Enhancing Swarm Efficiency: Exploration-Exploitation Tuning for Fast Target Tracking

A thesis submitted
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

in Mechanical Engineering

By
Julien Philippot

2024-06-06

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
Department of Mechanical Engineering

University of Ottawa
Ottawa, Ontario, Canada K1N 6N5

Abstract

Is it possible for multi-robot systems to efficiently search and track one or more fast-moving targets? While individual robots on their own are simply unable to mount a proper pursuit of a target moving faster than they can, leveraging swarm intelligence is key to achieving this goal. In this thesis, the question of how to optimize a swarm's design to achieve such a task efficiently is raised and answered through the lens of balancing exploration— i.e., the gathering of information—and exploitation—i.e., the use of said information.

Through the combination of simple movement rules, namely repulsion between agents, promoting exploration, and attraction towards a target, which generates exploitation, we generated the decentralized operation of swarms of agents. To communicate, these agents rely on a k-nearest neighbors network, which can be adjusted to calibrate the swarm's exploration-exploitation balance. Other parameters, such as implementing short-term memory and the introduction of fast agents—i.e., heterogeneity—are also used to affect this balance with the ultimate goal of optimizing swarm performance. By adjusting the size of the search-space the swarm operates in, we vary its agent density, which is introduced as a novel method to study swarm behaviors and improve the optimization process.

The identification of density phases characterized by a swarm's performance reveals the mechanisms behind the previously discovered critical minimum and maximum densities. By exploring how to shift the 'transition' phase, where performance is maximized for the number of agents present, a framework is provided to optimize swarm performance and harness swarm intelligence. Furthermore, using metrics to describe the swarm's exploration-exploitation balance allows an optimum level of connectivity k to be found for each specific tracking scenario, revealing a path towards the design of swarms that perform over wider ranges of conditions. The replacement of agents by faster ones was also shown to be universally beneficial, although extra care must be given as the exploration-exploitation balance is affected. Thus, the development of truly adaptive strategies able to tune this balance according to local conditions, as was manually done throughout this work, is discussed as the next step toward the development of successful swarms.

Preface

This thesis presents previous results, discussions, and conclusions from the following co-authored publications:

- H. L. Kwa, J. Philippot, and R. Bouffanais, "Effect of swarm density on collective tracking performance," Swarm Intelligence, vol. 17, no. 3, pp. 253–281, 2023. [1]

  http://dx.doi.org/10.1007/s11721-023-00225-4

- H. L. Kwa, J. Philippot, and R. Bouffanais, "The impact of agent density and environmental factors on target tracking swarms," in Artificial Life Conference Proceedings 35, vol. 2023, no. 1. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info@ mit.edu, 2023, p. 88. [2]

  http://dx.doi.org/10.1162/isal_a_00584

- H. L. Kwa, V. Babineau, J. Philippot, and R. Bouffanais, "Adapting the exploration-exploitation balance in heterogeneous swarms: Tracking evasive targets," Artificial Life, vol. 29, pp. 1–16, 2022. [3]

  http://dx.doi.org/10.1162/artl_a_00390

- H. L. Kwa, J. L. Kit, N. Horsevad, J. Philippot, M. Savari, and R. Bouffanais, "Adaptivity: a path towards general swarm intelligence?" Frontiers in Robotics and AI, vol. 10, p. 1163185, 2023. [4]

  http://dx.doi.org/10.3389/frobt.2023.1163185

This work was all done within the scope of Dr. Hian Lee Kwa's PhD research. As such, he was the principal contributor, responsible for coordinating our progress. The methodology employed was developed over the years by Dr. Kwa and other members of Dr. Bouffanais's lab, as detailed in Chapter 3, and under Dr. Bouffanais's scientific supervision. The work of different authors, as well as my specific contributions, are outlined in the following paragraphs. In addition, the publications upon which each chapter is based are listed at the beginning of each chapter.

Our research on density was published in the article titled "Effect of swarm density on collective tracking performance" [1], as well as a conference paper, "The impact of agent density and environmental factors on target tracking swarms" [2]. In these publications, I worked with Dr. Kwa, under the supervision of Dr. Bouffanais. I used the previously mentioned methodology to simulate the impact of modifying swarm connectivity, swarm size, and agent memory length on the swarm density - swarm performance curve. Dr. Kwa, in addition to assisting me in the generation and analysis of these results, also developed the concepts of local density difference and exploration ratio. He also tested the system's response to variations in target parameters. I joined Dr. Kwa, who was the principal author, and Dr. Bouffanais, serving as editor, in discussions over the interpretation of these results and the drafting of our publications.

On the topic of heterogeneity, we published an article titled "Adapting the exploration-exploitation balance in heterogeneous swarms: Tracking evasive targets" [3]. For this publication, I worked with Babineau and Dr. Kwa, under the supervision of Dr. Bouffanais. Babineau and I contributed to the simulation and interpretation of data on the relationship between the introduction of fast agents, swarm density, agent memory, and swarm performance. Dr. Kwa studied the impact of the level of network connectivity on heterogeneous swarms, as well as the impact of the proportion of fast agents on swarm performance generally. He also developed other heterogeneous strategies based on different memory lengths and levels of connectivity before testing them. I joined Babineau, Dr. Kwa, who was the principal author, and Dr. Bouffanais, serving as editor, in discussions over the interpretation of these results and the drafting of our publication.

On adaptivity, we published a perspective paper titled "Adaptivity: a path towards general swarm intelligence?". For this publication, I worked with Kit, Dr. Horsevad, Savari, and Dr. Kwa, under the supervision of Dr. Bouffanais. I contributed, alongside Dr. Horsevad and Savari, to the definition of flexibility, adaptivity, general intelligence, and swarm intelligence. Dr. Kwa described the current state-of-the-art work on adaptivity, as well as the path towards general adaptivity. Kit described the need for and developed a proposal for a swarm intelligence benchmark framework. Dr. Kwa combined these different perspectives into one publication, with Dr. Bouffanais's editing support.

Acknowledgements

First, I would like to thank my supervisor, Dr. Bouffanais, for always being understanding and never wavering in your confidence in me. I would not have been able to complete this thesis without your support and guidance, and I will never forget your passion and everything you've taught me.

Thanks to Dr. Spinello and Dr. Irani for accepting to be my thesis examiners, I appreciate your time.

Special thanks to Hian Lee, I was very lucky to work with such a good researcher. I learned a lot from you, both in terms of your selflessness and talent for writing excellent research papers.

Thanks to Léo for enduring my endless storm of complaints, and the rest of the group for suffering through my terrible scheduling during these last few months. Getting to escape for D&D on Sunday nights helped me get through it, even when I was busy stressing out about this thesis.

Thanks to my parents and brothers for their love and support. I appreciate your willingness to show interest in my work, even if I could never really explain what I was working on!

Finally, thanks to my partner, Brianna, without whom I would have never gotten to the finish line. Your love and support have kept me going, and I am forever thankful to have you and Ogie by my side.

Table of Contents

Nomenclature

Λ	Exploration ratio
ρ	Agent density
Θ	Engagement ratio
Ξ	Tracking performance
EED	Exploration-Exploitation Dynamics
k	Number of neighbors
L	Search space width
M	Agent memory
N	Number of agents
$S_{\mathrm{exp}}[t]$	Agent exploratory state
T	Number of simulation time-steps
t_{mem}	Agent memory length
$v[t]$	Agent velocity
v_{max}	Maximum agent speed
$x[t]$	Agent position
CMOMMT	Cooperative Multi-Robot Observation of Multiple Moving Targets
MAS	Multi-Agent System
MIPS	Motility-Induced Phase Separation
MRS	Multi-Robot System
PSO	Particle Swarm Optimisation

List of Figures

Chapter 1

Introduction

Biomimetics, or the emulation of the natural world, has been at the inception of many complex ideas and concepts. One such concept, swarm intelligence, was discovered through the observation of the behavior of large groups of social animals, and became a new field of study. It has been defined formally time and time again, but one of its earliest definitions still holds true: swarm intelligence is the collective behavior emerging from the decentralized interactions between multiple agents [5]. The concept of emergence is key, as swarm intelligence describes the general behavior exhibited by a swarm, not the individual actions of any of its members. It's only by combining those actions through repeated interaction that a swarm can become truly intelligent, becoming greater than the sum of its parts.

Just like a flock of starlings can form a complex dance by following only three basic rules—attraction, alignment, and repulsion [6]—robots can be given very simple instructions and somehow solve complex dynamic problems. This ability to adapt to rapidly changing circumstances is one of the foundational characteristics of swarms, in addition to the lack of central control and the global distribution of locally collected information [7]. Applying these characteristics to robotic agents created the field of swarm robotics, which seeks to solve many of the problems we currently face.

When confronted with distributed tasks in large environments, like object retrieval, search and rescue, or surveillance [8], many practitioners have turned towards multi-robot systems MRS. However, endowing such MRS with swarm intelligence is an opportunity to go even further by enabling robots to perform better, especially when placed in unstructured and dynamic environments [4]. While some aspects, like modularity, which allows for performance increases and cost reductions by mixing robots of different abilities together [9], are only newly embraced by the field, three main advantages of decentralized swarming MRS are commonly accepted [10]:

- Scalability: The capacity of a system to continue functioning properly when the number of its components varies.
- Flexibility: The capacity to solve problems/perform tasks that depart from those chosen at design time.

- Robustness: The capacity to continue to work efficiently in environmental conditions different from those considered at design time.

A popular application where these benefits can be exploited is for target tracking problems. Thanks to its wide array of uses, such as search and rescue [11] and environmental monitoring [12], target tracking is a mainstay amongst the swarm robotics community. In particular, the search and track of fast targets, i.e., faster than the pursuing agents themselves, has opened up even more opportunities to exploit the potential of swarm intelligence [13]. However, a significant problem persists: how to design a swarm to optimize performance?

To accomplish this goal, practitioners have turned towards different tools and approaches, chief amongst which is the balancing of exploration and exploitation behaviors. These general concepts represent the two main actions carried out by a swarm: the gathering of information, and the utilization of the said information [14]. By identifying the existence of an optimal exploration-exploitation balance, multiple studies have confirmed that swarm performance can be maximized for a specific set of operating parameters. Although this is a far cry from the touted concept of general swarm intelligence, where a swarm is so adaptive it can respond to any environmental change without outside input [4], it offers a first foray into the benchmarking of more general problems.

In this work, two main design considerations are explored in a simulated environment for the target tracking problem: (i) swarm density, which is the density of agents within a swarm's search-space, and (ii) heterogeneity, which is the combination of agents with different abilities within a single swarm. By combining density and heterogeneous analysis with the concept of exploration-exploitation dynamics (EED), the effect of possible design decisions on the intricate balance between these two behaviors can be observed. Other design parameters, like memory, various target parameters, and swarm size, are also considered under the lenses of both analyses. This allows for the optimization of the searching and tracking strategy developed by Kwa et al. [13–16], as applied in the fast-target problem.

1.1 Outline

This thesis is structured into six chapters as follows:

- Chapter 2 introduces the topic of swarms and decentralized MRS by discussing seminal works in the literature. The problem of target tracking is also described, and the past approaches to solving this problem are presented and discussed.

- Chapter 3 describes and justifies the strategies employed to simulate swarm behavior. The metrics used to evaluate said behavior are also defined, and the methodology itself is validated through a variability analysis.

- Chapter 4 analyzes the impact of varying density on a swarm's exploration-exploitation balance. The existence of three distinct phases, representing three levels of swarm performance, is demonstrated. Other parameters are evaluated for their ability to affect the density at which these phases occur.

- Chapter 5 pushes the density analysis done previously by studying the impact of adding faster agents at varying densities. The aforementioned parameters are also investigated in conjunction with swarm heterogeneity, and different heterogeneous strategies based on these parameters are also considered.

- Chapter 6 concludes this work by summarizing important findings. The ultimate goal of swarm intelligence research, general adaptivity, is also discussed, as well as how future work stemming from these results can lead the way there.

Chapter 2

Literature Review

In this review, the origins of the topics explored in the thesis are discussed, as well as the reasoning behind the selection of the target tracking problem. Past approaches used to improve the various searching and tracking strategies are also brought into focus, leading to the specific strategy and focus set forth in the following chapters. This review is mainly compiled from previous work [1, 3], with references to previous reviews done by Kwa et al. [14, 17]. It is worth noting that additional references will be given in the following chapters in relation to some technical details specific to these chapters, while this chapter serves as a more general introduction to the concepts covered.

2.1 Multi-Robot Systems

The development of our understanding of swarm intelligence has led to its implementation on MRS, bringing about a new field in the study of robots: swarm robotics. Those swarming MRS have been trialed to accomplish a vast array of tasks, amongst which are environment classification [18, 19], area mapping [20–22], target tracking [23, 24], and area protection [25, 26]. The term swarm intelligence needs to be defined to distinguish swarming MRS from their traditional counterparts. When aspiring to do so, Şahin [27] put forth 5 criteria:

1. **Autonomous Robots**: Agents need to be autonomous and be able to both interact with their environment on their own and make their own decisions.
2. **Large Number of Robots**: Swarms need to be large and scalable, and even experimentation limited to a few robots should be made with the vision to scale up.
3. **Few Homogeneous Groups of Robots**: If different types of agents are used, it should be in a manner that leaves no more than a few homogeneous groups with multiple agents each. Too many specific pre-defined roles go against the concept of swarm robotics.
4. **Relative Incapable or Inefficient Robots**: The assigned task cannot be achievable efficiently by a single agent. The existence of cooperation with other swarm members needs to be the enabling factor to accomplish the task.
5. **Robots with Local Sensing and Communicating Capabilities**: The swarm's agents must be limited to their surroundings for both sensing and communication. This decentralizes coordination, enabling swarm intelligence.

While this was a useful endeavor that still guides the design of swarms to this day, there is still currently no universally accepted definition. By comparing and contrasting the different proposals to do so made by the swarm robotics community over the years, we defined Swarm Intelligence as follows: "Swarm Intelligence is the emergent ability of a decentralized system of agents to make the appropriate adjustments to its collective behavior, thereby allowing the system to achieve changing goals in dynamic environments" [4].

We obtain the benefits outlined earlier by applying these criteria to an MRS, thus endowing it with this elusive concept of swarm intelligence. While each of these characteristics is relevant to making a swarming MRS, decentralization, which manifests itself through the communication strategy, is what sets swarms apart [4]. Indeed, communication allows agents to coordinate, which gives rise to emergent collective behaviors, the source of a swarm's ability to accomplish more than the sum of its parts.

In the literature, communication methods can take different shapes. The most popular are predictably direct communication methods, where agents send each other messages containing information on their local environment, like positions or measurements. This is often accomplished through the use of radio communications [28]. Nonetheless, indirect communication methods, such as stigmergy, which involve agents modifying their environment by leaving behind a signal, are emerging as a valid alternative [29]. These methods are often directly inspired by the natural environment, as an agent leaving a tag behind is a direct parallel to using pheromones in an ant colony [30].

Intuitively, direct communication methods might be considered more efficient, as they provide the means to communicate easily with the entire swarm. However, careful examination has shown that in the natural world, an individual starling only follows cues from its 7 nearest neighbors on average, discarding visible signals from further neighbors [31]. Through the use of swarming MRS, it was shown that increasing the agents' number of neighbors too much detracts from the system's response, as both too many or too few neighbors can reduce the level of system response [32, 33]. The concept of actively limiting communication is integral to future discoveries, as it hints at an ideal level of communication between agents.

2.2 Target Tracking Problems

Although we are still removed from real-world applications of swarming MRS [29], the potential applicability of different swarming problems has guided research into a few particular areas. Specifically, the searching and tracking of one or many targets is one of the more popular problems in this field [17]. Indeed, target tracking has applications in civilian contexts, for example, with environmental monitoring [34], but also on defense grounds, including protecting an area from intruders [26, 35].

A few frameworks have been developed over the years to help define target tracking problems, including Parker and Emmons' Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) [36], and Huang and Tseng's k-coverage problem [37]. In such studies, agents navigate a search-space and react to a signal their neighbors communicate to alert them of a nearby target [38, 39]. In most cases, although one might argue it contradicts one of Şahin's criterion, these problems have emphasized cases with targets that are slower or just as fast as the pursuing agents. This is because tracking a target with slower agents is unfeasible, as it will always outmaneuver its followers [36].

Despite that, a new type of problem emerged when it was shown that swarms can capture a target, even if it is faster than any of its individual agents [40]. This was later showcased both using simulations and through experimental results [13, 15], as harder and harder problems were sought to evaluate the potential of different methods to maximize a swarm's overall performance. In fact, it's only by increasing the problem's difficulty so far past the ability of any single agent that we can make sure to properly measure the results of swarm intelligence, as any given agent on its own is rendered obsolete. This has guided the development of the strategy described throughout this thesis.

2.3 Past Approaches

While the Particle Swarm Optimisation (PSO) algorithm was first devised by Kenney and Eberhart [41] to optimize network weights amongst social collaborators, a few modifications enabled it to be implemented with swarming MRS as a way to solve the target tracking problem [23, 42, 43]. To achieve this conversion, some changes were made, such as limiting the communication network [23, 44], adding some kind of collision avoidance [45], and relying on non-global agent positioning [44].

One of the main challenges noted when implementing such an algorithm was using agent-based memory [17]. Indeed, the exploitation of outdated information by agents initially proved to be detrimental to a swarm's operations [13, 23, 24], until the target was upgraded with an evasion mechanism [15]. Interestingly, the quest for an increasingly complex target-tracking problem thereby brought back memory as a foremost consideration in swarm design.

Another ongoing challenge with PSO-based target tracking algorithms is the tendency of agents to over-exploit, leading to a skewed exploration-exploitation balance [17, 46]. Such an imbalance hinders agent performance in the long run, as the lack of exploration reduces the ability of the swarm to keep tracking the target past their initial encounter. Diverse approaches have been considered to alleviate this problem, but many of them, as is the case for the constant repulsion employed by Blackwell and Bentley [47], rely on centralization in one form or another.

To avoid such a pitfall, it is important to consider the concept of exploration and exploitation more generally. Although these terms are specifically applied to the target tracking problem throughout this work, they come from very general concepts that are useful in many fields. To generalize the distinction between exploration and exploitation, Melhorn et al. [48] defined three central methods:

1. **Behavioral patterns**: used when a specific behavior can be observed directly, as in a target tracking problem.
2. **Values and uncertainty**: based on reinforcement learning literature, choosing randomly represents exploration while selecting the known highest value option is exploitation.
3. **Outcomes obtained**: a behavior providing information is defined as exploration, while one done for an explicit reward is exploitation.

In the previously discussed PSO target tracking problem, feature 1 is particularly useful to characterize the system's EED, as agents observed forming a cluster or moving towards a common point can be considered to be exploiting information from the same source [15, 46]. Conversely, agents randomly moving about the search space or away from each other can be identified as exploring. This was used by Kwa et al. [13, 16] to define a state change associated with the knowledge of information. This change happens when an agent obtains a point of interest through communication or direct observation, which causes an agent to move toward the position and shift its state from exploring to exploiting. The process is repeated throughout the task until completion [14], and allows for the development of specific metrics characterizing the amount of exploration and exploitation undertaken by a swarm's agents [13, 16].

Coming back to the over-exploitation issue characteristic of PSO problems, the main control method available in such algorithms is to limit the swarm's connectivity, which can be achieved in one of three main ways, as detailed by Kwa et al. [14]: (1) reducing the number of neighbors, (2) limiting the communication range, and (3) changing the communication network itself. Of those, modifying the number of neighbors k was repeatedly shown to be key to the control of the level of exploration and exploitation [13, 15, 16, 49–52]. Indeed, increasing k allows information to spread throughout more of the swarm, promoting exploitation. On the contrary, reducing k increases exploration by reducing the spread of information, therefore alleviating over-exploitation. When implementing such control in k-nearest neighbors networks, it was found there exists an optimal value of k for which the balance of exploration and exploitation generates the best possible swarm performance

[13, 15, 16, 20, 32, 33]. Above this value, the abovementioned over-exploitation manifests itself, which decreases the overall target tracking performance.

More generally, when studying any swarming problem, the number of agents to use is one of the first considerations practitioners encounter. In most cases in the literature, an environment of fixed dimensions is used to study the size of MRS, which allows for the addition or removal of agents to either increase or decrease the swarm's agent density. Although this might seem like a simple problem of scale, a swarm's size does impact its EED, providing another window into studying the exploration-exploitation balance.

For those studying optimization, the quantity of agents is a stand-in for the system's level of exploration [53, 54]. Logically, having more agents at its disposition enables the system to explore better the search space, which reduces the odds of optimizing the algorithm for a narrow, local, optima. Returning to the natural world, the study of bee colonies has shown that a larger swarm allows for more exploration through a higher number of scouts, which leads to higher accuracy for its final consensus [55]. The practical use of MRS also confirmed this theory, as larger swarms achieved higher performance through increased exploration [20, 26, 56, 57].

Consequently, marginal utility gain generated by adding agents has been the topic of multiple studies. It's well-established in foraging problems specifically, starting with the work of Lerman and Galstyan [58], who demonstrated that as the overall foraging performance increases by adding agents, the robots' individual share of that performance decreases. Indeed, adding more agents to the operating space increases the number of collisions between agents, causing physical interference. This trend was also shown in other tasks, such as for collective area mapping [20] and target tracking [59, 60].

It must be noted that target tracking differs from other tasks like resource foraging and area mapping due to the inherent cap to performance. Once the swarm is large enough to enable a constant tracking of the target, performance ceases to improve with the addition of more targets, which hastens the decrease in individual efficiency. However, this is not restricted to target tracking, as it's also been shown that adding too many agents in multi-agent system (MAS) optimization problems results in marginal accuracy benefits while disproportionately lengthening computation times [61].

The concept of reducing efficiency with increased size is familiar to the study of economics, where it is known as the "Law of Diminishing Marginal Returns". Its application to MAS was shown by Hamann [62], who modeled it factoring in inter-agent cooperation and interference through the use of an exponential function.

Later, this approach was combined with a costing model, allowing for an optimal number of agents to be found that maximizes system performance while minimizing costs [63].

In particular, the concept of diminishing marginal returns is exacerbated by problems involving common resource exploitation. For example, it was shown that making agents deposit their foraged resources in a central location caused a further decrease in individual robot efficiency in large swarms [64], which is caused by the longer distances traveled to get to resources. Higher agent interference caused by the exploitation of common resources was also shown to be a factor, to the point of causing negative marginal utility or a net decrease in performance through the addition of agents [65]. Indeed, so much interference is generated that it can completely negate the extra exploration undertaken by the swarm.

Taking the analysis further, Hamann [66] came up with the same conclusions in a stick-pulling task, which led him to define three phases resulting from altering a swarm's size: (1) a phase where the resources are under-exploited due to a lack of collaboration, (2) a phase where too much available information causes shared resources to be depleted, and (3), an intermediate phase where optimal performance is achieved. These findings, while general, paved the way for more precise definitions of swarm density phases.

Another potential consequence of adding too many agents in a swarm is the snowball effect of clustering around points of interest, which is integral to tasks such as target tracking. This phenomenon, known as Motility-Induced Phase Separation (MIPS) [67], occurs due to the positive feedback loop created when agents aggregate into an area of high local agent density. Consequently, information is over-exploited, which results in worse overall swarm performance.

Different strategies have been employed to counteract MIPS, typically by using the local density of the swarm around individual agents. For instance, the BEECLUST algorithm, used to identify areas of high light density, was modified by enabling robots to identify the surrounding agent density and leave their position. This adaptation, by limiting clustering and increasing the level of exploration, made the swarm, as a whole, more responsive to dynamic environment changes [68]. Other examples of such strategies include controlling the distribution of a swarm's agents based on local swarm density and the strength of a signal field [69], varying the step size taken by each robot in a foraging task based on its perceived local swarm density [70], or the implementation of inter-agent repulsion in target searching and tracking tasks [13, 15, 71, 72].

Finally, heterogeneity in swarm composition is an ideal tool to use the modularity of swarming MRS. Mixing agents of different abilities has been undertaken for many distinct goals and has proven quite successful, at least theoretically. For example, Christensen et al. [73] utilized agents of different communication ranges, which effectively reduced the overall system cost, paving the way for larger-scale applications. In the case of Shi et al. [74], it was the task of the agents themselves to adapt to match their different levels of movement and sensing abilities.

Modularity also allows for the partial upgrade of a swarm's agents, thus evoking the possibility of maintaining performance with fewer agents, i.e., for a reduced overall cost [9]. This technique has shown promise in target searching and tracking tasks, both as a way to improve tracking performance and to reduce target capture times [16, 75]. However, it was demonstrated generally by Althusler [76] that upgrading certain agents in a swarm does not automatically and necessarily yield an improvement in the system's performance. This hints at a deeper alteration to the swarm's overall EED. Indeed, the modification of agent abilities to modify a swarm's exploration-exploitation balance has been discussed by Kwa et al. [14]. This provides an opportunity to study potential improvements to target tracking strategies utilizing heterogeneity in the swarm composition.

Chapter 3

Methodology

In this chapter, the strategies used to simulate the dynamics of MRS are detailed. Simple equations guide each agent's behavior based on its proximity to the target and other agents. Adding in the connectivity aspect, we generate a complex dynamic that allows the different agents to cooperate in unforeseen ways while making their own decisions. Other mechanics, like agent memory and target evasion, are implemented to increase the complexity of the problem at hand. This strategy was developed by Kwa et al. [13, 15, 16, 77] and used in our simulation work [1–3].

3.1 Search and Track

The search and track strategy used in this work was first developed by Kwa et al. [13] to model the tracking of targets by a homogeneous swarm. It was later expanded [14] to allow for variation in the maximum speed of pursuing agents, thus enabling the study of heterogeneous swarms. Agents were also endowed with short-term agent-based memory, enabling them to track evasive targets [15].

Generally, our strategy relies on the interplay of two behaviors within each agent: (1) agent aggregation towards a target, which is responsible for exploitation, and (2) inter-agent repulsion, which is adaptive and drives exploration. Each of these components generates a velocity vector every time step, which is added to create an agent's final velocity vector as per the following equation:

$$\mathbf{v}_i[t] = \mathbf{v}_{i,\mathrm{att}}[t] + \mathbf{v}_{i,\mathrm{rep}}[t], \tag{3.1}$$

where $\mathbf{v}_{i,\mathrm{att}}[t]$ is the velocity vector generated by the attractive component, $\mathbf{v}_{i,\mathrm{rep}}[t]$ is generated by the repulsive component, and t is the time-step at which each component was generated. Each agent's maximum speed, v_{max}, scales its final velocity vector $\mathbf{v}_i[t]$.

To modify the swarm's overall behavior or EED, this strategy allows the use of multiple levers: (1) agent connectivity level, (2) agent-based memory, (3) agent speed, (4) target evasion method, and (5) various other target parameters. The general strategy is described in Alg. 1, and the specific levers are detailed in the rest of this chapter.

3.2 Inter-Agent Connectivity Method

As previously mentioned, one of the core components of an MRS is its agents' communications network [78], which transforms it from a collection of robots working individually to a bonafide swarm, defined by its collective action [10]. In the literature, most research has concentrated on agents endowed with either an unlimited communications range [24, 79], or a fixed physical range, including all agents found within [72, 80, 81].

However, tuning the network's connectivity by restricting the number of agents being communicated with instead was shown to have a substantial impact on the col-

Algorithm 1 : Dynamic k-Nearest Network Search and Tracking Strategy

Set $t = 0$, $k = 15$
while System active **do**
  for All agents **do**
    Set point of attraction
    Calculate $\mathbf{v}_{i,\text{att}}[t]$
    Calculate $\mathbf{v}_{i,\text{rep}}[t]$
    $\mathbf{v}_i[t] \leftarrow \mathbf{v}_{i,\text{att}}[t] + \mathbf{v}_{i,\text{rep}}[t]$ // Apply Eq. (3.1)
    $\mathbf{v}_i[t] \leftarrow (v_{\max}/v_i[t]) \cdot \mathbf{v}_i[t]$ // Ensure magnitude of velocity vector does not exceed maximum speed
    $\mathbf{x}_i[t+1] \leftarrow \mathbf{x}_i[t] + \mathbf{v}_i[t]$ // Update agent position
  end for
  $t \leftarrow t + 1$
end while

lective dynamics of a swarm [32]. Indeed, too much communication was found to be detrimental to swarm performance when responding to a simulated predator attack. Subsequent research showed that systems with a higher level of connectivity respond better to slow-changing perturbations, while lower levels of connectivity are more fitted to fast-changing ones [33, 82]. Swarms tasked with accurately characterizing static and noisy environments behaved similarly [21, 22, 77].

Based on these conclusions, it was decided to opt for a k-nearest neighbor communication network. Here, the neighborhood is used in the sense of a topological network, where the k agents closest to any given agent are its neighbors. k thus represents the swarm's level, or degree, of connectivity. As the system studied here exists in a time-varying network, each agent's neighborhood is dynamic and redefined at every time step. It is also worth specifying that, in this work, information is transmitted perfectly with no quality degradation, regardless of the distance between agents. Furthermore, agents can only communicate a position they identified themselves, as allowing information to spread freely throughout the swarm counteracts the limitations placed on communication.

Using such a network, Kwa et al. [13, 15, 16] established the existence of an optimal level of connectivity k^*, for which a swarm tracking a fast-moving target sees its performance maximized. At this optimal level of connectivity, an MRS can achieve the perfect balance between the exploration above and the exploitation behaviors of its agents. Of course, the value of k^* depends on the swarm's specific conditions and the general system it evolves in.

3.3 Memory

A particular aspect of some swarms in the literature is agent-based memory. Its introduction increased performance in a swarm in a non-destructive foraging scenario, especially when the environment contains densely clustered resources [83, 84]. When the target is immobile, memory allows agents to return to the same resource repetitively, saving it from having to rediscover said resource over time. Predictably, the advantages of using memory disappeared when tracking a non-evasive moving target as the swarm's performance decreased [13, 23, 24]. Indeed, memory was found to cause agents to cluster around a target's past location, thereby reducing effective target tracking.

Even so, agent-based memory M, of a duration of t_{mem}, was added to the general swarm strategy by Kwa et al. [15]. It was then shown that memory is essential to improve swarm performance when tracking an evasive target. This is due to the added complexity created by the target's evasion mechanism, which renders any information, even outdated, valuable to the swarm. While agents clustering around a past location of the target seems wholly futile, it does allow them to close in on its general location, making them better able to respond when new, more accurate information is communicated. Therefore, the extra exploitative actions memory generates prove to be effective and enable agents to improve the tracking of their target.

3.4 Agent Velocity

3.4.1 Attraction Velocity

To generate the agents' velocity vector, the aforementioned attractive component is calculated based on each agent's position relative to the point of interest $\mathbf{p}[t]$, which is chosen using Algorithm 2. By driving agents to pursue the target, the attraction velocity is the main force behind exploitation. It works in conjunction with the communication protocol and memory, as they both affect how the point of attraction is chosen.

For every time step in the simulation, each agent begins in the exploratory state, scanning its local environment and looking for a target. If a target is detected, its state goes from exploratory to tracking, and $\mathbf{p}[t]$ is set to the target's location, which is then transmitted to its k-nearest neighbors (detailed further in Sec. 3.2). If no target is detected, the agent relies instead on target positions and encounter times received from its network. Agents are also endowed with a memory (detailed further in Sec. 3.3), allowing them to remember the position and time a target was found for an amount of time t_{mem}.

By comparing the values an agent receives from its neighbors and its own encounter information, the most recent location of the target can be determined and used as a point of attraction $\mathbf{p}[t]$. Therefore, the target can track its own past or present encounter or a neighbor's past or present encounter, depending on the most recent information available to it. However, in the absence of any target positions, the point of attraction is set to $x_i[t]$, which is the agent's location, thereby rendering the attractive component null.

As previously mentioned, the agents' neighborhood communication network evolves over time. Therefore, the agents' neighbors are redefined at each time step. Overall, each agent independently sets its $\mathbf{p}[t]$ using Algorithm 2.

Once the point of attraction is established, an agent's attractive velocity can be calculated. The equation uses the agent's velocity at the previous time-step and its location relative to $\mathbf{p}[t]$ as follows:

$$\mathbf{v}_{i,\text{att}}[t] = \omega \mathbf{v}_i[t-1] + cr\big(\mathbf{p}[t] - \mathbf{x}_i[t]\big). \tag{3.2}$$

This equation is based on the previously discussed Social-Only PSO model originally proposed by Engelbrecht et Al.[85]. Therein, ω represents the velocity inertial weight, c is the social weight, and r is a number randomly drawn from the unit interval. In a computational optimization problem, this attractive velocity is the main force behind a system's exploitative behavior. For the strategy we propose, it is used to drive agents toward the target, which is our problem's stand-in for exploitation. A key difference is the memory length used, as the Social-Only PSO model implemented an infinite memory length. To avoid the undesirable side effect of exploitation of outdated information, the memory implemented for this strategy is capped.

3.4.2 Adaptive Repulsion

The second component of our agents' velocity vector is the adaptive repulsion. Pushing agents away from each other counteracts the natural tendency to aggregate around points of interest generated by the attraction velocity. This makes it the main driver of exploration, as agents are pushed to explore the search space. Repul-

Algorithm 2 : Point of Attraction Update Algorithm

Initialise $M = t_{\text{mem}}$
if Agent detects target **then**
  $\mathbf{p}_{\text{self}} \leftarrow$ Target's position
  $t_{\text{best}} \leftarrow t$
end if
Determine $\mathcal{N}_i = \{j \in [1, N]$ s.t. agent j is a topological k-nearest neighbor of agent $i\}$
Get list of all neighbors' $\mathbf{p}$ and t_{best}
$\mathbf{p}_{\text{neigh}} \leftarrow$ Most recent entry in all neighbors' $\mathbf{p}$
$t_{\text{neigh}} \leftarrow$ Most recent entry in all neighbors' t_{best}
if $t_{\text{best}} + M < t$ **then**
  $\mathbf{p}_{\text{self}} \leftarrow \emptyset$
end if
if $t_{\text{neigh}} + M < t$ **then**
  $\mathbf{p}_{\text{neigh}} \leftarrow \emptyset$
end if
if $\mathbf{p}_{\text{self}} = \emptyset$ **and** $\mathbf{p}_{\text{neigh}} = \emptyset$ **then**
  $\mathbf{p}[t] \leftarrow \mathbf{x}_i[t]$
else if $t_{\text{best}} > t_{\text{neigh}}$ **then**
  $\mathbf{p}[t] \leftarrow \mathbf{p}_{\text{self}}$
else
  $\mathbf{p}[t] \leftarrow \mathbf{p}_{\text{neigh}}$
end if

sion also makes the simulation more accurate to real-life experiments by providing an anti-collision measure.

The inter-agent repulsion developed by Kwa et al. [13] is based on the one implemented in past studies on a BoB swarm [9, 34], which was itself directly inspired by the repulsion exerted between atomic particles. Repulsion is thus calculated, for an Agent i with topological neighbors j, using the following equation:

$$\mathbf{v}_{i,\text{rep}}[t] = -\sum_{j \in \mathcal{N}_i} \left(\frac{a_R[t]}{r_{ij}[t]} \right)^d \frac{\mathbf{r}_{ij}[t]}{r_{ij}[t]}, \tag{3.3}$$

where $\mathbf{r}_{ij}$ is the vector from Agent i to Agent j and $r_{ij} = \|\mathbf{r}_{ij}\|$. As such, two parameters serve to control the repulsion's magnitude: repulsion strength a_R, which dictates the distance between agents at equilibrium, and exponent d in the pre-factor term (a_R/r_{ij}). In our work, d is fixed at 6, which was found to have a limited impact on the strategy's performance [13]. When (a_R/r_{ij}) and d are sufficiently large, the agents' repulsion strength roughly equals the nearest-neighbor distance in equilibrium.

The most important particularity of defining inter-agent repulsion is its dynamic aspect. Indeed, each agent can adjust its own repulsion strength $a_R[t]$ in response to its local environment and neighborhood. To do so, we exploit the aforementioned state, which is defined by agents according to whether or not they are currently aware of any target's location. When an agent is in an exploratory state ($S_{i,\text{exp}}[t] = 1$), or unaware of the target, it gradually increases its a_R until a predefined maximum is reached. Reciprocally, an agent in tracking state ($S_{i,\text{exp}}[t] = 0$) will reduce its a_R until it falls to its predefined minimum. The overall adaptive repulsion is summarized in Algorithm 3.

Algorithm 3 Adaptive Repulsion

```
Initialise a_{R,min}, a_{R,max}, d = 6, δ_explore = 0.1, and δ_track = 0.75
while System active do
  if S_{i,exp}[t] = 0 then
    if a_R > a_{R,min} then
      a_R ← a_R − δ_track
    end if
  else if a_R < a_{R,max} then
    a_R ← a_R + δ_explore
  end if
  Calculate v_{i,rep} using Eq. 3.3
end while
```

3.5 Target Parameters

To model the targets being tracked, we use a disc-shaped binary objective function with fixed radii of $p = 1$, and a location denoted by $x[t]$ and $y[t]$ at the center of the disc. Targets also have a maximum velocity $v_{o,\max}$, which is higher than the velocity of both types of agents used in this work, hence $v_{a_0,\max} < v_{a_1,\max} < v_{o,\max}$. When an agent is within a target's radius, it is considered to be tracked, thus giving the formal equation:

$$\mathrm{cov}(o_m, t) = \begin{cases} 1 & \exists i \in A \text{ s.t. } \|\mathbf{x}_i - \mathbf{x}_m\| \leq \rho, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

By modeling the target as a binary objective function, we mimic a purely visual search task in which an agent needs to see the target to be considered tracked [39]. This serves to make the task more challenging, as opposed to systems that allow a target to be considered tracked from the intensity of an emitted signal (e.g. radio signal strength, chemical plume, etc.). Such systems can use various gradient-descent tracking strategies, which, although widely used, are rendered completely ineffective in a binary case. Therefore, using a binary objective function allows us to use swarm intelligence to its full potential by implementing it in the most challenging scenarios without relying on simplistic tracking techniques.

With the target now defined, the final component of executing our strategy is its movement policy. By default, the target uses a non-evasive policy. In practice, this means that the target generates a random waypoint within the search space to head towards at its maximum speed.

Should the target's movement policy be set to evasive, the same initial strategy is employed. However, if an agent is found within its radius (i.e., if the target is tracked), the target uses a repulsion mechanism similar to the one described in Sec. 3.4.2 to attempt to evade its pursuers. When this repulsion is active, the target's velocity is calculated as described in Eq. (3.5):

$$\mathbf{v}_{\mathrm{rep},m}[t] = -\sum_{i \in \mathcal{N}_i} \left(\frac{a_R}{r_{mi}[t]} \right)^d \frac{\mathbf{r}_{mi}[t]}{r_{mi}[t]}, \tag{3.5}$$

where $\mathbf{r}_{mj}[t]$ is the vector from target m to agent i at time-step t, and agent i is in the group of agents $\mathcal{N}_i$, located within the target's radius. The repulsion vector's magnitude is dictated by a_R, the repulsion strength of the target, and d, the prefactor exponential term. These are respectively set at 7.5 and 6.

The target's final evasion mechanism is to travel in a straight line for t_{evade} time steps. This is only triggered after the target is tracked consecutively for t_{limit} time-steps, representing its last resort to evading its pursuers. In our work, $t_{\text{evade}} = 30$ and $t_{\text{limit}} = 10$.

3.6 Problem Statement

The general strategy defined earlier was specifically built within the CMOMMT framework put forth by Parker and Emmons [36], which first defined our target tracking problem.

As such, in this work, a set of N tracking agents $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$, and a set of M targets $\mathcal{O} = \{o_1, o_2, \ldots, o_M\}$ move within the boundaries of a two-dimensional $L \times L$ sized square search-space without any obstacles. To vary the swarm density $\rho = N/L^2$, the length L of the search-space is modified, and $L \in [10^{0.6}, 10^{2.65}]$. Varying number of agents and targets are used, with $N \in [20; 50]$ and $N \in [1, 2, 3]$.

Two types of agents are simulated: slower agents, representing the BoB-0 units, and faster-upgraded agents, standing in for BoB-1 units, with respective maximum velocities $v_{a_0,\max} = 0.1$, and $v_{a_1,\max} = 0.26$ [3]. The target is always faster than agents, as it is set to $v_{o,\max} \in [0.15, 0.30]$ depending on whether faster agents are considered or not.

The goal of the system is to maximize its tracking performance (Ξ) within the environment, which is measured by the reward function:

$$\Xi = \frac{1}{T} \sum_{t=1}^{T} \text{cov}(o, t), \tag{3.6}$$

where T is the total time period of interest. To generate initial conditions, this method relies on using a random seed. Therefore, for any observations to be generally valid, i.e., outside a specific set of initial conditions, the time period T—i.e., the total number of dimensionless time-steps t representing each iteration of the simulation— over which the system is analyzed must be sufficiently long for the system to be ergodic. In this work, the length of time needed is tested by setting $T \in [1000, 10000, 100000]$.

3.7 Metrics

3.7.1 Exploration and Engagement Ratio

In addition to the aforementioned tracking performance, we defined a pair of complementary metrics that measure the system's exploration-exploitation balance. These are the engagement ratio, first introduced by Kwa et al. [15], and the exploration ratio, which was first used as part of this work [1, 2].

As its name indicates, the exploration ratio measures the proportion of agents not engaged with a target, i.e., exploring, averaged over time. It is, therefore, used as a proxy for the amount of exploratory actions carried out by the swarm. As previously defined, an agent that is currently in an 'exploratory' state sets $S_{i,\exp}[t] = 1$. As such, we can calculate the swarm's overall exploration ratio in the following way:

$$\Lambda = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} S_{i,\exp}[t], \tag{3.7}$$

where N is the total number of agents within the system. When using the exploration ratio, a higher Λ signifies that a higher proportion of agents are spending more time looking for the target. The swarm is thus understood to be biased towards carrying out exploration. Meanwhile, the opposite, a lower Λ, indicates agents are spending more time exploiting target information.

The engagement ratio is simply the exploration ratio's opposite, meaning they can be used interchangeably to come to the same conclusions. Therefore, the engagement ratio can be calculated with the following equation:

$$\Theta = 1 - \Lambda. \tag{3.8}$$

As the tracking performance is also a measure of exploitation, the exploration ratio is typically preferred in this work. However, the engagement ratio renders the difference between 'effective' exploitation, represented by the tracking performance, and the total amount of exploitation carried out much clearer. It is thus preferred in some analyses.

Using the engagement ratio in past work also allowed for discovering an optimum engagement ratio, for which tracking performance is maximized [15]. It is important to note that such an optimum is specific to a particular problem. Consequently, the relationship between the optimum engagement ratio and its mirror, the optimum exploration ratio, is explored in the following work.

3.7.2 Swarm Density

Finally, we introduce the concept of a system's *swarm density* or *agent density*. Both of these terms are used interchangeably throughout this work, and they are calculated using the following equation:

$$\rho = N/L^2. \tag{3.9}$$

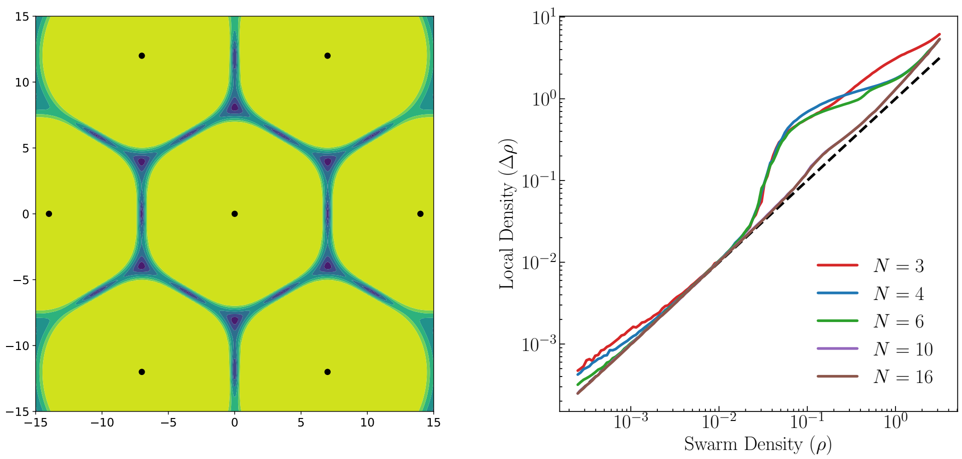
To add onto this notion of density, the *local swarm density* is also calculated using the average distance between each agent and their 6 nearest neighbors, as explained below. By calculating the local density of each of the swarm's agents in this manner, its overall local density can be averaged for all agents for the entire simulation period:

$$\rho_L = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{7}{\pi L_i^2}, \tag{3.10}$$

where L_i is the aforementioned average distance from an agent to its six closest neighbors. This metric allows us to quantify the swarm's level of clustering, as tracking agents tend to coalesce in a cluster around the target. To normalize this result, the density difference, representing the difference between the swarm density and the local density, is calculated: $\Delta\rho = \rho_L - \rho$.

The movement behavior of agents is analyzed to select the number of neighbors to use for the local swarm density. As shown in Fig. 3.1(a), the agent's inter-agent repulsion results in a trend towards forming a hexagonal pattern. The number of neighbors an agent will have in equilibrium is, therefore, typically 6, excluding edge cases.

Of course, any number of neighbors could be selected to calculate the local swarm density. This is shown in Fig. 3.1(b), as similar trends in local density respective of global density can be observed when varying the number of neighbors. However, using a smaller number increases the local density at a small swarm density, as random encounters between agents artificially increase the local density. When using a very large number, the opposite is seen, as the impact of clustering is almost completely erased from the local density curve, reducing the density difference to almost nothing. Consequently, 6 was chosen as an intermediate value to avoid these issues.



(a) **(b)**

Figure 3.1: 3.1(a): Positioning of agents (black dots) in relation to each other and the repulsion fields generated by the individual agents while in their equilibrium positions. Areas in yellow represent areas of high repulsion potential while those in blue represent areas of low repulsion potential. Given these repulsion potential fields, agents tend to fall into a hexagonal packing pattern around each other. 3.1(b): Local agent density calculated with different numbers of neighboring agents for a swarm comprised of 50 agents, connected using a $k = 20$ communications network. The system tracks a non-evasive target traveling at a maximum speed of $\mathbf{v}_{o,\max} = 0.15$. The local agent density is compared against the global average swarm density (dashed line) [1].

3.8 Validation

To investigate the variability of results produced by this method over different periods of interest, we first run simulations with 10 randomly selected seeds over time periods of $T = 1,000$, $10,000$, and $100,000$. In Fig. 3.2, it is shown that the standard deviation of results decreases dramatically when augmenting the time period, leading to a negligible variation between results when using $T = 100,000$. The swarms simulated were operating in a middling performance range, as it's where the highest variation in results was noticed in this work [1, 2].

In our initial work [3], the higher value $T = 400,000$ was used to ensure results were ergodic and thus truly statistically representative. However, as increasing the time period leads to an exponential increase in computing time for diminishing improvements in precision, this investigation reveals using values this high is unnecessary. Subsequent work [1, 2] was thus carried out with $T = 100,000$.

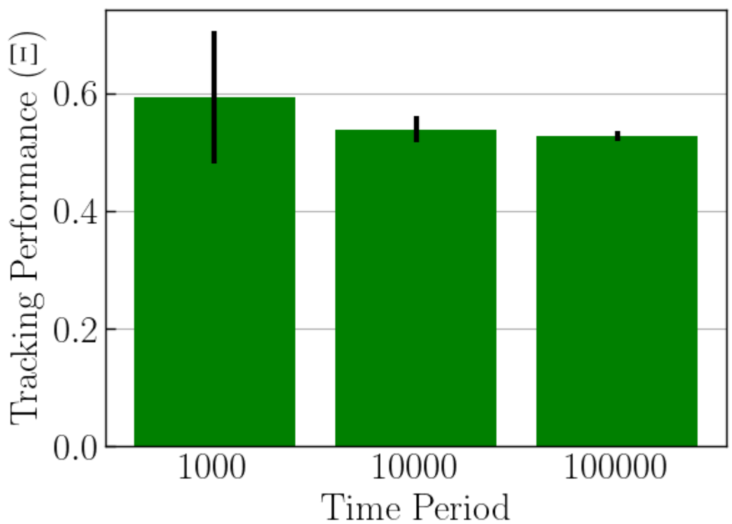


Figure 3.2: Average tracking performance of a swarm over a time period of $T = 1,000$, $10,000$, and $100,000$ with the standard deviation of each data set.

Chapter 4

Density Analysis

Does swarm size matter? Although it is trivial to determine that the number of units in a swarm is a key parameter in its design, there is currently no guiding principle to help in its selection. This chapter shows that swarm size is fundamentally related to swarm density. Examining the relationship between swarm density and performance reveals how the number of agents influences group dynamics, which manifests in trade-offs between exploration and exploitation. Different levers to modify those dynamics, namely connectivity, memory, and different target parameters, are then explored. This chapter compiles and re-frames results presented previously [1, 2].

4.1 Introduction

In the design of a swarming MRS, few questions are as critical as the number of robots to use to achieve the expected results [86]. In fact, by using too few agents, a practitioner can completely miss the benefits of swarms, as "the advantages of swarm intelligence algorithms can only be exploited if a critical mass of swarm members is reached" [87]. In addition to the swarm's ability to coalesce into collective, other important parameters must be considered when selecting the number of agents to compose it. As stated by Schroeder et al. [63], these include:

- System scalability
- Technical capabilities of the individual agents making up the swarm, such as communication range, sensor range, maximum speed, etc.
- Financial or logistical constraints, such as the number of robots that can be bought, built, and operated with available resources.

While practitioners may consider some or all of these parameters, the selection of the number of units used in past experiments is often nebulous. Three orders of magnitude exist within the literature, starting at less than 10 units for a hunter drone swarm [88], to the 150 unit subCULTron [28], all the way to the 1024 unit Kilobot swarm[89]. As previously mentioned in Sec. 2.1, many consider the need for a certain minimum size to even be considered a swarm, but there is no consensus on the required number [86]. Because of the lack of analysis on how such swarms were sized, there's often no evidence showing a smaller swarm could not have been used in their stead. Even the smallest systems studied have been argued to be capable enough [15, 20, 34], but whether this is done for technical or more practical reasons is typically left unsaid.

When considering size, a more apt metric for the dynamic environments MRS are often designed for is the density of agents throughout the search space, as defined in Sec. 3.7.2. Rephrasing the aforementioned critical mass of agents reveals a minimum critical density, which is hinted at throughout the literature. This is often linked to communication networks, as is the case for Kilobots, which can only communicate up to 10 cm [89], and in the case of the CIMAX algorithm, where it was found that at least 3 agents are needed within an agent's communication range to achieve the desired performance [90].

Consequently, the analysis of agent density can help guide the question of how to select the MRS size necessary for a specific problem, which has not been systematically and satisfactorily answered [1]. Currently, knowledge is limited to the aforementioned minimum density, which was demonstrated for the case of a system of self-propelled particles [91]. In particular, reaching the critical minimum density generated a phase change from disorder to order with the particles aligning. Testing more complicated problems with different behaviors and protocols for a constant surface area, a superlinear performance increase was found with increasing density until a critical maximum density was attained [62]. Increasing density farther actually caused a reduction in performance, attributed to the increase in interference, manifesting through more collision avoidance behavior.

In this chapter, the effects of density on the performance of the MRS defined in Chapter 3 for the tracking of a fast non-evasive target is studied. This is complemented by an analysis of its influence on the exploration-exploitation balance. Other parameters, such as memory, various target parameters, and swarm size, are also investigated at varied densities using both our performance and EED metrics. Three density 'phases', common to all swarms examined, are thus revealed: (1) the low-density phase, in which agents are unable to track the target, (2) the high-density phase, in which the target is consistently tracked, and (3) a transition phase, throughout which the swarm's performance increases rapidly with density. While these phases occur no matter the swarm size or other parameters, how such modifications affect the density at which they occur is explained through the analysis of their impact on the exploration-exploitation balance. Although most of the work hinges on the previously described concept of agent density, swarm size still has a determining impact due to its influence on the number of available units for the carrying out of exploration.

4.2 Density Phases

First, swarms between $N = 20$ and $N = 50$ are simulated, tracking a non-evasive target in various environment sizes to study the effects of swarm density on tracking performance. These results, which were presented by Kwa et al. [1] are shown in Fig. 4.1. There, three distinct regimes, hereafter called 'phases,' are distinguished: (1) low density, (2) transition, and (3) high-density phase.

When density is low, in this case, $\rho \lesssim 10^{-3}$, the simulated agents are completely incapable of tracking the target. The main culprit is the large inter-agent distance, as it reduces the odds of any single agent coming into contact with the target, thus preventing the target's position from spreading from an agent to its neighbors. Additionally, even if the target is encountered, the agents are too spread out to

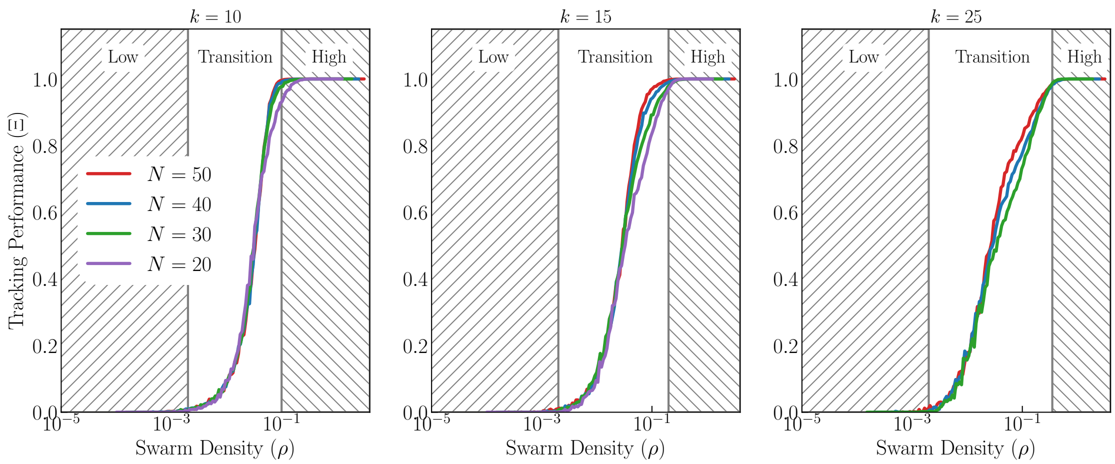


Figure 4.1: A swarm of $N = 20$, 30, 40, and 50 agents with varying levels of connectivity, k, and swarm density, ρ, tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$. In the low-density phase, the system is unable to track the target. In the high-density phase, the system can track the target continuously without interruptions. Between these two phases, there is a transition phase in which the tracking performance of systems rapidly increases with swarm density, ρ [1].

band together and cluster around it effectively. Thus, encounters are short-lived; the individual agents get outpaced immediately without getting the chance to follow the target based on information sharing from other nearby agents. However, as explained in 3.2, the model used for this simulation does not reduce the flow of information with distance. Therefore, the inter-agent distance negatively affects only the use of this information, not its quality.

When density is high ($\rho \gtrsim 10^{-1}$), the swarm tracks the target with near absolute accuracy. As the inter-agent distance is very small, agents are almost always on the target. This means the target has practically no open space to occupy, leaving it always in agent range.

In the middle transition phase, tracking performance (Ξ) increases quickly with ρ. Its start coincides with the local density reaching a level sufficient for the agents to cluster around a point of attraction, as shown by Fig. 4.2. Clustering enables agents to lengthen encounters, as they are sufficiently close to each other to relay the tracking task to new agents as the original one that found the target gets outpaced.

Fig. 4.1 also shows that the density at which the transition phase starts is independent of N, while the density at which it stops is not. This is explained by the fixed level of connectivity k, which draws a constant number of agents to the target. At low density, agents cannot translate target information into effective tracking through clustering. Thus, the system is limited by a lack of exploitation, because its agents only explore, no matter the overall swarm size. However, nearing the high-density phase, the performance-density curves of the differently sized swarms separate as larger N swarms outperform smaller ones. This is due to smaller swarms having fewer spare, or non-exploiting, agents—i.e., agents outside of the connectivity range of those in contact with the target— to allocate to the exploration of the search area. Therefore, smaller swarms are limited by their lack of exploration.

To summarize, swarms in the low-density range are exploitation-limited, as opposed to high-density ones, which are exploration-limited. In the next sections, the implications of different parameters, like swarm size, level of connectivity, and memory, are explored.

4.3 Exploration-Exploitation Dynamics

As described in Sec. 3.7, the system's EED is characterized by a system-level metric, the *exploration ratio*, Λ, first introduced by Kwa et al [1]. By definition, high Λ demonstrates a swarm prone to carry exploration, while a low Λ indicates the bias is shifting to exploitation. However, it is important to note that not all exploitation is 'useful', in the sense of contributing to carrying out the task at hand, which will be discussed further in this section. Since the swarm's tracking performance Ξ is essentially a measure of exploitation resulting in actual target tracking, Ξ can be used to indicate said 'useful' exploitation. As Λ is only a tool to understand results, Ξ is still the parameter sought to be optimized in this section.

Although a trade-off between exploration and exploitation in collective tasks has been shown in different multi-agent swarms [77, 92], no clear understanding of this process has been set forth. Using the aforementioned tracking performance and exploration ratio throughout the density spectrum allows us to provide a quantitative analysis of the EED.

Beginning with the previously identified low-density phase 4.2, Fig. 4.3 presents an EED balance that is fully biased towards exploration. This is irrespective of the connectivity level k, which favors exploitative behaviors. Thus, skewing the swarm towards exploitation at such low density is insufficient to generate any target tracking. Consequently, it can be said that the swarming strategy cannot be effective in this density range.

Moving to the transition phase, Fig. 4.3 renders the trade-off between explo-

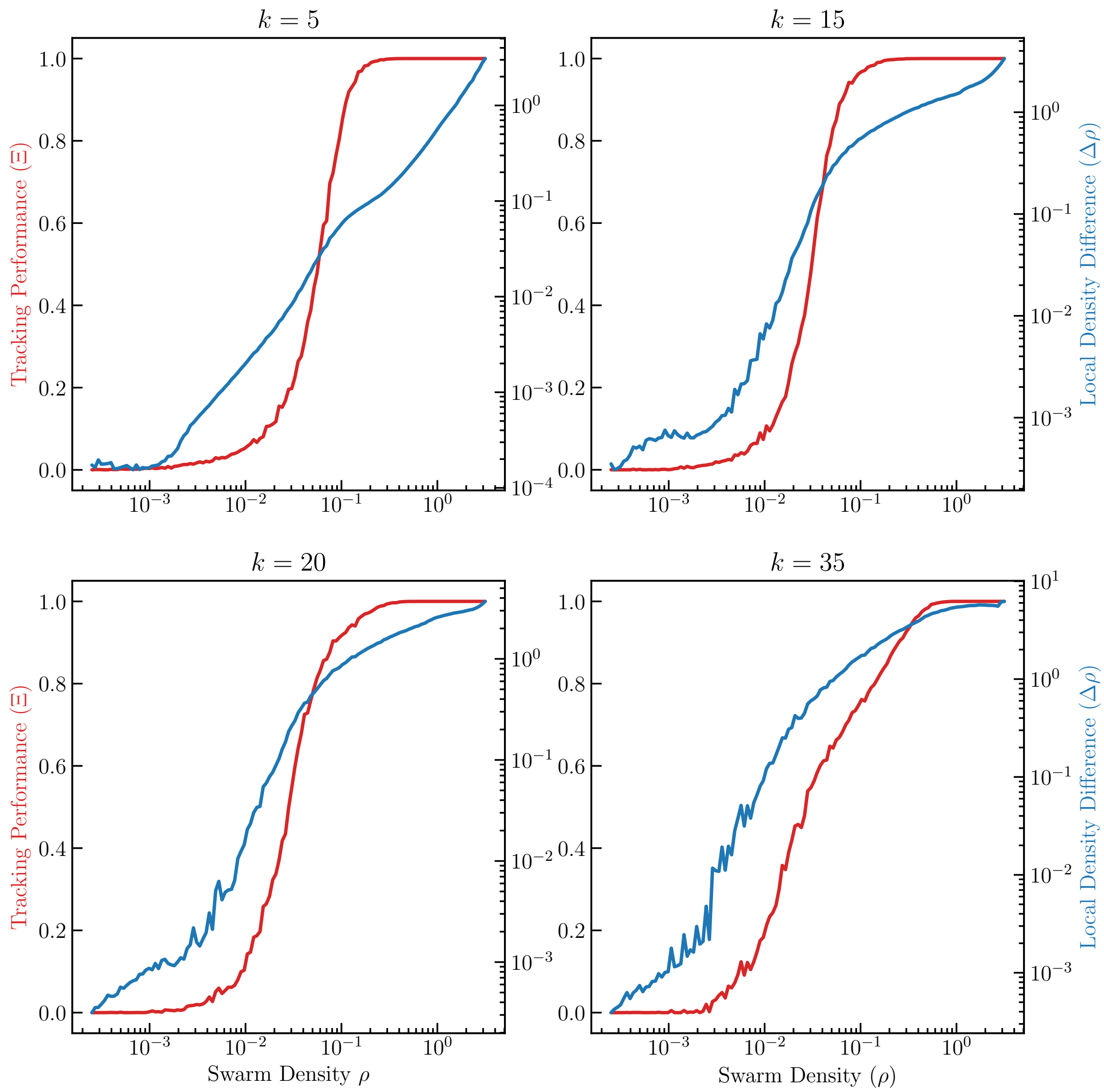


Figure 4.2: Density difference, $\Delta\rho$, of a swarm comprised of $N = 50$ agents with various degrees of connectivity k, tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$ [1].

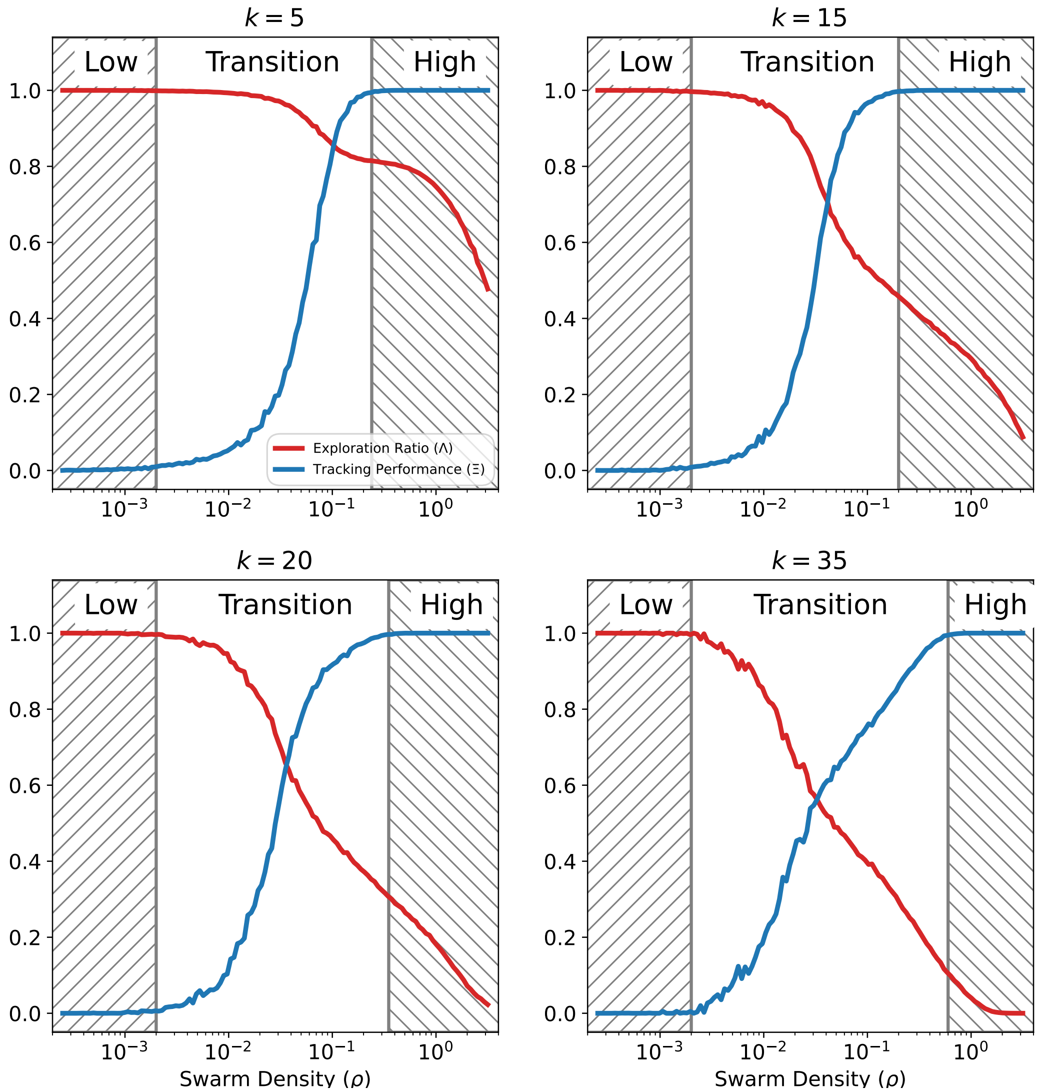


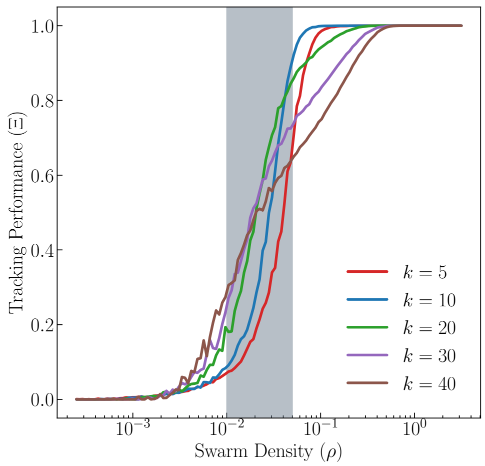
Figure 4.3: Tracking performance, Ξ, and exploration ratio, Λ, of a swarm of $N = 50$ agents using with levels of connectivity $k \in [5, 15, 25, 35]$, while tracking a non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$. The low, transition, and high-density phases have been marked out on the plots [1].

ration and exploitation obvious. As inter-agent distances get smaller, it is easier for individuals to flock, or cluster, together. This promotes the exploitation of target information shared by exploring agents. Increasing density, Ξ, leads to more resources being allocated to exploitation, which simultaneously reduces the exploration ratio, Λ.

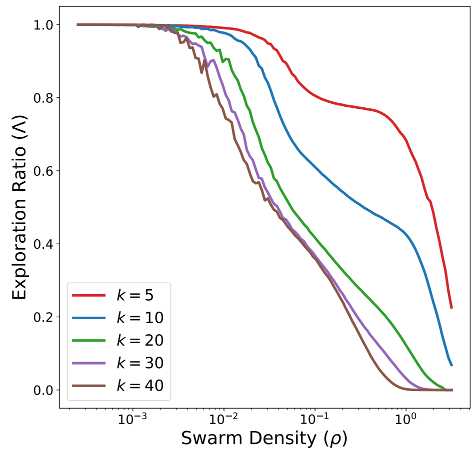
During the transition phase, the impact of connectivity, k, is highly noticeable. As discussed in Sec. 4.2, the transition phase itself begins at lower densities for high k systems relative to low k ones, as the sharp rise in performance is shifted. For example, looking at the case of $k = 35$ system in Fig. 4.3, Λ quickly decreases throughout the transition phase until almost no exploration is carried out. In contrast, $k = 5$ shows much higher exploration Λ at higher densities, while high Ξ values are still achieved. Interestingly, the trade-off in this latter case is not associated with mutually exclusive actions, as high levels of exploration can be maintained while carrying out high levels of exploitation in the medium-to-high ($8 \times 10^{-2} \lesssim \rho \lesssim 1 \times 10^{0}$) density range. This result matches current literature, which has shown that exploration and exploitation can be carried out simultaneously [14].

Finally, the exploration-limited high-density phase of the swarms of Fig. 4.3 clearly shows why low k systems perform better than high k ones. Indeed, at low connectivity, systems achieve a maximum tracking performance of 1 while resources are still used for exploration. This EED balance is thus much more effective, as the system keeps the ability to look out for the target, which is essential to achieving the desired high tracking scores. Comparatively, when $k = 35$, the aforementioned increase in performance across the transition phase begins earlier, but is not as rapid. In the right half of the transition phase, where $\rho \gtrsim 10^{-1}$, the increase in Ξ slows down, which causes higher swarm densities to be needed to get to a perfect tracking. As this regime change coincides with a rapid drop in Λ, the slowed performance can be directly attributed to the system's poor vigilance.

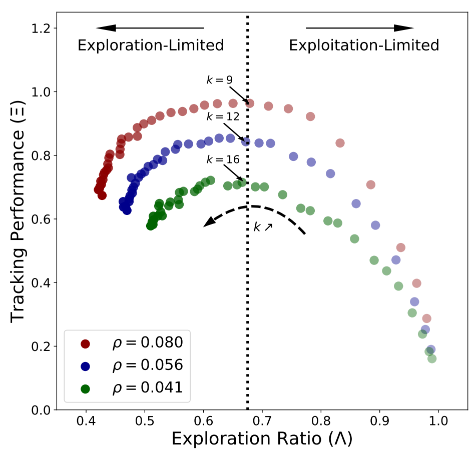
The complex relationship between exploration and exploitation reveals a very important fact: a MAS's behavior is required to be adaptive when varying swarm density. Given a specific density, an adaptive swarm could achieve the optimal EED balance required to yield maximal tracking. In this searching and tracking problem, we use k as our 'adaptivity' lever, but other parameters affecting the EED balance could be used. Thus, one question remains: how can the optimum exploration-exploitation balance (here, in value of k) be identified at any given density ρ? Considering each combination of swarm densities and levels of connectivity has a different exploration-exploitation balance, we know there is an optimum connectivity level for each density, as illustrated in Fig. 4.4(a). The gray-shaded area, where the best-performing curve crossovers from one value of k to the next, presents where this optimum level of connectivity shifts. In practice, the higher tracking performance is



(a`



´b)



(c)

Figure 4.4: A swarm of $N = 50$ agents tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$. 4.4(a): Relationship between a swarm's tracking performance (Ξ) and the average swarm density (ρ) while operating at different levels of connectivity (k). The crossover region, where the optimum k maximizes Ξ, is highlighted in gray. 4.4(b): Relationship between a swarm's exploration ratio (Λ) and the average swarm density while operating at different levels of connectivity. 4.4(c): Relationship between the system's tracking performance and exploration ratio. Lighter-shaded points represent systems with lower k, while darker-shaded points represent systems with higher k. The optimum level of exploration, Λ^*, is represented by the dotted line, with exploration-limited systems lying on the left of that line while exploitation-limited systems lie on the right. All three densities considered fall within the transition region (Fig. 4.1) [1].

achieved with a higher level of connectivity at the lower end of the transition phase.

Then, the opposite becomes true as the density increases.

To further analyze these phenomena, we again turn towards the relationship of the exploration ratio, Λ, with the density ρ, as seen in Fig. 4.4(b). At low levels of connectivity k, swarms achieve higher levels of exploration throughout the density spectrum. This is expected since reduced connectivity limits the spread of the target's location to a smaller group of agents. Unaware of the target's position, other agents maintain their exploration, increasing the swarm's overall exploration ratio.

For these low k exploitation-limited swarms, the worse range of operations is low ρ, which is naturally exploration-heavy or exploitation-limited. Instead, high k values can be used, as they promote the reach of social transmission of information between agents, making the swarm's agents better equipped to collaborate and exploit information by clustering around the target. Before the crossover, these exploitation-heavy swarms are thus better equipped to the conditions and outperform low k systems.

Predictably, increasing the density in the transition phase promotes exploitation by decreasing inter-agent distances, which allows for the clustering of the agents around the target. As briefly noted previously, past the crossover, we see low k systems performing better than high k ones. This is explained by over-exploitation, which happens when the higher k swarms generate excessive clustering around the target, which does not improve its tracking performance. Such exploitation is considered 'redundant' and takes away from the swarm's ability to explore, rendering it exploration-limited.

The previously established connection between the optimal level of connectivity k^* and ρ is made clearer with Fig. 4.4(c), which also shows there exists an optimal level of exploration Λ^*. Achieving this Λ^* allows for our swarm's tracking performance to be maximized for a given density. Therefore, we have an optimal connectivity k^* such that $\Xi(k^*, \rho) = \max_k(\Xi(\rho))$ (Fig. 4.4(a)), which also gives us an optimal exploration ratio $\Lambda^* = \Lambda(\max_k(\Xi))$.

Interestingly, while the optimum exploration level Λ^* is almost completely hidden in Fig. 4.4(b), it is easily represented by a simple vertical dotted line in Fig. 4.4(c). Surprisingly, Fig. 4.4(c) suggests that Λ^* stays constant even for different densities. This requires further study, as only a narrow range of densities is considered, which gives us a small glimpse of the location of Λ^*. Additionally, this visualization of Λ^* makes the previously discussed notion of exploration-limited and exploitation-limited systems very clear, as it delineates between the two regions of the graph. Indeed, systems falling on the left side of the optimum exploration ratio are limited by their lack of exploration, while systems on the right-hand side are limited by their lack of exploitation.

This conclusively shows that the system's level of connectivity has to be adapted to manage EED and, therefore, achieve an optimum balance between its two principal behaviors: exploration and exploitation [1]. Such a use for an 'adaptivity' lever was suggested by several previous studies [13, 15, 82, 93].

4.4 Memory

Although the level of connectivity k is the main 'adaptivity' lever presented in this work, the potential of other parameters to affect the EED of the swarm, manifesting itself in the different phases presented in Sec. 4.2, were also studied. In particular, the agents' memory has been identified as having some limited potential to act as an 'adaptivity' lever.

4.4.1 Evasive Target Tracking

Beginning with the tracking of an evasive target, Fig. 4.5 shows that implementing agent-based memory is a helpful way to push the beginning of the transition phase to the left, i.e., promoting an increase in performance at lower densities. Indeed, it indicates that a swarm with memory will begin its transition phase at a much lower density when compared to one that does not utilize memory ($\rho \sim 1.2 \times 10^{-2}$ with memory compared to $\rho \sim 5 \times 10^{-2}$ without memory) [2].

To explain this effect, we must return to Sec. 3.4.2, where the point of attraction was defined. By providing an agent with memory through the mechanism detailed in Sec. 3.3, each sighting of the target becomes a persistent point of attraction, to which agents will congregate, even if the target's location has been outside of the agents' range for multiple time-steps. Consequently, the cooperation between agents is promoted, boosting their exploitative behavior, i.e., tracking the target. Previous literature had shown memory to increase performance when tracking an evasive target, which this result confirms [15].

Additionally, Fig. 4.5 shows that the length of memory does not affect the density at which the swarm enters its high-density phase, while the memory-less swarm's transition phase occurs over a narrow range of densities. As a swarm cannot be effective at all in the low-density phase, and its individual agents can barely move

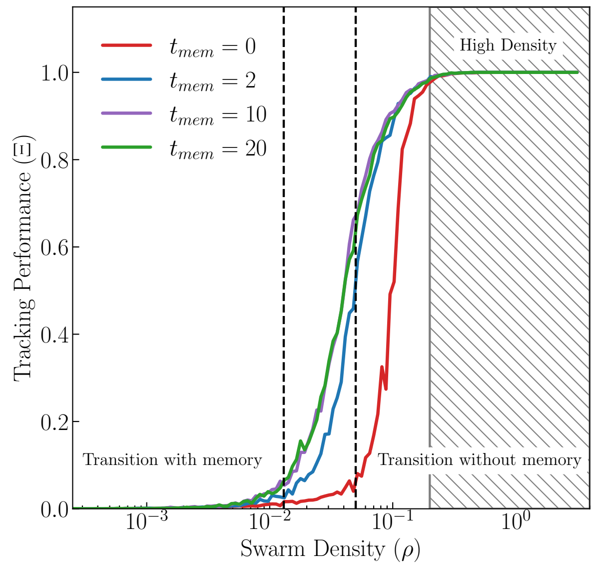


Figure 4.5: The effect of memory on a system's transition phase when tracking an evasive target. The start of the transition phase in a memory-less system occurs at a higher swarm density than systems with memory. Memory usage does not affect the start of the high-density phase [2].

in the high-density one, these results suggest that the range of operation to conduct a target-tracking task for a memory-less swarm is narrower.

Although implementing memory can reduce the minimum density necessary to start the transition phase, Fig. 4.5 also shows that performance cannot simply be increased further by increasing memory length. Indeed, higher memory quickly loses its benefits to the swarm's tracking performance. As mentioned, going from a memory length $t_{\text{mem}} = 0$ to $t_{\text{mem}} = 2$ vastly reduces the starting density, but the increase in performance when going from $t_{\text{mem}} = 2$ to $t_{\text{mem}} = 10$ is much smaller. Furthermore, the transition phase of the swarms with a memory length $t_{\text{mem}} = 10$ and $t_{\text{mem}} = 20$ are similar, with $t_{\text{mem}} = 10$ even having a better performance at higher densities, showing a clear limit to returns obtained by increasing the swarm's memory.

While giving a certain persistence to the information uncovered when the target is located was shown to be beneficial, the longer the agents' memory is, the more outdated the information exploited becomes. Since the target is always on the move, this older information causes the agents to cluster at a position that the target has left, as shown in Fig. 4.6. While clustering near a slightly older location can increase target tracking by increasing inter-agent collaboration, this quickly becomes counterproductive as the information becomes too old. Thus, increasing the agents' memory

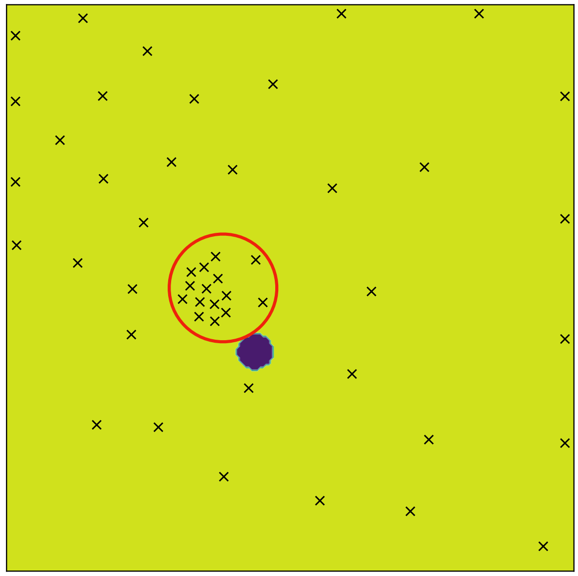


Figure 4.6: A swarm with a memory length of $t_{\text{mem}} = 20$ tracking an evasive target. Agents clustering in the wrong location due to the exploitation of outdated information circled in red when using excessive memory lengths [2].

further does not benefit their performance, even if it boosts exploitative behavior.

4.4.2 Non-Evasive Target Tracking

In the literature, agent memory has often been removed entirely from a swarming strategy when tracking non-evasive targets [13, 23, 94]. That choice is typically made to prevent the exploitation of outdated information, which, as shown in the evasive case, can cause clustering around a target's past location. This problem can be aggravated in cases where the target or the environment evolves faster [46].

However, Fig. 4.7 shows that implementing agent memory can cause a slight increase in performance during the transition phase. Similarly to the evasive case, there is no notable effect on the high-density phase. On the low-density side, adding memory slightly lowers the density required to enable tracking. Thus, adding memory to a swarm's agents increases the range of densities for which the MRS is effective, thereby increasing its potential use cases.

To analyze the mechanisms behind this increase in performance, we turn to Fig. 4.8, which shows the direction of each agent's velocity vector [2]. The effect of memory can clearly be seen as agents converge towards the target's old location,

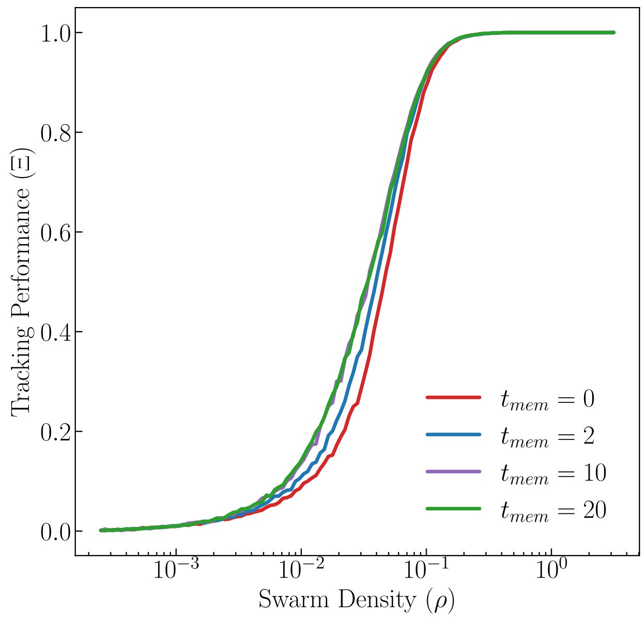


Figure 4.7: The effect of memory on a system's transition phase when tracking a non-evasive target [2].

even if its new location is unknown. Thus, the target's speed advantage, which allowed it to outrun the agents that found it in the first place, is partly negated, as the agents are driven to an area that increases their likelihood of finding it. Memory also generates reductions in inter-agent distances, which, as previously discussed, is another important parameter with the potential to increase swarm performance. Indeed, such a cluster of agents can more readily aggregate and exploit new information on the target's location once another agent happens upon it.

As seen in the evasive case, Fig. 4.7 also presents clear evidence that there is an optimal memory length past which tracking performance is no longer improved. As performance is capped at a memory length $t_{\text{mem}} = 10$, this optimal memory appears to be the same in both the evasive and non-evasive cases. Similarly, again, to Fig. 4.5, Fig. 4.7 shows a slight increase in performance for some densities with $t_{\text{mem}} = 10$ over $t_{\text{mem}} = 20$. Although implementing agent-based memory has clear advantages when tracking either an evasive or non-evasive target, caution must be taken to identify the optimal memory length and avoid the false clustering memory that can be caused when used in excess. Having too many agents that are neither exploring nor tracking the target's current location renders further improvements to the swarm's tracking performance through the use of memory impossible.

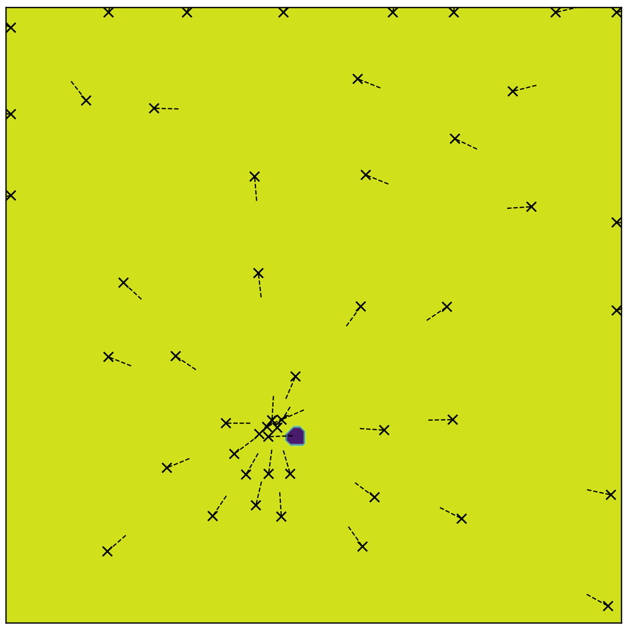


Figure 4.8: A swarm with a memory length of $t_{\mathrm{mem}} = 20$ tracking a non-evasive target at a swarm density of $\rho = 10^{-2}$. Dotted lines indicate the direction of travel of the individual agents [2].

4.5 Target Parameters

4.5.1 Target Movement Policy

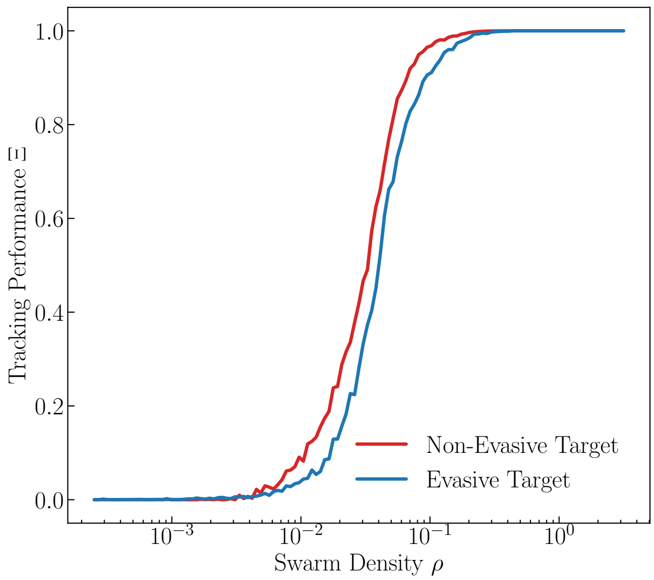


Figure 4.9: The effect of a target's movement policy on a tracking swarm's performance at different densities [2].

In addition to the aforementioned connectivity and memory, the target's tracking properties can also influence the transition phase's density range. One of these parameters, which was explored in relation to agent memory in Sec. 4.4, is the target's movement policy. Fig. 4.9, by comparing the tracking performance when simulating an evasive and a non-evasive target, demonstrates that the evasive target is harder to track. Indeed, even using memory, which was set at $t_{\mathrm{mem}} = 10$ for both swarms, the resulting performance curve for the evasive case is shifted to higher densities.

To understand why the target's evasive movement policy impacts the transition phase, it's important to consider how it affects the agents' tracking actions. With everything else being equal, the swarm's agents can track the target as usual upon the initial encounter. Only after this, where agents can establish exploitation of the target, its evasion mechanism kicks in, allowing it to try to distance itself from the pursuing agents. By doing so, the agents' exploitation potential is partly negated, as they must rely on inaccurate information on the target's position.

Furthermore, even if the target's evasion does not fully succeed in getting its pursuers off its trail, it does make it harder to follow in the long run. Having fewer agents transmitting its location to the rest of the swarm than its non-evasive counterpart means the evasive target has fewer agents drawn to it, which reduces the swarm's exploitation even further. This reduction in effective exploitation lowers the swarm's score, thus shifting the performance-density curve to higher-density ranges.

4.5.2 Target Velocity

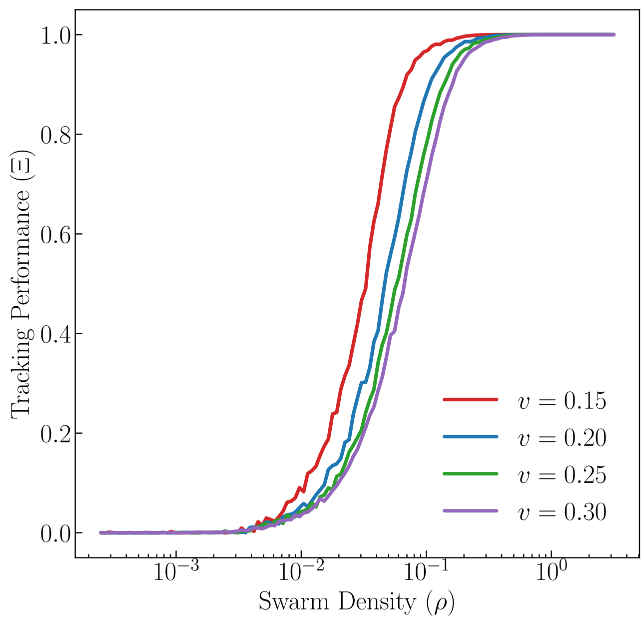


Figure 4.10: The effect of a non-evasive target's velocity on a tracking swarm's performance at different densities [2].

Another target parameter that can be modified to increase the tracking task's difficulty is the target's velocity. By making it easier for the target to outrun its pursuers, the swarm's agents spend more time trailing it, which reduces the amount of effective tracking they can accomplish. As shown in Fig. 4.10, this is the case even if the target is non-evasive. The same mechanism described when comparing the evasive and non-evasive movement policy is also in action here, as increasing the target's velocity means that its positional information is disseminated to fewer overall agents, again reducing the exploitation the swarm can carry out. Similarly, less effective exploitation has the direct consequence of shifting the range of densities over which the swarm's transition phase occurs.

4.5.3 Number of Targets

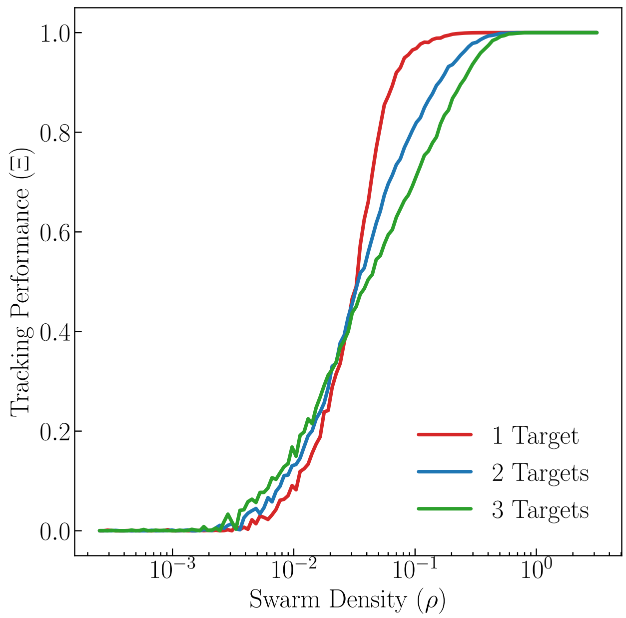


Figure 4.11: The effect of the number of non-evasive targets to be tracked simultaneously on a tracking swarm's performance at different densities [2].

The final method trialed to increase the tracking problem's difficulty and thus change the swarm's resulting transition phase is simply multiplying the number of targets to be tracked. Interestingly, Fig. 4.11 shows that having one or two extra targets to track does not simply shift the density range over which the system's transition phase occurs, in contrast to modifying the target's movement policy or velocity [2]. Indeed, more targets to be tracked results in a transition phase that begins at a lower density, as systems with more targets to track outperform the others. Then, as density increases, the performance curves hinge about a middle point, after which a higher number of targets results in a lower performance.

What might initially seem counter-intuitive can be simply understood by considering that our swarm density metric does not consider *target* density. As targets wander about the search-space at random, increasing their number increases the likelihood of encountering one for the same density. Therefore, when the swarm is limited by its lack of exploitation, i.e., at low density, adding targets adds opportunities for exploitation to occur, which lowers the density necessary for the transition phase to begin. We then observe a direct link between the amount of effective exploitation, represented by the score, and the number of agents.

However, this reasoning no longer holds true as we transition from an exploitation to an exploration-limited swarm. Then, the increased target density still boosts exploitation, which further reduces the amount of exploration carried out by the system's agents. As the system's focus is now divided between multiple targets, having more agents dedicated to over-exploiting the targets has an outsized impact on performance at higher densities. This reduction in performance delays the start of the high-density phase, making the transition phase of a multi-target swarm much wider than a single-target one.

A distinction can be made here between parameters that affect the system's EED and those that uniformly affect its exploration or exploitation. While changing the target's movement policy to evasive or increasing its speed makes exploitation harder along the entire density spectrum, thus changing the ratio of effective to non-effective exploitation, modifying the number of targets actively impacts the exploration-exploitation balance by biasing agents towards generating more exploitation. The difference is obvious on the performance-density curve, as we can shift the transition phase left or right by making exploitation easier or harder, in contrast to widening or narrowing it by generating an exploitation bias or an exploration one.

4.6 Swarm Size

The work presented in previous sections of this chapter has unequivocally shown the necessity of considering swarm density to analyze a MAS. In particular, the importance of the minimum density needed for collective behavior to emerge cannot be understated, as it is the starting point of what we deem swarm 'intelligence'. Density is also essential to analyze the balance between a swarm's exploitation and exploration behaviors throughout its transition phase. Any change in density has consequences that reverberate through every aspect of a swarm's actions.

Despite this evidence, substituting the number of agents by agent density for an analysis cannot make said analysis completely abstract. Indeed, finite-size effects should be considered, as they are unavoidable. For example, the logistical and financial challenges that prompted this density analysis in the first place restrain many practitioners from working with small numbers of agents [63, 95]. Therefore, we must consider the effect of swarm size, which was briefly discussed in Sec. 4.2, and use it to complete our density analysis.

As stated in Sec. 2.3, optimization problems have typically utilized an increase in the number of agents in a system as a means to carry out more exploration. This

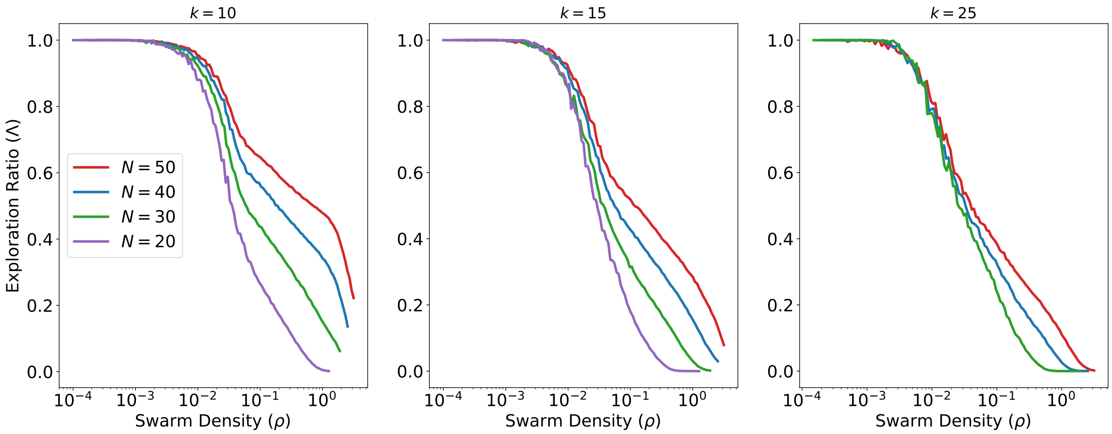


Figure 4.12: Exploration ratio of a swarm of $N = 50$, 40, 30, and 20 agents for three different levels of connectivity k and varying swarm densities ρ tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$ [1].

is shown to be an effective method for our target tracking problem in Fig. 4.12, as, especially at higher densities, an increase in the overall number of agents leads to an increase in the swarm's overall exploration ratio. The effect is even more striking for systems operating at a low level of connectivity, as larger swarms maintain a high level of exploration throughout the density spectrum.

To shed some light on that phenomenon, Fig. 4.13 can be used to show the level of exploration a swarm achieves varies in function of k/N, and not the previously used swarm density. When an agent switches from actively exploring to exploiting, a small swarm loses more of its exploration capacity than a larger one. For example, it would be a 10% decrease in a 10-agent swarm and only 2% in a 50-agent one. This means that, when using a fixed level of connectivity, the larger swarm has a smaller proportion of its agents engaged with the target, leaving more of its agents free to realize its exploration potential. From these results, it can be concluded that exploration is a system-level action, making it dependent on the proportion of agents exploring [1].

To further investigate exploration's system-level classification, we can analyze tracking performance instead. Inspecting Fig. 4.14 confirms the existence of the aforementioned optimal level of connectivity (k^*), which maximizes a swarm's performance [1]. For $k < k^*$, the swarm is found to be exploitation-limited, while it is exploration-limited for $k > k^*$. In that phase, reduction in performance is observed to be proportional to k/N irrespective of swarm size, thus supporting the conclusion that a system's exploration capacity is a global property [1].

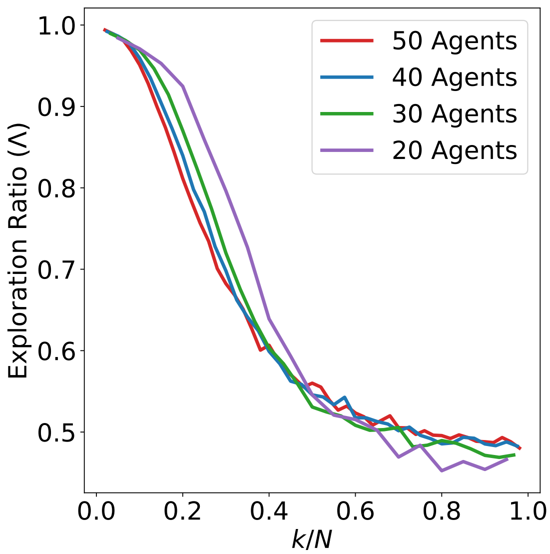


Figure 4.13: Exploration ratio of a swarm of $N = 50$, 40, 30, and 20 agents with varying levels of connectivity at a swarm density of $\rho = 0.0444$, tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$ [1].

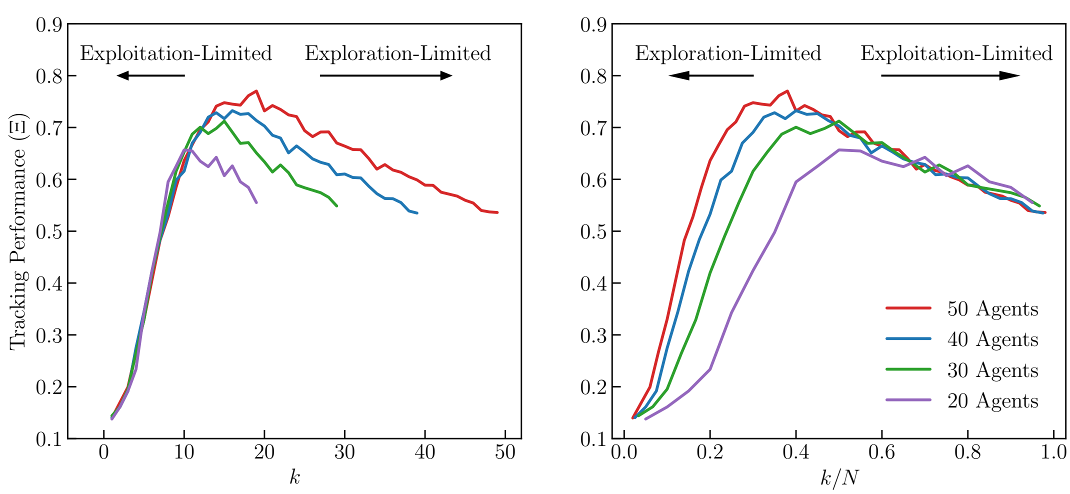


Figure 4.14: Tracking performance of a swarm of $N = 50$, 40, 30, and 20 agents with varying levels of connectivity at a swarm density of $\rho = 0.0444$, tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$. Tracking performances are plot against k (left) and k/N (right) [1].

We can oppose the system's exploitation-limited phase, where the swarms' performances are directly proportional to the value of k. As this is the case for all swarm sizes, we can conclude that exploitation is a local-level task [1]. Indeed, when the swarms are exploitation-limited, having a fixed level of connectivity means that a fixed number of agents are informed of the target's location, thus keeping the resulting level of exploitation constant. Furthermore, the effectiveness of exploitation is wholly unrelated to the overall number of agents, making the number of agents that can form a cluster around the target the only relevant metric, which is dictated by k.

In addition to the general EED trends outlined above, Fig. 4.14 shows a general rule about swarm sizes: the maximum tracking performance achievable is lower for smaller swarms. Predictably, their k^* is also lower. To explain this rule, we must recall that exploitation and exploration are not mutually exclusive actions on a system-wide level, as observed in Sec. 4.3. Since small swarms have a lower exploitation potential or exploration capacity, they have a lower capacity to conduct both exploration and exploitation. Indeed, by increasing the smaller swarm's level of connectivity, it is forced to allocate a higher proportion of its resources to exploitation. This limits the resources available to explore, leaving the swarm to transition into an exploration-limited system faster than its larger counterpart. Meanwhile, the large swarm can keep dedicating more and more agents to exploitation without running out of agents to explore the space, making its overall tracking potential much higher.

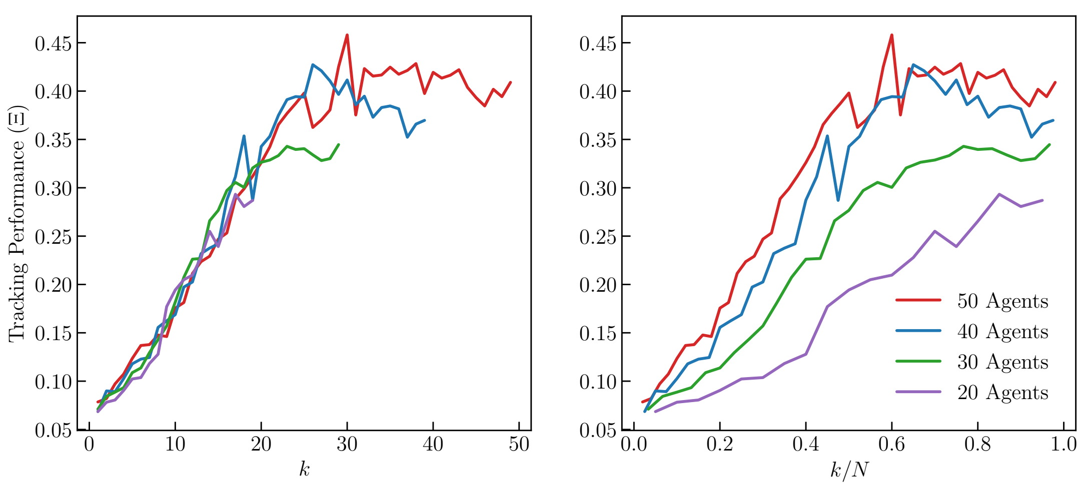


Figure 4.15: Tracking performance of a swarm of $N = 50$, 40, 30, and 20 agents with varying levels of connectivity at a swarm density of $\rho = 0.02$, tracking a fast-moving non-evasive target traveling at $\mathbf{v}_{o,\max} = 0.15$. Tracking performances are plot against k (left) and k/N (right) [1].

Figure 4.15 shows another consequence of the lack of capacity of smaller swarms, which is seen at this density close to the low-density phase: it can completely run out of exploitation potential. Indeed, we've talked up to this point of exploitation potential and exploration capacity, as k dictates how much the swarm is dedicated to exploiting, the rest being available for exploration capacity or 'unrealized' exploitation potential. However, if $k = N$, then there are simply no agents left to explore or no 'unrealized' exploitation potential. In this low-density case, for swarms of $N = 20$ and $N = 30$, tracking performance rises with k and does not reach an optimal k^* value. Therefore, the agents are so few and far between that, even when maxing out their exploitation potential, there are not enough agents to properly exploit the target, i.e., to reach the performance expected of a swarm of such a density. In other words, the optimal level of connectivity k^* does not exist for such a swarm; it does not even have the capacity to *be* exploration-limited.

4.7 Discussion

In this chapter, the testing of our target tracking strategy revealed the existence of three phases, each defined by the performance of the swarm for the corresponding density range. These are the low-density phase, in which the swarm cannot track the target, the transition phase, throughout which performance rises rapidly with density increases, and the high-density phase, in which the target is continuously tracked.

In particular, the transition phase reveals that the optimum level of connectivity k^* changes inversely with the density, as it serves as a lever to compensate for the swarm's over or under-exploitation. When exploration is abundant, at low densities, k biases the swarm towards exploitation. The opposite holds as the swarm becomes exploration-limited at higher densities, necessitating a lower value of k to reach the optimal exploration-exploitation balance. In that case, using a k higher than the optimal value generates excess exploitation, which does not increase performance.

Next, memory implementation was shown to shift the transition phase to a lower density range when tracking both evasive and non-evasive targets. In practice, this allows for a design to include fewer agents to obtain the desired results, which, by lowering the cost and resources necessary for implementation, is a shared goal of practitioners. This is due to the increased level of clustering generated by the longer-lasting point of attraction, which can also become detrimental past a certain memory length.

Different target parameters, including their number, velocity, and movement policy, were also trialed to measure their impact on the swarm performance over a large range of densities. An evasive target was found to shift the transition phase to higher density, as was using a higher target velocity. Interestingly, multiplying the number of targets was found to increase performance at low densities due to the higher odds of encountering a target boosting exploitation and reducing it at higher densities as the swarm becomes exploration-limited.

Finally, focusing on swarm size instead of swarm density allowed for the characterization of exploration and exploitation as, respectively, global and local tasks. Indeed, exploitation relies on the number of agents in a local cluster around the target, while exploration depends on the proportion of agents exploring the search space. Therefore, swarm size greatly impacts exploration, as a large swarm will be able to allocate a larger portion of its agents to exploration, even at the same overall density as a smaller swarm.

Overall, these results provide guidance to practitioners by characterizing the mechanisms behind the critical minimum and maximum densities evoked in the literature. Of course, not every single one will be useful, depending on the task and the specific strategy employed to solve it. However, identifying the transition phase and shifting it to lower densities is vital to harnessing the benefits of swarm intelligence, thus reducing implementation costs. The identified 'adaptivity' levers also reveal a path towards designing swarms that perform over wider ranges of conditions, hinting at the possibility of having agents alter their memory or connectivity level based on the swarm operating in either an exploration or exploitation-limited regime.

Chapter 5

Heterogeneous Analysis

Modularity, although often overlooked, is a great potential feature of MRS, as utilizing a decentralized swarm renders the addition, removal, or replacement of robots inconsequential to the overall swarming strategy. To study the repercussions of introducing such heterogeneity in a swarm, we consider the replacement of slow agents with faster ones. Density, connectivity, and memory are all used to uncover the changes brought forth by introducing fast agents onto the swarm's exploration-exploitation balance. Different heterogeneous strategies are also considered as means to improve tracking performance further. The work presented in this chapter has been partially published in Kwa et al. [3] and further integrates later developments.

5.1 Introduction

In addition to more practical considerations, like slowly rolling out an upgrade while leaving functional robots in the field, or replacing destroyed agents with whatever is available, modularity also has much potential to optimize a swarm's performance. This can be done with a focus on cost [73], energy [25], or directly on performance [74]. Although it seems straightforward to replace a few agents with faster ones to improve the swarm's overall performance [16], caution must be employed, as it was also shown that mixing different types of agents could result in worse overall performance [96].

As done in the previous chapter, modifying the connectivity level of a swarm's agents is a well-known tool to adjust its exploration-exploitation balance [23, 93]. Furthermore, memory was determined to have a similar impact [84, 97]. As shown previously, increasing either memory lengths or connectivity levels increases the amount of exploitation a swarm conducts. However, modifying the communication network of a single agent is the least well-known technique, which has been identified as a possible avenue to implement heterogeneity and improve performance [14]. This hints at the possibility of implementing more complex heterogeneous strategies by varying the number of neighbors of certain agents and other parameters, like agent memory length.

Therefore, in this chapter, we combine the work previously accomplished in Chapter 4 [1, 2] with the concept of replacing faster agents with slower ones. As such, agent density, the level of connectivity, memory, and the number of fast agents are all studied for their impact on a heterogeneous swarm's exploration-exploitation balance, as well as overall performance. In addition, the other aforementioned heterogeneous strategies, involving the introduction of varying levels of connectivity and memory lengths, are also studied. Of course, these strategies aim to maximize a swarm's performance, thereby unlocking its potential when dealing with fast-evolving circumstances [3].

To do so, the strategy described in Chapter 3 is used to simulate swarms with agents that can have one of two maximum speeds. Of course, the target's speed is faster than both, as neither agent can be allowed to track the target by itself constantly. Therefore, we confirm the existence of the previously studied density phases and some interesting opportunities to reduce swarm sizes while increasing performance through the use of faster agents. Introducing faster agents is also shown to generally increase performance without altering the tracking strategy, but lowering the optimal level of connectivity k^* makes the strategy less effective than if it was modified to consider the newer, faster agents. This result again raises the need for a truly adaptive strategy that can better exploit a swarm's modularity by optimizing its own parameters to respond to changes in composition.

5.2 Density Phases

One of the possible advantages of modularity is reducing the overall number of agents necessary to accomplish the task at hand by upgrading a select few of them. Of course, reducing the size of the swarm will reduce its density, which, as we've seen, will affect the swarm's exploration-exploitation balance. A density analysis comparing homogeneous swarms with their heterogeneous counterparts was thus undertaken to validate the potential of said heterogeneity.

Beginning with the performance density curves shown in Fig. 5.1, the previously established three phases are clearly visible [3]. Indeed, there is a low-density phase ($< 3 \times 10^{-2}$ agents/unit area) where the swarm is ineffectual at tracking the target. It is then followed by the transition phase, where the swarms' performance improves quickly as density rises. Once performance maxes out (> 1 agent/unit area), swarms enter the high-density phase, where the target is perfectly tracked. These observations confirm the phases framework developed by Kwa et al. [1–3] that was discussed throughout Chapter 4.

In addition to this validation, Fig. 5.1 shows that heterogeneity counteracts the finite-size effects studied in Sec. 4.6. In fact, going from a homogeneous swarm of 50 agents to a heterogeneous one comprised of 30 slow agents and 10 fast ones leads to a large increase in tracking performance. This is due to the fast agents' much-improved exploitation abilities, as they have an easier time keeping up with the target. Therefore, this boost to exploitation counteracts the aforementioned reduction in performance caused by smaller swarms. The increase in performance brought on by heterogeneity goes hand in hand with a similar increase in the swarm's engagement ratio, showing that this extra exploitation potential for the swarm is very effective.

Furthermore, the decrease in performance resulting from removing slow agents from the swarm confirms the importance of considering swarm size in density analysis. Although the proportion of fast agents is increased by removing slow ones, it forces a higher proportion of the swarm's agents to be allocated to exploitation, leaving less room for exploration, a global task, to be undertaken, thus reducing the swarm's overall performance. This suggests that the actual number of fast agents is more important than their proportion respective to the total swarm size, confirming

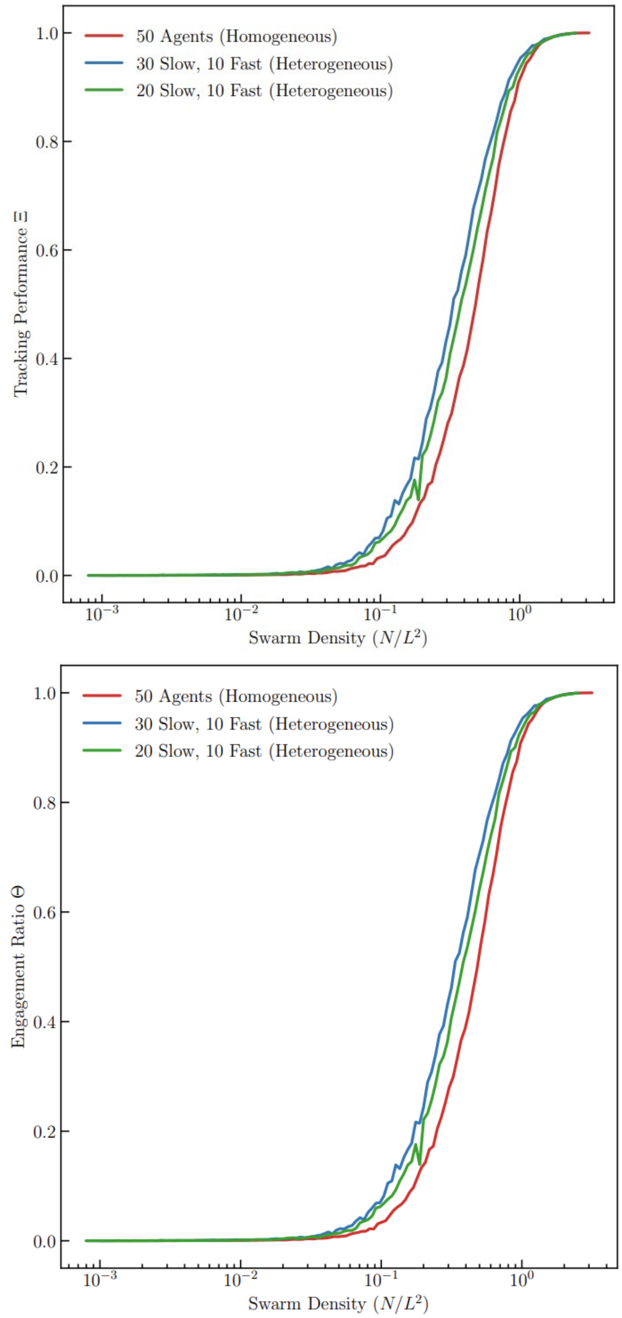


Figure 5.1: Evasive tracking performance (top) and engagement ratio (bottom) of a homogeneous swarm comprised of 50 agents and two heterogeneous swarms comprised of a total of 40 and 30 agents each. Simulations were carried out over various area sizes to vary swarm density with agents using a $k = 14$ network [3].

exploitation status as a local metric.

5.3 Exploration-Exploitation Dynamics

As the existence of an optimum level of connectivity k^* was previously established, it is important to investigate how, if any, heterogeneity affects it. In Fig. 5.2, we notice this k^*, as the tracking performance is maximized for a specific value of k in both the homogeneous and heterogeneous swarms. Again, for $k < k^*$, the swarms end up exploitation-limited, as most of their agents are dedicated to exploration, and few manage to engage with a target. However, for $k > k^*$, the engagement ratio keeps rising as performance falls. This is indicative of the agents over-exploiting the targets, i.e., clustering around them too much, leaving the entire swarm exploration-limited, and hindering its performance. In fact, this extra-clustering makes agents easier to outrun, as they have a harder time finding a target again once they lose track of it.

Next, Fig. 5.2 demonstrates that tracking performance always increases when swapping a homogeneous swarm for a heterogeneous one, no matter the value of k. Thus, it is possible to simply replace slower agents with faster ones without necessitating any adjustments to maintain the swarm's productivity. Such a result had been suggested in the literature, as it was established by Vallegra et al. [9] that no modifications were needed to take advantage of a heterogeneous swarm for a dynamic area coverage task. However, other studies showed that the overall strategy needed to be modified, for example, by changing the communications network, or that specific conditions needed to be initialized to actually realize benefits from a heterogeneous swarm [76, 96]. A more complete explanation of why that is the case is found in Sec. 5.5, as the faster agents' impact on the exploration-exploitation balance is explored in detail.

Figure 5.2 also clearly shows that introducing faster agents to the swarm causes the optimum level of connectivity to go down. While the general tracking strategy is shown to be effective even with these heterogeneous swarms, this reduction in k^* nonetheless signifies that altering the level of connectivity specifically is necessary to maximize the heterogeneous swarm's performance. Therefore, a swarm that takes advantage of the modularity of MRS should possess some kind of mechanism to adapt its level of connectivity when agents of different abilities are added or removed from it. This would enable such a swarm to fully exploit the extra potential endowed to it by the addition of faster agents. The mechanisms behind this phenomenon are further analyzed in Sec. 5.5.

The number of targets is another parameter that was shown to impact a swarm's exploration-exploitation balance. In Fig. 5.2, it is demonstrated that adding targets decreases the swarm's overall performance. This suggests, based on our previous un-

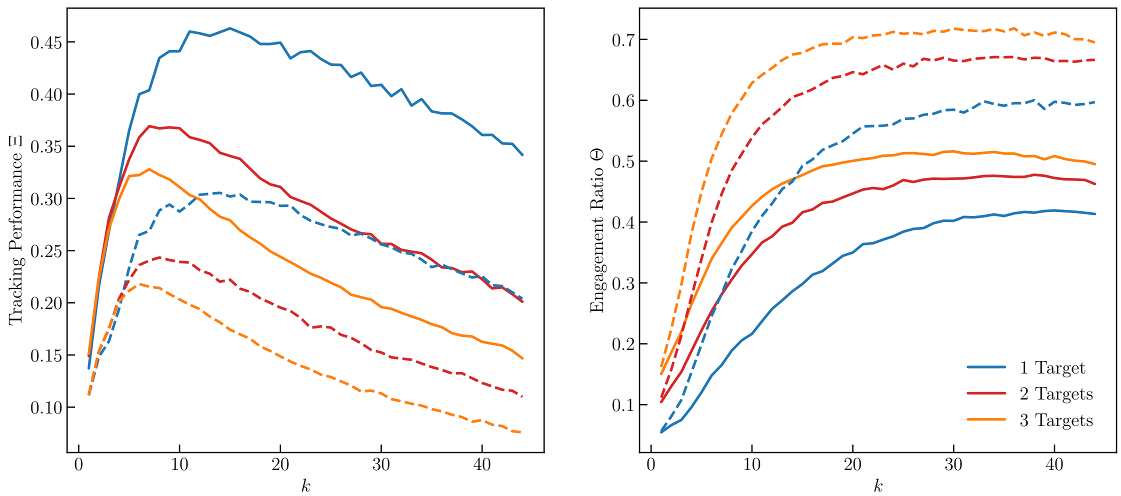


Figure 5.2: Tracking performance (right) and engagement ratio (left) of a heterogeneous swarm comprised of 45 fast agents and 5 slow agents (solid line), as well as a homogeneous swarm comprised of 50 slow agents (dashed line) tracking multiple fast-moving evasive targets. Simulations were performed using various levels of network connectivity k and a memory length of $t_{\mathrm{mem}} = 20$ [3].

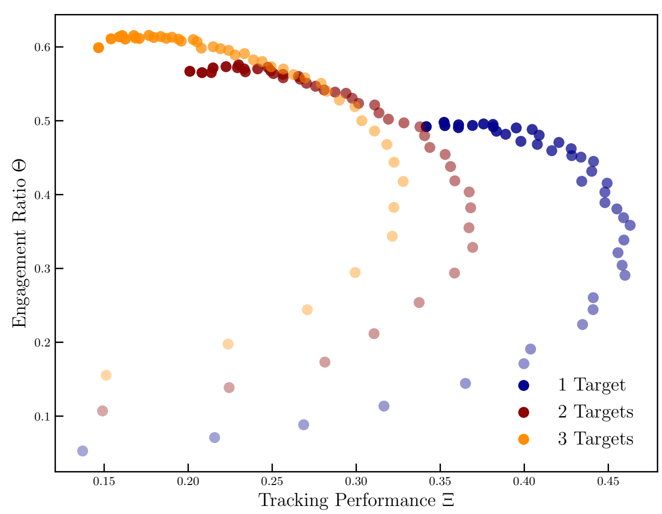


Figure 5.3: Engagement-Tracking plots of a heterogeneous swarm comprised of 15 fast agents tracking multiple fast-moving evasive targets. Darker shaded points indicate swarms communicating through networks with higher values of k [3].

derstanding of multi-target tracking developed in Sec. 4.5, that the swarms simulated are found after the middle point of the transition phase, i.e., in their exploration-limited phase. Indeed, having multiple targets to track is easier for the swarm than a one-target case when this swarm is limited by its exploitation. This is due to the many targets being easier to find and thus exploit, hampering the swarm's exploration abilities.

Additionally, Fig. 5.2 shows that both adding targets and increasing the swarm's level of connectivity results in a higher engagement ratio Θ. This is coherent with our analysis, as tracking multiple targets increases the likelihood of an agent encountering one, therefore increasing the ratio of agents engaged to a target. As the engagement ratio is a global variable that does not differentiate between targets, adding more always results in an increase. In the case of extra connectivity, the target information uncovered by agents gets spread to more of the swarm, resulting in more agents transitioning from exploration to exploitation. Although achieving optimal performance requires a balancing act, Θ allows us to identify excessive exploitation by comparing the gap between overall engagement and performance.

Interestingly, Fig. 5.3 shows that the optimum exploration-exploitation balance remains constant even when targets are added. Yet, as adding more targets increases exploitation while reducing exploration, exploration needs to be increased to achieve this optimal balance and thus maximize performance. This is shown in Fig. 5.2, as the swarm's k^* goes down when adding extra targets.

5.4 Memory

To further study the use of heterogeneity as a means to improve swarm performance, it is important to consider the different memory lengths that can be implemented for the swarm's agents. As making the agents' point of attraction more persistent tends to boost exploitation, which is also the main advantage of introducing faster agents, the interplay between those two upgrades and their impact on the exploration-exploitation balance has to be studied. Furthermore, excessive memory was previously found to have the potential to be detrimental thus great care has to be taken when designing with memory.

Notably, Fig. 5.4 demonstrates that, when keeping the same memory length throughout the swarm, heterogeneity in agent speed does not change how the swarm responds to memory, as the optimal memory length is the same for both the homogeneous and heterogeneous swarms. However, the importance of implementing

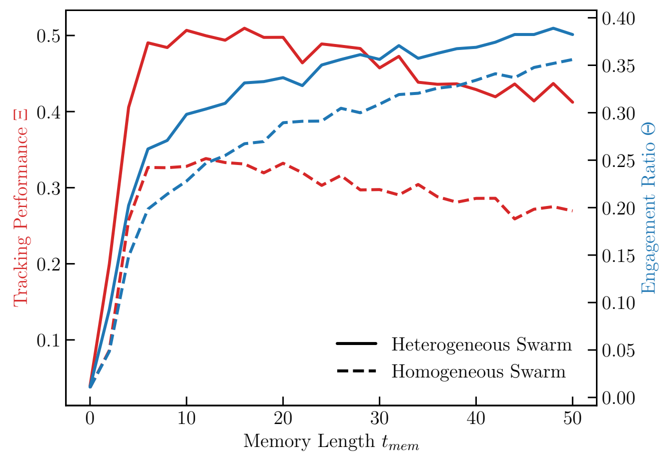


Figure 5.4: Tracking performance (red line) and engagement ratio (blue line) of a heterogeneous swarm comprised of 15 BoB-1 units and 35 BoB-0 units (solid line) and a homogeneous swarm comprised of slow agents (dashed line) tracking a single fast-moving evasive target. Simulations were performed using network connectivity of $k = 14$ and various memory lengths [3].

short-term memory when tracking an evasive target is also made obvious, as swarms with $t_{\text{mem}} = 0$) can simply not track the target for the density chosen. Adding even minimal amounts of memory greatly increases performance and engagement, enabling the swarm to track the target better and better. Indeed, as previously described in Sec. 4.4, memory enables the agents to utilize even very short encounters to their advantage, as the point of attraction generated lasts longer. Without this persistence, the swarm's agents disperse into their static equilibrium, and any encounters with the target are too short-lived to trigger a coordinated response. The target quickly evades its pursuing agent, and the lack of memory means the swarm immediately starts to get back in position.

Nonetheless, Fig. 5.4 also demonstrates the impact of the aforementioned excessive memory. The swarm's tracking performance plateaus around $t_{\text{mem}} = 10$, before degrading as memory length increases. As in Sec. 5.3, the cause behind this decrease is obvious by comparing the tracking performance curves with the engagement ratio. Indeed, as memory length increases past the optimum values, the engagement ratio keeps rising, creating a gap of ineffective exploitation. This excessive exploitation is generated by the age of exploited information, as information becomes too dated past a certain point, thus rendering it harmful to exploit. This shows the importance of maintaining the optimal exploration-exploitation balance mentioned throughout this work.

5.5 Impact from the Introduction of Fast Agents

As previously alluded, the proportion of fast agents in a swarm can be considered to properly generalize our work. This is done in Fig. 5.5, where a linear increase in both the tracking performance and engagement ratio can be observed due to increasing the proportion of fast agents. As the tracking strategy used for both homogeneous and heterogeneous swarms is the same, we can establish that modifying the strategy itself is unnecessary to obtain performance increases.

Analyzing the behavior of the simulated agents gives us a few hints as to why that is the case. In fact, Fig. 5.6 demonstrates that faster agents tend to cluster around the target as a group to exploit, but the adaptive inter-agent repulsion is sufficient to compel the faster agents to disperse around the search space and explore once sight of the target is lost. This increases the odds that a fast agent will encounter the target, as distributed fast agents can deploy quicker after a target is encountered.

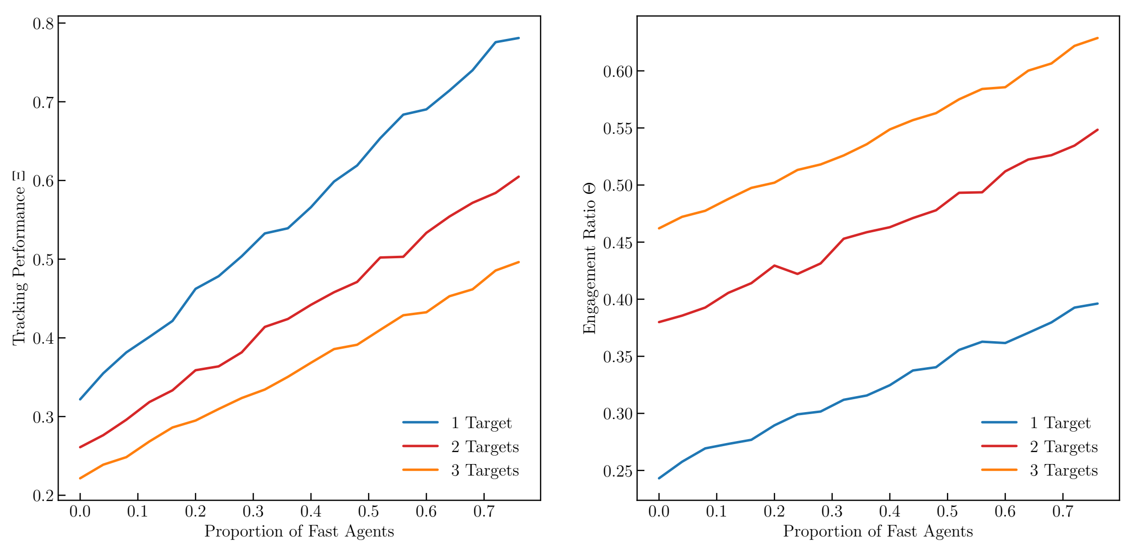


Figure 5.5: Tracking performance (right) and engagement ratio (left) of swarms comprised of 50 agents with various proportions of BoB-1 agents tracking a fast-moving evasive target. Simulations were performed using a network connectivity of $k = 12$ and a memory length of $t_{\text{mem}} = 20$ [3].

From these quantitative results and associated observations, we can conclude that the increased mobility of the faster agents throughout the search space is the main factor behind the increase in exploitation, itself resulting in both an increase in engagement and performance. Since the faster agents are better equipped to track down discovered targets, adding them to a swarm directly increases its tracking

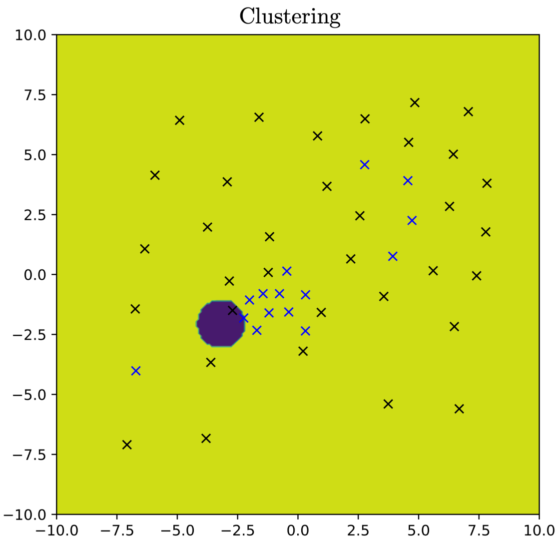


Figure 5.6: Heterogeneous swarm comprised of 35 BoB-0 units (black crosses) and 15 BoB-1 units (blue crosses) tracking an evasive target (dark blue disc). (Left) BoB-1 units clustering while pursuing target; (Right) BoB-1 units redistributed around the environment due to inter-agent repulsion [3].

abilities. As was discussed in Chapter 4, making agents better at tracking down the target compounds throughout the swarm since agents that spend more time on the target can disseminate the information to more of the swarm. This also applies to slower agents, as they get access to better location data, enabling them to remain engaged with the target for longer.

Furthermore, it can be concluded from the increase in the engagement ratio that adding faster agents biases the swarm's exploration-exploitation balance towards exploitation. While this might seem problematic initially, Fig. 5.7 demonstrates this is not the case, as replacing slow agents with faster ones increases the optimum engagement ratio, for which performance is maximized. It can thus be said that the swarm's optimum exploration-exploitation balance is also shifted towards higher exploitation. This phenomenon is explained by the behavior of exploring agents, as the swarm's higher mobility makes the non-engaged agents more efficient at exploring, thus accomplishing more with less.

Regardless of what could be expected from this apparent shift in favor of exploitation in the optimum exploration-exploration balance, Fig. 5.7 also demonstrates a reduction in the optimum level of connectivity k^*. Indeed, this optimum level goes from $k^* \sim 15$ when using all slow agents to $k^* \sim 10$ when replacing 45 slow agents with fast ones. Such a decrease in k^* indicates that the swarm needs to boost its exploration by lowering its level of connectivity. This is congruent with previous work in the literature on non-evasive targets [16], although the change is shown to be less extensive here with evasive ones. It indicates that replacing slow agents with fast ones increases the swarm's capability, but also propensity, to exploit target information [3]. Thus, to balance the extra exploitative actions being carried out, the swarm needs to increase its level of exploration. As a result, the optimum level of connectivity k^* is decreased.

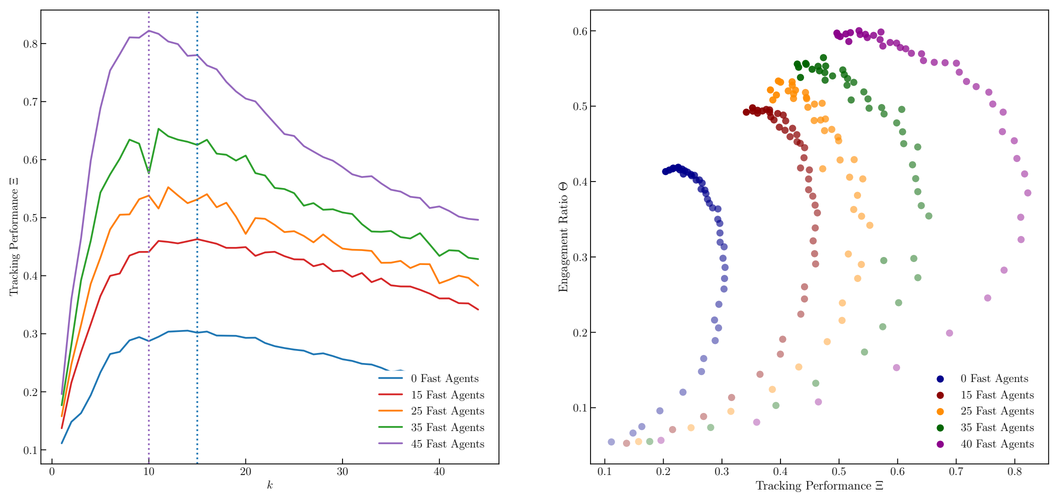


Figure 5.7: Swarms comprised of 50 agents with different agent compositions of BoB-1 and BoB-0 units tracking a single fast-moving evasive target. A system with 0 fast agents is a homogeneous swarm comprised solely of BoB-0 units. (Left) Tracking system performance while operating with different levels of connectivity k. (Right) Engagement-Tracking plots where darker shaded points indicate swarms communicating through networks with higher values of k [3].

5.6 Heterogeneous Strategies

Another strategy under consideration to increase swarm performance is to upgrade or downgrade the memory of faster agents, thus developing a heterogeneous memory strategy. To investigate this possibility, the optimum memory lengths of swarms with agents of different maximum speeds were tested, as demonstrated in Fig. 5.8. Interestingly, the optimum memory length is constant throughout, thus suggesting that such a heterogeneous memory length strategy would not improve a swarm's performance.

Finally, using a heterogeneous strategy, mixing different levels of connectivity was

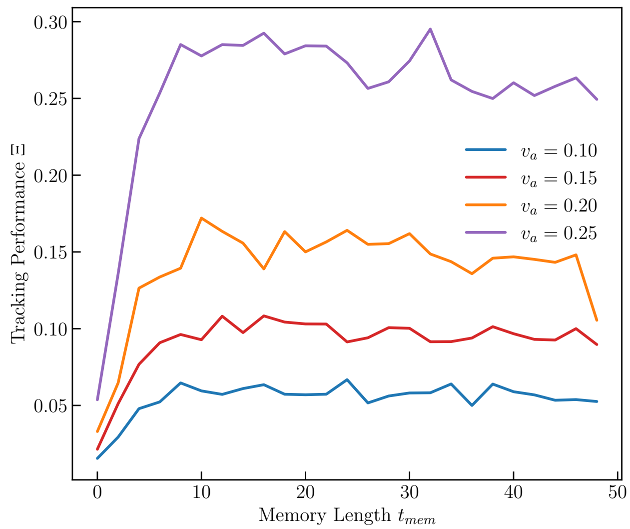


Figure 5.8: Evasive tracking performance of homogeneous swarms with different v_a, max and memory lengths. Simulations were run at $k = 14$ and $v_{o,\max} = 0.3$ [3].

considered. Indeed, it was shown in Sec. 5.5 that adding faster agents to a swarm decreases its optimal level of connectivity k^*, and the literature presents evidence that the optimal level of connectivity also varies with the agents' driving speed [13, 33]. Therefore, using the same general strategy, we proceeded to simulate swarms with a distinct slow (k_s) and fast (k_f) agent level of connectivity.

Using Fig. 5.9, an optimal combination of levels of connectivity $k_s = k_f = 10$ can be identified. Of all the simulated cases, none provided any credence to the theory that using different values for k_s and k_f could positively impact the swarm's performance. Consequently, this strategy was abandoned as well.

Considering the failure of both heterogeneous strategies trialed, it is clear that improving the heterogeneous tracking strategy further needs to involve more complex changes. A way forward would be to modify the more intrinsic rules behind agent behaviors. Possible examples include differentiated task allocation based on agent's placement, prediction of target movement patterns, etc [3].

5.7 Discussion

In this chapter, the trialing of our heterogeneous tracking strategy confirmed the previously defined three density phases. By adding faster agents, the finite-size

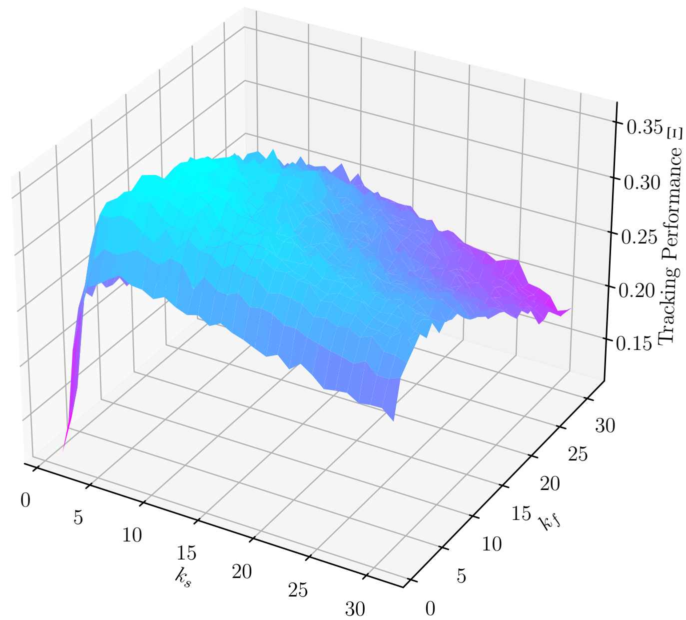


Figure 5.9: Single evasive target tracking performance of a heterogeneous comprised of 30 slow agents and 10 fast agents with slow and fast agents, each having different levels of connectivity [3].

effects described in Sec. 4.6 were quelled by the increased efficiency of exploitation, thus opening the door to smaller, more performant swarms. As faster agents boost exploitation, which was shown to be a local-level task in Chapter 4, their overall number was demonstrated to be more important than their proportion to boost performance.

When considering adding faster agents, the first consideration should be whether or not it will improve the target tracking strategy, and what needs to be changed to take advantage of them. It was clear that replacing slower agents with faster ones increases performance. However, the exploration-exploitation balance needed to be considered to optimize the swarm strategy. Varying the level of connectivity k revealed that its optimum value, k^x, actually decreases as faster agents are added. Therefore, an adaptive strategy, which modulates its level of connectivity in function of the speed of its agents, should be considered. However, modifying the number of targets did not change the exploration-exploitation balance, although the boost of exploitation of extra targets needs to be counteracted with a lower level of connectivity.

Conversely, the optimal memory length was shown not to change when introducing heterogeneity. However, the importance of memory when tracking evasive targets was shown once again, as small amounts of memory proved vital to overall swarm performance. By pushing memory even further than in Chapter 4, its potential to decrease performance was demonstrated without a doubt, as it keeps raising the engagement ratio past its optimum point.

By analyzing the proportion of fast agents, it was shown that the extra speed makes faster agents better and more likely to exploit target information. This increases the engagement ratio, but the extra speed also makes agents better at exploring. However, the overall impact of these agents is to bias the swarm towards exploitation, which is congruent with the reduction in the optimal level of connectivity.

Finally, different heterogeneous strategies were considered, either modifying the faster agent's memory or level of connectivity. As was previously mentioned, the optimum memory length does not seem to change with the mixing of agents of different maximum speeds, leading to the abandonment of that strategy. Varying the level of connectivity did not yield better results, as the best-performing strategy ended up being the one where the level of connectivity of slow and fast agents matched. These results, while disappointing, point towards the consideration of more adaptive strategies that can modify agents' parameters throughout operations based on environmental conditions.

Chapter 6

Conclusions and Future Work Recommendations

6.1 Summary of Findings

This work shows how to optimize the performance of swarming MRS in the context of searching and tracking a fast target. While the problem of more adaptive optimization will be addressed in the subsequent sections, our results have demonstrated how to combine different strategies to maximize results in varied operating environments through the use of the exploration-exploitation balance.

By analyzing performance in relation to swarm density, three density 'phases', which correspond to the MRS's tracking ability at any given density, were uncovered: (1) the low-density phase, where agents are unable to cooperate and meaningfully track the target, (2) the high-density phase, where the omnipresence of agents causes the target to be consistently tracked, and (3) the transition phase, where emergent behavior starts to appear in the form of clustering with increased swarm density, leading to rapidly increasing performance. Identifying the phase a swarm is operating in is vital to optimizing performance, as the operating strategy can only truly be optimized in the transition phase. It also raises the key design consideration of using available means to shift the transition phase to lower densities, therefore increasing performance for the same number of robots or maintaining performance while using fewer.

Using both the exploration and engagement ratios, optimal levels of connectivity k and the parameters affecting them were identified. This allowed for the characterization of an 'exploitation-limited' and an 'exploration-limited' regime, corresponding respectively to the transition phase's lower and higher density halves. As an increased level of connectivity biases the swarm towards exploitation, performance is increased in the exploitation-limited regime and decreased in the exploration-limited one. Similarly, memory, which was shown to be vital to the tracking of a fast agent, also increases exploitation, thereby reducing the optimal level of connectivity needed to achieve balance, i.e., maximizing performance. By allowing the characterization of parameters within this framework, this understanding of the exploration-exploitation balance provides an important design tool to practitioners, as it is the key to maximizing tracking performance [17].

While most work done throughout this thesis considered the normalized swarm density or proportion of fast agents, it is well known that finite-size effects are an important consideration, especially in the smaller swarms considered here. This was substantiated by our results, as exploitation was shown to be a local task, dependent on the number of agents currently exploiting. At the same time, exploration is a global task that depends on the proportion of the swarm allocated to exploration. Counter-intuitively, this results in a larger swarm's increased capacity for exploration, as a larger portion of its agents can explore. Therefore, it follows that smaller systems, which require a higher proportion of their agents committed to exploitation, have a lower potential maximum performance than larger ones at equal swarm density.

The previously mentioned findings were conclusively verified by introducing faster agents to the simulated swarm. Interestingly, the potential for a reduction in performance when introducing heterogeneity to a swarm that was identified for other tasks was not found in the searching and tracking context. As faster agents are much more fitted to tracking a target than their slower counterparts, both in exploiting its information and continuously staying in its vicinity, a one-to-one replacement of a slower agent with a faster one was found to be universally positive. However, such a replacement impacts the swarm's exploration-exploitation balance, as these agents bias the swarm's abilities towards exploitation. Therefore, optimizing swarm performance requires the searching and tracking strategy to be altered to boost exploration and bring back balance, which can be achieved through a reduction in the level of connectivity, as demonstrated through the heterogeneous swarm's lower k^*.

6.2 Discussion

Although we believe the general notions behind our conclusions will hold true for other tasks, our work has been limited to simulations of our specific searching and tracking problem. In fact, we are confident that such concepts as density phases [66], local and global behaviors [53, 54], or exploration-exploitation balance [13, 15, 16, 20, 32, 33], which were all hinted at in the literature, are general to many swarming problems. However, this remains to be proven, and conclusions reached on more specific topics, such as using memory to boost exploitation and the degradation in performance provided by excess memory, are already known to be fairly task-dependent.

Even in the specific realm of searching and tracking problems, our specific strategy was only tested in simulations. More practical issues, such as imperfect transfer of information and limitations to communication ranges, were thus not considered in this thesis. This could prove problematic when tracking a physical target, especially at the very low agent densities that a large environment might require. This adds a layer of complexity to selecting an optimum level of connectivity, as taking a strategy developed for a simulated swarm to a physical one will require handling such issues. The simplest way to achieve this would be to use another parameter to boost exploitation, though doing so brings back the issue of centralization, as local agent knowledge is, and has to be, limited. Other practical considerations have even deeper ramifications on the strategy employed, such as equipment failing, the sensing of noise, and even the presence of obstacles in the search space.

Another important limitation to our work is that all MRS were simulated in a two-dimensional search-space. While the methodology was originally developed to simulate the dynamics of a group of land robots [13], the recent surge in the popularity of drones and the far-reaching implications of our conclusions point to the necessity of the development of a more complete three-dimensional analysis. We expect most of our conclusions to extend to three-dimensional problems, as some intrinsic properties of the swarming system are unaffected by this extra degree of freedom. Nonetheless, it is well known that any extension of the dimensionality of the problem is never truly straightforward, and hence further studies would be required.

6.3 Adaptivity

Analyzing these results, their limitations, and the literature, it is clear that the future outlook of current swarming strategies to solve problems is limited by their lack of adaptivity, or the ability to learn or change behavior to respond to new operating conditions [4]. Swarms mostly exhibit flexibility or the capacity to perform tasks that depart from those chosen at design time. At the same time, adaptivity is limited to the so-called 'narrow' adaptivity, where a swarm's ability to adapt is still limited to a narrow subset of the general problem that it is tasked to solve [4].

This is due to the current paradigm of optimizing operations over specific conditions, which limits the swarm's overall ability to be adaptive. Therefore, there is still a wide gap between current strategies and the so-called 'general' swarm intelligence that has been the target of many practitioners. When comparing the typical methods used to generate adaptive artificial intelligence models to the ones used to optimize MRS in swarming problems, the reason for this bias towards 'narrow' adaptivity is clear: centralization. Indeed, the very definition of swarms relies on the decentralization of the system, which, by making the collective actions of a swarm so remote from its programming, makes SI more complicated to develop than AI to achieve equivalent results [98–100].

6.4 Leveraging Adaptivity

To solve the problem of narrow optimization and implement truly adaptive swarm strategies, practitioners need to move away from the classically used robust controllers [101], or the 'offline' methods [98], which enable swarms to adapt their behavior when operating conditions change within the expected range [4]. Instead, adaptive controllers [101], or the 'online' methods [98], which adjust the aforementioned robust controller itself by implementing new rules or goals, would allow for the development of more general adaptivity or swarm intelligence. The difference between these two approaches is shown in Fig. 6.1 [4].

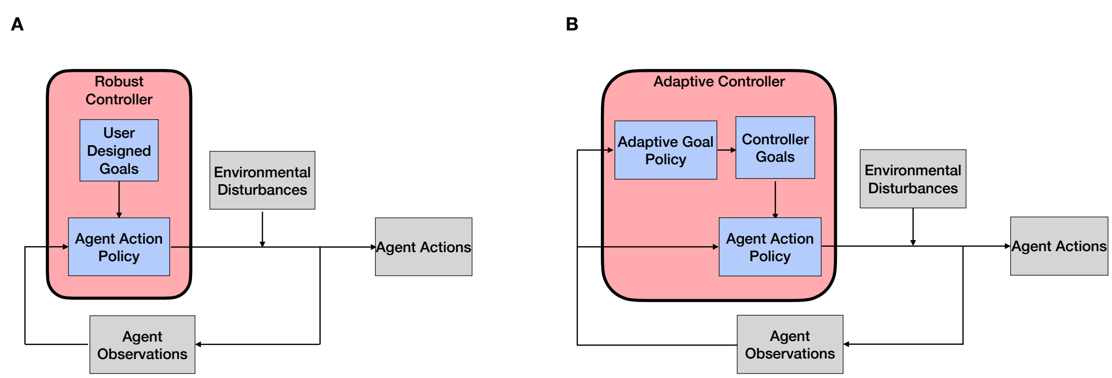


Figure 6.1: Flowchart of controllers used in swarming MRS to promote adaptivity. **A**, Robust controllers. **B**, Adaptive controllers [4].

As a next step towards such advanced controllers, multi-agent reinforcement learning (MARL) is expected to enable swarms to respond to the effects of dynamic environments, allowing for their use in more open-world situations. This is already being undertaken by some groups [12, 102–104], while others are developing ways to use transfer learning from a task to another [105, 106]. Combining these approaches is key to reaching the next level of adaptivity, as general SI requires a swarm to be able to transition between tasks without external input. Automated reinforcement learning (AutoRL) also offers such potential, hoping to enable a swarm to train itself [107, 108].

In conjunction with these advancements, quantifying adaptivity and comparing swarms executing various tasks is severely needed. This problem was solved in the AI and computational optimization communities by implementing a series of evolving benchmark problems [4], such as OpenAI's Procgen Benchmark [109]. Similarly, it is imperative to get past the current restrictive testing comparing the performance of different strategies over a narrow range of conditions and develop a complete standardized suite of benchmark problems. To do so, we propose the benchmark framework seen in Fig. 6.2, in which benchmark problems are associated with their respective parameters and performance [4]. Such a framework would allow practitioners to evaluate the adaptivity of different strategies in varied contexts. By combining a new approach favoring adaptivity and the development of effective means to test it, the community will unlock a new pathway towards the design of successful swarms.

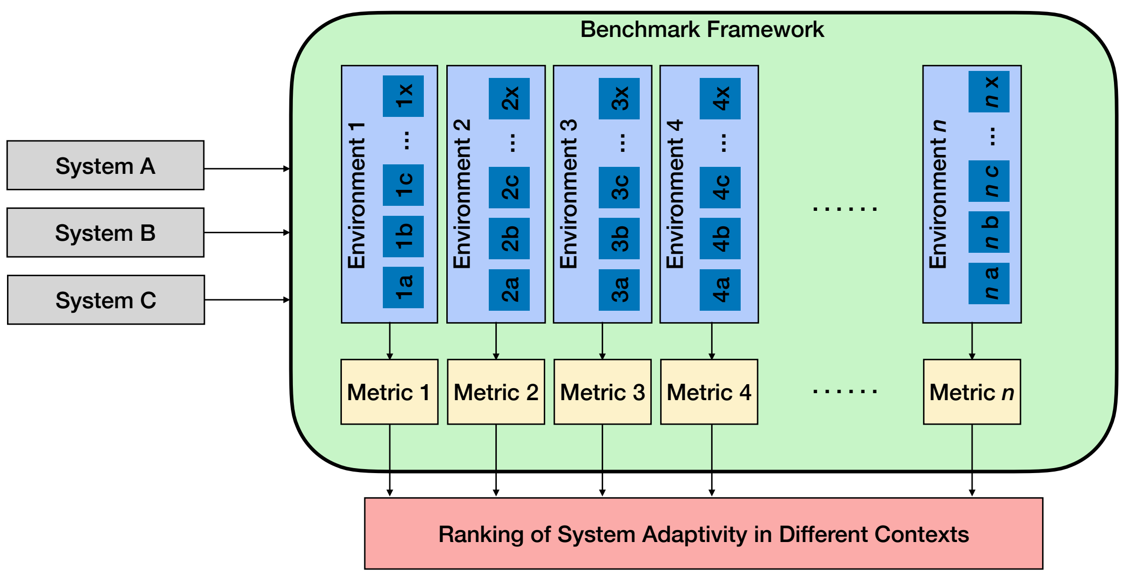


Figure 6.2: A flowchart describing our proposed benchmark framework. The systems to be compared are tested in multiple environments (Environments 1, 2, ..., n), each with different environmental parameters $(a, b, \ldots, x)$. Their performances are determined by various metrics specific to each environment, thereby allowing the ranking of the systems' adaptivity in different contexts [4].

Bibliography

[1] H. L. Kwa, J. Philippot, and R. Bouffanais, “Effect of swarm density on collective tracking performance,” *Swarm Intelligence*, vol. 17, no. 3, pp. 253–281, 2023.

[2] ——, “The impact of agent density and environmental factors on target tracking swarms,” in *Artificial Life Conference Proceedings 35*, vol. 2023, no. 1. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info@ mit. edu, 2023, p. 88.

[3] H. L. Kwa, V. Babineau, J. Philippot, and R. Bouffanais, “Adapting the exploration-exploitation balance in heterogeneous swarms: Tracking evasive targets,” *Artificial Life*, vol. 29, pp. 1–16, 2022.

[4] H. L. Kwa, J. L. Kit, N. Horsevad, J. Philippot, M. Savari, and R. Bouffanais, “Adaptivity: a path towards general swarm intelligence?” *Frontiers in Robotics and AI*, vol. 10, p. 1163185, 2023.

[5] 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.

[6] 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.

[7] R. Bouffanais, *Design and Control of Swarm Dynamics.* Singapore: Springer Singapore, 2016.

[8] R. N. Darmanin and M. K. Bugeja, “A review on multi-robot systems categorised by application domain,” in *2017 25th mediterranean conference on control and automation (MED).* IEEE, 2017, pp. 701–706.

[9] 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, 2018.

[10] 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.

[11] D. S. López, G. Moreno, J. Cordero, J. Sanchez, S. Govindaraj, M. M. Marques, V. Lobo, S. Fioravanti, A. Grati, K. Rudin *et al.*, "Interoperability in a heterogeneous team of search and rescue robots," *Search and Rescue Robotics-From Theory to Practice*, pp. 93–125, 2017.

[12] M. Kouzehgar, M. Meghjani, and R. Bouffanais, "Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys," in *IEEE-MTS Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–8.

[13] 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.

[14] ——, "Balancing collective exploration and exploitation in multi-agent and multi-robot systems: A review," *Frontiers in Robotics and AI*, vol. 8, no. 771520, 2022.

[15] H. Kwa, J. 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*. MIT Press, 2021.

[16] 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*. Singapore: IEEE, 2020.

[17] H. L. KWA, "The exploration and exploitation dynamics of target tracking swarms," Ph.D. dissertation, Singapore University of Technology and Design, 2022.

[18] 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.

[19] 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.

[20] 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)*. New Brunswick, NJ, USA: IEEE, 2019.

[21] 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).* Springer Cham, 2021.

[22] Z. Liu, M. Crosscombe, and J. Lawry, "Imprecise fusion operators for collective learning," in *ALIFE 2021: The 2021 Conference on Artificial Life*, 2021.

[23] 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.

[24] 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.

[25] L. Strickland, K. Baudier, K. Bowers, T. P. Pavlic, and C. Pippin, "Bio-inspired role allocation of heterogeneous teams in a site defense task," in *Distributed Autonomous Robotic Systems 2018.* Boulder, CO, USA: Springer International Publishing, 2018.

[26] 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.

[27] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics.* Springer, 2004, pp. 10–20.

[28] 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. Morgantin, 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.

[29] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019.

[30] 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.

[31] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1232–1237, 2008.

[32] D. Mateo, Y. K. Kuan, and R. Bouffanais, "Effect of correlations in swarms on collective response," *Scientific Reports*, vol. 7, no. 10388, 2017.

[33] 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.

[34] 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, 2018.

[35] 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)*. IEEE, 2020, pp. 2486–2492.

[36] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 3, Alburquerque, NM, 1997, pp. 2082–2089.

[37] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 115–121.

[38] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 187–198, 2018.

[39] L. Esterle and P. R. Lewis, "Distributed autonomy and trade-offs in online multiobject k-coverage," *Computational Intelligence*, vol. 36, no. 2, pp. 720–742, 2020.

[40] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek, "Group chasing tactics: how to catch a faster prey," *New Journal of Physics*, vol. 19, no. 5, p. 053003, 2017.

[41] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*. IEEE, 1997, pp. 303–308.

[42] 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.

[43] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 200–213, 2014.

[44] 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.

[45] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization algorithms," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 327–332.

[46] A. R. Jordehi, "Particle swarm optimisation for dynamic optimisation problems: A review," *Neural Computing and Applications*, vol. 25, pp. 1507–1516, 2014.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] F. Van Den Bergh and A. P. Engelbrecht, "Effects of swarm size on cooperative particle swarm optimisers," in *GECCO'01: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, San Fransisco, CA, USA, 2001, pp. 892–899.

[54] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 58, p. 100718, 2020.

[55] T. M. Schaerf, J. C. Makinson, M. R. Myerscough, and M. Beekman, "Do small swarms have an advantage when house hunting? the effect of swarm size on nest-site selection by apis mellifera," *Journal of the Royal Society Interface*, vol. 10, no. 87, 2013.

[56] J. Prasetyo, G. De Masi, and E. Ferrante, "Collective decision making in dynamic environments," *Swarm Intelligence*, vol. 13, pp. 217–243, 2019.

[57] M. Jurt, E. Milner, M. Sooriyabandara, and S. Hauert, "Collective transport of arbitrarily shaped objects using robot swarms," *Artificial Life and Robotics*, 2022.

[58] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, 2001.

[59] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar, "Distributed simultaneous action and target assignment for multi-robot multi-target tracking," in *Proceedings - IEEE International Conference on Robotics and Automation*. Brisbane, Australia: IEEE, 2018, pp. 3724–3729.

[60] ——, "Distributed assignment with limited communication for multi-robot multi-target tracking," *Autonomous Robots*, vol. 44, pp. 57–73, 2020.

[61] O. Roeva, S. Fidanova, and M. Paprzycki, "Population size influence on the genetic and ant algorithms performance in case of cultivation process modeling," in *Recent advances in computational optimization*. Springer, 2015, pp. 107–120.

[62] H. Hamann, "Towards swarm calculus: Universal properties of swarm performance and collective decisions," in *Swarm Intelligence: 8th International Conference, ANTS 2012. Volume 7461 of LNCS.* Berlin, Germany: Springer, 2012, pp. 168–179.

[63] A. Schroeder, B. Trease, and A. Arsie, "Balancing robot swarm cost and interference effects by varying robot quantity and size," *Swarm Intelligence*, vol. 13, no. 1, pp. 1–19, 2019.

[64] J. P. Hecker and M. E. Moses, "Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms," *Swarm Intelligence*, vol. 9, no. 1, pp. 43–70, 2015.

[65] A. Rosenfeld, G. A. Kaminka, and S. Kraus, "A study of scalability properties in robotic teams," in *Coordination of Large-Scale Multiagent Systems*, P. Scerri, R. Vincent, and R. Mailler, Eds. Boston, MA, USA: Springer, 2006, pp. 27–51.

[66] H. Hamann, "Superlinear scalability in parallel computing and multi-robot systems: Shared resources, collaboration, and network topology," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10793 LNCS, pp. 31–42, 2018.

[67] M. E. Cates and J. Tailleur, "Motility-induced phase separation," *Annual Review of Condensed Matter Physics*, vol. 6, no. 1, pp. 219–244, 2015.

[68] 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.

[69] H. Li, C. Feng, H. Ehrhard, Y. Shen, B. Cobos, F. Zhang, K. Elamvazhuthi, S. Berman, M. Haberland, and A. L. Bertozzi, "Decentralized stochastic control of robotic swarm density: Theory, simulation, and experiment," in *IEEE International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, 2017, pp. 4341–4347.

[70] 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.

[71] S. Zhang, M. Y. Liu, X. K. Lei, and Y. K. Huang, "Dynamics and motion patterns of a k-capture game with attraction-repulsion interaction," *EPL (Europhysics Letters)*, vol. 128, no. 10003, 2019.

[72] M. Dadgar, M. S. Couceiro, and A. Hamzeh, "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.

[73] A. L. Christensen, S. Oliveira, O. Postolache, M. J. de Oliveira, S. Sargento, P. Santana, L. Nunes, F. Velez, P. Sebastião, V. Costa *et al.*, "Design of communication and control for swarms of aquatic surface drones," in *International Conference on Agents and Artificial Intelligence*, vol. 2. Scitepress, 2015, pp. 548–555.

[74] Y. Shi, N. Wang, J. Zheng, Y. Zhang, S. Yi, W. Luo, and K. Sycara, "Adaptive informative sampling with environment partitioning for heterogeneous multi-robot systems," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 11 718–11 723.

[75] T. Iwama and M. Sato, "Group chase and escape with some fast chasers," *Physical Review E*, vol. 86, no. 6, p. 067102, 2012.

[76] Y. Altshuler, "To add with caution - decreasing a swarm robotics' efficiency by imprudently enhancing the robots' capabilities," in *ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents*. Wellington, New Zealand: IEEE, 2009, pp. 351–356.

[77] H. L. Kwa and R. Bouffanais, "The effect of network connectivity on exploration and exploitation during decentralized collective learning," in *2022 International Workshop on Agent-Based Modelling of Human Behaviour (ABMHuB)*, Online, 2022. [Online]. Available: http://abmhub.cs.ucl.ac.uk/2022/camera_ready/Kwa_Bouffanais.pdf

[78] 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.

[79] G. Rossides, B. Metcalfe, and A. Hunter, "Particle swarm optimization-an adaptation for the control of robotic swarms," *Robotics*, vol. 10, no. 58, 2021.

[80] 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.

[81] E. A. Jensen, L. Lowmanstone, and M. Gini, "Communication-restricted exploration for search teams," *Distributed Autonomous Robotic Systems. Springer Proceedings in Advanced Robotics, vol 6*, pp. 17–30, 2018.

[82] N. Horsevad, D. Mateo, R. E. Kooij, A. Barrat, and R. Bouffanais, "Transition from simple to complex contagion in collective decision-making," *Nature Communications*, vol. 13, no. 1442, 2022.

[83] 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.

[84] 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.

[85] A. P. Engelbrecht, "Heterogeneous particle swarm optimization," in *7th Int. Conf. ANTS 2010*, Dorigo, M. et al, Ed. Brussels, Belgium: Springer, Berlin, Heidelberg, 2010, pp. 191–202.

[86] H. Hamann, *Swarm Robotics: A Formal Approach.* Springer International Publishing, 2018.

[87] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, and M. Sende, "Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends," *Swarm and Evolutionary Computation*, vol. 60, no. 100762, 2021.

[88] C. de Souza, P. Castillo, and B. Vidolov, "Local interaction and navigation guidance for hunters drones : A chase behavior approach with real-time tests," *Robotica*, vol. 40, no. 8, pp. 1–19, 2022.

[89] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, "Kilobot: A low cost robot with scalable operations designed for collective behaviors," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 966–975, 2014.

[90] 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*, vol. 29, no. 3, 2020.

[91] 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, pp. 132–135, 1995.

[92] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-filho, and R. Menezes, "Better exploration-exploitation pace, better swarm: Examining the social interactions," in *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI).* Arequipa, Peru: IEEE, 2017.

[93] 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.

[94] 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.

[95] N. Horsevad, H. L. Kwa, and R. Bouffanais, "Beyond bio-inspired robotics : How multi-robot systems can support research on collective animal behavior," *Frontiers in Robotics and AI*, vol. 9, no. 865414, 2022.

[96] N. Nishikawa, R. Suzuki, and T. Arita, "Coordination control design of heterogeneous swarm robots by means of task - oriented optimization," *Artificial Life and Robotics*, vol. 21, pp. 57–68, 2016.

[97] 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.

[98] M. Birattari, A. Ligot, and K. Hasselmann, "Disentangling automatic and semi-automatic approaches to the optimization-based design of control software for robot swarms," *Nature Machine Intelligence*, vol. 2, pp. 494–499, 2020.

[99] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[100] G. Bianconi, A. Arenas, J. Biamonte, L. D. Carr, B. Kahng, J. Kertesz, J. Kurths, L. Lü, C. Masoller, A. E. Motter, M. Perc, F. Radicchi, R. Ramaswamy, F. A. Rodrigues, M. Sales-Pardo, M. San Miguel, S. Thurner, and T. Yasseri, "Complex systems in the spotlight : next steps after the 2021 Nobel Prize in Physics Complex systems in the spotlight : next steps after the 2021 Nobel Prize in Physics," *J. Phys. Complex*, vol. 4, no. 010201, 2023.

[101] J. D. Hasbach and M. Bennewitz, "The design of self-organizing human–swarm intelligence," *Adaptive Behavior*, vol. 30, no. 4, pp. 361–386, 2022.

[102] G. Wang, F. Wei, Y. Jiang, M. Zhao, K. Wang, and H. Qi, "A multi-AUV maritime target search method for moving and invisible objects based on multi-agent deep reinforcement learning," *Sensors*, vol. 22, no. 21, p. 8562, 2022.

[103] Y. Wang, M. Damani, P. Wang, and Y. Cao, "Distributed Reinforcement Learning for Robot Teams: A review," *Current Robotics Reports*, vol. 3, pp. 239–257, 2022.

[104] M. Kouzeghar, Y. Song, M. Meghjani, and R. Bouffanais, "Multi-target pursuit by a decentralized heterogeneous uav swarm using deep multi-agent reinforcement learning," in *2023 International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[105] T. G. Karimpanal and R. Bouffanais, "Experience replay using transition sequences," *Frontiers in Neurorobotics*, vol. 12, p. 32, 2018.

[106] T. 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.

[107] A. Faust, A. Francis, and D. Mehta, "Evolving Rewards to Automate Reinforcement Learning," in *6th ICML Workshop on Automated Machine Learning*, Long Beach, CA, USA, 2019. [Online]. Available: http://arxiv.org/abs/1905.07628

[108] J. Parker-Holder, R. Rajan, X. Song, A. Biedenkapp, Y. Miao, T. Eimer, B. Zhang, V. Nguyen, R. Calandra, A. Faust, F. Hutter, and M. Lindauer, "Automated reinforcement learning (AutoRL): A survey and open problems," *Journal of Artificial Intelligence Research*, vol. 74, pp. 517–568, 2022.

[109] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging procedural generation to benchmark reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2048–2056.