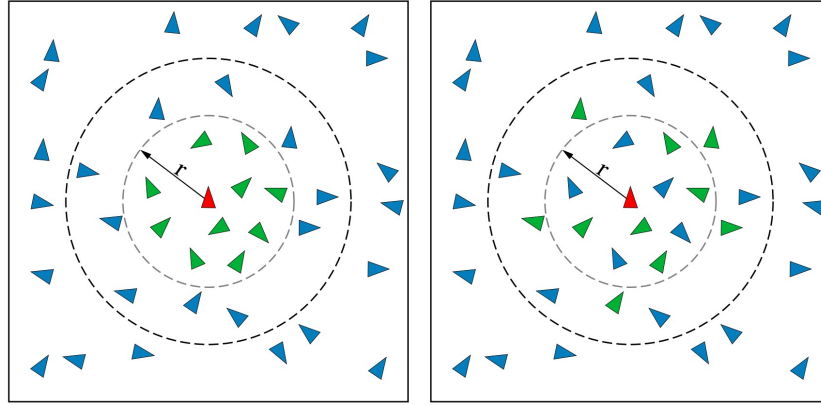


FIGURE 4.3: (Left) A standard communication model with an agent in communication with its 10 nearest neighbors. (Right) A forced switching communication model where $k_s = 5$ of the nearest neighbors are substituted with a more distant set of neighbors within the black circle with a probability of p . [1]

more by allowing an agent to sample from a larger set of information. This in turn leads to higher levels of consensus, especially in slow evolving environments.

Lastly, [288] analyzed the influence of switching agents on the topology of the network of interaction when considering simulations of self-propelled particles. The overall coherence of the swarm—measured through its polarization—was found to increase with agent switching.

4.4.2 Stigmergy

Should an MAS operate in an environment where direct agent-to-agent communications are denied, indirect methods, such as stigmergy, can be used. In this framework, an agent is able to elicit a response from other agents in the system by leaving traces in the environment [302]. This concept was first introduced in [303] and has been applied in various MAS.

The most notable application of stigmergy is the Ant Colony Optimization (ACO) algorithm, first proposed by [291]. It has been used in various optimization tasks ranging from vehicle routing problems to electrical grid management [304]. Using the ACO algorithm, agents deposit markers, also known as pheromones, as they traverse the environment. These pheromones are most commonly used attractively, drawing agents to points of interest, with most commonly used pathways being reinforced with higher pheromone concentrations, hence resulting in a positive feedback loop encouraging exploitation. To prevent over-exploitation and encourage exploration, these pheromones are subject to evaporation, causing the attractive strength of a given pathway to decrease over time [305]. In addition, pheromones can also be subject to diffusion, allowing agents to explore adjacent solutions [292, 293]. As previously mentioned in

Sec. 4.3.1, Ant Colony algorithms can also be used in systems comprised of two different levels of pheromone sensitivity [224, 225]. This allows for a dedicated sub-group tasked with exploration and another for exploitation.

In addition to computational optimization tasks, these algorithms have also been implemented in various MRS. [294] simulated the ACO algorithm's use in a robot swarm tasked with finding the shortest path between two points in a dynamic environment. However, in addition to the rate of pheromone evaporation, they also re-initialize the system and erases the existing pheromone trail when the system is determined to be approaching stagnation. This strategy essentially triggers a large increase in the amount of exploratory activity carried out by the system's agents. During this period, the system re-explores the solution domain before it is able to exploit a new set of information gathered through the exploration process and finally converges on a new solution. This is similar to the re-initializing of PSO strategies seen previously in Sec. 4.3.1.

Several groups have worked to bring the ACO algorithm from the virtual to the physical world, showing the feasibility of using such exploration and exploitation control strategies in real-world problems. While using robots to carry out a target search task, [295, 296, 297] spread out RFID data carrier tags around the physical operating space to act as pheromones within the environment. A similar strategy was also employed by [298, 299], who used a physical RFID grid to generate a map exploitable by a swarm of robots, guiding them towards a target. Using this type of strategy, robots are equipped with an RFID writer/reader and update the pheromone levels of the tags when they are encountered. Doing so allows an agent to exploit the environmental information by following a gradient descent algorithm towards the target. Similar to their counterparts in computational optimization, these pheromones are also subject to evaporation, preventing the robots from exploiting outdated information. [300, 301] later built on this work by supplementing the strength of the pheromone with a vector pointing the direction of areas with higher pheromone concentrations. This allows an agent to carry out more effective (targeted) exploitative actions, improving search efficiency and success rates when searching for both dynamic and static targets.

Instead of using pheromones attractively, [289, 290] have also shown that they may be used in a repulsive manner in their area mapping and exploration scenarios. Similar to "no-entry" signals in ant foraging patterns [306], these negative pheromones facilitate area coverage and exploration by preventing agents from taking overlapping paths, thereby encouraging agents to take paths not previously explored. However, [290] has shown that agent exploration is limited using this strategy when there is an agent located in an area with higher swarm densities. This is because the large amounts of repellent pheromone in the vicinity of an agent limits its movement, preventing it from carrying out any useful exploration.

4.5 Conclusion

The so-called exploration–exploitation dilemma is a common challenge faced by many systems operating on the basis of collective decision-making, including those encountered by various multi-agent and multi-robot systems. This dilemma stems from the

fact that, in general, exploration and exploitation tend to be mutually exclusive tasks.

The allocation—be it automated or human driven—of agents to these exploration and exploitation tasks is nontrivial; a large proportion of agents carrying out exploration would indeed result in many points of interest being identified, however, the system would not be able to fully benefit from these points. Conversely, while a large proportion of agents carrying out exploitation would increase the rate at which the system profits from the information currently available, the system would not be able to adapt to a dynamic environment—in particular a rapidly evolving one—and may not be able to maximize the amount it profits from the environment.

Given the challenging nature of this problem and the diverse range of applications for MAS and MRS systems, there is a large body of literature dedicated to studying the balance between exploration and exploitation in such systems. These studies and works often originate from different disciplines, (e.g., autonomous multi-agent dynamics, field robotics, and service robotics) and utilize various tools and methods to develop solutions and strategies in relation with their specific problems (e.g., control theory, simulations of self-organized behaviors, and physical experimentation). Given the multidisciplinary nature of this problem and the large body of work reported in the literature over the past decade, in this review, we identify and classify the commonalities among different solutions and frameworks, as well as between the different definitions of exploration and exploitation.

To this end, we have categorized the methods used to control the level of exploration and exploitation carried out by an MAS into two broad categories, namely “agent response methods” and “information dissemination methods.” The latter can be further broken down into “network topology methods”, where a system’s communications network parameters are changed to regulate the flow of information across the MAS, and stigmergy, where an agent is able to indirectly elicit a response from other agents in the system by leaving behind a signal in the environment. Similarly, the broad category of “agent response methods” can be further broken down into: (i) methods that cause an agent to display large and drastic changes in the way it deals with the environment, (ii) methods that result in smaller changes in an agent’s response to its environment, and (iii) the assignment of agents to areas of interest or tasks found in the environment.

In addition, we have also summarized the methods used in various studies to quantify the level of exploration and exploitation carried out by an MAS. Such quantitative analyses rest upon the proper definition of metrics, which are key to estimating an MAS’ exploration–exploitation balance, thereby allowing one to see if a proposed strategy is working as intended, and make adjustments as needed. However, as is apparent in this review, metrics tend to be closely tied to their specific applications/problems and there is no generalized metric available to measure exploration and exploitation. This important point has also been raised in previous reviews, and a similar observation was also made in reviews about swarm robotics [307] as well as human-swarm interaction [308]. The lack of a single/absolute metric applicable to different problems limits our ability to truly rank different approaches and strategies. Even if there is no guarantee that such a universal metric exists, we strongly believe that the community working

on these problems would benefit from ways to compare and rank current MAS strategies.

On top of the metric issue just discussed, it is worth stressing that the fact that most methods/strategies are developed for a specific problem is also an issue. The effectiveness of any approach should clearly be scenario independent. A swarm's flexibility and adaptivity—i.e., swarm intelligence—is the key concept underpinning this, and it has never been understood as limited to one particular task. In our view, a true swarm should exhibit flexibility and adaptability in vastly different scenarios. Although, seeking swarm intelligence in its most general aspect is a laudable objective, it is still an extremely challenging task in practice, and multi-agent reinforcement learning (MARL) is attempting to do just that [309, 310]. Nonetheless, the design of a benchmark problem would offer the possibility to quantitatively compare the various approaches considered for this problem of exploration–exploitation balance. In the process of designing such a benchmark problem, one can also identify a range of scenarios where swarm intelligence is advantageous for the task. In other words, it is critical to determine the set of environmental parameters that calls for the deployment of a swarm, and for swarm intelligence to enable the effective cooperation of the MAS. For instance, it could be debated if swarm intelligence would really be necessary when dealing with a static problem.

We argue that to probe and quantify swarm intelligence, an MAS has to be subjected to challenging circumstances, which would be equivalent to some sort of “stress test” for flexibility and adaptivity. It is only then can one assess and comment on the level of swarm intelligence demonstrated by a system. We propose that such a “stress test” could take the form a benchmark problem with a range of specific features. As previously explained, the environment in which the MAS is operating should be dynamic and preferably with a relatively fast rate of evolution when compared to the dynamics of a single agent. Specifically, a single-agent system should be unable to cope with the pace of changes in the environment, whereas an MAS made of several of these agents could cooperate to navigate such fast-evolving circumstances by almost continuously adapting its exploration–exploitation balance. The system's exploratory behavior can be evaluated as it attempts to find or accomplish multiple objectives (e.g., tracking of multiple targets, mapping of multiple areas). The system's exploitative behavior can be evaluated as it attempts to exploit multiple spatial objectives—i.e., objectives with spatially distributed tasks within the environment. Indeed, a single spatial objective can conveniently be tackled with reduced exploration since the exploitation is highly localized within the environment. From the perspective of a virtual MAS, it is possible to consider ideal sensors (without noise), localization, and inter-agent communication. This therefore eliminates a number of unknowns as the MAS algorithm is being evaluated on a selected benchmark problem, allowing for a more accurate evaluation of the algorithm's true flexibility.

In this review, we deliberately avoided making a distinction between theoretical analysis, simulations, and actual robotic experiments. However, it is important stressing that performing large-scale experiments with a large number of robots still remain a challenging and complex task [130]. While an ideal robotic system will act in the same way as its simulated counterpart, various factors (e.g., robot kinematics, environmental

conditions, and noise) will affect the final behavior of an MRS. The differences between simulated and physical systems will only grow as the number of MRS agents increase. Currently, a large proportion of experimental results available tend to be limited in size, scope, and robustness. This highlights the need for more extensive experimentation involving larger systems operating in more complex environments [311].

Obviously, this particular area of research is a fast-moving one, given the ever increasing technological readiness level of MRS. As detailed in two recent comprehensive reviews [129, 108], one can expect to see a growth in the number of large-scale MRS deployed to tackle a number of tasks requiring a fine balance between exploration and exploitation, which has been shown throughout this review to be rather complex and challenging.

Reference	Metric	Task
[160]	Area observed by at least two agents	Area Coverage
[154, 165, 141, 138]	Euclidean distance between solutions	Optimization
[155]	Euclidean distance between solution and mean system solution	Optimization
[156]	Euclidean distance between solution and median system solution	Optimization
[157]	Standard deviation of all candidate solutions	Optimization
[174]	Fitness value between sub-swarm leaders and mean fitness value	Optimization
[172, 173]	Local agent density	Optimal Area Aggregation
[152, 110, 158, 159]	Agent sensor overlap region	Target Search
[166, 167, 168, 169, 134, 170]	Euclidean distance between agents	Target Search
[132]	Number of agents within visual range	Target Search
[161, 162, 163]	k -Coverage	Target Tracking
[164]	Distance between agents, distance between targets and agents & distance moved by agents in one time-step	Target Tracking

TABLE 4.1: List of works using spatial distribution metrics to quantify MAS exploration and exploitation.

Reference	Metric	Task
[183, 184, 185]	Lévy parameter	Area Mapping
[187, 188]	Opinion switching probability	Best-of- n Problem
[178, 180]	Lévy parameter	Resource Foraging
[186, 179, 146]	Lévy parameter and memory utilization probability	Resource Foraging
[189, 190, 137, 191]	Task switching probability threshold	Task Allocation
[176, 177, 181, 182]	Lévy parameter	Target Search

TABLE 4.2: List of works using probability based metrics to quantify MAS exploration exploitation dynamics.

Reference	Metric	Task
[187, 188, 192]	Number of stubborn agents	Best-of- n Problem
[193]	Number of unique messages broadcast & proportion of agents broadcasting unique messages	Collective Decision-Making
[128, 194]	Correlation between an agent's direction of travel and directional bearing to target	Target Tracking
[196]	Tracking fairness	Target Tracking
[195, 144]	Engagement ratio	Target Tracking
[197]	Tracking Fairness and time-based occupancy grid map	Target Tracking

TABLE 4.3: List of works using their own developed methods or metrics to quantify the level exploration and exploitation carried out by an MAS.

Reference	Strategy	Task
[199, 200]	Agent behaviors selected using AIS	Dynamic Shepherding
[201]	Random walk search phase & agent assignment exploitation phase	Resource Foraging
[179, 146]	Lévy Walk exploration & memory driven exploitation	Resource Foraging
[202, 178]	Dedicated exploration & exploitation Lévy parameters	Resource Foraging
[203, 204]	Random search & PSO tracking strategy	Target Capture
[205, 206, 161, 162]	Random search & target following strategy	Target Tracking (CMOMMT)
[163]	Pattern search & target following strategy	Target Tracking (CMOMMT)
[128, 194]	Repulsion based exploration & PSO tracking strategy	Target Tracking
[195, 144]	Repulsion based exploration, PSO tracking strategy & adjustable memory length	Target Tracking
[164]	Agent behaviors selected using AIS	Target Tracking
[181]	Dedicated exploration & localization Lévy parameters	Target Suppression
[207, 208]	Agent behaviors selected using AIS	Task Assignment

TABLE 4.4: List of works using self-determined state change strategies to influence a system's exploration and exploitation dynamics.

Reference	Strategy	Task
[215]	Dynamic sub-swarm membership	Area Exploration
[187, 188, 192]	Stubborn agents that do not change opinion of site	Best-of- n Problem
[216, 217, 218, 219]	Dedicated environment exploration and long term monitoring (exploitation) agents	Environment Monitoring
[220, 221, 157]	Forced re-initialization of a subset of candidate solutions	Optimization
[222, 154, 223]	Dedicated exploratory and exploitative particles	Optimization
[224, 225]	Ant Colony System paired with agents with preset pheromone sensitivity	Optimization (Traveling Salesman Problem)
[226, 227, 228, 229]	Dynamic sub-swarm membership	Target Search
[230]	Search strategies assigned based on agent's distance to target and target-to-searcher ratio	Target Search
[231]	Forced re-initialization of robots	Target Search
[232, 233]	Predetermined static agents for target search and mobile agents for target tracking	Target Tracking (CMOMMT)
[234, 235, 236, 237, 238, 239]	Specified number of agents closest to target chosen to utilize a target tracking strategy	Target Tracking

TABLE 4.5: List of works using assigned agent responses to influence a system's exploration and exploitation dynamics.

Reference	Strategy	Task
[242]	Exponential inter-agent repulsion strength	Area Coverage
[243, 244]	Exponential inter-agent repulsion & attraction gradient	Area Coverage
[245]	Exponential inter-agent repulsion with "selfishness" term to drive exploration	Area Coverage
[120]	Varying agent environment sampling time	Area Characterization
[118]	Varying time interval at which robots regroup to trade map information	Area Exploration
[246, 247, 248, 249]	Adaptive step size	Optimization
[222]	Exponential inter-agent repulsion strength	Optimization
[171, 172, 250, 251, 252, 253, 173]	Varying agent wait time based on local fitness value	Optimal Area Aggregation
[182, 180]	Variable Lévy parameter	Resource Foraging
[254]	Variable Lévy parameter & interruption of Lévy walks	Resource Foraging
[201]	Random walk that gradually becomes more directed	Resource Foraging
[255]	Exponential inter-agent repulsion strength	Target Search
[256]	Using Lévy walks and Firefly optimization algorithm to generate points of attraction for each agent with different attraction weights	Target Search
[257]	SRLA and SRLAMR agent interaction schemes	Target Search
[134, 170, 128, 194, 195, 144]	Exponential inter-agent repulsion and attraction strength	Target Tracking
[258, 259, 205, 206]	Linear inter-agent repulsion strength & variable target attraction strength	Target Tracking (CMOMMT)

TABLE 4.6: List of works using small response changes to influence the level of exploration and exploitation of an MAS.

Reference	Strategy	Task
[264]	Area partitioning	Patrolling
[265]	Agent deployment probability threshold	Resource Foraging
[266]	Agent utility distribution to drive robot movement	Task Allocation
[190, 191]	Belief maps & task transition probability	Task Allocation

TABLE 4.7: List of works using agent area and task assignments to influence the level of exploration and exploitation of an MAS.

Reference	Strategy	Task
[269]	Adjusting communications range	Collective Decision Making
[148]	Changing number of communication neighbors	Distributed Consensus
[270]	Changing attention limit of agents	Distributed Consensus
[271]	Changing interaction radius and field of vision angle	Distributed Consensus
[272]	Changing interaction radius	Distributed Consensus
[1]	Forced switching network model	Distributed Consensus
[273, 274, 275, 276]	Changing number of communication neighbors	Optimization
[241]	Dynamic sub-swarm membership	Optimization
[277]	Adjusting communications range	Optimal Size Aggregation
[278]	Adjusting communications range	Resource Foraging
[279]	Adjusting communications range based on internal workload state	Task Allocation
[174]	Changing an agent's number of neighbors in network based on fitness value	Target Search
[128, 194, 280, 195, 144]	Changing number of communication neighbors	Target Tracking
[161]	Varying communication link strength and number of communication neighbors	Target Tracking (CMOMMT)

TABLE 4.8: List of works using different information dissemination strategies to influence the level of exploration and exploitation of an MAS or MRS.

Reference	Strategy	Task
[289, 290]	Repulsive pheromones	Area Exploration
[291]	Ant Colony Optimization with pheromone evaporation	Optimization
[292, 293]	Ant Colony Optimization with pheromone diffusion	Optimization
[224, 225]	Ant Colony System with agents of different pheromone sensitivities	Optimization (Traveling Salesman Problem)
[294]	ACO with solution re-initialization	Path Finding
[295, 296, 297, 298, 299]	Gradient decent with physical data carriers	Target Search
[300, 301]	Gradient decent with vectorial pheromones and physical data carriers	Target Search

TABLE 4.9: List of works using stigmergy to influence the level of exploration and exploitation carried out by an MAS.

Chapter 5

Towards a benchmark framework that probes the flexibility of swarming systems.

5.1 Introduction

Often the collective behaviors exhibited by natural swarms like the coherence of a flock of birds changing direction as a whole during their flight, the army of ants traversing the ground bringing food back to its nests, and the school of fishes evading the attacks of predators are found to be intriguing. These captivating collective behaviors does not derived from instructions of a central entity as the natural swarms are fully decentralized, instead the individual entities follows a set of predefined rules to guide their actions. These rules are "written" in the insects and animals genetic code—e.g. a worker ant have a different set of rules that they operate within as compared to a queen ant.

On a system level, these collective behaviors are the effect of self-organization. This self-organizing collective behaviors that we observe in nature, emerges from repeated simple local interactions among the entities [3]. Often these collective behaviors suggest the existence of a form of intelligence—i.e. swarm intelligence (SI). SI is appealing in the physical world applications, especially when our world's problems are becoming complex.

The idea of "automation" has been around to provide solutions to past generations of problems where it aims to automate existing processes that are already performed by humans, so that the operations can scale up. However, as the operations increased in complexity, the solutions (or systems) applied will be required to perform information collection, computation, communication, decision making, and acting on these information and results. In order to foster system independence—i.e. not micro-managing the systems, SI is a promising avenue to offer self-organization to the systems. It offers "an alternative way of autonomy, emergence, and distributed functioning replace control, pre-programming, and centralization" [38].

SI algorithms are often metahuristic, being able to effectively find sub-optimal solutions, especially with incomplete information or limited computation capacity [312].

This characteristic of SI is applicable to an increasing prominence of NP-hard problems where real-time locating of global optima is almost impossible. In such problems, the number of potential solutions that exists tends to be infinite. Therefore, finding a workable or sub-optimal solution within time limitations becomes important.

The use of SI spans across a vast range of problems, in almost all area of sciences, engineering and industries, from data mining to optimization, artificial intelligence, business planning, robotics, and in industrial applications. Some other areas include, internet of things [313], disease diagnosis and detection [314], optimization for engineering modelling [315], navigation control, multi-robot system control, malignant tumour detection and control and image processing technologies [40].

Although SI solutions have found their way into the physical world, and some have already been implemented, Dorigo et. al. [36, 129] stated that the lack of a standardized design methodology for swarming systems persists. One of the contributing factors that we reckon is the lack of a quantification and comparison of SI. As the use cases for SI increases, the development of SI will ramp up. In Dorigo et. al. [36, 129] own words, in order “to concretely support the research community, benchmarks are invaluable tools to measure the progress in a quantitative way and can be used to challenge the researchers on tasks that grow in complexity.” Therefore, in this work, we proposed a benchmark framework that can allow for SI systems or algorithms to be evaluated for its SI. This benchmark framework can be use in future SI development work as a form of standardization when quantifying and evaluating SI. Similar to the field of reinforcement learning where there are standardize “toy” problems—e.g. cart-pole problem—for developers to evaluate their reward functions for effective learning. However, to the best of our knowledge, this form of comparison has not yet exist within the SI research community.

Even during the design process of a MRS or a robot swarm, there is a fundamental question of whether or not is SI even beneficial or required in solving current problems [37]. It is not a trivial to answer the latter question. In [37], Dorigo et. al. proposed a guideline to evaluate if there is a need for the application of swarm. First question: Is there a need to use of multi-robot system to (efficiently) solve the problem? In cases where the task is too complex or demanding, multi-robot system can be used, for example the use of multi-robot systems in mapping larger areas. Second question: What type of control approach would be most appropriate for the considered problem? For example, if the solution requires decentralization, robustness, and/or flexibility then swarming approach can be considered. This provided a systematic thinking process to evaluate the need for applying SI—i.e. the question of effectiveness of SI application. The proposed benchmark framework with the possibility of SI quantification and comparison, will be able to assist in evaluating the effectiveness of the applied SI solution. In [316], it is acknowledged that the design and implementation of an effective robotic swarm is among the greatest challenges in the area of robotics.

Before we discuss on the design of the proposed benchmark framework and the results obtained, there is a need to understand the definition of SI and why flexibility can be a measure of SI. In the following sections, we will explore i) the various ideas of measuring SI, ii) explore the current benchmark problems, iii) SI benchmark framework,

finally iv) results and discussions.

5.1.1 What is swarm intelligence?

The concept of swarm intelligence was first being introduced, in the 1990s, by Beni G. et. al. [13]. Surprisingly, it is not a work specifically on swarming system or swarm intelligence, but a work on cellular robotics. Beni G. attempts to define cellular robots' intelligence (which was identified as Swarm Intelligence) as follows.

In [13], Definition 2.2.1 ("Swarm Intelligence"):

"Systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to unpredictably (#) produce specific (=not in a statistical sense) ordered patterns of matter in the external environment.

Unpredictable is meant as globally 'intractable' or 'externally not-representable'."

Over the years, swarm intelligence has gained popularity and many researchers, from various field of studies, followed Beni et. al. in attempt to provide a definition from their perspectives. Largely, there are two broad perspectives of SI definition. It is eminent that these definitions were inspired by the early research works of biologist who observe and study natural swarming systems, and computer scientists who studied artificial intelligence, adopted biomimicry approach to gain inspiration to develop optimization algorithms. From the biologist-inspired definitions [38, 39, 34, 40, 41, 42], although there are some variations, the common features of these SI definitions seem to involve i) groups of simple agents, and ii) global emergent or self-organization, and three of these definitions included iii) inter-agents local interactions. These features mentioned made up the main features of SI [43]. Beni et al. definition did capture a few important essence of self-organization. Firstly, it mentioned "systems of non-intelligent robots". This reflects the often simple units we find in nature—e.g. an ant, fish, bird, bee, etc.—existing in a colony or a group which display an intelligence for beyond the individual's. Secondly, it mentioned "the ability to unpredictably produce specific ordered patterns of matter in the external environment.". This is in fact the emergence of the collective behavior which can be observed as "ordered patterns". However, it lacks in capturing the interactions of the units which is crucial in a system of social insects or social animals.

The algorithm-inspired definitions [44, 45, 46, 47] mainly describe SI as a bio-inspired computational methods(or algorithms) to solve optimization problems. Swarming systems have often been described as a collective information processing unit. In this case, SI governs the process at which information is acquired, processed, making collective decision, and acting on it. In [48], a collective operation and computation framework is proposed, in which a part is "local computation" that each agent performs using only information from a certain local neighborhood. This emphasize the importance of local communication and computation—i.e. local information processing and decision making via information from local neighborhood.

Although there are similarities in the definitions, the perspectives are still vastly different, and within each perspective, the definitions also hold differences. This presents

a long-standing challenge in SI study, that is the issue of non-unifying SI definition [43], which can also be inferred from earlier work that calls for quantification of SI [38]. The lack of a unifying definition, hence theoretical foundations, would pose significant problems in quantifying SI and the future development of SI systems. We could not make statements like "SS A is more (swarm) intelligent than SS B". The lack of quantification of SI will not allow SI comparison among system, noting that there is a difference between system's performance and system's SI. In many of the works, operational performance of the systems are often measured for comparison—e.g. machine learning, optimization research, etc. This provided a way for the improvement and further development of the technology. For example, in one of the most use case of SI, optimization, in a particular task like foraging, we can compare the performance of the use of Artificial Bee Colony(ABC) and Ant Colony Optimization (ACO) algorithm. If we use a general metric of "total amount of food gathered at the end of the simulation", most likely one algorithm will outperform the other. However, operational performance does not necessarily correlate with SI—i.e. the system can perform well and still not displaying SI.

The confusion that stems from the non-unified definitions of SI also entails missed opportunities as well. Now, it is not easy to identify the "right" conditions such that SI can be the most effective. Without a clear understanding of what SI is and how and why it arises, it is very difficult to envision how to take advantage of its true potential. Not having the method to compare SI will cause an eminent effect of artificial swarming systems not being able to improve and be more intelligent, which will cause the slow-down of real life applications and commercialization of this technology.

5.1.2 Quantifying swarm intelligence

After studying the various SI definitions, it is evident that self-organization (SO) or emergent behavior is a central and important feature that describes SI, also mentioned in [43]. SO, being the central feature of SI, that causes the display of collective behaviors, arises from the interplay of four lower-level components: 1) positive feedback, 2) negative feedback, 3) amplification of fluctuations, and 4) multiple interactions [38].

This brings us to another question, "What is self-organization or emergent behavior?". Still these terms are ambiguous as there are many interpretations. Serra et al. [50] describes the concept of self-organization generally as "highly organized behaviour even in the absence of a pre-ordained design." They go on to further describe examples such as the resonance phenomenon in lasers, and in cellular automata where "unexpected and complex behaviours can be considered as self-organized." Bonabeau et al. [38] described SO as a "key feature of a swarm system which results global level (macroscopic level) response by means of low- level interactions (microscopic level)."

Often, the noticeable effect of self-organization we observed in nature is the order, even in nature's migration of birds and schooling of fishes, etc. One method of quantifying order is by global consensus. For example the measurement of the direction of travel of the birds compared to the global direction of travel to find the order of the system

[51]. If the system is in order or coherent—i.e. the difference between all the agents' direction and the global system's direction is low—then it suggests a high level of order. On the other hand, a high difference would suggest a low level of order. However, the challenge in using order as a measure is that order can be derived from either a centralized or decentralized paradigms.

This is important as distinguishing between a centralized order and a self-organized order is not trivial. As pointed out by Cavagna in [52], systems' hierarchical and distributed control are often a combination of centralized and self-organizing (decentralized) control mechanism. Even distinguishing the difference between the two extreme cases of order from a top-down centralized control mechanism and a bottom-up self-organized feature emerging from local behavioral rules is not trivial. Furthermore, there is an interesting work by Alessandro A. et. al. [53] that documented the discovery of a display of collective behavior—i.e. the emergence of global order or pattern—without collective order in wild swarms of midges. Despite the lack of order, the swarms display high level of correlations, totally incompatible with models of non-interacting particles. This work challenges the long-standing self-organization definition or perspective of order or pattern.

There is another self-organization perspective by Prigogine et al. [54] that described self-organization in chemical reactions and thermodynamic contexts. His notion is quite enlightening and emphasizes the element of fluctuations in far-from-equilibrium system states. Non-linear system changes, fluctuations as he refers to them, due to either random events or chaotic dynamics, are then amplified by positive feedback mechanisms. This results in structures that emerge spontaneously which often are presented as examples of self-organization. In a similar perspective, Cavagna et al. proposed a method on distinguishing and evaluating self-organization paradigm from centralized paradigm—i.e. the display of collective behaviors (which emerges due to self-organization)—is not the collective behavior of order that we observed of swarming system, instead it is the collective response of the system. It is a measure of how well a system reacts to its environment—i.e. the system's response to perturbations. If the system reacts and adapt to the environment changes, that would provide a mean to suggest the existence of a self-organizing feature of the system. Both Prigogine et al. and Cavagna et al. think about self-organization as the ability to adapt (structures that emerge spontaneously) to a changing environment (external perturbations).

This collective response that Cavagna et. al. mentioned is the flexibility of the system, which is an important feature of social insects [38]. From the literature, flexibility also have various meaning. Firstly, some define flexibility as the ability for the system to perform as well in different environment or scenario by maintaining the behaviors of the individuals of the system [55]. It is true that flexibility is the system's performance in various environment by maintaining the behaviors of the individuals of the system, but we disagree with the view that the system should perform "equally well" in all environments. It is debatable as swarming systems are highly sensitive, some described it to be on the edge of chaos, they will perform differently when exposed to different environment. Given a different environment, they may be able to perform relatively as well as compared to other solutions, but will not be able to perform as well in an absolute sense. In order to perform "equally well" in different environments, the system

needs to be tuned—i.e. it possesses adaptability.

Secondly, the flexibility definition that we use in our work, is the ability of a system to adapt to a changing environment—i.e. how the system collectively respond to external perturbations [56, 52, 41, 38, 57, 27]. Fukada et al. [56] had a similar idea of flexibility definition however he evaluated it from the perspective of flexibility being a trait of robustness. General studies in swarming systems have paid lots of attention to features like scalability and robustness. However the key feature of flexibility has not gain as much attention. Reason being, it is not an easy feature to probe and measure.

The system's ability to be flexible or collectively respond to a changing environment requires effective information processes. How well the system adapt will depends on how well the system process these information. Some would describe a swarming system simply as an information processing unit—i.e. acquiring information to processing information to collective decision making and finally to acting on it. We can draw an analogy to a robot that acquires information through its sensors, process or compute these information and finally acting on it. This information process could be link to the idea of Exploration-Exploitation. In such a system, acquiring information can be seen as exploration which processing and acting on these information can be seen as exploitation. When the environment is static, the system can settle in an ideal exploration-exploitation balance that acquires enough new information and acting them such that they are efficient. It becomes interesting when the environment changes, similar to kicking the rope when an acrobat is on it. We can then observe how the system (the acrobat) regain its stability, a new exploration-exploitation balance since the environment changed. The system's flexibility—i.e. self-organization that gave rise to swarm intelligence—can be said as the ability to restore an exploration-exploitation balance suited to the current environmental conditions.

5.1.3 Our research question

Now that we understand the relationship between SI and flexibility, it is possible to access SI of a system via flexibility. In order for such an assessment, a benchmark framework will be required. This allows for various algorithms—e.g. swarm-inspired algorithms, centralized algorithm, etc.—to be assessed and compared among each other in various environments, via a set of metrics. In order to do this, the following research questions need to be answered.

1. What is a benchmark framework that can probe and evaluate the flexibility of the system?
2. In the context of the chosen benchmark problem, what is a set of metrics that we can use for assessment and comparison?

5.1.4 Our contributions

In this work, with the understanding that flexibility is an outcome of collective behavior that can be an avenue to evaluate SI, we aim to provide a benchmark framework

that specifically probe and measure the flexibility of the systems such that it will be possible for SI quantification and comparison among systems with a set of metrics. The proposed benchmark framework will allow users to further understand the effectiveness of the various strategies (SI or not) in the various environments, even reveal the need for SI or a centralized strategy can suffice.

Through our simulations, we have learn about the varying environmental difficulty levels, the effect of workload distribution and the effect of dynamic environment have on the system's performance. We have also evaluated the systems when it is subjected to the environmental change at different times. The early dynamics of the systems are compared to further explain some of the trends that we have observed. These results will be further explained in section 5.6.1.

5.2 Existing Benchmark Problems

The idea of benchmark problem has been around in many area of studies. It brings about a form of standadization of performance assessment which is often useful for a deeper understanding of the study and as a reference for future development. One such area of study is in optimization. In [317], Yao X. and Liu Y. consolidated 23 functions, claiming to have the largest amount at that time, as benchmark functions to evaluate the performance of fast evolutionary programming and classical evolutionary programming. After which, work like [318, 319] continued to expand the list of benchmark functions to evaluate differential evolution, particle swarm optimization and other evolutionary algorithms. In the area of global optimization, [320] also consolidated an extensive list of 73 benchmark functions.

In Artificial Intelligence study, specifically in reinforcement learning (RL), toy problems—i.e. benchmark problems—are being used extensively. Similar to the latter example, there are numerous benchmark problems—i.e. environments. The reason for such a long list of problems, is that there will not be one problem that can test "everything"—i.e. each benchmark problem is designed specifically for testing a specific part of the algorithm or system. In the case of RL, these problems exist in the form of environments. Gym by OpenAI [321] is one such toolkit for developing and comparing RL algorithms. It is a collection of test problems that users can use to work out their RL algorithms. In Gym, it attempts to solve two problems that is said to slow down RL research. Firstly, the existing open-source collections of RL environments do not have enough variety. Secondly, the lack of standardization of environments used in publications leads to difficult reproduction of RL results—i.e. a subtle difference in the problem definition, such as reward function or the set of actions will drastically alter the task's difficulty.

Unfortunately in SI there exist no such tools or platform for the development of SI. We recognize the importance of such benchmark problems and aspire to bring such useful tools into SI study. Since it is difficult to design artificial swarming system due to the fact that the collective behaviors [38] are not entirely predictable as these are the emergence of frequent interactions among the agents, a benchmark framework with a deeper understanding of SI can give artificial swarming system a head-start in the

design process. With our proposed benchmark framework, it is possible to understand the environmental parameters that can affect the performance of SI algorithms. This will allow users to better tune the system to operate effectively for an intended environment. The following section 5.3 will dwell in-depth into the design of our proposed benchmark problem that can probe the SI of a system.

5.3 Design Methodology of SI Benchmark Framework

The instances when natural swarming systems solve dynamic problems—e.g. fish and birds evading predators, ants and bees foraging (with changes in the environment)—SI can be observed. When the environment becomes static like when ants and bees found their food (that last for a long time), the system reaches some sort of stable state, and their collective actions experience little evolution. These are the times when we have realized that the swarming capabilities and its intelligence are not necessary—i.e. not being utilized to their fullest potential. To some extent, most (static) problems could not probe the actual flexibility of the swarming system.

This suggests that the level of challenge—e.g. static vs. dynamic—of a given problem can help probe different levels of SI. If a problem is deemed not challenging “enough”, the system would not exhibit swarming behaviors (intelligence), if any. Therefore, these problems could potentially be solved with non-SI strategies. However, more work is needed to understand the effectiveness of deploying SI to solve different problems, as exposing the system to the most challenging environments does not guarantee a maximum display of SI. In this work, we attempt to bridge this knowledge gap, to understand the effectiveness of the application of SI in various environments with varying difficulty levels.

Our aim for this SI benchmark framework is to support the SI research community, in terms of allowing for SI of a system to be observed and measured, and be compared to other systems’ SI. The proposed benchmark framework will be a “invaluable tool(s) to measure the progress in a quantitative way and can be used to challenge the researchers on tasks that grow in complexity” [37]. Since the SI level is intrinsically dependent on its environment, different environments can yield different levels of SI. The benchmark framework should have different environments challenging “enough” to facilitate probing a vast range of strategies.

In this benchmark framework, we have chosen a problem of collective mapping. The objective, for the system, is to search and detect obstacles in the domain. This problem is analogous to ants foraging for food, however with a few modifications. In this problem, the food source(s) can be static or dynamic. When the food source(s) is dynamic, it will disappear and re-appear instantaneously at another location without any overlap. This is similar to food source(s) washed away by rain or carried away by other insects or animals. In our proposed problem, the agents discover and collect the food, but are not required to return to a nest—i.e. the agents have an unlimited carrying capacity.

Considering the varying property of SI with its environment, the benchmark framework will be limited if the candidate system can only be subjected to one environment. Therefore, we have proposed to design an environment that allows for an adjustable "difficulty" level to cater for a fair assessment of SI. In this design we have considered the tuning of the "difficulty" level via the dynamic of change and the workload distribution. Even in the same environment (problem), the dynamic of change can be very different—i.e. the changes can occur earlier in the beginning or later towards the end of the operation. Will the system still be as intelligent (swarm) if the dynamic of the change differs? In the same way, the distribution of workload can be very different, from concentrated workload to dispersed workload. In these different settings, it is not necessary for the same algorithm to perform well in all of them. Given that both algorithm *A* and *B* exude SI through a problem, in a different setting of the problem, algorithm *A* may perform better than algorithm *B* in one setting, while in another setting algorithm *B* may perform better.

Therefore in our proposed SI benchmark framework, it consists of one problem (environment) and various tasks (different settings of the environment) for the system SI to be exposed and evaluated in these different settings. As discussed in Section 5.1.2, flexibility (or collective response) will indicate the measure of SI that our work will focus on. Flexibility of a system—i.e. part of adaptivity—can be observed by its exploration and exploitation actions. This important concept of exploration-exploitation balance of the system is important when observing an adaptive process, and it will aid as a supplement to the evaluation of the benchmark results—i.e. performance and flexibility of the system. As the systems operates in a changing environment (and distribution of workload)—i.e. an unknown environment, it readjust its exploratory and exploitative actions to adapt to the unknown.

This proposed SI benchmark problem, in the form of a simulation, is designed for Multi-agent system (MAS), Multi-robot system (MRS), non-cooperative and centralized system algorithms to be evaluated and thus the possibility of comparison. The benchmark problem will be designed to be modular with 1) environment, 2) agent, 3) strategy (algorithm), and 4) communication network modules that can be easily modified. The modular design provides users the flexibility to simulate different types of agents in different environments using various strategies, even non-SI algorithms. In the following subsections, we will discuss each module of our SI benchmark framework..

5.3.1 Environment

In order to design an effective benchmark problem to probe the flexibility of the systems, we need an answer to the question of "What is a difficult enough problem?" or "What is a range of difficulty of the problem?". As such, we have considered the design of an environment with various levels of temporal and spatial severity which will attribute to the difficulty of the problem. Temporal severity described in [322] as Type I, II, III is a method of classifying the difficulty of an optimization problem. In general severity increases from static to slow changing to fast changing, and finally instantaneous changing environment being the most difficult problem. If the changes

are drastic—i.e. instantaneous—in a dynamic environment, then any prior knowledge will not be beneficial to the future operation as the new objectives do not depend on the previous ones. This will make the problem extremely challenging to show the benefit of swarm cooperation and the flexibility (or adaptability) to a drastic problem.

There is another perspective about the temporal severity, that we have proposed, that is the “time” at which the change (instantaneous) is introduced. A flexible (or adaptable) system that explores and exploits will be affected by how the changes are implemented—i.e. slow changes to instantaneous changes—and when the changes are introduced—i.e. towards the beginning, middle or end of the simulation. When the changes are introduced at the beginning of the operation, the system is largely in its exploration phase which means it is more ready to react to a change. However, if the changes are introduced later in the operation, it will require more time for the system to exit its current exploitative state to return to a more exploratory state in order to search for the changes.

Spatial consideration of the workload within the environment is also important. As compared to the environment, the workload should be substantial and distributed, to increase the difficulty of the problem. When the workload is distributed, it requires the system to distribute itself to find and handle the workloads such that it can be efficient. However, if the workload is distributed too much—i.e. the individual workload pieces are too small, an approach like random walk could outperform a swarm algorithm. As a pure exploration approach could find the small workload pieces randomly and since the workload pieces are small, exploitation is not necessary. On the other hand, when the workload is large, less exploration is needed since the system can quickly identify the workload, while exploitation is more important to handle the workload. Therefore, it is important to find a balance of the workload’s amount and distribution such that the system has to balance their exploration and exploitation strategy to exude SI.

With the temporal and spatial environmental considerations, we choose to create a benchmark problem environment with variable i) distribution of workload and ii) time of instantaneous change of workload. We have also chosen to use occupancy grid map (OGM) as the representation of our environment. Each cell in the OGM holds the belief of the agent(s) whether it is a clear, occupied or unknown cell. The stochastic belief system allows for more accurate and reinforced truth of the environment. This can also account and reduce the effect of noise in the environment, the robot sensors, odometry etc.

OGM is widely used especially in the robotic systems where they perform mapping operations [48, 27]. Many algorithms and strategies have been based on operating using OGM, for example obstacle avoidance and shortest path (i.e. A-star search) algorithms. This allows for a future wide adaptation of the applied algorithms to physical platforms if they are found to be suitable for particular environment (or problem). Existing algorithms that are operating in problems that are represented in OGM can also be tested on our benchmark problem simulation. In our benchmark problem environment, there exists pillars of occupied cells that are the “workload” for the system to discover during the collective mapping operation, as seen in fig. 5.1. At a given iteration, a change will

be injected, and the pillars will change to their new locations. The system will then attempt to discover the new pillars.

Another benefit of using OGM is that the maps are represented in pixels, commonly represented by value [0-1] which corresponds to various shades of grey from white (clear cells) to black (occupied cells). These grey scale maps are images that are readily suited for image processing and machine learning to improve system's capability.

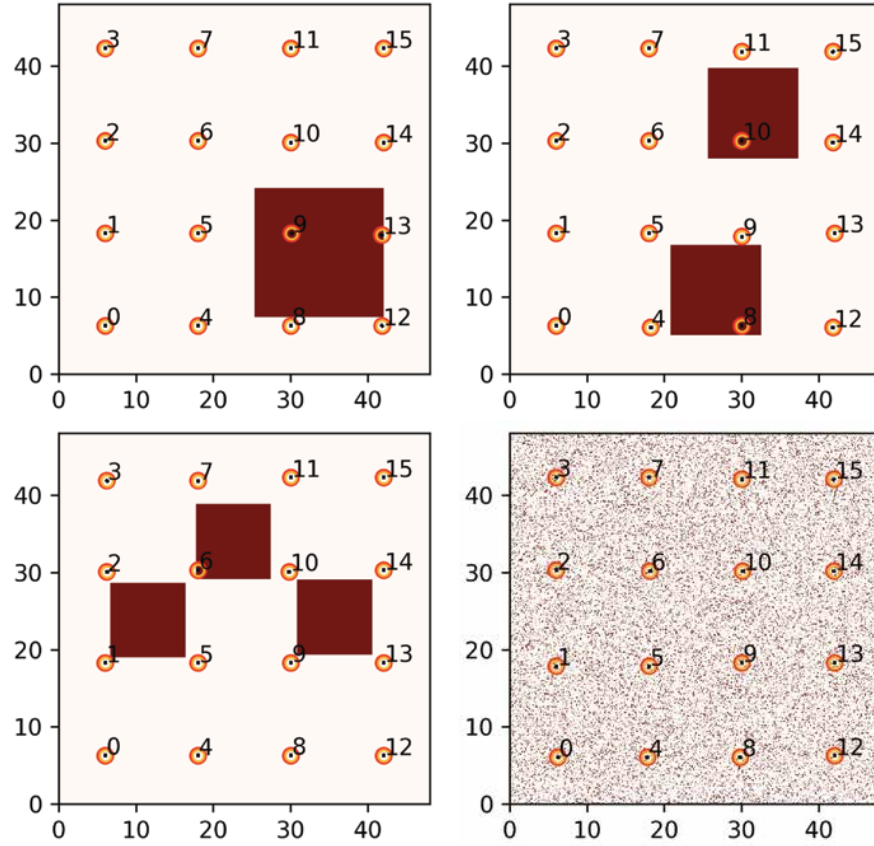


FIGURE 5.1: Simulation environment true maps: 1, 2, 3 pillars and scattered workload distribution

In order to focus on the evaluation of the flexibility of the system, we have made the following assumptions and simplifications to the problem. In this benchmark problem simulation, we have assumed that there are no noises from the system's perception of the environment, and its perception of itself within the environment (odometry). This will allow for a perfect mapping of the environment, eliminating the probabilistic nature (belief) of OGM. In this case, the OGM cells will only hold the value of 0, 0.5 or 1.0 which represents a cleared cell, an unknown cell, and an occupied cell respectively. In order to allow the system to focus on finding the occupied cells, not affected by an initial unknown environment, the agents will start off with a "cleared" environment instead of an "unknown" environment. This is to simulate that the system has already existed in this location for a while and have known this environment (as "cleared" prior

to this. This is to simulate an on-going process of monitoring of an environment, in attempt to detect the dynamic changes occurring in the environment. Therefore the OGM cells will only hold a value of 0 or 1.0.

We further simplify the environment to contain no obstacles, removing a layer of navigation algorithm that can create a sense of pseudo-exploration. As we study the flexibility of the system, we want to keep the agents' motion as close to its intention (way-point generated by SI algorithms) as possible to reduce any complications. Therefore in our problem, the OGM cells with value 1.0 means the existence of "food" instead of "obstacles". We assume that the agents can traverse through the occupied cells and pick up the "food". Although our problem draws similarity to dynamic food sources of ants, the agents do not have a nest and will not be required to deposit the "food" at their nest—i.e. the agents can carry an unlimited amount of "food". This allows for the agents to focus on discovering new dynamic cells.

TABLE 5.1: Benchmark Simulation Parameters

Environmental parameters	
Parameter	Quantity(units)
1. Domain x	48m
2. Domain y	48m
3. Resolution	0.1
4. Total # of occupied cells	27648 (12% of total cells)
5. # of occupied pillars	1, 2, 3, <i>scattered</i>
6. Pillars location	random but do not overlap
7. Change occurs at iteration	<i>static</i> , 100(<i>early</i>), 600(<i>mid</i>), 1000(<i>late</i>)
Agent parameters	
1. Field of view	360°
2. Speed	0.1m/s
3. Sensor range	0.5m
Strategy parameters	
1. # of agents	16
2. Repulsion (agent-agent) range	8m
3. Repulsion ratio	$rep_{explore}/rep_{exploit} = 2.4$
4. Strategy	KNN-R, ABC, PSO, PSO adaptive repulsion
5. Communication network	e.g. k-nearest neighbour

In Table 5.1, the environmental parameters of the benchmark problem are listed. The domain size and resolution are often chosen with a trade-off between desired quality—i.e. granularity—of the results and the simulation run-time. The domain size can limit the number of agents deployed, and many would proposed that only with a high number of agents that the system can be considered as scalable and as a swarm. However, the scalability of the swarm should not be determine by the mere number of agents, instead the density of the agents within an environment should be considered. For example, a system of 10 agents will not be able to scale as much in a smaller environment as in a larger environment. Without mentioning the size of the environment, it will not be

conclusive if the system of 10 agents can scale further or not. It is important to also note that choosing the resolution or granularity of the environment is crucial. This affects the balance between the realism of the simulation in terms of agents movement—e.g. speed and direction (limited to the granularity of the environment)—and the simulation run-time.

In the design of the "food source(s)"—i.e. occupied cells, we have also attempted various formations of the occupied cells in the form of vertical gates, horizontal gates, and finally square pillars. After investigation, we have settled with $x = y$ pillars as it provide the fairest evaluation of the strategies, not allowing any x -bias or y -bias strategy to gain an advantage during exploration and exploitation. In order for a system to meet the objective of our benchmark problem, there is a need for the system to perform both exploration and exploitation. The balance of this collective (or not) exploration and exploitation will determine the performance of the system. The temporal characteristic of the dynamic features will encourage the need for 'better' exploration strategy to rapidly detect new changes in the environment. While the spatial characteristic of the dynamic features will encourage the need for both exploration and exploitation to rapidly detect as much of these features. In Dorigo et. al. work [36], it is mentioned that future swarms(robotic) will exist and operate at many different scales, of which includes both temporal scale—i.e. from swarms that display quick adaptation in a rapidly evolving environment to robots that operate on months-long missions(quasi-static)—and spatial scale of operation domain size.

5.3.2 Agents

There are two perspective that we can take on the design of the agents, either a micro or a macro perspective. The difference lies in the perspective of agents as multiple individual agent or a system of agents. However, when we study swarming systems, we need to take a macro view. In an example of a bee's manoeuvre (flight) back to its hive, the micro perspective will suggest that the flight is complex, while the macro perspective suggests the flight as a simple agent movement [323]. Therefore in our agent design, there are only three parameters considered as shown in Table 5.1. However, we still aim for an agent design that is versatile for both MAS and MRS algorithm and also suitable for possible physical applications beyond simulation.

In order to design versatile agents, we have considered three tunable parameters like speed, peripheral angle (field of vision) and perception range. This can facilitate transition of algorithm from a simulation to a MRS in future development. Specific to our benchmark problem, we have set the agents to move $0.1m$ or 1 cell per iteration, and each agent possess an omni-directional peripheral of $0.5m$ range. A short perception range is chosen to mimic swarming in nature where each agent has limited field of vision—e.g birds and fishes, where collectively they can see "more". This can also translate to cheaper sensors for MRS and making use of SI to improve its performance. It is important to note that the speed and the accuracy of the direction of travel of the agents are limited to the resolution or granularity of the environment.

5.3.3 Communication network

The communication network is established to facilitate possible information propagation, if any, in a centralized or decentralized manner. In our implementation, the communication network will receive all the relevant information at every iteration, by default all the agents can get access to these information. In a centralized approach, the "central control station"—i.e. the computer running the simulation script—will communicate with all the agents throughout the main simulation script. However in a decentralized approach, there is no "central control station", instead the agents are supposed to communicate directly among each other. In order to create a pseudo limited network, all the agents can get an access to the information of their neighborhood via the communication network—e.g. an agent *A* will decide which other agents (*B* and *C*) get access to its information, agent *B* and *C* will tap into the communication network and only access information from agent *A*. This implementation limits the communication to simulate decentralized communication. When the agents are communicating in an all-to-all manner (decentralized), this can have the same effect as a centralized communication approach whereby all agents have access to all the information via the communication network.

5.4 Strategies-Swarming Algorithms

5.4.1 Artificial Bee Colony

Artificial Bee Colony algorithm was first proposed by Karaboğa D. in 2005 [324], describing a minimal model of bee forage selection that consists of three essential components: 1) food sources, 2) employed foragers, and 3) unemployed foragers. The food sources are distinguished by various factors such as distance to the nest, concentration of food and ease of extraction—e.g. presence of predators. However in the model, instead of capturing these factors with various variables, a single variable of "profitability" of food sources can be used as a representation. Associated with these food sources are the employed foragers, where they are exploiting or described as being employed at. These employed foragers will share information about the food sources to the unemployed bees. As the employed foragers deem their associated food sources as depleted, they will become unemployed as scouts, exploring the domain in search for new food sources so as to be employed again. While employed foragers share information about the food sources, a group of unemployed onlookers are waiting to compare and choose the most profitable food source. This exchange of information, in the case of bees are in the form of dance, brings about the collective knowledge and the emergence of collective foraging behavior. The general algorithm is as follows in Algorithm 2.

In Karaboğa D. approach, the first half of the colony consists of employed artificial bees and the second half of the colony consists of unemployed onlooker artificial bees. For every food sources, there is only one employed bee. When the food source is exhausted, the employed bees will become unemployed scout. Using [325] allow us to further dive

Algorithm 2: Karaboğa D. ABC algorithm

```

Send the scouts onto the initial food sources
while requirements are met do
    Send the employed bees onto the food sources and determine their nectar
    amounts
    Calculate the probability value of the sources with which they are preferred by
    the onlooker bees
    Send the onlooker bees onto the food sources and determine their nectar amounts
    Stop the exploitation process of the sources exhausted by the bees
    Send the scouts into the search area for discovering new food sources, randomly
    Memorize the best food source found so far
end while

```

into the details of the deployment of this strategy. In each cycle, the employed bees will continue to exploit their food sources as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}), \quad (5.1)$$

where x_k is a randomly selected candidate solution ($i \neq k$), j is a random dimension index, and $\phi_{i,j}$ is a random number within $[-1, 1]$. After computing $v_{i,j}$, a greedy selection is used to reject or accept the new solution. If the fitness value of $v_{i,j}$ is better than the previous solution $x_{i,j}$, then update $x_{i,j}$ with $v_{i,j}$; otherwise reject new solution $v_{i,j}$. After the search process, the employed bees share information of their food sources with the onlooker bees through the wiggle dance. Based on these information, the onlooker bees will be placed on the food sources based on a probability based selection process described as follows:

$$p_i = \frac{fit_i}{\sum_{j=1}^N fit_j}, \quad (5.2)$$

where fit_i is the fitness value of the i th solution in the swarm. The higher fitness value represents a high quality food source which will have a higher probability of being selected. If a solution cannot be improved over a predefined time, then the food source will be abandoned and the employed bee associated will become an unemployed scout bee. The scout bee will attempt to find a new food source to replace the abandoned as follows:

$$x_{i,j} = lb_j + rand(0, 1) \cdot (ub_j - lb_j), \quad (5.3)$$

where $rand(0, 1)$ is a random number between $[0, 1]$ and lb and ub are lower and upper bound of the j th dimension.

In our application of ABC, we have made the following adjustments to the ABC algorithm to accommodate the limitations of our benchmark environment. The initial

state of the bee agents are split half employed and half unemployed (onlookers), this is due to the unknown environment, and we do not assign one employed to one food source. Since the total amount of food take up 12% of the environment, each agent sensor area is only 0.035%, with a total simulation run-time of 1200 iterations, it will not be sufficient and effective for one employed bee to tend to one food source 342.857 times its sensing area. Therefore in our simulation, we did not limit the number of employed bees to one food source to encourage an increase in the number of onlookers drawn to the area of interest, assisting exploitation work. In order to further increase the chance of onlookers being selected to assist the employed bees, we have kept the probability p_i , from eq. 5.2, to 0.5. In our environment, due to the simulation limitation, the cells only take value 0 (clear) or 1 (occupied), their fitness value can only be 0 or 0.5. During the probability based selection process, the employed bees with fitness value of 0 will have a close to 0 chance of having an onlooker assistance. While the employed bees with fitness value of 0.5 have an equal chance of having an onlooker assistance, the more employed bees with 0.5 fitness value—ie. the same food quality—, the lesser chance that the onlookers will assist, eventually each employed bees will have a $1/employed_bees$ probability of being pick by the onlookers. Therefore, to increase the probability of exploitation, we assign p_i to be 0.5.

In ABC algorithm, the employed bees will abandon their role when the food source is depleted, changing to a scout role. In our case, when the food source(s) changes or when employed bees wondered off the food source—i.e. agents not detecting dynamic features, then these employed bees will be unemployed to become scout bees after a certain number of iteration instead. This helps to reset the system and allowing it to gain exploration. The scout bees will perform a random walk instead of the method stated in eq. 5.3. Although both resort to exploring the environment randomly, random walk focuses on short range exploration while eq. 5.3 could encourage both short and long range exploration. Since half of our agent population have the chance to become scout bees, a short range exploration may be sufficient for discovering new food sources. In addition, there are cases where employed bees wandered off from the food source and becomes unemployed scout bees, the short range exploration can help these scout bees to recover and re-engaged the food source quickly.

5.4.2 Particle Swarm Optimization

In 1995, Kennedy et. al. intended to use Particle Swarm Optimization (PSO) as an algorithm to simulate social behaviors [326]—i.e. movement—of agents in a flock of birds or a school of fish. It was observed that this algorithm to be performing optimization, searching for global optimal solutions. PSO is a metaheuristic that makes less or few assumptions about the optimization problem which allows it to be applied to wide range of optimization problem. However, PSO does not guarantee that an optimal solution can be ever found. In general, PSO agents make use of its current velocity vector, personal best (in terms of fitness value) location and global best location to compute its next velocity vector that guide its waypoint. It is described in eq.[4, 5] of [42] by [326, 327]. As PSO evolves to become commonly applied in the field of optimization, the expression of the agents' velocity vector equation does have slight differences.

In which, the following expression gave users or operators the ability to fine tune the weight of every term in v_i^{t+1} .

$$v_i^{t+1} = \omega \cdot v_i^t + \phi_{pb} \cdot rand(0, 1) \cdot (p_i^t - x_i^t) + \phi_{gb} \cdot rand(0, 1) \cdot (p_g^t - x_i^t) \quad (5.4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5.5)$$

Each agent will have a x_i that is a position in the domain while the velocity v_i vector. p_i being the current best personal solution and p_g being the best global solution of the swarm. The various terms in eq. 5.4 describe the current velocity vector, the vector towards the personal best and the vector towards the global best, with ω , ϕ_{pb} , ϕ_{gb} as the weights to tune the system. ω describes the inertia of the agent in its current direction of travel, the higher the value of ω , the more inertia the agent possessed to be steered away with the other two terms. ϕ_{pb} and ϕ_{gb} would determine the influence of personal best and global best on the agent's new velocity vector. Notice that randomness $rand(0, 1)$ is also applied to these two terms which is important, as mentioned in [38], that "SO relies on the amplification of fluctuations" as randomness enables the discovery of new solutions. Finally, the agents will compute its next location as in eq. 5.5.

5.4.3 K-Nearest Neighbor with repulsion (KNN-R)

As the name suggest, this strategy primarily works with local neighborhoods of agents to gather, share and act upon the environmental information. The local neighborhoods of agents are determined by the use of metric distance to include or exclude agents from the neighborhoods. Within the local neighborhoods, information is shared via the k-nearest neighbor directed network, where agent A can be a neighbor of agent B but vice versa is not true. During each iteration, the neighborhoods can alter depending on the current positions of the agents, suggesting that the agents' local knowledge of the environment can vary with different groups of neighbors.

This seemingly "limited" information does yield benefits as the systems' clustering emerges from the various k-value behaviors, which can indirectly affect the exploration and exploitation balance of the system. The more information available to the agents, the more clustering—i.e. bringing the agents closer and into fewer clusters, the system will tend to be "tied-down" to exploit the available information. Although it can seem to benefit the system in terms of finding more local information, the system will not be available in other parts of the environment to find new information—i.e. causing lesser or limited exploration. Therefore when the environment changes, the system might take a longer time to adapt to the change.

In an extreme case like $k = 0$ (0-nearest neighbor), the agents do not share any information with each other—i.e. they operate independently to acquire and act on the information gathered by themselves. While $k = N - 1$ (N is the number of agent), the system is constantly engaging in an all-to-all communication. The all-to-all information sharing can be similar to that of a centralized information sharing where all the agents

will get access to all the information. However the difference lies in the method of disseminating information via a centralized node or the individual agents broadcasting to each other.

In an initially unknown environment, when the agents started in an "exploration" mode, the deployment of a random walk make sense as it only require "luck" to find the food sources. In [328], various random walks were evaluated as exploration strategies and Levy walk yields the best performance. However due to the limitation of the simulation environment, where the agents can only take one step (cell) every iteration, we have chosen to use a generic random walk instead. At every iteration, agents will send its sensor data to their neighbors and update their individual local map with sensor data of their own and their own set of neighbors, according to the eq. 5.6. In the equation, the adjacency matrix $a_{i,j}$ will determine the sets of sensed data that will be relevant for the individual agent based in k nearest neighbors and itself.

$$S_t^i = \bigcup_{t'=1}^t \{s_j(t'; j | a_{i,j}(t') = 1)\}, \quad (5.6)$$

with $s_j(t')$ the sensed data of agent j at instant t' , $a_{ii}(t) = 1, \forall t$, and $a_{ij}(t) = 1$ if unit i is connected with agent j at instant t' , and $a_{ij}(t) = 0$ otherwise. It is worth stressing that this approach was designed to be compatible with temporal interaction networks, which have recently been found to be necessary to achieve effective collective responses by MRS [75].

If there is no food source—i.e. occupied cells—detected in both its own and its neighbors' sensed data, the agents will continue in "exploration" mode performing random walk. If there are new occupied cells on the updated individual local map, the agent will choose the nearest food source and make it its waypoint—i.e. the agent will move towards it until it is within its sensor radius before registering a new waypoint. After a given number of iterations, if agents are still not able to find new occupied cells, they will exit "exploitation" mode and return to "exploration" mode increasing the chance of finding new occupied cells by increasing agents' repulsion radius. The described process is presented in the following algorithm 3.

In the *dir_travel()* function, it will take in the agent's and its neighbors' sensed data to evaluate the presence of cells of interest—e.g. dynamic cell. If there is cells of interest detected, the function will compute a vector considering a set Xm repulsion from its neighbors, repulsion from the domain edges and attraction to the nearest cell of interest detected. This vector will be processed and *dir_travel()* will output the next waypoint for the agent. In cases where cells of interest is not detected by itself or its neighbors, then the function will compute the next waypoint based on a random walk algorithm.

Algorithm 3: K-nearest neighbor

```

1:  $num\_of\_agent \leftarrow N$ 
2:  $k\_nearest\_neigh \leftarrow k$ 
3: Initialize Environment
4: Initialize Agents  $a_i \forall i \in [1, 2, \dots, N]$ 
5: for  $iteration < 1250$  do
6:   for  $i \leftarrow 1, 2, \dots, N$  do
7:      $neighs \leftarrow find\_neighbours(a_i, [a_1, a_2, \dots, a_N], k)$ 
8:     if  $POI$  in  $a_i.sensor\_data$  then
9:        $send\_sensor\_data(a_i.sensor\_data, neighs)$ 
10:    end if
11:     $neighs\_sensor\_data \leftarrow receive\_sensor\_data(neighs)$ 
12:     $Update\_localmap(a_i.sensor\_data, neighs\_sensor\_data)$ 
13:     $next\_waypoint = dir\_travel(a_i.sensor\_data, neighs\_sensor\_data)$ 
14:     $a_i.move(next\_waypoint)$ 
15:   end for
16: end for

```

5.5 Design of Metrics

In this benchmark framework, we have proposed two metrics to evaluate the overall operational performance and the flexibility of the system, namely the Map Similarity Index (MSI) and the Swarm Intelligence Index (SII) respectively.

The Map Similarity Index (MSI) is obtained from the comparison between the collective map and the true map. The collective map represents the system knowledge of the environment and the true map represents the ground truth with "clear" cells and "occupied" food cells. Since the objective of the system is to search only for the "occupied" cells, the MSI only focuses on comparing area(s) where it is "occupied" in the true map.

The SII is computed from the MSI. The SII is obtained by performing an integration from the last 250 iterations of simulation's MSI. The last 250 iterations are chosen to represent a settled system and that a collective response has taken effect after the change. The values are normalized to the maximum SII.

Through the results discussion, exploration and exploitation concept will be used to further understand and give value to the results.

5.6 Benchmark Framework: Collective Mapping Simulation

The benchmark framework provides a simulated experimental environment for users to evaluate and examine the swarm intelligence of the systems. The modular implementation allows for variation of environment, agents, communication networks and strategies.

In order to understand the challenges that the environment can pose to the system of agents, we will examine two variables of the benchmark environment—i.e. i) workload distribution and ii) dynamics of the environment. Various swarming algorithms will be employed to observe their performances and their response in these environments.

5.6.1 Results and Discussion

The analysis of swarming systems' emergence behavior tend to be challenging, often the results are not repeatable as the systems' actions are not deterministic. In [13], Beni et. al. mentioned one of the characteristics of SI is its unpredictability, which is due to swarming systems' self-organization(emergence) behavior. Some described SI as systems that are on the edge of chaos, where the systems' performance are highly sensitive to any change in environment or agent. The factors that affect the vast difference in the systems' performance are initial conditions of the agents and environment, distribution of workload, number of agents, iteration when dynamic change is introduced, etc. This sensitivity is reflected in the results as standard deviation, describing the spread of the results.

In order to reduce the standard deviation of the results, we employed a method to select 5 simulation runs for each case before evaluating the systems' performance. The mean is plotted in each graph. In the future, more cases can be run to further improve the accuracy of the results.

Effect of workload distribution on the flexibility of the system

In this section, the effect of workload distribution will be examined. In order to isolate this effect, and not be affected by other factors like dynamic environment, the systems will be subjected to a static environment with a range of workload distribution—i.e. 1, 2 and 3 pillars, and scattered workload. The other simulation parameters are kept constant as shown in Figure 5.2.

TABLE 5.2: Simulation parameters: Effect of workload distribution

Parameter	Quantity (units)
1. Total # of iteration	1250
2. Total # of agents	16
3. Agent-agent repulsion radius	8 m
4. Agent sensor radius	0.5 m
5. Total # of pillars	1, 2, 3, <i>scattered</i>
6. Change occurs at iteration	<i>static</i>

Fig. 5.2 shows a comparison of all the performances of the systems operating in a scattered, 3 pillars, 2 pillars and 1 pillar environment, from a vastly dispersed to a concentrated workload. The intention of this simulation is to understand the effect of workload distribution has on the performance of the various systems. In each of these

environment, strategy KNN-R ($k = 0$) is chosen to be the baseline for comparison as it is a fully independent system where agents are independently performing both exploration and exploitation. In each of the cases, although the systems are all exposed to the same number of dynamic cells (12% of the area of the environment), the distribution of these workload (dynamic cells) are different, it is notable that different strategies excelled under different environment.

This is an indication that different distribution of workload can pose level of challenge to the various systems, requiring different level of SI to excel. Also, although the environment is static, SI strategies has shown to be effective in the different static environment providing higher Map Similarity Index (MSI). Self-organization of these agents is still effective in allowing for the discovery of the 'food' cells when the systems lack any prior environmental knowledge—i.e. unknown environment.

In fig. 5.2, it is observed that the generic PSO employed scores the lowest MSI as compared to all the other strategies. As the generic PSO was designed to locate local or global, minimum or maximum point—i.e. the gathering of agents in a single location—within a search space, and is not suitable for a problem with an objective(s) that spans an area. The initial phase of PSO before stagnation indicates the convergence of the agents towards a point of interest and hovering around it. As the system enters into a steady-state, it loses its exploration effort and remains stagnant as its exploitative actions does not cover an area.

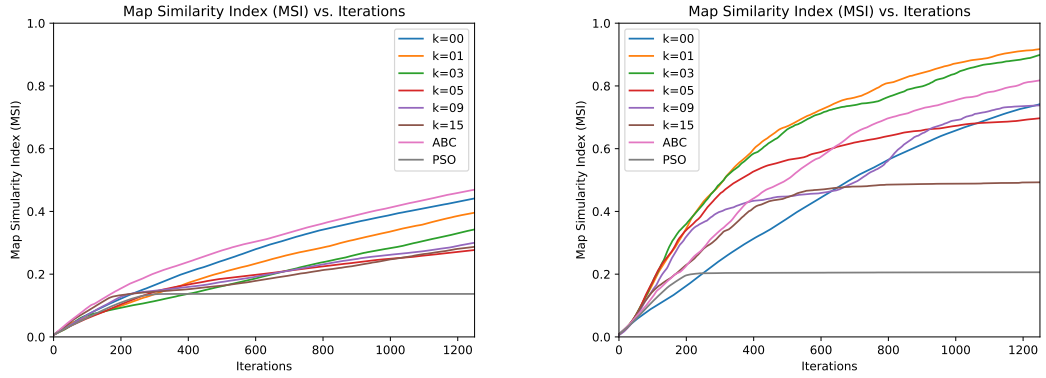
The result from the scattered environment shows that strategy ABC and $k = 0$ performing the best and strategy $k = 5$, $k = 9$ and $k = 15$ performing the worst. In such an environment, where the objectives are scattered randomly, a high exploratory strategy can perform better. Comparing top performing ABC and $k = 0$ strategy, ABC has got an advantage of a cooperative exploration over $k = 0$ non-cooperative—i.e. independent agents performing random walk. In such a randomly scattered objectives, exploitation is not favorable because the information gathered from the environment will not help the agents in searching for other cells—i.e. the location of a cell does not help the agents in locating the consolidated cells (continuous area of objectives). Therefore it is observed that strategies with a hint of exploitation does not perform well in this case. Although visually this environment might seem challenging, it is actually not complicated. It does not require any level of intelligence to perform fairly well in this environment as shown by strategy $k = 0$.

On the opposite end of the workload distribution spectrum, in the 1 pillar case, exploitation is favored. Once the pillar is detected, the agents can exploit the information to gather around the area since there is only 1 pillar of workload. Therefore it can be observed that strategies with higher exploitation like $k = 5$, $k = 9$ and $k = 15$ performing well in this case. Since $k = 0$ does not perform cooperative exploitation, the system does not make use of the information to gather around the objective and causes it to perform the worst. In fig. 5.2(D), ignoring PSO generic, the benefit of cooperation can be clearly seen as compared to a fully independent $k = 0$ system. Now, the power of cooperation within a system can be observed and measured in this environment.

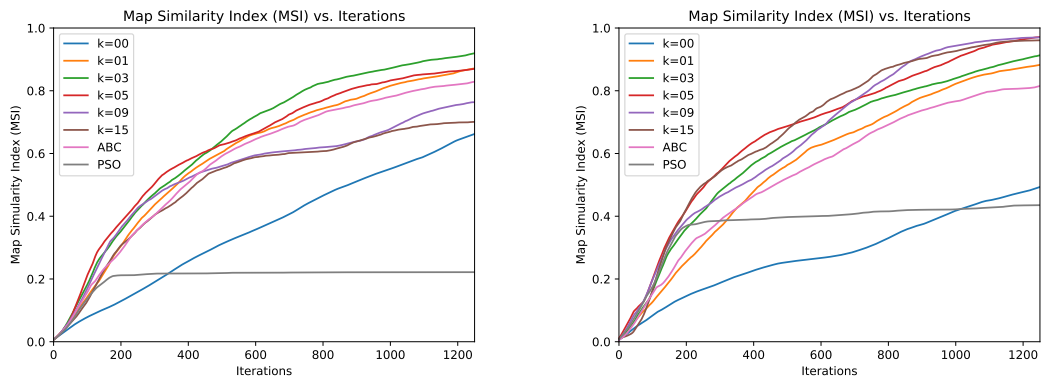
The case of 2 pillars and 3 pillars requires more from the system, that is to have both

exploration and exploitation, forcing the system to have a right balance of exploration-exploitation. This can be seen in both the cases result showing $k = 3$ having higher MSI. $k = 1$ showed the benefit of exploration in higher workload distribution, for example in 3 pillars as compared to 2 pillars environment. Although the other strategies (except for PSO) possess a blend of exploration and exploitation, their EE balance does not suit this environment as well.

It is also observed that, in fig. 5.2(B), after iteration 600, higher k systems start to score lower MSI than our $k = 0$ baseline which shows that there are environments where high level of swarm cooperation—i.e. exploitation—does not guarantee a better performance. In our KNN-R strategy, swarm cooperation can be seen as the exploitation of the system. Unique to our strategy, the exploration is purely random via random-walk algorithm whereas the cooperation resides within exploitation described by k value—i.e. the flow of information, the higher the flow of information, the higher system exploitation or swarm cooperation.



(A) Rep_rad=8m; scattered; sen_range=0.5m; static (B) Rep_rad=8m; 3 pillars; sen_range=0.5m; static



(C) Rep_rad=8m; 2 pillars; sen_range=0.5m; static (D) Rep_rad=8m; 1 pillar; sen_range=0.5m; static

FIGURE 5.2: Static environment; comparing 1p, 2p, 3p, scattered

Effect of dynamic100 environment on the flexibility of the system.

In this simulation, the systems are subjected to an environment with change occurring at iteration 100. This increases the difficulty of the problem drastically as the change introduced is instantaneous and does not overlap with the past pillars/cells—i.e. prior knowledge of the environment is deemed futile, which can be observed in the drastic drop of the Map Similarity Index (MSI) in fig. 5.4 (B, D, F, H) at iteration 100. Furthermore, the systems in the dynamic100 environment are exposed to two sets of occupied cells, one set before the change and the other set after the change—i.e. having to collectively map out twice the number of cells as compared to a static case. The maximum MSI of 1.0 represents the maximum similarity of one set of pillar(s) of "occupied" cells, as MSI is based on the similarity of areas with "occupied" cells. Therefore the MSI is the similarity index of the "occupied" pillar(s) before and after the environmental change—i.e. drop in MSI. In fig. 5.4, the results are presented in a static-dynamic100 pair (A-B, C-D, E-F, G-H). The simulation parameters are presented in table 5.3.

TABLE 5.3: Simulation parameters: Effect of dynamic100 environment

Parameter	Quantity(units)
1. Total # of iteration	1250
2. Total # of agents	16
3. Agent-agent repulsion radius	8m
4. Agent sensor radius	0.5m
5. Total # of pillars	1, 2, 3, <i>scattered</i>
6. Change occurs at iteration	100 <i>iteration</i>

The case of scattered objectives, in fig. 5.4(A-B), showed little difference in the systems' performance relative to each other, between the static and dynamic100 environment. Since the objectives are already randomly scattered—i.e. extremely unstructured environment—displaying no pattern, the agents will not be required to process any information regarding the pattern of the objectives. Therefore injecting an environment change will not drastically affect the system in a different way and the strategies with high exploratory actions are still favored in both cases.

From dynamic environment 1 pillar onward, it can be observed that these environments pose a challenge to the systems. In 1 pillar case, in fig. 5.4(C-D), comparing static to dynamic100, strategy $k = 9$ and $k = 15$ predominantly out-perform other strategies given their high exploitative actions which is advantageous for a 1 pillar objective. Nonetheless, for both cases, strategy with cooperation does provide an advantage over independent $k = 0$ system. However, the dynamic100 environment takes a toll on the systems' performances and it affects strategies with lower exploitative actions—i.e. ABC and $k = 1$, creating a larger spread of systems' performances.

In fig. 5.4(E-F), strategies $k = 3$ seems to have the right EE balance to operate well in this dynamic environment, giving the best performance. It is also observed that strategies $k = 1$ and ABC performance reduces drastically in the dynamic100 environment, similar to the case in dynamic100 1 pillar case. Although ABC is a highly

exploratory strategy as observed from its performance in the scattered case, the environmental change causes its performance drop drastically. We can deduce that there are other factors that affect the recovery of the strategy, despite its high exploration.

The inability for *ABC* to quickly recover in its performances has got to do with the different state the the system is subjected to—i.e. initial condition (locations and mode of agents) for the static case and the condition at iteration 100 for the dynamic case. Since in the static case, the initial condition for all the agents are that the agents are spread evenly across the domain and are in exploration mode—i.e. the agents are not preoccupied with exploitation work, these contributed to the performance of *ABC*. In the dynamic100 case, at iteration 100, the agents will not be evenly spread out and some may be preoccupied by exploitation work which can cause *ABC* exploration to reduce, thus the drastic performance drop of *ABC* in dynamic100 environment.

Fig. 5.3 compares the various strategies operating under 1p, 2p, 3p, scattered dynamic100 environments through SII. It is observed that for KNN-R strategy, as the environment becomes more challenging—i.e. increase in workload distribution (1p, 2p and 3p)—, the optimal k value decreases with SII. Cases 2p and 3p shows a similar optimal $k = 3$, however a higher granularity of simulation runs (e.g. $k = 2$ and $k = 4$) may reflect the reduction in optimal k value in these 2 cases. The scattered case shows the best performing strategy *ABC* which possess the highest exploration suited for this challenging environment. Therefore, as the environment becomes more challenging, the higher exploratory strategies will score a higher SII.

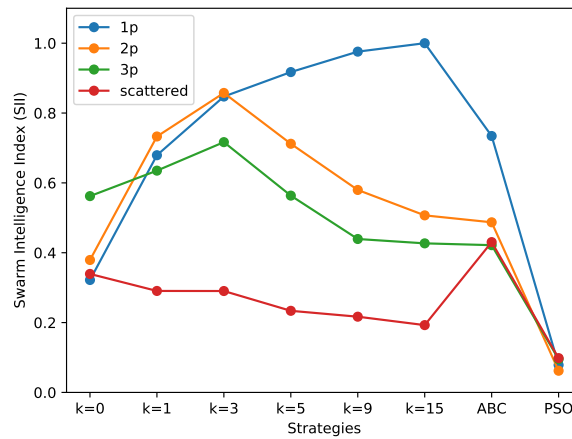


FIGURE 5.3: Swarm Intelligence Index (SII) of systems when change is injected @100 iteration

Effect of various dynamic environmental change (early to late)

In fig. 5.6, in a 2 pillars objective environment, the systems are subjected to various dynamics of the environment, from as early to late change. In this case, a slow change

signifies a change that occurs towards the end of the simulation, allowing the systems to stabilize and settled before the change is injected. An early change then signifies that the change occurs towards the start of the simulation, giving the systems less to no time to stabilize and settled before the change is injected. An early change is similar to a high frequency change while a late change is similar to a low frequency change in [?]. In this evaluation, we can examine the systems' settled level and their responses to the change.

TABLE 5.4: Simulation parameters: Effect of (slow to fast) dynamic change

Parameter	Quantity(units)
1. Total # of iteration	1250
2. Total # of agents	16
3. Agent-agent repulsion radius	8m
4. Agent sensor radius	0.5m
5. Total # of pillars	2
6. Change occurs at iteration	100, 600, 1000, <i>static</i>

In the 4 cases illustrated in fig. 5.6, a change in the environment is indicated when a drastic drop in MSI is observed in the results. These systems started with the same state—i.e. location and exploration state—across static, dynamic100, dynamic600, dynamic1000 environment. Therefore, the relative performances of the systems before the change are similar as compared to that of the static case. However, the systems start to perform differently after the change due to the different "initial state after the change" of the systems and the limit of the length of the simulation run.

In fig. 5.6(B), since there is only 250 iterations left after the change, a higher exploitative strategy $k = 9$ is required to rapidly make use of the existing environmental knowledge to gain more cells. $k = 15$ is overly exploitative and is not the right EE balance for this case. In fig. 5.6(C and D), the effect of the "settled-ness" (before the change) has on system (after the change) can be observed via the optimal k computed and displayed in fig. 5.5. In dynamic100, since the change occurs at iteration 100 when the systems are in the initial exploration phase, the states of the systems did not vary much from the initial state of the systems at iteration 0. However, the difference in the starting states of the systems (iteration 0 and 100) does pose a challenge to the systems as the relative performances of the systems decrease when comparing dynamic100 to static case. In dynamic600, a higher exploratory optimal $k = 1$ is essential to perform well after the change. Since the systems' performance are approaching a plateau—i.e. becoming stagnant—at iteration 600, the systems can be described as becoming stiff. Therefore, after the change, a strategy with a high exploration can outperform the others.

In fig. 5.5, it is observed that in a 2 pillars environment, when the problem increases in its temporal severity from dynamic100 to dynamic600, the optimal k reduces with SII. It is consistent with the trend seen in increasing the spatial severity of the problem from 1p, 2p to 3p. However, when the problem becomes even more challenging at dynamic1000, a higher exploitative strategy $k = 9$ performs the best. Interestingly, it

is not the case for the most challenging problem by spatial severity (scattered) which a higher exploratory strategy ABC performs the best.

As the systems operate longer (before the change) in an environment, it stabilizes and it tends to be in a more clustered state due to exploitation, which also contributes to the "settledness" of the systems. This state of the system poses a challenge for the system to self-organize to adapt to the environmental change and to escape the steady state. In the static environments, the systems started in a state where all the agents are evenly spread across the domain and they are all in exploration mode. It is not difficult to find the pillars and exploit the static pillars as compared to the systems being engaged in exploitation prior to the dynamic change. This requires the system to self-organize and search for the new pillars. Therefore the problem can get even more challenging when the environment becomes dynamic. Since a fast change occurs way before the system settles, it is more likely to adapt better to the change as the agents might be spread out and lesser are engaged in exploitation.

5.7 Conclusion

The purpose of this work is to bridge a significant gap between Swarm Intelligence research and physical world application, and also to understand SI in various difficult environments. The design phase of a swarming system is important to allow researchers and engineers reduce the resources needed to implement swarming system solutions. One of the tools that can aid in the process of understanding a particular SI strategy, measure it, and compare it with existing strategies is a benchmark problem. Inspired by research in e.g. artificial intelligence(machine learning) and optimization, where they have access to multiple standard benchmark or toy problems that can aid users in understanding the performance of their solutions, it can also present itself as a tool for fast prototyping to reduce overall implementation cost and to ensure effectiveness of solution.

In this work, we have designed an SI benchmark framework to probe the flexibility of a system such that it can be measured and be assessed against other solutions. From our study and perspective, flexibility is an outcome of collective behavior which can be an indication of SI and it is a property of a system that evolves with its environment—i.e. flexibility of a system will be different if it is subjected to different environment. Therefore in order to capture the full flexibility potential of a system, it needs to be subjected to multiple environments such that a suitable challenging problem can probe its flexibility. Note that it is not necessarily the most challenging problem that will probe the flexibility of the system.

The SI strategies ($KNN - R$, ABC , PSO) are being evaluated in the various environments spanning across different levels of workload distribution and dynamics of the environment. The objective in all the environments are kept constant—i.e. the occupied 'food' cells that the systems are required to discover is 12% of the domain area—while the different levels of workload distribution describe the spread of the objective, the dynamics describe when the change occurs (earlier or later) in the simulation run time.

The concept of exploration and exploitation is important in the discussion of the benchmark framework results. Since flexibility part of adaptability, exploration and exploitation is a suitable concept to better understand such adaptive process. For the strategies discussed in this work, they possessed a fix exploration-exploitation strategy, although $KNN-R$ can be further developed to encompass an adaptive exploration-exploitation strategy. As the environment changes, the adaptive $KNN-R$ can then change its k accordingly to suit the environment.

The metrics that we deployed in this work to for comparison is the *Map Similarity Index (MSI)* and the *Swarm Intelligence Index (SII)* of the system. The *MSI* describes the overall operational performance of the system in discovering occupied "food" cells. The *SII* of the system captures the flexibility of the system as it adapts to the change. The ranking of different system's flexibility is not trivial, although a system might recover the fastest from an environmental change, over time its performance might not sustain and be the best overall performing strategy. In the end, it depends on the profile of the problem that is posted to the swarming system. Is the system subjected to a short or long (time)

mission? If it is subjected to a long-term mission, is quick recovery still required and beneficial? If it is a short-term mission, will the strategy with the fastest recovery be able to maintain as the best performing overall strategy?

Therefore by utilizing our proposed benchmark framework, the user will be able to understand his/her SI solution and the environment in which it can perform the best. The user can also alter the benchmark environment and agents that best described the problem that is to be solved, and design a solution for it. This can be a starting point for the solution implementation into the physical world, and thereafter, tuning can be done to achieve better performance.

In our future work, we will examine the transferability of SI results gathered from our proposed benchmark problem to a different problem. If the results are not transferable to another problem, in other words, results are confined within this benchmark problem, the proposed benchmark problem will be rendered as impractical. The idea of SI transferability will be explored in the following Chapter 6.

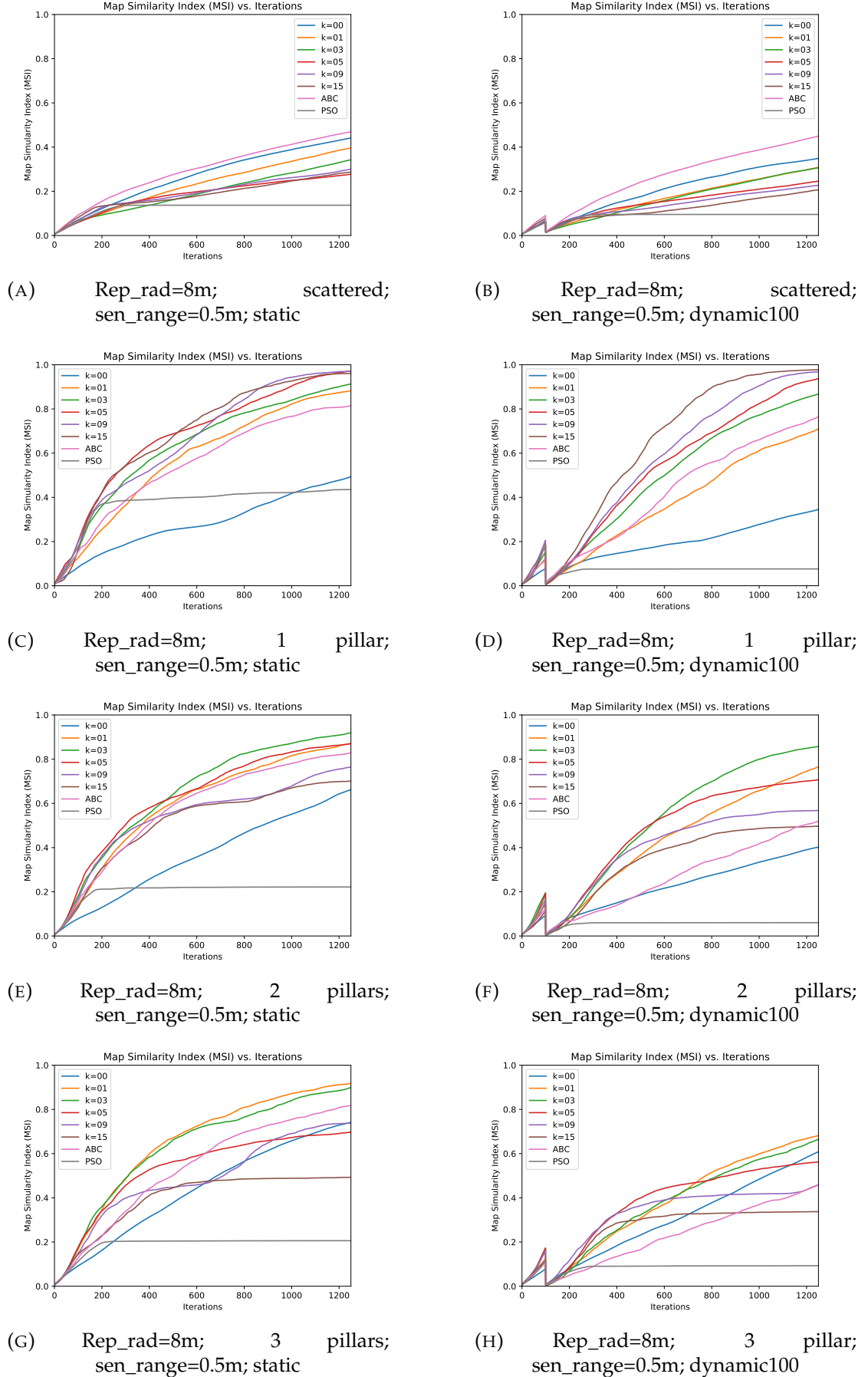


FIGURE 5.4: Dynamic environment change @ iteration 100; comparing 1p, 2p, 3p, scattered

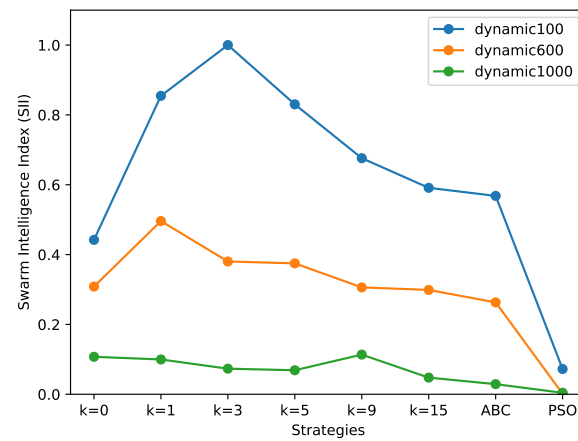
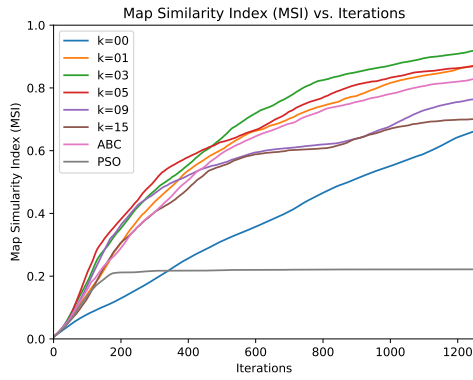
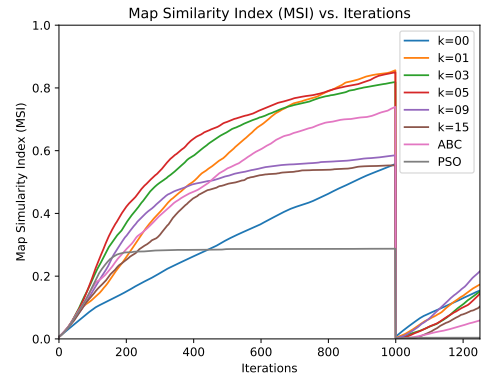


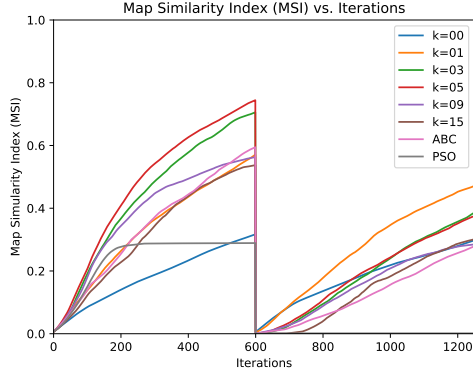
FIGURE 5.5: Swarm Intelligence Index (SII) of systems in 2 pillars environments



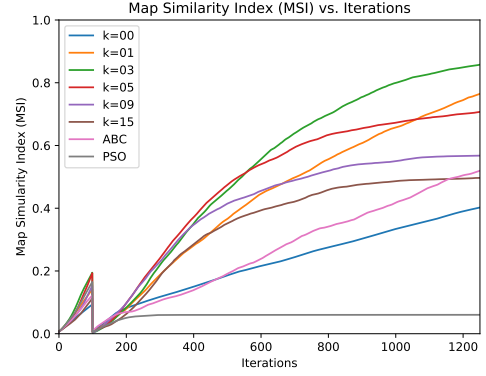
(A) Rep_rad=8m; 2pillars; sen_range=0.5m; static



(B) Rep_rad=8m; 2pillars; sen_range=0.5m; dynamic1000



(C) Rep_rad=8m; 2pillars; sen_range=0.5m; dynamic600



(D) Rep_rad=8m; 2pillars; sen_range=0.5m; dynamic100

FIGURE 5.6: Slow to fast environmental changes; comparing static, dynamic1000, dynamic600, dynamic100

Chapter 6

Qualitative study on the Transferability of Swarm Intelligence

6.1 Introduction

In Chapter 5, we have proposed a benchmark framework for evaluating and comparing the flexibility of systems. The benchmark framework is built upon a problem of collective mapping, where the objective is for the system of agents to search and map out the occupied cells—i.e. ‘food’ source(s). The systems will be evaluated through a series of different environment—i.e. different level of workload distribution and dynamics of the environment—, representing different level of difficulty specific to the individual systems. Since flexibility is a measure that is highly dependent on the system and its environment, by varying the difficulty of the environment it will probe the system differently, ideally allowing users to understand the best solution (SI strategy / algorithm) suited for the specific environments.

The overall operational performance (Map Similarity Index), Swarm Intelligence Index, and the exploration-exploitation balance of the systems in the various environments are studied to provide us with the insights into the best performing strategies. In future, with these information, users can select the best performing strategy for a given scenario as a starting point for physical implementation work, hopefully shortening the system tuning process to achieve better results. Now, having these results and insights, it begs the question of how useful they are given that the strategies are deployed to solve a different problem—i.e. other than collective mapping. This is essential to the concept of a benchmark framework, that the results and insights can be extracted and make sense of even when its applied to a different problem. If the findings and results are constrained within the benchmark problem, it will only provide limited contribution in terms of the evaluation of SI. This can render our proposed benchmark framework as ineffective.

The concept of applying the results and insights we gained from one problem to another is the study of transferability. In many research work, generalizability is a more common concept that is applied than transferability. In generalization, the findings from a large population is applicable to a specific case. However, in transferability,

it is the study of findings attained from a specific case applicable to another specific case. This concept of transferability are applicable in many area of research like in social sciences [329], medicine [330], robotics [331], and more. In terms of SI, which is highly dependent on the environment, it will be useful to learn about its transferability to benefit future development in SI.

In this work, we will be evaluating a SI algorithm in two different problems of collective mapping (benchmark problem) of multiple dynamic objectives and multiple fast target tracking (in [332]). The latter problem is selected as target tracking is a common operation for both multi-agent and multi-robot systems. Especially when there are multiple targets that are moving fast—i.e. faster than the agents, it is considered a NP-hard which was first formalized in the Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) framework by Parker and Emmons [258]. In this scenario, an application of a meta-heuristic solution like swarming algorithm can overcome the NP-hard problem by providing a sub-optimal performance. Considering the two problems, we will attempt to apply the strategy, PSO adaptive repulsion [332], and evaluate its performance and discuss the insights (i.e. trends) that can be observed in both the problems. The transferability will be studied qualitatively. The work will be presented in the following sections 1) Strategy: PSO adaptive repulsion, 2) Benchmark problem, 3) Multiple fast targets tracking, 4) Results and discussion, and 5) Conclusion.

6.2 Strategy: PSO adaptive repulsion

In this work, we are employing a PSO variant strategy, that is inspired by Kwa et. al. [144] PSO algorithm, into the two different proposed problem—i.e. collective mapping of dynamic distributed workload and multiple fast target tracking. This PSO variant is modified to include agent-to-agent adaptive repulsion and altering the system's communication network topology to overcome issues of over-exploitation—i.e. to encourage exploration in an environment where gaining new information is important to adapt to the unknown and dynamic environment.

Particle Swarm Optimization (PSO) algorithm is one of the earliest natural swarm inspired algorithm developed by Kennedy et. al. [326] to model after birds and fishes, for the purpose of studying the behavior of the systems' movement. Eventually, PSO has found its utility in solving optimization problem. It is not surprising as natural swarms perform constant optimization in their behaviors. PSO is a metaheuristic that makes less or few assumptions which allows it to be applied to wide range of optimization problem. However, it does not guarantee an optimal solution.

In general, the original PSO algorithm is a centralized approach in which agents make use of its current velocity vector, personal best (in terms of fitness value) location and global best location to compute its next velocity vector that guides the agents' way-point. It is described in eq. [4,5] of [42] by [326, 327]. The following eq. 6.1 described the velocity vector that each agent has to compute at every time step, also giving operators the ability to fine tune the weight of every term.

$$\mathbf{v}_i^{t+1} = \omega \cdot \mathbf{v}_i^t + \phi_{pb} \cdot \text{rand}(0, 1) \cdot (p_i^t - x_i^t) + \phi_{gb} \cdot \text{rand}(0, 1) \cdot (p_g^t - x_i^t) \quad (6.1)$$

Each agent will have a x_i that is a position in the domain while the velocity v_i vector. p_i being the current best personal solution and p_g being the best global solution of the swarm. The various terms in eq. 6.1 describe (1) the current velocity vector, (2) the vector towards the personal best and (3) the vector towards the global best, with $\omega, \phi_{pb}, \phi_{gb}$ as the weights to tune the system. ω describes the inertia of the agent in its current direction of travel, the higher the value of ω , the more inertia the agent possessed to be steered away by the other two terms. ϕ_{pb} and ϕ_{gb} would determine the influence of personal best and global best on the agent's new velocity vector. Notice that randomness $\text{rand}(0, 1)$ is also applied to these two terms which is important

However it is noted that in the application of PSO, agents tend to swarm towards and cluster at a single point of interest, often at the local minima or maxima instead of a global minimum or maximum—i.e. the system gets stuck in a local point of interest. This is an issue with PSO inherent over-exploitation which is generally caused by (1) outdated memory, and (2) excessive exploitation of the search environment at the expense of exploration [333]. In order to overcome over-exploitation and encourage exploration, there are various techniques that we can employ to include agent-to-agent adaptive repulsion and altering the system's communication network topology.

There are also other techniques that suit computational employment of PSO which randomizes the particle positions and fully restarting the entire PSO process when the 'global best' (gBest) value has not been updated in several iterations [334, 335] to encourage more exploration. However, these approaches are not ideal for physical world deployment—e.g. on a multi-robot system. It will not be effective to collect the physical system and redeploy them randomly when gBest stays stagnant. Therefore Jatmiko et. al. have proposed another technique of dispersing the agents if the gBest value has not changed for a specified number of iterations [336]—i.e. adaptive repulsion. One can also implement a PSO strategy with a constant agent-to-agent repulsion to reduce clustering [222].

Here is the repulsion vector [27] that we use in addition to the above 6.1 to repel away from other agents.

$$\mathbf{v}_{\text{rep},i}[t] = - \sum_{j \in \mathcal{N}_i} \left(\frac{a_R[t]}{r_{ij}[t]} \right)^d \frac{\mathbf{r}_{ij}[t]}{r_{ij}[t]}, \quad (6.2)$$

where $\mathbf{r}_{ij}[t]$ is the vector from agent i to agent j at time-step t . The level of inter-agent repulsion is controlled by (1) the dynamic repulsion strength, $a_R[t]$, represented in physical distance (m) that is to be kept amongst the agents, (2) the exponential term d in the term (a_R/r_{ij}) .

The adaptive repulsion works as describe in 4, where the system's repulsion radius is changed depending on the situation that they are in. As the system explores the environment, the agents that have detected points of interest will keep track of their individual $pbest$, p_i^t . In the event when an agent have an empty $pbest$ and its neighbors did not have $pbest(s)$ to share, the particular agent will start to increase its repulsion

radius towards the maximum $a_{R,\max}$ to encourage exploration in the swarm. In the event when the agent have a p_{best} or its neighbors share their p_{best} , the agent will then decrease its repulsion radius towards the minimum $a_{R,\min}$ to exploit and draw the swarm towards it. This provides the system with a level of adaptation when presented with a varied environment as illustrated in fig. 6.1.

Algorithm 4: Adaptive Repulsion

```

Set  $a_{R,\min} = 4$ ,  $a_{R,\max} = 12$ ,  $d = 6$ , and  $\delta = 1$ 
while System active do
  if  $p_i^t == []$  or  $\exists j \in \mathcal{N}_i$  s.t.  $p_j^t == []$  then
    if  $a_R < a_{R,\max}$  then
       $a_R \leftarrow a_R + \delta$ 
    end if
  else
    if  $a_R > a_{R,\min}$  then
       $a_R \leftarrow a_R - \delta$ 
    end if
  end if
end while

```

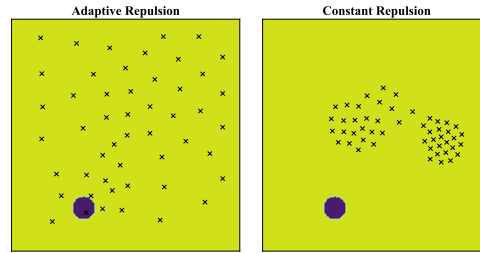


FIGURE 6.1: (Left) Agents using adaptive repulsion being blocked by the search-space boundaries and neighboring agents; (Right) agents remaining in place after losing track of the target with the low exploratory behavior in the presence of low constant repulsion.

Next, PSO network topology defines the channels at which information flows amongst the agents. Interestingly network topology does solve part of problem (1) and (2) mentioned above by restricting the information flow within the system. When the system is less connected, information will only be available within an exploiting sub-system, thereby causing only the sub-system to possess out-dated memory, if any, and allowing the rest of the system to continue exploring. This information exchange represented in the form of network topology is important in determining the collective response of a swarm [337]. The most commonly studied network structures are the static lbest (ring) and gbest (all-to-all) network topologies, with the gbest topology being the predominant approach [338]. In general, these networks are fixed during initialization of the system and do not change throughout the PSO simulation. In a recent study, it is recognized that agents in PSO networks with lower amounts of connectivity (i.e. low degree) perform more exploratory actions achieving higher global search performance.

In contrast, agents in networks with greater connectivity (i.e. high degree) perform more exploitative actions and fare better at local search tasks [339].

The use of a static network has not been a problem when performing PSO simulation, however this will pose a challenge to MRS, deploying PSO strategy, operating in an unstructured physical world. The limited communication range and information processing capabilities of a robotic system will constrain the number of connections it can establish and sustain. Therefore, dynamic PSO networks based on metric-distance neighbors have been proposed and studied [340, 341]. The findings showed that a larger communication range will result in faster convergence for the swarm [340]. As the communication range increases to cover the entire swarm, an all-to-all network is established and the system has reached its maximum convergence time [341].

The study of swarm network shows that there is an optimal level of connectivity amongst the agents. Interestingly, this optimal connectivity is directly related to speed of the driving signal [280]. Furthermore, it has also shown that although initial increase of k does improve the system's performance, beyond the optimal k , it will be detrimental to the system's performance—i.e. collective respond to external perturbations [272]. This implies that while tracking a dynamic target, instead of relying on all agents within a set communication range, tuning the level of connectivity of a swarm may result in a better collective tracking performance.

Finally, in Kwa et. al. [144] work, the concept of memory of agent was also explored. Since one of the issue with traditional PSO application is the (1) out-dated memory, an improved PSO was presented with a memory erase function. During implementation, if the agent's *lbest* does not change or improve for a specified number of iterations, then this particular agent will "forget" its *lbest*, in combination with the adaptive repulsion described earlier, to prevent the agent from over-exploitation. Kwa et. al.'s improved version of PSO includes (1) adaptive repulsion illustrated in fig. 6.1, and (2) memory erase.

The version of PSO adaptive repulsion with memory erase that will be use in this work is presented in algo. 5. It will be applied to two different problem of collective mapping and collective tracking, for the evaluation of the transferability of SI.

6.3 Problem 1: Collective Mapping

The first problem that the system will be solving is a collective mapping problem. In order to create a challenging mapping problem, we consider increasing the workload distribution and adding dynamics to the environment. The system of agents will traverse the environment to map out occupied pillar(s) of cells, which are dynamic and will change its location after a specified time interval. It could be an analogue to a sub-colony of worker ants foraging food sources in an unknown environment, noting the difference in the simulated agents that does not require them to transport the foraged food back to the nest, as there is not nest. In the physical world, the food sources could change due to various reasons like other animals or insects depleted the food source, the rain wash it away, or humans remove them. It is a problem where the

Algorithm 5: PSO with adaptive repulsion and memory erase

```

 $N = 16, k = 3, M = t_{mem}$ 
while  $t < max\_iteration$  do
  for each agent  $i$  do
    Receive  $pbest_j$  from  $k$  neighbors
    Select most updated  $pbest_i$  and  $pbest_j$  to be  $lbest_i$ 
    Compute  $\mathbf{v}_i^{t+1}$  shown in eq.(6.1)
    Compute  $\mathbf{v}_{rep,i}^t$  shown in eq.(6.2)
     $\mathbf{v}_{total,i}^{t+1} \leftarrow \mathbf{v}_i^{t+1} + \mathbf{v}_{rep,i}^t$ 
    Adjust agent  $i$   $a_R$  with algo. 4
    Move agent  $i$  with  $x_i^{t+1} \leftarrow x_i^t + \mathbf{v}_{total,i}^{t+1}$ 
    Update  $[pbest_i, t]$  if any
    if  $(t - t \text{ of } pbest_i) == M$  then
      Empty  $pbest_i$ 
    end if
  end for
end while

```

agents constantly explore and exploit the known and unknown food sources within the domain. Exploration is required to search for new food sources during initial exploration phase or when environmental changes are introduced. On the other hand, exploitation is required to map out (or collect) as much of the pillar(s) of "food sources" as possible.

From the perspective of collective mapping, we have made modification to the problem such that it is simplified for us to better evaluate SI of the system. Firstly, pillar(s) of interest will not pose as environmental obstacles, which means that the agents will not need to perform obstacle avoidance. This allows for an observation of the direct exploration and exploitation, as obstacle avoidance can alter the display of the agents behavioral intent. For example, an agent intent is to move towards a computed waypoint to perform exploitation, however, when an obstacle stands between itself and the waypoint, it can cause the agent to take a detour and may portray a less exploitative action as it intended.

Secondly, the agents' sensor data like odometry, orientation measurement, and short range obstacle sensor data are considered without errors—i.e. perfect sensor data. This means that the occupancy grid map(OGM) does not need to hold the belief(of the probability of the existence of obstacles) of the agents reducing the chance of compounded errors negatively influencing the system. Since the orientation and odometry sensor data are without error, after initialization of a common reference frame, the local map merging process of the shared local maps(OGM) according to the defined network is simplified. There is no need for applying map merging techniques to combine local maps. Since there is no errors, the values in the OGM cells are 0, 0.5, or 1.0, representing *cleared*, *unknown*, or *occupied* cells respectively. Furthermore, we assumed that the agents start in a known but cleared environment which further simplify the OGM to take only values 0 or 1.0. The merging of the local maps will simply be as follows,

shown in eq. (6.3). If we were to remove these assumptions, there are techniques and solutions in research to deal with each of our assumptions like implementing Simultaneous Localization and Mapping, map merging techniques without common reference frame, etc..

$$C_{ij} = \begin{cases} 1, & \text{if } A_{ij} = 1 \text{ or } B_{ij} = 1 \\ 0, & \text{if } A_{ij} = 0 \text{ and } B_{ij} = 0 \end{cases}, \forall (i, j) \in \{1, 2, \dots, 480\} \quad (6.3)$$

The simulation environment is $48m$ by $48m$, with a resolution of 0.1 —i.e. 480×480 grids—, represented in an occupancy grid map (OGM). There are 2 pillars of occupied cells (or food sources) that makes up 12% of the domain, in this case, each pillar makes up 6% of the domain. The pillars' *length = width* will also be maintained to ensure "fairness" when searching in both x and y direction. The multiple pillars are designed to not overlap, even if there is a change in the environment, the pillars before and after will also not overlap. In this problem, the change in the environment will be injected at iteration 100.

In this simulation, a total of $N = 16$ agents are deployed, each of the agents move with a constant speed of $1\text{cell}/\text{iteration}$ or $10\text{cm}/\text{iteration}$. Due, to the granularity of the environment, the agents are limited to moving 1cell in one of the eight direction—i.e. $[0^\circ, 360^\circ]$ with increment of $+45^\circ$. The agents are equip with a variable range all-around sensor—e.g. lidar—, default set to $0.5m$. The agents will be connected and share information locally based on the k-nearest neighbor network described in section 6.2.

6.4 Problem 2: Multiple fast tracking

The second problem that the system will be solving is a multiple target tracking problem. The objective is for the agents to stay on target—i.e. within the radius of the target. In order to increase the difficulty of the problem, the targets are set to move faster than the agents. This could be an analogue to a military defense mission where drones are deployed to track multiple fast enemies movement.

In order to make the problem more challenging, the agents possess a near-zero-range sensor tracking a target of radius $1m$ that does not emit a gradient field. In other words, the agents will only know that they are tracking if they are within the target radius. Furthermore, the targets are moving faster than the agents which increased the difficulty of the task drastically.

The targets move in accordance to a "non-evasive" or a "evasive" policy. In the "non-evasive" policy, the target traverse to random way-points within the environment. In the "evasive" policy, the target follow the "non-evasive" policy until it comes into contact with an agent. Upon contacting an agent—i.e. positioned within the target's radius—, the target computes its velocity using a repulsion equation [144], with all the agents within its radius as repulsion neighbours. After encountering agents for

t_{limit} consecutive iterations, the target will travel in one direction for t_{evade} iterations, in attempt to escape its pursuers.

There agents operate in a continuous search space of $25m$ by $25m$. A total of $N = 50$ agents are deployed in an environment with two "evasive" targets. The agents are communicating according to a communication network designed within the system's strategy described in section 6.2.

6.5 Results and discussion

In this simulation, the strategy of PSO adaptive repulsion is applied to two different problem of collective mapping and multiple target tracking problem. Despite the vast difference in the assigned tasks, the trends of the performance of the systems observed from these results are common across both problems, suggesting a level of transferability of SI.

Firstly, it is observed that there is a common trend of the presence of an optimal k that yields the best performance in their respective solutions. In figure 6.2, it shows the collective mapping performances of the various k value system over the period of simulation (1250 iteration). The performance of the system refers to the cumulative number of occupied cells detected throughout the operation, both before and after the environmental change is introduced. It is observed that as k increases from 0, the worst performing k value system, the system's performance increases up to its best performing $k = 3$. After which, the system's performance deteriorate as k continues to increase. This showed an optimal performing k of the system, further illustrated in figure 6.3b (dotted green line), reiterating the optimal best performing $k = 3$.

Similarly, this trend of optimal k appears in all the cases with different environmental settings—i.e. dynamic change at 100 and 600, and 2 and 3 pillars. Result in figure 6.3a also showed that in both "non-evasive" and "evasive" target(s) scenario, there exist an optimal best performing k value system. However, both the 1 pillar cases (in fig. 6.3b) have seen to plateau instead, as there is only 1 objective (pillar) the high exploitative strategies $k = 9$ and $k = 15$ behaves similarly—i.e. $k = 9$ were not given a environmental condition to form more than one clusters, both $k = 9$ and $k = 15$ form one single cluster in this environment.

Secondly, as the problem becomes more challenging—i.e. increasing workload distribution (targets and pillars), or dynamic of the environment (evasive targets and fast/rapid change)—, both the overall performance and the optimal k of the systems reduce. In fig. 6.3a and 6.3b, although they represent two very different problems, this trend persists. This shows that the optimal k , for the same system, changes as the system operates under various environmental settings. It suggests a constant choosing (or balancing) of optimal as the system operates in a dynamic environment. In fig. 6.3b, the three pillars cases (green dotted and solid lines) does not show a reduction in k , from a less challenging dynamic600 to more challenging dynamic100 environment. This could be due to the limitation of the collective mapping simulation environment where it cannot accommodate more agents leading to a low granularity of possible k

and its performance. In this work, we have compared and discussed about PSO adaptive repulsion with k as a tunable variable to alter the system's performance to suit the environment, k is merely a representation of the system's exploration-exploitation balance. When the system's connectivity k is low, the system is loosely connected, which means that it possess more exploratory actions than exploitative actions. Therefore, in order for a system to adapt to changes in the environment, it is ideal for the strategy to be tunable—i.e. ability to tune the exploration and exploitation of the system. It does not need to be the k value, which is specific to PSO adaptive repulsion. The tuning is beneficial as there exist an optimal exploration-exploitation balance. In an environment that is more challenging, the system should increase its exploratory actions (reduce exploitation)—e.g. reducing k for PSO adaptive repulsion strategy—, and vice versa.

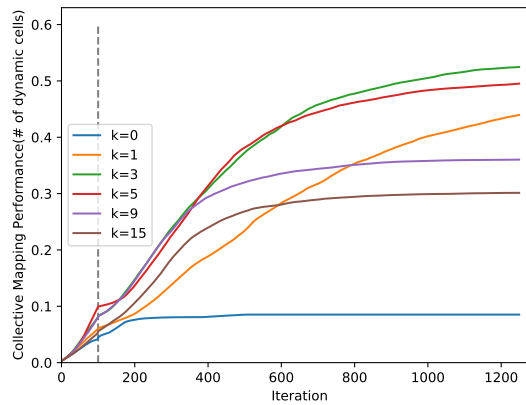
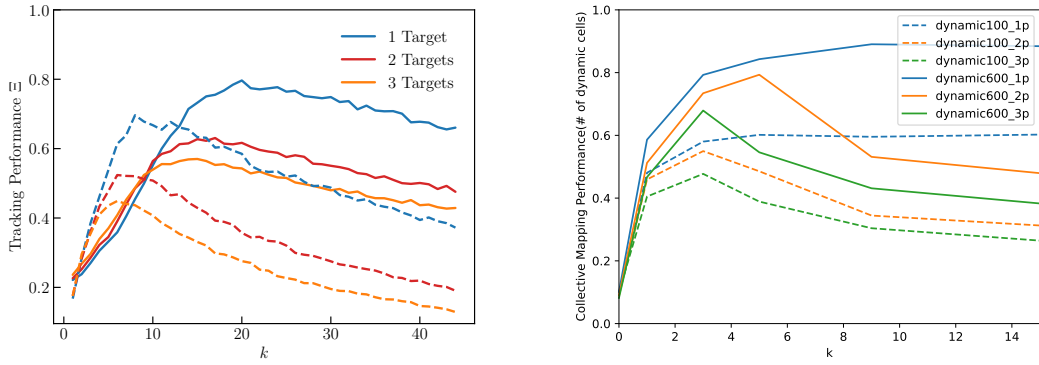


FIGURE 6.2: PSO adaptive repulsion collective mapping performance, varying $k = [0,1,3,5,9,15]$ (of pillars = 2, dynamic change @iteration100, sensor_range = 0.5m, [min rep, max rep] = [3m, 12m]

6.6 Conclusion

We have examined PSO adaptive repulsion with memory erase strategy operating in two different problems of collective mapping and multiple fast targets tracking. This work attempts to show the transferability of Swarm Intelligence. It may not be an exact system performance transfer as Swarm Intelligence is environment (or problem) dependent, but the trends of SI that we have observed are consistent with the two problems.

A level of transferability of SI trends can be observed across fig. 6.3a and fig. 6.3b. There were two trends that persists in both the results, (1) there exist an optimal k for all cases with varying environmental parameters, and (2) when the difficulty of the environment is increased, it is observed that the performance of the system and the optimal k reduces.



(A) Multiple non-evasive (solid lines) and evasive (dashed lines) target tracking performance (maximum agent's velocity $v_{a,max} = 0.1$, memory length of $t_{mem} = 20$, and maximum targets' velocity $v_{o,max} = 0.2$).

(B) Dynamic change at 600 (solid lines) and Dynamic change at 100 (dashed lines) collective mapping performance (# of pillars = 2, dynamic change at iteration 100 and iteration 600, sensor_range = 0.5m, [min rep, max rep] = [3m, 12m]).

FIGURE 6.3: Performance vs. k

The trend of an optimal k for each system in different environment, brings about an insight into a tuning a tunable SI strategy. As the environment changes, in this case, the system can tune k such that it can aim to maintain the optimal k for various environment. In our strategy, tuning k will indirectly affect the exploration and exploitation of the system—i.e. a representation of exploration-exploitation balance. When a system can be tuned to shift its exploration-exploitation balance, it will be truly an adaptive strategy. However, when will the system tune its exploration-exploitation and to what level?

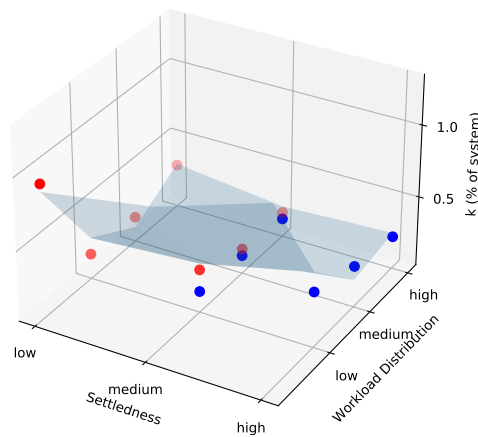


FIGURE 6.4: A concept of swarming system characteristic map

In order to predict the exploration-exploitation balance of a system, we would proposed a concept of a swarming system characteristic map illustrated in fig. 6.4. The purpose of this characteristic map can form a basis for users or systems to first select

a starting exploration-exploitation balance point, thereafter fine-tuning can be made to further improve the performance of the systems. In order to build this characteristic map, the system can be subjected to a series of environment in our benchmark framework [Chapter 5]. For each combination of environment temporal and spatial severity, the optimal k with the best system performance will be plotted onto the characteristic map. In fig. 6.4, the environmental parameters used were "Settledness" and "Workload Distribution" representing the difficulty of the environment and the exploration-exploitation parameter used was " k ".

Nonetheless, this work can be expanded to further study the transferability of SI in a quantitative manner and with different environments.

Chapter 7

Conclusions

This work began by exploring the behaviors of the swarming system, where social insects and animals display a form of order in their daily maintenance of hives, escaping predators, foraging, etc. The order displayed in natural swarming systems is intriguing due to the lack of a central command providing the system of agents with instructions, often these "instructions" are rules embedded within the system such that the agents can carry out their functions. This phenomenon is often said to be possible with a form of intelligence—i.e. Swarm Intelligence.

In order to learn from nature and apply this intelligence to solve the many problems faced by humans with the advantages that swarming systems can offer, it is important to understand (SRQ1) What is Swarm Intelligence (SI)? Currently, there are multiple perspectives on the definition of SI from various groups of researchers—e.g. biologist, roboticist, computer scientist, and more. Although these multiple SI definitions suggest non-unifying definitions, these perspectives highlighted the multiple facades of SI.

As the research community progresses on the development of SI, there is a need for the analysis of SI systems. In the near future, as the development space of SI application gets saturated, by showing that SI solution works to solve a particular problem will no longer suffice. As SI development progresses, it requires an analysis that can measure and compare SI among the systems so as to determine a better suited solution for the problem. Note that a high overall system performance does not necessarily suggest a high SI. Therefore, in this work, we have addressed (SRQ2) How can SI be quantified, such that it can be measured and compared?

We have learned that SI is a result of self-organization (SO), which is a widely accepted. SO is a collective behavior that emerges from the constant local interactions among the individual agents. It is the first phenomenon that many recognizes as SO is the ordered pattern that emerges as the agents communicate, interact, and move. Many works attempted to measure SO with order. However in our study, interestingly order is not the only possible indication of SO, flexibility can be the other indication. Cavagna et. al. suggested that order might not be a good measure as it can be from both a centralized or a decentralized system. Furthermore, it is not trivial to measure order and determine if it arises from one of these two types of system. Therefore, the group proposed to use collective response (a.k.a flexibility)—i.e. which is also a feature of

swarming systems—as an indication of SO as it is a trademark of self-organized order as opposed to a centralized one. This was further explained in Chapter 2.

Now we have considered flexibility as a possible indication of the collective response of a self-organizing system, which can describe the SI of the system. Therefore we proposed the use of flexibility as a way to quantify SI. Flexibility is the system's ability to adapt to a change or unknown in the environment. For a system to be evaluated on its SI (or flexibility), it requires an environment that is challenging such that SI can be probed. Therefore, besides having a set of metrics to measure SI, we have proposed a benchmark framework that provides a range of environment (of the same problem) that increases in difficulty to evaluate the systems' SI ability.

The design of the environment is important as the same system can display different levels of SI in different environment. Our benchmark framework, in Chapter 5, revolves around the problem of Collective Mapping. In which, the difficulty of the environment can be adjusted with two environmental variables—i.e. workload distribution and the iteration at which change is injected. As we subject the various SI strategies to the different environments, we have observed that the different strategies adapt better in one environment than the other. The operational performance is captured with the Map Similarity Index (MSI) and the Swarm Intelligence Index (SII) is deduced from MSI.

The concept of exploration and exploitation is also introduced in this work, to facilitate the understanding of the flexibility and adaptability of the systems. However, this concept stretches beyond the recovery of a system, it allow us to understand how the systems behave initially when the environment is unknown, how the systems recover from an environmental change, and how the systems detect and exploit the change. This provides an explanation and understanding of a system recovering well but not necessarily perform well overall—e.g. a strategy with high exploration might detect a change earlier, but might not have enough exploitative actions to gather the cells from the pillars quickly. In our KNN-R and PSO adaptive repulsion strategies, the tunable k nearest neighbor communication network control the information propagation within the system. This value of k indirectly controls the level of exploration-exploitation of the system, for example a lower k would suggest a high exploration and low exploitation, and a higher k would suggest a low exploration and high exploitation. In our work, these k values are fixed and does not change through the course of the operation. However in our study, we reckon that a tunable k —i.e. exploration-exploitation balance—should be change in order to maximize the overall performance of the system. From this work, it is evident that a fix value k —i.e. a fix exploration-exploitation balance—will not yield the overall optimal results. As the environment and the state of the system evolves, the optimal exploration-exploitation balance tend to change. The next possible extension to our work can be the development of a strategy with an adaptive exploration-exploitation balance.

In Chapter 6, the transferability of SI (from benchmark framework results) is evaluated qualitatively. It is important to consider transferability of the benchmark framework, as the SI performance is often study within an environment, so that even if SI is applied to a different problem the findings and trends will remain largely true. If the results are

constrained within the benchmark framework, then it will render the benchmark as ineffective. We have evaluated PSO adaptive repulsion on both problems of Collective Mapping and Multiple tracking fast targets. It is observed that in both problems, despite the various environmental settings, there exists an optimal k that gives the best overall performance. As the environment becomes more challenging, both the optimal k and the overall performance of the system is reduced. This suggests that when a system is in a more challenging environment, a more exploratory strategy—i.e. lower k —is preferred, while in a less challenging environment, a more exploitative strategy—i.e. higher k —is preferred.

We have proposed a way to represent the transferability of SI via a system characteristic surface plot similar to a typical system characteristic curve, illustrated in fig. 6.4 showing a surface graph of *Settledness* vs. *Workload distribution* vs. k . 'Settledness' is a measure of the system's activity before the environmental change is introduced, and determine if the system is in a steady state or if it is still active performing its task. This is a more practical generic quantity than a direct measure of the signal of the dynamic change in the environment to describe the dynamic element in the environment. Since 'settledness' is a system quantity, it is a practical approach for future machine learning or reinforcement learning application, rather than measuring the 'dynamicity' of the environment from the system's perspective which can be almost impossible. 'Workload distribution' is used specifically in our benchmark framework to describe the challenging environment, and how challenging it is can include features beyond workload distribution. For example in the problem of multiple tracking fast targets, the factors contributing to a challenging environment are number of targets, target size, target speed. In future, we can incorporate all these factors to determine how challenging is the environment to the system. This is important as we have seen that by changing the level of difficulty of the environment, the system needs to adapt and prefer a different k connection value. Although ' k ' is specific to strategies like KNN-R and PSO adaptive repulsion, it represents the exploration-exploitation balance of the system. If other tunable strategies that balances their exploration-exploitation via other parameters, it can replace k .

In future, the idea is to put the system through various environment of different *challenges* and to log the system *settledness* when the change occurs, and the optimal *exploration-exploitation* balance. This system characteristic map, built with a specific problem, can then be used to predict the optimal *exploration-exploitation* balance for the system to be effective in the environment presented to it, based on the *settledness* of the system and the level of *difficulty* of the environment. However, SI is extremely sensitive to the differences in the operating environment, therefore it will not be trivial to relate the *settledness* and the *difficulty* level of a different problem to a baseline problem. One approach could be to determine the range of difficulty levels of the new problem, then map it onto the system characteristic surface plot. Then there will be a reference to determine how the difficulty levels in problem *A* relates to the difficulty levels in problem *B*.

This compliments the possibility of applying reinforcement learning to tune the exploration-exploitation balance of a system. The system could start off its operation with system characteristic map from our benchmark framework. As it performs its operation, it

can update and refine this system characteristic map on-the-go to better suit its operational environment. At the same time, it can use this map to continuously shift its exploration-exploitation balance in attempt to achieve a higher performance.

In conclusion, this thesis has addressed the main and sub-research questions. Throughout this thesis, we have contributed to (C1) Quantification of Swarm Intelligence, (C2) Application of Swarm Intelligence in multi-robot collective mapping problem, (C3) Benchmark framework to compare and analyze flexibility of swarming systems, and (C4) Qualitative study on the transferability of Swarm Intelligence.

Bibliography

- [1] Y. K. Kuan, "Design of an effective swarming system for the pervasive monitoring of aquatic environments," PhD Thesis, Singapore University of Technology and Design, Singapore, 2018.
- [2] M. Maeterlinck *et al.*, "Life of the white ant," 1930.
- [3] R. Bouffanais, *Complexity and Swarming Systems*. Singapore: Springer Singapore, 2016, pp. 1–3. [Online]. Available: https://doi.org/10.1007/978-981-287-751-2_1
- [4] E. Schultner and U. Pulliainen, "Brood recognition and discrimination in ants," *Insectes Sociaux*, vol. 67, no. 1, pp. 11–34, 2020.
- [5] C. R. Fonseca, "Nesting space limits colony size of the plant-ant pseudomyrmex concolor," *Oikos*, pp. 473–482, 1993.
- [6] F. Bastin, H. Chol  , G. Lafon, and J.-C. Sandoz, "Virgin queen attraction toward males in honey bees," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.
- [7] A. Schrempf, S. Aron, and J. Heinze, "Sex determination and inbreeding depression in an ant with regular sib-mating," *Heredity*, vol. 97, no. 1, pp. 75–80, 2006.
- [8] J. H. Fewell and M. L. Winston, "Regulation of nectar collection in relation to honey storage levels by honey bees, *apis mellifera*," *Behavioral Ecology*, vol. 7, no. 3, pp. 286–291, 1996.
- [9] C. Collison, "A closer look–foraging behavior," *Bee Culture*, 2018.
- [10] M. D. Rivera, M. Donaldson-Matasci, and A. Dornhaus, "Quitting time: When do honey bee foragers decide to stop foraging on natural resources?" *Frontiers in Ecology and Evolution*, vol. 3, p. 50, 2015.
- [11] B. Cartwright and T. S. Collett, "Landmark learning in bees," *Journal of comparative physiology*, vol. 151, no. 4, pp. 521–543, 1983.
- [12] B. Kheradmand and J. C. Nieh, "The role of landscapes and landmarks in bee navigation: A review," *Insects*, vol. 10, no. 10, p. 342, 2019.
- [13] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and biological systems: towards a new bionics?* Springer, 1993, pp. 703–712.
- [14] K. Tirri and P. Nokelainen, *Measuring multiple intelligences and moral sensitivities in education*. Springer Science & Business Media, 2012, vol. 5.
- [15] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale, "Scale-free correlations in starling flocks," *Proceedings of the National*

- Academy of Sciences of the United States of America*, vol. 107, no. 26, pp. 11 865–11 870, jun 2010.
- [16] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.
- [17] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995.
- [18] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, vol. 4. Citeseer, 1995, pp. 1942–1948.
- [19] M. Moussaïd, D. Helbing, and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6884–6888, 2011.
- [20] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.
- [21] S. Garnier, C. Jost, J. Gautrais, M. Asadpour, G. Caprari, R. Jeanson, A. Grimal, and G. Theraulaz, "The embodiment of cockroach aggregation behavior in a group of micro-robots," *Artificial life*, vol. 14, no. 4, pp. 387–408, 2008.
- [22] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tâche, I. Saïd, V. Durier, S. Canonge, J. M. Amé *et al.*, "Social integration of robots into groups of cockroaches to control self-organized choices," *Science*, vol. 318, no. 5853, pp. 1155–1158, 2007.
- [23] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in self-organized robot colonies," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 695–711, 2009.
- [24] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano *et al.*, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2, pp. 223–245, 2004.
- [25] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneuborg, and M. Dorigo, "The cooperation of swarm-bots: Physical interactions in collective robotics," *IEEE Robotics & Automation Magazine*, vol. 12, no. 2, pp. 21–28, 2005.
- [26] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy *et al.*, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.
- [27] J. L. Kit, A. G. Dharmawan, D. Mateo, S. Foong, G. S. Soh, R. Bouffanais, and K. L. Wood, "Decentralized multi-floor exploration by a swarm of miniature robots

- teaming with wall-climbing units,” in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019, pp. 195–201.
- [28] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
- [29] B. M. Zoss, D. Mateo, Y. K. Kuan, G. Tokić, M. Chamanbaz, L. Goh, F. Vallegra, R. Bouffanais, and D. K. Yue, “Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies,” *Autonomous Robots*, vol. 42, no. 8, pp. 1669–1689, 2018.
- [30] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira, and A. L. Christensen, “Evolution of collective behaviors for a real swarm of aquatic surface robots,” *PloS one*, vol. 11, no. 3, p. e0151834, 2016.
- [31] P. Zahadat and T. Schmickl, “Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition,” *Adaptive Behavior*, vol. 24, no. 2, pp. 87–101, 2016.
- [32] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments,” *Science Robotics*, vol. 3, no. 20, 2018.
- [33] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, “Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment,” *Science Robotics*, vol. 4, no. 35, 2019.
- [34] C. Meyer and E. Bonabeau, “Swarm intelligence: a whole new way to think about business,” *Harvard Business Review*, vol. 5, pp. 107–114, 2001.
- [35] D. Bozhinoski and M. Birattari, “Towards an integrated automatic design process for robot swarms,” *Open Research Europe*, vol. 1, no. 112, p. 112, 2021.
- [36] M. Dorigo, G. Theraulaz, and V. Trianni, “Reflections on the future of swarm robotics,” *Science Robotics*, vol. 5, no. 49, 2020.
- [37] —, “Swarm robotics: Past, present, and future [point of view],” *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.
- [38] E. Bonabeau, M. Dorigo, and G. Theraulaz, “From natural to artificial swarm intelligence,” 1999.
- [39] Y. Liu and K. M. Passino, “Swarm intelligence: Literature overview,” *Department of electrical engineering, the Ohio State University*, 2000.
- [40] D. Karaboga and B. Akay, “A survey: algorithms simulating bee swarm intelligence,” *Artificial intelligence review*, vol. 31, no. 1-4, p. 61, 2009.
- [41] S. Roy, S. Biswas, and S. S. Chaudhuri, “Nature-inspired swarm intelligence and its applications,” *International Journal of Modern Education & Computer Science*, vol. 6, no. 12, 2014.
- [42] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, “A comprehensive review of swarm optimization algorithms,” *PloS one*, vol. 10, no. 5, p. e0122827, 2015.

- [43] M. Fleischer, "Foundations of swarm intelligence: From principles to practice," *arXiv preprint nlin/0502003*, 2005.
- [44] A. Abraham, H. Guo, and H. Liu, "Swarm intelligence: foundations, perspectives and applications," in *Swarm intelligent systems*. Springer, 2006, pp. 3–25.
- [45] G. Folino and A. Forestiero, "Using entropy for evaluating swarm intelligence algorithms," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010, pp. 331–343.
- [46] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," *Nature-Inspired Computing and Optimization*, pp. 475–494, 2017.
- [47] R. Wanka, "Swarm intelligence," *it-Information Technology*, vol. 61, no. 4, pp. 157–158, 2019.
- [48] J. L. Kit, D. Mateo, and R. Bouffanais, "A decentralized mobile computing network for multi-robot systems operations," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018, pp. 309–314.
- [49] G. Beni, "From swarm intelligence to swarm robotics," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 1–9.
- [50] R. Serra and G. Zanarini, *Complex systems and cognitive processes*. Springer Science & Business Media, 2013.
- [51] Y. Inada and K. Kawachi, "Order and flexibility in the motion of fish schools," *Journal of theoretical Biology*, vol. 214, no. 3, pp. 371–387, 2002.
- [52] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale, "Scale-free correlations in starling flocks," *Proceedings of the National Academy of Sciences*, vol. 107, no. 26, pp. 11 865–11 870, 2010.
- [53] A. Attanasi, A. Cavagna, L. Del Castello, I. Giardina, S. Melillo, L. Parisi, O. Pohl, B. Rossaro, E. Shen, E. Silvestri *et al.*, "Collective behaviour without collective order in wild swarms of midges," *PLoS computational biology*, vol. 10, no. 7, p. e1003697, 2014.
- [54] G. Nicolis, "Self-organization in nonequilibrium systems," *Dissipative Structures to Order through Fluctuations*, pp. 339–426, 1977.
- [55] L. Mikaël, "Formal verification of flexibility in swarm robotics," Ph.D. dissertation, Citeseer, 2012.
- [56] T. Fukuda, D. Funato, K. Sekiyama, and F. Arai, "Evaluation on flexibility of swarm intelligent system," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 4. IEEE, 1998, pp. 3210–3215.
- [57] J. Harwell and M. Gini, "Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm," *arXiv preprint arXiv:1907.03880*, 2019.

- [58] N. D. Daw, J. P. O’doherly, P. Dayan, B. Seymour, and R. J. Dolan, “Cortical substrates for exploratory decisions in humans,” *Nature*, vol. 441, no. 7095, pp. 876–879, 2006.
- [59] J. D. Cohen, S. M. McClure, and A. J. Yu, “Should i stay or should i go? how the human brain manages the trade-off between exploitation and exploration,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, no. 1481, pp. 933–942, 2007.
- [60] O. Berger-Tal, J. Nathan, E. Meron, and D. Saltz, “The exploration-exploitation dilemma: a multidisciplinary framework,” *PloS one*, vol. 9, no. 4, p. e95693, 2014.
- [61] J. G. March, “Exploration and exploitation in organizational learning,” *Organization science*, vol. 2, no. 1, pp. 71–87, 1991.
- [62] J. Uotila, M. Maula, T. Keil, and S. A. Zahra, “Exploration, exploitation, and financial performance: analysis of s&p 500 corporations,” *Strategic management journal*, vol. 30, no. 2, pp. 221–231, 2009.
- [63] M. Yogeswaran and S. Ponnambalam, “Reinforcement learning: exploration-exploitation dilemma in multi-agent foraging task,” *Opsearch*, vol. 49, no. 3, pp. 223–236, 2012.
- [64] S. Leonardos, G. Piliouras, and K. Spendlove, “Exploration-exploitation in multi-agent competition: Convergence with bounded rationality,” *arXiv preprint arXiv:2106.12928*, 2021.
- [65] S. Leonardos and G. Piliouras, “Exploration-exploitation in multi-agent learning: Catastrophe theory meets game theory,” *arXiv preprint arXiv:2012.03083*, 2020.
- [66] J. R. Sampson, “Adaptation in natural and artificial systems (john h. holland),” 1976.
- [67] C. Shapiro, S. Carl, H. R. Varian *et al.*, *Information rules: a strategic guide to the network economy*. Harvard Business Press, 1998.
- [68] W. W. Powell, K. W. Koput, and L. Smith-Doerr, “Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology,” *Administrative science quarterly*, pp. 116–145, 1996.
- [69] R. Katila and G. Ahuja, “Something old, something new: A longitudinal study of search behavior and new product introduction,” *Academy of management journal*, vol. 45, no. 6, pp. 1183–1194, 2002.
- [70] K. Mehlhorn, B. R. Newell, P. M. Todd, M. D. Lee, K. Morgan, V. A. Braithwaite, D. Hausmann, K. Fiedler, and C. Gonzalez, “Unpacking the exploration-exploitation tradeoff: A synthesis of human and animal literatures.” *Decision*, vol. 2, no. 3, p. 191, 2015.
- [71] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, 2005.

- [72] R. Bouffanais, *Design and Control of Swarm Dynamics*, ser. SpringerBriefs in Complexity. Heidelberg: Springer, 2016.
- [73] W. Rone and P. Ben-Tzvi, "Mapping, localization and motion planning in mobile multi-robotic systems," *Robotica*, vol. 31, no. 1, pp. 1–23, 2013.
- [74] A. Sekunda, M. Komareji, and R. Bouffanais, "Interplay between signaling network design and swarm dynamics," *Network Sci.*, vol. 4, no. 2, pp. 244–265, 2016.
- [75] D. Mateo, N. Horsevad, V. Hassani, M. Chamanbaz, and R. Bouffanais, "Optimal network topology for responsive collective behavior," *Science Advances*, vol. 5, p. eaau0999, Apr 2019.
- [76] J. L. Kit, D. Mateo, and R. Bouffanais, "A decentralized mobile computing network for multi-robot systems operations," in *2018 9th IEEE UEMCON International Conference*. New York City, N.Y.: IEEE, 2019, in Press.
- [77] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proc. IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.
- [78] J. Sundram, V. D. Nguyen, G. S. Soh, R. Bouffanais, and K. L. Wood, "Development of a miniature robot for multi-robot occupancy grid mapping," in *3rd IEEE Int. Conf. Adv. Rob. Mechatron. (ICARM)*, Jul 2018, pp. 414–419.
- [79] J. Djugash, S. Singh, and B. Grocholsky, "Decentralized mapping of robot-aided sensor networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2008, pp. 583–589.
- [80] M. Turduev, Y. Ataş, P. Sousa, V. Gazi, and L. Marques, "Cooperative chemical concentration map building using decentralized asynchronous particle swarm optimization based search by mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 4175–4180.
- [81] M. Duarte, J. Gomes, V. Costa, T. Rodrigues, F. Silva, V. Lobo, M. M. Marques, S. M. Oliveira, and A. L. Christensen, "Application of swarm robotics systems to marine environmental monitoring," in *Proc. OCEANS 2016-Shanghai*, 2016, pp. 1–8.
- [82] B. M. Zoss, D. Mateo, Y. K. Kuan, G. Tokić, M. Chamanbaz, L. Goh, F. Vallegra, R. Bouffanais, and D. K. P. Yue, "Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies," *Autonomous Robots*, vol. 42, pp. 1669–1689, Feb 2018.
- [83] F. Vallegra, D. Mateo, G. Tokić, R. Bouffanais, and D. K. P. Yue, "Gradual collective upgrade of a swarm of autonomous buoys for dynamic ocean monitoring," in *2018 MTS/IEEE OCEANS, Charleston*. Charleston, S.C.: IEEE, Oct 2018, pp. 1–7.
- [84] K. Cesare, R. Skeeel, S.-H. Yoo, Y. Zhang, and G. Hollinger, "Multi-uav exploration with limited communication and battery," in *Proc. IEEE Int. Conf. Robotics Autom. (ICRA)*, 2015, pp. 2230–2235.

- [85] T. Abukhalil, M. Patil, and T. Sobh, "A comprehensive survey on decentralized modular swarm robotic systems and deployment environments," *Int. J. Eng.*, vol. 7, no. 2, p. 44, 2013.
- [86] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.
- [87] A. G. Ozkil, Z. Fan, J. Xiao, S. Dawids, J. K. Kristensen, and K. H. Christensen, "Mapping of multi-floor buildings: A barometric approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 847–852.
- [88] M. Karg, K. M. Wurm, C. Stachniss, K. Dietmayer, and W. Burgard, "Consistent mapping of multistory buildings by introducing global constraints to graph-based slam," in *Proc. IEEE Int. Conf. Robotics Automat. (ICRA)*, 2010, pp. 5383–5388.
- [89] J. A. Delmerico, J. J. Corso, D. Baran, and P. David, "Toward autonomous multi-floor exploration: Ascending stairway localization and modeling," Army Research Lab Adelphi MD, Computational And Information Sciences Directorate, Tech. Rep., 2013.
- [90] J. A. Delmerico, J. J. Corso, D. Baran, P. David, and J. Ryde, "Ascending stairway modeling: A first step toward autonomous multi-floor exploration," in *2012 IEEE/RSJ Int. Conf. Intell. Robots Sys. (IROS)*. IEEE, 2012, pp. 4806–4807.
- [91] G. J. Wenger, A. M. Johnson, C. J. Taylor, and D. E. Koditschek, "Semi-autonomous exploration of multi-floor buildings with a legged robot," in *Unmanned Systems Technology XVII*, vol. 9468, 2015, p. 94680B.
- [92] J.-G. Kang, S.-Y. An, and S.-Y. Oh, "Navigation strategy for the service robot in the elevator environment," in *2007 Int. Conf. Control Automat. Sys. (ICCAS)*. IEEE, 2007, pp. 1092–1097.
- [93] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *2011 IEEE Int. Conf. Robotics Automat. (ICRA)*. IEEE, 2011, pp. 20–25.
- [94] H. H. Hariri, D. C. Y. Koh, H. C. Lim, A. G. Dharmawan, V. D. Nguyen, G. S. Soh, S. Foong, R. Bouffanais, H. Y. Low, and K. L. Wood, "ORION-II: A miniature climbing robot with bilayer compliant tape for autonomous intelligent surveillance and reconnaissance," in *15th IEEE Int. Conf. Contr Automat. Rob. Vis. (ICARCV)*, Nov 2018, pp. 1621–1626.
- [95] A. G. Dharmawan, P. Xavier, D. Anderson, K. B. Perez, H. H. Hariri, G. S. Soh, A. Baji, R. Bouffanais, S. Foong, H. Y. Low, and K. L. Wood, "A bio-inspired miniature climbing robot with bilayer dry adhesives: Design, modeling, and experimentation," in *ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. (IDETC/CIE)*. American Society of Mechanical Engineers, 2018, p. V05BT07A036, (Best Paper Award).

- [96] D. C. Y. Koh, A. G. Dharmawan, H. H. Hariri, G. S. Soh, S. Foong, R. Bouffanais, H. Y. Low, and K. L. Wood, "Design and analysis of a miniature two-wheg climbing robot with robust internal and external transitioning capabilities," in *IEEE Int. Conf. Rob. Automat. (ICRA)*. IEEE, May 2019, in Press.
- [97] A. G. Dharmawan, P. Xavier, H. H. Hariri, G. S. Soh, A. Baji, R. Bouffanais, S. Foong, H. Y. Low, and K. L. Wood, "Design, modeling and experimentation of a bio-inspired miniature climbing robot with bilayer dry adhesives," *J. Mech. Rob.*, vol. 11, p. 020902, Feb 2019.
- [98] A. G. Dharmawan, D. C. Y. Koh, G. S. Soh, S. Foong, R. Bouffanais, and K. L. Wood, "Tail design of a miniature two-wheg climbing robot for external transitioning," in *Advances in Mechanism and Machine Science*, T. Uhl, Ed. Cham: Springer International Publishing, 2019, pp. 2139–2148.
- [99] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. P. Yue, "Swarm-enabling technology for multi-robot systems," *Front. Robot. AI*, vol. 4, p. 12, 2017.
- [100] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.
- [101] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. 2nd Int. Conf. Auton. Agents*, vol. 98, 1998, pp. 47–53.
- [102] R. Bouffanais, "YouTube video – Multi-Robot System ORION: Collective Exploration of Unknown and Dynamic Environments," <https://youtu.be/GUvKQzOO48Y>, 2019, [Online; accessed 08-April-2019].
- [103] H. L. Kwa, J. Leong Kit, and R. Bouffanais, "Balancing collective exploration and exploitation in multi-agent and multi-robot systems: A review," *Frontiers in Robotics and AI*, p. 438.
- [104] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.
- [105] A. Rehman, M. M. Rathore, A. Paul, F. Saeed, and R. W. Ahmad, "Vehicular traffic optimisation and even distribution using ant colony in smart city environment," *IET Intelligent Transport Systems*, vol. 12, pp. 594–601(7), September 2018.
- [106] U. Madhushani and N. E. Leonard, "Heterogeneous explore-exploit strategies on multi-star networks," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1603–1608, 2020.
- [107] A. Quattrini Li, "Exploration and mapping with groups of robots: Recent trends," *Current Robotics Reports*, 2020.
- [108] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekerçioğlu, and M. Sende, "Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends," *Swarm and Evolutionary Computation*, vol. 60, no. 100762, 2021.

- [109] O. Berger-Tal, J. Nathan, E. Meron, and D. Saltz, "The exploration-exploitation dilemma: A multidisciplinary framework," *PLoS ONE*, vol. 9, no. 4, 2014.
- [110] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, 2016.
- [111] R. Azoulay-Schwartz, S. Kraus, and J. Wilkenfeld, "Exploitation vs. exploration: Choosing a supplier in an environment of incomplete information," *Decision Support Systems*, vol. 38, no. 1, pp. 1–18, 2004.
- [112] K. Mehlhorn, B. R. Newell, P. M. Todd, M. D. Lee, K. Morgan, V. A. Braithwaite, D. Hausmann, K. Fiedler, and C. Gonzalez, "Unpacking the exploration-exploitation tradeoff: A synthesis of human and animal literatures," *Decision*, vol. 2, no. 3, pp. 191–215, 2015.
- [113] C. Stadler, T. Rajwani, and F. Karaba, "Solutions to the exploration/exploitation dilemma: Networks as a new level of analysis," *International Journal of Management Reviews*, vol. 16, no. 2, pp. 172–193, 2014.
- [114] L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.
- [115] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, no. 36, 2020.
- [116] P. G. F. Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, "Swarm robotics: A perspective on the latest reviewed concepts and applications," *Sensors*, vol. 21, no. 6, p. 2062, 2021.
- [117] A. Khan, B. Rinner, and A. Cavallaro, "Multiscale observation of multiple moving targets using micro aerial vehicles," in *IEEE International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015, pp. 4642–4649.
- [118] K. Okumura, Y. Tamura, and X. Défago, "Amoeba exploration: Coordinated exploration with distributed robots," in *2018 9th International Conference on Awareness Science and Technology, iCAST 2018*. Fukuoka, Japan: IEEE, 2018, pp. 191–195.
- [119] J. T. Ebert, M. Gauci, and R. Nagpal, "Multi-feature collective decision making in robot swarms," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, Stockholm, Sweden, 2018, pp. 1711–1719.
- [120] J. T. Ebert, M. Gauci, F. Mallmann-trenn, and R. Nagpal, "Bayes bots : Collective bayesian decision-making in decentralized robot swarms," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 7186–7192. [Online]. Available: <https://ssr.seas.harvard.edu/files/ssr/files/icra2020-ebert.pdf>
- [121] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

- [122] G. Valentini, E. Ferrante, and M. Dorigo, "The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives," *Frontiers in Robotics and AI*, vol. 4, p. 9, 2017.
- [123] R. Takahashi, M. Takimoto, and Y. Kambayashi, "Cooperatively transporting unknown objects using mobile agents." in *ICAART (2)*, 2014, pp. 60–68.
- [124] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. P. Yue, "Swarm-enabling technology for multi-robot systems," *Frontiers in Robotics and AI*, vol. 4, p. 12, 2017. [Online]. Available: <https://doi.org/10.3389/frobt.2017.00012>
- [125] D. Shishika and D. A. Paley, "Mosquito-inspired distributed swarming and pursuit for cooperative defense against fast intruders," *Autonomous Robots*, vol. 43, no. 7, pp. 1781–1799, 2019.
- [126] A. Kamimura and T. Ohira, "Group chase and escape," *New Journal of Physics*, vol. 12, no. 5, p. 053013, 2010.
- [127] K. Shah and M. Schwager, "Multi-agent cooperative pursuit-evasion strategies under uncertainty," in *Distributed Autonomous Robotic Systems. Springer Proceedings in Advanced Robotics*, vol 9, Boulder, CO, USA, 2019, pp. 451–468.
- [128] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Optimal swarm strategy for dynamic target search and tracking," in *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, Auckland, New Zealand, 2020, pp. 672–680.
- [129] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.
- [130] G. Francesca and M. Birattari, "Automatic design of robot swarms: Achievements and challenges," *Frontiers Robotics AI*, vol. 3, 2016.
- [131] J.-B. Mouret and K. Chatzilygeroudis, "20 years of reality gap: a few thoughts about simulators in evolutionary robotics," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2017, pp. 1121–1124.
- [132] J. Yang, X. Wang, and P. Bauer, "Extended PSO based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76 328–76 341, 2019.
- [133] L. Matignon and O. Simonin, "Multi-robot simultaneous coverage and mapping of complex scene - Comparison of different strategies," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 1, Stockholm, Sweden, 2018, pp. 559–567.
- [134] C. Coquet, C. Aubry, A. Arnold, and P.-J. Bouvet, "A local charged particle swarm optimization to track an underwater mobile source," in *OCEANS 2019 - Marseille*. Marseille, France: IEEE, 2019.

- [135] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 865–875, 2015.
- [136] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek, "Group chasing tactics: How to catch a faster prey," *New J. Phys.*, vol. 19, 2017.
- [137] V. A. Kazakova, A. S. Wu, and G. R. Sukthankar, *Respecializing swarms by forgetting reinforced thresholds*. Springer US, 2020, vol. 14, no. 3.
- [138] A. R. Jordehi, "Particle swarm optimisation for dynamic optimisation problems: A review," *Neural Comput. Appl.*, vol. 25, pp. 1507–1516, 2014.
- [139] L. Pitonakova, R. Crowder, and S. Bullock, "Information exchange design patterns for robot swarm foraging and their application in robot control algorithms," *Frontiers in Robotics and AI*, vol. 5, no. 47, 2018.
- [140] H. D. Phan, K. Ellis, J. C. Barca, and A. Dorin, "A survey of dynamic parameter setting methods for nature-inspired swarm intelligence algorithms," *Neural Computing and Applications*, vol. 32, no. 2, pp. 567–588, 2020.
- [141] B. J. Leonard, A. P. Engelbrecht, and A. B. van Wyk, "Heterogeneous particle swarms in dynamic environments," in *2011 IEEE Symposium on Swarm Intelligence*. Paris, France: IEEE, 2011.
- [142] P. Ghassemi and S. Chowdhury, "Decentralized informative path planning with balanced exploration-exploitation for swarm robotic search," in *Proceedings of the ASME Design Engineering Technical Conference*, Anaheim, CA, USA, 2019.
- [143] M. Senanayake, I. Senthooan, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.robot.2015.08.010>
- [144] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Tracking multiple fast targets with swarms : Interplay between social interaction and agent memory," in *ALIFE 2021: The 2021 Conference on Artificial Life*, 2021.
- [145] A. K. Gupta, K. G. Smith, and C. E. Shalley, "The interplay between exploration and exploitation," *Academy of management journal*, vol. 49, no. 4, pp. 693–706, 2006.
- [146] J. Nauta, P. Simoens, and Y. Khaluf, "Memory induced aggregation in collective foraging," in *ANTS2020, The 13th International Conference on Swarm Intelligence.*, M. Dorigo, T. Stützle, M. J. Blesa, C. Blum, H. Hamann, M. K. Heinrich, and V. Strobel, Eds. Barcelona, Spain: Springer International Publishing, 2020, pp. 176–189.
- [147] A. Wolek, S. Cheng, D. Goswami, and D. A. Paley, "Cooperative mapping and target search over an unknown occupancy graph using mutual information," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1071–1078, 2020.
- [148] M. Crosscombe and J. Lawry, "The impact of network connectivity on collective learning," in *Proceedings of the 15th International Symposium on*

- Distributed Autonomous Robotics Systems (DARS21)*, 2021. [Online]. Available: <http://arxiv.org/abs/2106.00655>
- [149] Z. Liu, M. Crosscombe, and J. Lawry, "Imprecise Fusion Operators for Collective Learning," in *ALIFE 2021: The 2021 Conference on Artificial Life*, 2021.
- [150] A. Tversky and W. Edwards, "Information versus reward in binary choices," *Journal of Experimental Psychology*, vol. 71, no. 5, p. 680, 1966.
- [151] D. J. Navarro, B. R. Newell, and C. Schulze, "Learning and choosing in an uncertain world: An investigation of the explore-exploit dilemma in static and dynamic environments," *Cognitive psychology*, vol. 85, pp. 43–77, 2016.
- [152] N. Palmieri and S. Marano, "Discrete firefly algorithm for recruiting task in a swarm of robots," in *Studies in Computational Intelligence*, 2016, vol. 637, pp. 133–150.
- [153] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, 2017.
- [154] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459–472, 2006.
- [155] S. Cheng, Y. Shi, Q. Qin, Q. Zhang, and R. Bai, "Population diversity maintenance in brain storm optimization algorithm," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 4, no. 2, pp. 83–97, 2015.
- [156] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "On the exploration and exploitation in popular swarm-based metaheuristic algorithms," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7665–7683, 2019.
- [157] Z.-h. Zhan, J. Zhang, Y. Li, and H. S.-h. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [158] M. Dadgar, M. S. Couceiro, and A. Hamzeh, "RDPSO diversity enhancement based on repulsion between similar ions for robotic target searching," in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*. Shiraz, Iran: IEEE, 2017, pp. 275–280.
- [159] —, "RbRDPSO: repulsion-based RDPSO for robotic target searching," *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*, vol. 44, no. 1, pp. 551–563, 2020. [Online]. Available: <https://doi.org/10.1007/s40998-019-00245-z>
- [160] H. Sardinha, M. Dragone, and P. A. Vargas, "Combining lévy walks and flocking for cooperative surveillance using aerial swarms," in *17th European Conference on Multi-Agent Systems 2020*. Thessaloniki, Greece: Springer International Publishing, 2020.

- [161] L. Esterle and P. R. Lewis, "Online multi-object k-coverage with mobile smart cameras," in *Proceedings of the 11th International Conference on Distributed Smart Cameras*, 2017, pp. 107–112.
- [162] L. Esterle and P. Lewis, "Distributed autonomy and trade-offs in online multiobject k-coverage," *Computational Intelligence*, vol. 36, no. 2, pp. 720–742, 2020.
- [163] M. Frasheri, L. Esterle, and A. V. Papadopoulos, "Cooperative multi-agent systems for the multi-target κ -coverage problem," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020*. Valletta, Malta: Springer International Publishing, 2020, pp. 106–131.
- [164] M. Stogiannos, A. Alexandridis, and H. Sarimveis, "An enhanced decentralized artificial immune-based strategy formulation algorithm for swarms of autonomous vehicles," *Applied Soft Computing Journal*, vol. 89, p. 106135, 2020. [Online]. Available: <https://doi.org/10.1016/j.asoc.2020.106135>
- [165] J. M. Hereford and M. A. Siebold, "Bio-inspired search strategies for robot swarms," in *Swarm Robotics from Biology to Robotics*, Ester Martinez Martin, Ed. IntechOpen, 2010.
- [166] S. Meyer-Nieberg, E. Kropat, S. Pickl, and A. Bordetsky, "Intercepting a target with sensor swarms," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, Wailea, Maui, HI, USA, 2013, pp. 1222–1230.
- [167] J. Lv, M. Liu, H. Zhao, and B. Li, "Maritime static target search based on particle swarm algorithm," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Y. Bi, S. Kapoor, and R. Bhatia, Eds., vol. 16. London, UK: Springer Cham, 2016, pp. 917–927.
- [168] S. Meyer-Nieberg, "Coordinating a team of searchers: of ants, swarms, and slime molds," in *Proceedings of GECCO '17*. Berlin, Germany: ACM, 2017, pp. 27–34.
- [169] X.-X. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robotics and Autonomous Systems*, vol. 112, pp. 123–136, 2019.
- [170] C. Coquet, A. Arnold, and P.-J. Bouvet, "Control of a robotic swarm formation to track a dynamic target with communication constraints : analysis and simulation," *Applied Sciences*, vol. 11, no. 3179, 2021.
- [171] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.
- [172] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymski, and K. Crailsheim, "Get in touch: Cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, pp. 133–155, 2009.
- [173] M. Wahby, J. Petzold, C. Eschke, T. Schmickl, and H. Hamann, "Collective change detection: Adaptivity to dynamic swarm densities and light conditions in robot

- swarms," in *Artificial Life Conference Proceedings*. Newcastle, United Kingdom: MIT Press, 2019, pp. 642–649.
- [174] Q. Tang, F. Yu, Z. Xu, and P. Eberhard, "Swarm robots search for multiple targets," *IEEE Access*, vol. 8, pp. 92 814–92 826, 2020.
 - [175] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205–1213, 2007.
 - [176] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, S. Havlin, M. G. E. Da Luz, E. P. Raposo, and H. E. Stanley, "Lévy in random searches," *Physica A*, vol. 282, pp. 1–12, 2000.
 - [177] G. M. Viswanathan, E. P. Raposo, and M. G. da Luz, "Lévy flights and superdiffusion in the context of biological encounters and random searches," *Physics of Life Reviews*, vol. 5, no. 3, pp. 133–150, 2008.
 - [178] O. Zedadra, A. Guerrieri, H. Seridi, and G. Fortino, "A lévy walk and firefly based multi-robots foraging algorithm," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer International Publishing, 2019, vol. 11874 LNCS, pp. 213–222. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-34914-1_{ }21
 - [179] J. Nauta, Y. Khaluf, and P. Simoens, "Hybrid foraging in patchy environments using spatial memory," *Journal of the Royal Society Interface*, vol. 17, no. 166, 2020.
 - [180] J. Nauta, S. Van Havermaet, P. Simoens, and Y. Khaluf, "Enhanced foraging in robot swarms using collective levy walks," in *24th European Conference on Artificial Intelligence - ECAI 2020*, Santiago de Compostela, Spain, 2020.
 - [181] K. Harikumar, J. Senthilnath, and S. Sundaram, "Multi-UAV oxyrrhis marina-inspired search and dynamic formation control for forest firefighting," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 863–873, 2019.
 - [182] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A swarm robotic exploration strategy based on an improved random walk method," *Journal of Robotics*, vol. 2019, 2019.
 - [183] R. K. Ramachandran, "Exploration, mapping and scalar field estimation using a swarm of resource-constrained robots," PhD Thesis, Arizona State University, 2018.
 - [184] M. Kegeleirs, D. G. Ramos, and M. Birattari, "Random walk exploration for swarm mapping," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11650 LNAI, pp. 211–222, 2019.
 - [185] R. K. Ramachandran, Z. Kakish, and S. Berman, "Information correlated lévy walk exploration and distributed mapping using a swarm of robots," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1422–1441, 2020.

- [186] A. Falcón-Cortés, D. Boyer, and G. Ramos-Fernández, "Collective learning from individual experiences and information transfer during group foraging," *Journal of the Royal Society Interface*, vol. 16, no. 151, 2019.
- [187] J. Prasetyo, G. De Masi, P. Ranjan, and E. Ferrante, "The best-of-n problem with dynamic site qualities: Achieving adaptability with stubborn individuals," *Swarm Intelligence. ANTS 2018. Lecture Notes in Computer Science*, vol. 11172, 2018.
- [188] J. Prasetyo, G. De Masi, and E. Ferrante, "Collective decision making in dynamic environments," *Swarm Intelligence*, vol. 13, pp. 217–243, 2019.
- [189] J. de Lope, D. Maravall, and Y. Quiñonez, "Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems," *Neurocomputing*, vol. 163, pp. 47–55, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.neucom.2014.08.094>
- [190] W. Lee and D. E. Kim, "Adaptive approach to regulate task distribution in swarm robotic systems," *Swarm and Evolutionary Computation*, vol. 44, pp. 1108–1118, 2019. [Online]. Available: <https://doi.org/10.1016/j.swevo.2018.11.005>
- [191] W. Lee, N. Vaughan, and D. Kim, "Task allocation into a foraging task with a series of subtasks in swarm robotic system," *IEEE Access*, vol. 8, pp. 107 549–107 561, 2020.
- [192] G. De Masi and E. Ferrante, "Quality-dependent adaptation in a swarm of drones for environmental monitoring," in *2020 Advances in Science and Engineering Technology International Conferences, ASET 2020*, Dubai, United Arab Emirates, 2020.
- [193] H. Hornischer, J. C. Varughese, R. Thenius, F. Wotawa, M. Füllsack, and T. Schmickl, "CIMAX: collective information maximization in robotic swarms using local communication," *Adaptive Behavior*, 2020.
- [194] H. L. Kwa, G. Tokić, R. Bouffanais, and D. K. P. Yue, "Heterogeneous swarms for maritime dynamic target search and tracking," in *Global OCEANS 2020: Singapore-U.S Gulf Coast*. IEEE, 2020.
- [195] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Tailoring exploration and exploitation in multi-agent systems with short-term memory and limited social interaction," in *2021 International Workshop on Agent-Based Modelling of Human Behaviour (ABMHuB)*, 2021. [Online]. Available: http://abmhub.cs.ucl.ac.uk/2021/camera_ready/Kwa_etal.pdf
- [196] J. Banfi, J. Guzzi, A. Giusti, L. Gambardella, and G. A. Di Caro, "Fair multi-target tracking in cooperative multi-robot systems," in *Proceedings - IEEE International Conference on Robotics and Automation*, Seattle, Washington, USA, 2015, pp. 5411–5418.
- [197] P. Yan, T. Jia, and C. Bai, "Searching and tracking an unknown number of targets: A learning-based method enhanced with maps merging," *Sensors*, vol. 21, no. 1076, 2021.

- [198] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "Multi-agent foraging: state-of-the-art and research challenges," *Complex Adaptive Systems Modeling*, vol. 5, no. 1, 2017.
- [199] S. Razali, Q. Meng, and S. H. Yang, "A refined immune systems inspired model for multi-robot shepherding," in *Proceedings - 2010 2nd World Congress on Nature and Biologically Inspired Computing, NaBIC 2010*, Kitakyushu, Japan, 2010, pp. 473–478.
- [200] S. Razali, Q. Meng, and S. Yang, "Immune-inspired cooperative mechanism with refined low-level behaviors for multi-robot shepherding," *International Journal of Computational Intelligence and Applications*, vol. 11, no. 1, p. 125007, 2012.
- [201] 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) Congress*. Hamburg, Germany, 2015, pp. 434–440.
- [202] O. Zedadra, M. Idiri, N. Jouandeau, H. Seridi, and G. Fortino, "Lévy walk-based search strategy: Application to destructive foraging," in *Proceedings of the 2018 13th International Symposium on Programming and Systems, ISPS 2018*. Algiers, Algeria: IEEE, 2018.
- [203] S. Li, L. Li, G. Lee, and H. Zhang, "A hybrid search algorithm for swarm robots searching in an unknown environment," *PLoS ONE*, vol. 9, no. 11, p. e111970, 2014.
- [204] G. Y. Zhang, J. C. Zeng, and S. D. Xue, "Research on task allocation of multi-target search with swarm robots," *International Journal of Wireless and Mobile Computing*, vol. 7, no. 3, pp. 297–304, 2014.
- [205] A. Kolling and S. Carpin, "Multirobot cooperation for surveillance of multiple moving targets - A new behavioral approach," in *Proceedings - IEEE International Conference on Robotics and Automation*. Orlando, FL: IEEE, 2006, pp. 1311–1316.
- [206] —, "Cooperative observation of multiple moving targets: an algorithm and its formalization," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 935–953, 2007.
- [207] D. W. Lee and K. B. Sim, "Artificial immune network-based cooperative control in collective autonomous mobile robots," in *Robot and Human Communication - Proceedings of the IEEE International Workshop*, Sendai, Japan, 1997, pp. 58–63.
- [208] J. H. Jun, D. W. Lee, and K. B. Sim, "Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, Tokyo, Japan, 1999, pp. 614–619.
- [209] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 187–204, 1986.

- [210] L. N. De Castro, L. N. Castro, and J. Timmis, *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media, 2002.
- [211] F. Liu, Q. Wang, and X. Gao, "Survey of artificial immune system," in *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, China, 2006.
- [212] D. Dasgupta, *Artificial immune systems and their applications*. Springer Science & Business Media, 2012.
- [213] A. Raza and B. R. Fernandez, "Immuno-inspired robotic applications: A review," in *Applied Soft Computing Journal*, vol. 37. Seoul, South Korea: Elsevier B.V., 2015, pp. 490–505.
- [214] S. J. Sun, D. W. Lee, and K. B. Sim, "Artificial immune-based swarm behaviors of distributed autonomous robotic systems," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3993–3998, 2001.
- [215] N. D. G. Sánchez, P. A. Vargas, and M. S. Couceiro, "A darwinian swarm robotics strategy applied to underwater exploration," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–6.
- [216] R. Thenius, D. Moser, J. C. Varughese, S. Kernbach, I. Kuksin, O. Kernbach, E. Kuksina, N. Mišković, S. Bogdan, T. Petrović, A. Babić, F. Boyer, V. Lebastard, S. Bazeille, G. W. Ferrari, E. Donati, R. Pelliccia, D. Romano, G. J. van Vuuren, C. Stefanini, M. Morgantini, A. Campo, and T. Schmickl, "subCULTron - Cultural development as a tool in underwater robotics," in *Artificial Life and Intelligent Agents*, vol. 732. Birmingham, UK: Springer International Publishing, 2016, pp. 27–41.
- [217] E. Donati, G. J. van Vuuren, K. Tanaka, D. Romano, T. Schmickl, and C. Stefanini, "aMussels: Diving and anchoring in a new bio-inspired under-actuated robot class for long-term environmental exploration and monitoring," in *Towards Autonomous Robotic Systems. TAROS 2017. Lecture Notes in Computer Science*, Y. Gao, S. Fallah, Y. Jin, and C. Lekakou, Eds. Guildford, UK: Springer Cham, 2017.
- [218] H. Hamann, *Swarm Robotics: A Formal Approach*, 2018. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-74528-2>
- [219] J. C. Varughese, R. Thenius, P. Leitgeb, F. Wotawa, and T. Schmickl, "A model for bio-inspired underwater swarm robotic exploration," *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 385–390, 2018.
- [220] A. Carlisle and G. Dozier, "Adapting particle swarm optimization to dynamic environments," *International Conference on Artificial Intelligence*, pp. 429–434, 2000.
- [221] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: Detection and response to dynamic systems," in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, Honolulu, HI, USA, 2002, pp. 1666–1670.

- [222] T. M. Blackwell and P. J. Bentley, "Dynamic search with charged swarms," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 2002, pp. 19–26.
- [223] T. Blackwell, "Particle swarm optimization in dynamic environments," in *Evolutionary computation in dynamic and uncertain environments*. Springer, 2007, pp. 29–49.
- [224] C. Chira, C. M. Pintea, and D. Dumitrescu, "Multi-population agent search: Stigmergy and heterogeneity," in *Proceedings of the 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2008*, Timisoara, Romania, 2008, pp. 256–261.
- [225] C.-M. Pintea, C. Chira, and D. Dumitrescu, "Sensitive ants: Inducing diversity in the colony," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*. Springer, 2009, pp. 15–24.
- [226] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, "Ensuring ad hoc connectivity in distributed search with robotic darwinian particle swarms," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 284–289.
- [227] —, "A novel multi-robot exploration approach based on particle swarm optimization algorithms," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 327–332.
- [228] M. S. Couceiro, C. M. Figueiredo, R. P. Rocha, and N. M. Ferreira, "Darwinian swarm exploration under communication constraints: Initial deployment and fault-tolerance assessment," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 528–544, 2014.
- [229] M. S. Couceiro and D. Portugal, "Swarming in forestry environments: collective exploration and network deployment," *Swarm Intell. Princ. Curr. Algorith. Methods*, vol. 119, p. 323, 2018.
- [230] P. Wang and M. Meghjani, "Lost at sea: Multi-searcher multi-target search," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*. IEEE, 2020.
- [231] Y. Yan, R. Zhang, J. Wang, and J. Li, "Modified PSO algorithms with "request and reset " for leak source localization using multiple robots," *Neurocomputing*, vol. 292, pp. 82–90, 2018.
- [232] Y. Wang and A. Cavallaro, "Prioritized target tracking with active collaborative cameras," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2016*. Colorado Springs, CO, USA: IEEE, 2016, pp. 131–137.
- [233] M. Schranz and T. Andre, "Towards resource-aware hybrid camera systems," in *International Conference on Distributed Smart Cameras (ICDSC '18)*, Eindhoven, Netherlands, 2018.

- [234] X. Cao and C. Sun, "A potential field-based PSO approach to multi-robot cooperation for target search and hunting," *At-Automatisierungstechnik*, vol. 65, no. 12, pp. 878–887, 2017.
- [235] X. Cao and C. Sun, "Multi-auv cooperative target hunting based on improved potential field in underwater environment," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2018, pp. 118–122.
- [236] J. Ni and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Transactions on Neural Networks*, vol. 22, pp. 2062–2077, 2011.
- [237] J. Ni, L. Yang, P. Shi, and C. Luo, "An improved DSA-based approach for multi-AUV cooperative search," *Computational Intelligence and Neuroscience*, vol. 2018, no. 2186574, 2018.
- [238] H. Ge, G. Chen, and G. Xu, "Multi-AUV cooperative target hunting based on improved potential field in a surface-water environment," *Applied Sciences*, vol. 8, no. 6, p. 973, 2018.
- [239] A. M. Farid, S. Egerton, J. C. Barca, and M. A. S. Kamal, "Search and tracking in 3D space using a species based particle swarm optimizer," in *Proceedings - 2018 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2018*. Shah Alam, Malaysia: IEEE, 2018, pp. 155–160.
- [240] T. G. Karimpanal and R. Bouffanais, "Self-organizing maps for storage and transfer of knowledge in reinforcement learning," *Adaptive Behavior*, vol. 27, no. 2, pp. 111–126, 2019.
- [241] J. Tillett, T. Rao, F. Sahin, and R. Rao, "Darwinian particle swarm optimization," in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, Pune, India, 2005.
- [242] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.
- [243] F. Vallegra, D. Mateo, G. Tokić, R. Bouffanais, and D. K. P. Yue, "Gradual collective upgrade of a swarm of autonomous buoys for dynamic ocean monitoring," in *IEEE-MTS OCEANS 2018*, Charleston, SC, USA, 2018.
- [244] B. M. Zoss, D. Mateo, Y. K. Kuan, G. Tokić, M. Chamanbaz, L. Goh, F. Vallegra, R. Bouffanais, and D. K. P. Yue, "Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies," *Autonomous Robots*, vol. 42, no. 8, pp. 1669–1689, 2018.
- [245] N. Ganganath, C. T. Cheng, and C. K. Tse, "Distributed antiflocking algorithms for dynamic coverage of mobile sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1795–1805, 2016.

- [246] M. Schumer and K. Steiglitz, "Adaptive step size random search," *IEEE Transactions on Automatic Control*, vol. 13, no. 3, pp. 270–276, 1968.
- [247] S. K. Azad and O. Hasançebi, "An elitist self-adaptive step-size search for structural design optimization," *Applied Soft Computing*, vol. 19, pp. 226–235, 2014.
- [248] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [249] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.
- [250] M. Bodi, R. Thenius, M. Szopek, T. Schmickl, and K. Crailsheim, "Interaction of robot swarms using the honeybee-inspired control algorithm BEECLUST," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 18, no. 1, pp. 87–100, 2012.
- [251] J. Hereford, "BEECLUST swarm algorithm: Analysis and implementation using a markov chain model," *International Journal of Innovative Computing and Applications*, vol. 5, no. 2, pp. 115–124, 2013.
- [252] M. Bodi, C. Möslinger, R. Thenius, and T. Schmickl, "BEECLUST used for exploration tasks in autonomous underwater vehicles," *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 819–824, 2015.
- [253] D. Kengyel, P. Zahadat, T. Kunzfeld, and T. Schmickl, "Collective decision making in a swarm of robots: How robust the BEECLUST algorithm performs in various conditions," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 264–271.
- [254] H. Sardinha, M. Dragone, and P. A. Vargas, "Towards an adaptive lévy walk using artificial endocrine systems," in *ADAPTIVE 2020: The Twelfth International Conference on Adaptive and Self-Adaptive Systems and Applications*, Nice, France, 2020, pp. 116–121.
- [255] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A PSO-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: Theory, simulation and measurement," *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 37–51, 2007.
- [256] D. Sutanty, P. Levi, C. Möslinger, and M. Read, "Collective-adaptive lévy flight for underwater multi-robot exploration," in *2013 IEEE International Conference on Mechatronics and Automation*. Takamatsu, Japan: IEEE, 2013, pp. 456–462.
- [257] S. Zhang, M. Liu, X. Lei, and Y. Huang, "Dynamics and motion patterns of a k-capture game with attraction-repulsion interaction," *EPL (Europhysics Letters)*, vol. 128, no. 1, p. 10003, 2019.
- [258] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," in *Proceedings of International Conference on Robotics and Automation*, vol. 3. IEEE, 1997, pp. 2082–2089.

- [259] L. E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets," *Autonomous robots*, vol. 12, no. 3, pp. 231–255, 2002.
- [260] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Distributed odor source localization," *IEEE Sensors Journal*, vol. 2, no. 3, pp. 260–271, 2002.
- [261] V. Gazi and K. M. Passino, "A class of attractions/repulsion functions for stable swarm aggregations," *International Journal of Control*, vol. 77, no. 18, pp. 1567–1579, 2004.
- [262] Z. Chen, H. Liao, and T. Chu, "Clustering in multi-agent swarms via medium-range interaction," *EPL (Europhysics Letters)*, vol. 96, no. 4, p. 40015, 2011.
- [263] M. E. Cates and J. Tailleur, "Motility-induced phase separation," *Annual Review of Condensed Matter Physics*, vol. 6, no. 1, pp. 219–244, 2015.
- [264] C. D. Alvarenga, N. Basilico, and S. Carpin, "Multirobot patrolling against adaptive opponents with limited information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 2486–2492.
- [265] A. S. Wu, R. P. Wiegand, and R. Pradhan, "Response probability enhances robustness in decentralized threshold-based robotic swarms," *Swarm Intelligence*, vol. 14, pp. 233–258, 2020.
- [266] B. Jung and G. S. Sukhatme, "Cooperative multi-robot target tracking," in *Distributed Autonomous Robotic Systems 7*. Springer, 2006, pp. 81–90.
- [267] D. J. T. Sumpter, *Collective Animal Behavior*. Princeton University Press, 2010.
- [268] A. Perna, B. Granovskiy, S. Garnier, S. C. Nicolis, M. Labédan, G. Theraulaz, V. Fourcassié, and D. J. Sumpter, "Individual rules for trail pattern formation in argentine ants (*linepithema humile*)," *PLoS Comput Biol*, vol. 8, no. 7, p. e1002592, 2012.
- [269] I. Rausch, A. Reina, P. Simoens, and Y. Khaluf, "Coherent collective behaviour emerging from decentralised balancing of social feedback and noise," *Swarm Intelligence*, vol. 13, no. 3, pp. 321–345, 2019.
- [270] P. Rahmani, F. Peruani, and P. Romanczuk, "Flocking in complex environments—attention trade-offs in collective information processing," *PLoS Computational Biology*, vol. 16, no. 4, pp. 1–18, 2020.
- [271] D. Strömbom, "Collective motion from local attraction," *Journal of Theoretical Biology*, vol. 283, no. 1, pp. 145–151, 2011.
- [272] D. Mateo, Y. K. Kuan, and R. Bouffanais, "Effect of correlations in swarms on collective response," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.
- [273] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1272–1282, 2005.

- [274] M. A. M. De Oca, T. Stutzle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: a composite particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, 2009.
- [275] Q. Liu, W. Wei, H. Yuan, and Z.-h. Zhan, "Topology selection for particle swarm optimization," *Information Sciences*, vol. 363, no. 1, 2016.
- [276] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 689–702, 2018.
- [277] M. S. Talamali, A. Saha, J. A. Marshall, and A. Reina, "When less is more: Robot swarms adapt better to changes with constrained communication," *Science Robotics*, vol. 6, no. 56, 2021.
- [278] L. Pitonakova, R. Crowder, and S. Bullock, "Information flow principles for plasticity in foraging robot swarms," *Swarm Intelligence*, vol. 10, no. 1, pp. 33–63, 2016.
- [279] T. Kano, E. Naito, T. Aoshima, and A. Ishiguro, "Decentralized control for swarm robots that can effectively execute spatially distributed tasks," *Artificial life*, vol. 26, no. 2, pp. 242–259, 2020.
- [280] D. Mateo, N. Horsevad, V. Hassani, M. Chamanbaz, and R. Bouffanais, "Optimal network topology for responsive collective behavior," *Science Advances*, vol. 5, no. 4, p. eaau0999, 2019. [Online]. Available: <https://doi.org/10.1126/sciadv.aau0999>
- [281] Y. Shang and R. Bouffanais, "Consensus reaching in swarms ruled by a hybrid metric-topological distance," *The European Physical Journal B*, vol. 87, no. 12, pp. 1–7, 2014.
- [282] D. Lazer and A. Friedman, "The network structure of exploration and exploitation," *Administrative science quarterly*, vol. 52, no. 4, pp. 667–694, 2007.
- [283] S. Judd, M. Kearns, and Y. Vorobeychik, "Behavioral dynamics and influence in networked coloring and consensus," *Proceedings of the National Academy of Sciences*, vol. 107, no. 34, pp. 14 978–14 982, 2010.
- [284] J. Lorenz, H. Rauhut, F. Schweitzer, and D. Helbing, "How social influence can undermine the wisdom of crowd effect," *Proceedings of the national academy of sciences*, vol. 108, no. 22, pp. 9020–9025, 2011.
- [285] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. Towards New Computational Principles for Robotics and Automation*. IEEE, 1997, pp. 146–151.
- [286] A. Czirók and T. Vicsek, "Collective motion," in *Statistical Mechanics of Biocomplexity*, D. Reguera, J. Vilar, and J. Rubí, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 152–164.

- [287] J. Buhl, D. J. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, "From disorder to order in marching locusts," *Science*, vol. 312, no. 5778, pp. 1402–1406, 2006.
- [288] A. Sekunda, M. Komareji, and R. Bouffanais, "Interplay between signaling network design and swarm dynamics," *Network Science*, vol. 4, no. 2, pp. 244–265, 2016.
- [289] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, *Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution*. Springer US, 2017, vol. 11, no. 1.
- [290] E. R. Hunt, S. Jones, and S. Hauert, "Testing the limits of pheromone stigmergy in high-density robot swarms," *Royal Society open science*, vol. 6, no. 11, p. 190225, 2019.
- [291] M. Dorigo, "Optimization, learning and natural algorithms," PhD Thesis, Politecnico di Milano, 1992.
- [292] J. Ji, Z. Huang, Y. Wang, and C. Liu, "A new mechanism of pheromone increment and diffusion for solving travelling salesman problems with ant colony algorithm," in *2008 Fourth International Conference on Natural Computation*, vol. 7. IEEE, 2008, pp. 558–563.
- [293] K. Krynicki, J. Jaén, and A. Catalá, "A diffusion-based aco resource discovery framework for dynamic p2p networks," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 860–867.
- [294] V. D. C. Santos, F. E. B. Otero, C. Johnson, F. S. Osório, and C. F. M. Toledo, "Exploratory path planning for mobile robots in dynamic environments with ant colony optimization," in *Genetic and Evolutionary Computation Conference (GECCO '20)*. Cancún, Mexico: ACM, 2020, pp. 40–48.
- [295] Herianto, T. Sakakibara, and D. Kurabayashi, "Artificial pheromone system using RFID for navigation of autonomous robots," *Journal of Bionic Engineering*, vol. 4, no. 4, pp. 245–253, 2007.
- [296] Herianto and D. Kurabayashi, "Realization of an artificial pheromone system in random data carriers using RFID tags for autonomous navigation," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2288–2293.
- [297] R. Johansson and A. Saffiotti, "Navigating by stigmergy: A realization on an RFID floor for minimalistic robots," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 245–252.
- [298] A. A. Khaliq, M. Di Rocco, and A. Saffiotti, "Stigmergic algorithms for multiple minimalistic robots on an RFID floor," *Swarm Intelligence*, vol. 8, no. 3, pp. 199–225, 2014.
- [299] A. A. Khaliq, "From ants to service robots: an exploration in stigmergy-based navigation algorithms," PhD Thesis, Örebro University, 2018.

- [300] Q. Tang, Z. Xu, F. Yu, Z. Zhang, and J. Zhang, "Dynamic target searching and tracking with swarm robots based on stigmergy mechanism," *Robotics and Autonomous Systems*, vol. 120, p. 103251, 2019.
- [301] Q. Tang, F. Yu, Y. Zhang, and J. Zhang, "A stigmergetic method based on vector pheromone for target search with swarm robots," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 32, no. 3, pp. 533–555, 2020.
- [302] G. Theraulaz and E. Bonabeau, "A Brief History of Stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97–116, 04 1999.
- [303] P.-P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis* et *cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes sociaux*, vol. 6, no. 1, pp. 41–80, 1959.
- [304] B. C. Mohan and R. Baskaran, "A survey: Ant colony optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, 2012.
- [305] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [306] E. J. Robinson, D. E. Jackson, M. Holcombe, and F. L. Ratnieks, "'no entry' signal in ant foraging," *Nature*, vol. 438, no. 7067, pp. 442–442, 2005.
- [307] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar 2013. [Online]. Available: <https://doi.org/10.1007/s11721-012-0075-2>
- [308] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2016.
- [309] M. Kouzehgar, M. Meghjani, and R. Bouffanais, "Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys," in *Global OCEANS 2020: Singapore-U.S Gulf Coast*, 2020.
- [310] S. Leonardos and G. Piliouras, "Exploration-exploitation in multi-agent learning: Catastrophe theory meets game theory," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 11 263–11 271.
- [311] K. Hasselmann, A. Ligot, J. Ruddick, and M. Birattari, "Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms," *Nature communications*, vol. 12, no. 1, pp. 1–11, 2021.
- [312] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on meta-heuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, no. 2, pp. 239–287, 2009.
- [313] M. Rath, A. Darwish, B. Pati, B. K. Pattanayak, and C. R. Panigrahi, "2 - swarm intelligence as a solution for technological problems associated with internet of things," in *Swarm Intelligence for Resource Management in*

- Internet of Things*, ser. Intelligent Data-Centric Systems, A. E. Hassanien and A. Darwish, Eds. Academic Press, 2020, pp. 21–45. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978012818287100005X>
- [314] D. Shrivastava, S. Sanyal, A. K. Maji, and D. Kandar, “Chapter 17 - bone cancer detection using machine learning techniques,” in *Smart Healthcare for Disease Diagnosis and Prevention*, S. Paul and D. Bhatia, Eds. Academic Press, 2020, pp. 175–183. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128179130000171>
- [315] S. Ranjbar, F. M. Nejad, H. Zakeri, and A. H. Gandomi, “3 - computational intelligence for modeling of asphalt pavement surface distress,” in *New Materials in Civil Engineering*, P. Samui, D. Kim, N. R. Iyer, and S. Chaudhary, Eds. Butterworth-Heinemann, 2020, pp. 79–116. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978012818961000003X>
- [316] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield *et al.*, “The grand challenges of science robotics,” *Science robotics*, vol. 3, no. 14, p. eaar7650, 2018.
- [317] X. Yao and Y. Liu, “Fast evolutionary programming.” *Evolutionary programming*, vol. 3, pp. 451–460, 1996.
- [318] J. Vesterstrom and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” in *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, vol. 2. IEEE, 2004, pp. 1980–1987.
- [319] S. P. Lim and H. Haron, “Performance comparison of genetic algorithm, differential evolution and particle swarm optimization towards benchmark functions,” in *2013 IEEE Conference on Open Systems (ICOS)*. IEEE, 2013, pp. 41–46.
- [320] S. K. Mishra, “Global optimization by differential evolution and particle swarm methods: Evaluation on some benchmark functions,” *Available at SSRN* 933827, 2006.
- [321] G. OpenAI. Gym documentation. [Online]. Available: <https://gym.openai.com/docs/>
- [322] T. Blackwell, “Swarms in dynamic environments.” in *In Genetic and Evolutionary Computation Conference*. Berlin, Heidelberg: Springer, 2003, p. 12.
- [323] E. Bonabeau, D. d. R. D. F. Marco, M. Dorigo, G. Théraulaz, G. Theraulaz *et al.*, *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999, no. 1.
- [324] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes university, engineering faculty, computer . . . , Tech. Rep., 2005.
- [325] Y. Xu, P. Fan, and L. Yuan, “A simple and efficient artificial bee colony algorithm,” *Mathematical Problems in Engineering*, vol. 2013, 2013.

- [326] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.
- [327] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on evolutionary computation*, vol. 12, no. 2, pp. 171–195, 2008.
- [328] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: an experiment with kilobots," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 185–196.
- [329] T. Hellström, "Transferability and naturalistic generalization: new generalizability concepts for social science or old wine in new bottles?" *Quality & Quantity*, vol. 42, no. 3, pp. 321–337, 2008.
- [330] R. Goeree, J. He, D. O'Reilly, J.-E. Tarride, F. Xie, M. Lim, and N. Burke, "Transferability of health technology assessments and economic evaluations: a systematic review of approaches for assessment and application," *ClinicoEconomics and outcomes research: CEOR*, vol. 3, p. 89, 2011.
- [331] S. Koos, J.-B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 122–145, 2012.
- [332] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Tracking multiple fast targets with swarms: Interplay between social interaction and agent memory," *arXiv preprint arXiv:2108.07122*, 2021.
- [333] A. R. Jordehi, "Particle swarm optimisation for dynamic optimisation problems: a review," *Neural Computing and Applications*, vol. 25, no. 7, pp. 1507–1516, 2014.
- [334] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, vol. 1. IEEE, 2001, pp. 94–100.
- [335] R. Liu, J. Li, C. Mu, L. Jiao *et al.*, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.
- [336] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement," *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 37–51, 2007.
- [337] S. B. Rosenthal, C. R. Twomey, A. T. Hartnett, H. S. Wu, and I. D. Couzin, "Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion," *Proceedings of the National Academy of Sciences*, vol. 112, no. 15, pp. 4690–4695, 2015.

- [338] Q. Liu, W. Wei, H. Yuan, Z.-H. Zhan, and Y. Li, "Topology selection for particle swarm optimization," *Information Sciences*, vol. 363, pp. 154–173, 2016.
- [339] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 689–702, 2018.
- [340] J. Pugh and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in *2007 IEEE swarm intelligence symposium*. IEEE, 2007, pp. 332–339.
- [341] J. Yang, X. Wang, and P. Bauer, "Extended pso based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76 328–76 341, 2019.