VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

INFORMATICS STUDY PROGRAM

Quantitative evaluation of multi-agent systems using the foraging ants model and automated simulation techniques.
Skaitinis daugiaagenčių sistemų įvertinimas naudojant maisto ieškančių skruzdžių modelį ir automatizuoto simuliavimo metodus.

Master's thesis

Author: Aymerick FIKEU ELAT
VU e-mail: aymerick.elat@mif.stud.vu.lt

Supervisor: Prof.dr Linas Laibinis
Reviewer: Prof.dr Linas Litvinas

Vilnius - 2025

Santrauka

Šiame tyrime pateikiamas kiekybinis daugiaveiksnių sistemų vertinimas, remiantis skruzdėlių maisto paieškos elgsena, įgyvendinta naudojant agentais pagrįstą modeliavimo sistemą su SimPy – „Python“ simuliacijos įrankiu. Modeliuojant pavienes skruzdėles kaip autonominius agentus, veikiančius 2D tinkle pagrįstoje ekosistemoje, tiriama, kaip paprastos elgsenos taisyklės lemia kolektyvinį intelektą. Sistema leidžia sistemingai testuoti aplinkos parametrus, tokius kaip plėšrūnų tankis, maisto pasiskirstymas ir feromonų skilimo greitis, siekiant kiekybiškai įvertinti jų poveikį maisto paieškos efektyvumui ir kolonijos atsparumui. Simuliacija suteikia įžvalgų apie decentralizuoto koordinavimo atsparumą priešiškomis sąlygomis ir kaip plėšrūnų spaudimas keičia paieškos elgseną. Galiausiai, testavimas ekstremaliomis sąlygomis patvirtina kolonijos atsparumą, tačiau taip pat parodo našumo sumažėjimą, kai sutampa keli streso veiksniai. Apskritai, simuliacija patvirtina, kad paprastos vietinės taisyklės ir efektyvus bendravimas gali sukurti tvirtas ir mastelio atžvilgiu pritaikomas maisto paieškos strategijas. Ši simuliacija ne tik atskleidžia saviorganizacijos principus biologinėse sistemose, bet ir tarnauja kaip universali platforma spiečiaus intelekto tyrimams taikomuose kontekstuose, tokiuose kaip robotika, dirbtinis intelektas ar optimizavimo algoritmai.

Raktiniai žodžiai: simuliacija, daugiaagentės sistemos, skruzdėlė, maisto paieška, elgsena, autonominiai, agentai, taisyklės, kolektyvinis intelektas, feromonas, decentralizuotas koordinavimas, mastelio keitimas, biologinės sistemos, saviorganizacija, spiečiaus intelektas.

Abstract

This study presents a quantitative evaluation of multi-agent systems through the lens of ant foraging behavior, implemented via an agent-based simulation framework using SimPy, a Python simulation tool. By modeling individual ants as autonomous agents interacting within a 2D grid-based ecosystem, we investigate how simple behavioral rules give rise to emergent collective intelligence. The framework allows systematic testing of environmental parameters, such as predator density, food distribution, and pheromone decay rates, to quantify their impact on foraging efficiency and colony resilience. The simulation provides insights into the resilience of decentralized coordination under hostile conditions and how predator pressure reshapes foraging behaviors. Finally, stress-testing under extreme conditions confirms the colony's resilience, but also underscores performance degradation when multiple stressors coincide. Overall, the simulation confirms that simple local rules and effective communication can generate robust and scalable foraging strategies. This simulation not only demonstrates the principles of self-organization in biological systems but also serves as a versatile foundation for studying swarm intelligence in applied contexts such as robotics, AI or optimization algorithms.

Keywords: simulation, multi-agent systems, ant, foraging, behavior, autonomous, agents, rules, collective intelligence, pheromone, decentralized coordination, scalable, biological systems, self-organization, swarm intelligence.

Acknowledgements

I would like to express my deepest gratitude to my academic supervisor, Prof.dr Linas Laibinis, for his invaluable guidance, unwavering support, and insightful expertise throughout this academic journey. his encouragements and constructive feedbacks have been instrumental in shaping this research, and I am truly appreciative of his dedication.

Furthermore, I extend my sincere appreciation to Vilnius University for providing an enriching academic environment that has fostered my growth and learning. The resources, MIF faculty, and fellow students have played a crucial role in making this experience both fulfilling and inspiring.

I also wish to acknowledge the Republic of Lithuania for the opportunities and support it has provided to me as an international student, enabling academic activities that contributed to knowledge and progress. The spirit of innovation and education upheld by the Baltic nation has been a motivating force behind this work.

Lastly, my heartfelt thanks go to my family, friends, and colleagues whose encouragement and support have been indispensable throughout this challenging yet rewarding endeavor.

Aymerick FIKEU ELAT
Vilnius University
May 28, 2025

Contents

List of Figures

List of Tables

General Introduction

In the realm of complex systems and artificial intelligence, the study of multi-agent systems (MAS) has gained significant attention for its potential applications in various domains such as robotics, economics, and transportation. MAS, composed of autonomous agents that interact with each other and their environment, exhibit emergent behaviors that are often challenging to comprehend and predict [82]. To unravel the intricacies of these systems, researchers have turned to nature-inspired models, with foraging ant algorithms standing out as a fascinating paradigm[43]. This dissertation delves into the realm of multi-agent systems, employing the innovative perspective of foraging ant models and simulation techniques using Simpy, a powerful and versatile simulation platform[51]. The foraging behavior of ants has captivated scientists for its efficiency and adaptability, making it an ideal inspiration for designing algorithms that can optimize resource allocation, task distribution, and decision-making in complex systems [40]. The foraging ant's behavior in the constructed model is controlled by a set of rules that describes the ant decision making process in specific situations and their inter-communication capabilities [58]. Experimenting with different variations of these rules and evaluating the impact these changes have on the overall behavior of the system is one of the objectives of this work [15]. The integration of Simpy, a leading simulation tool, provides a dynamic platform for modeling and analyzing the behaviors of multi-agent systems, it is a versatile multimethod simulation framework that allows users to model and simulate complex systems using a combination of agent-based, discrete event, and system dynamics simulation methods [79]. It relies on a wide range of simulation paradigms, including discrete event simulation, to provide a comprehensive and flexible platform for modeling diverse systems across various industries [81]. Leveraging its capabilities, this research aims to explore the nuances of agent interactions, resource utilization, and overall system dynamics within the context of foraging ant-inspired models. This study aims to enhance the understanding of emergent behaviors in complex systems by combining the adaptability of the foraging ants' model with the simulation capabilities of the AnyLogic framework [51]. The insights gained from this research hold promise for improving real-world applications, ranging from optimizing supply chain logistics to enhancing traffic flow in urban environments. In this exploration, the work aims to connect natural inspiration and artificial intelligence, revealing the complex interactions of autonomous agents in multi-agent systems[43].

Objective of the Study

The primary objective of this research is to leverage the insights gained from ant foraging behavior to enhance the capabilities of systems simulation in quantitatively evaluating the behavior of selected multi-agent systems By employing computational models inspired by the decentralized mechanisms observed in ant colonies, we aim to simulate and analyze the adaptive decision making processes within complex systems [42][25]. Ants, operating within a decentralized network, exhibit robustness, flexibility, and efficiency in responding to environmental challenges, traits that are increasingly relevant in the design of autonomous, fault tolerant and decentralized systems[27].

Scope and Significance

Through the integration of ant-inspired algorithms within automated simulation framework such as Simpy, we aim to develop techniques that can optimize resource allocation, enhance decision-making processes, and contribute to the advancement of adaptive, self-organizing systems.

Research Goal

The primary objective of this thesis is to quantitatively assess key characteristics of Ant foraging models through simulations within the Anylogic environment, specifically focusing on their effectiveness in efficiently transporting food to the nest within an optimal timeframe and their survival abilities.

Research Tasks

1. **Literature Review:** Conduct a thorough literature review on ant foraging behavior and its models, collective intelligence in ant colonies, and existing research in systems simulation. Identify and analyze key principles and mechanisms utilized by ants in foraging, emphasizing decentralized decision-making. Additionally, explore various simulation tools such as UPAAL, Simpy, and AnyLogic;

2. **System Modelling with Simpy**: Utilize Simpy, a versatile simulation software widely applied in diverse industries for modelling complex systems. Given its support for both discrete event simulation (DES) and agent-based modelling (ABM),

leverage AnyLogic to develop and execute simulations that reflect a broad spectrum of scenarios[79].

3. **Identification of Critical Aspects in a Foraging Ant System:** Explore the essential properties and parameters of the Simpy model depicting foraging ants, including aspects like ant communication via pheromones, the division of labor, and distributed decision making. Analyze the crucial factors that impact the behavior of individual ants and the overall dynamics of the colony [27].

4. **Establishment of Behavioral Rules in Specific Situations:** Set and formalize behavioral guidelines for ants in distinct scenarios within our simulated environment using Simpy. This involves specifying how ants react to alterations (adaptability), assess the challenges, and defend against risks, as well as seize opportunities throughout the foraging process. Then analyze the impact of dynamic environmental conditions on the emergent behavior of the foraging ant model [44];

5. **Global System Simulation and Outcome Evaluation:** Execute a comprehensive simulation of the entire system in Simpy, integrating all identified parameters and behavioral rules. Analyze the simulation results to evaluate the model's performance and effectiveness in returning food to the nest within an optimal timeframe. Assess whether the outcomes align with the anticipated expectations and objectives [44].

6. **Model Refinement and Parameter Optimization:** Refine and tweak the simulation model based on the evaluation results to enhance its accuracy and reliability. In particular, experiment with the behavioral rules controlling the ant decision process to achieve the best possible outcomes, aiming for optimal results that align with the goals of the research. Also, we optimize model parameters like the number of ants, the distance of the food from the ant to the nest and finally consider some environmental changes.

7. **Comparison with Other Methods:** Compare the results obtained from the simulated model with findings from other methods discussed in the literature review. Evaluate the effectiveness and uniqueness of the ant-inspired simulation approach in capturing the complexities of decentralized decision-making and collective intelligence.

Expected Results

In the culmination of this study, we will construct models representing ant foraging systems and execute these models within the Simpy environment. Subsequently, we will assess and scrutinize the simulated systems with respect to the predefined behavioral (decision) rules [44]. Following this evaluation, we will fine-tune the model parameters to identify optimal configurations, particularly focusing on the efficiency of the ant colony in transporting food back to the nest and enhancing their chances of survival. Ultimately, we will compare our research findings to those of other studies to assess the acceptability of our outcomes [66].

Structure of the Thesis

This research will be structured as follows. It will begin with a comprehensive literature review that delves into the existing research on ant colonies, collective intelligence, and systems simulation. The subsequent chapters will explore the development and validation of computational models inspired by ant foraging, offering a detailed analysis of their performance in various scenarios. The core chapters will present and evaluate our constructed models within the automated simulation environment of Simpy, as well as analyze and compare the obtained results.

1 Literature Review

1.1 Introduction

Multi-agent technology is a promising approach to development of complex decentralised systems that dynamically adapt to changing environmental conditions[45]. The study of multi-agent systems (MAS) has garnered significant attention due to its applications across various domains, including supply chain management, robotics, and collaborative frameworks[34]. In particular, the foraging ants model has emerged as a powerful metaphor for understanding complex interactions within MAS[6]. This literature review aims to synthesize recent research findings related to the quantitative evaluation of MAS, focusing on the foraging ants model and automated simulation techniques. Also, we will discuss some of the most prominent MAS simulation environments such as Simpy, UPPAAL and Anylogic and finally, we will highlight knowledge gaps and suggest future research directions. In this research, we will examine various aspects related to our research questions.

The remaining document is structured as follows: we begin with a background study on Multi-Agent Systems (MAS), discussing their concept, history, and evolution. Following this, we provide a brief overview of bio-inspired models, their connection to MAS, and introduce one of the most prominent examples: the foraging ants model. In the next section, we will present some theories behind the foraging ants model, exploring its various components. We will then discuss key metrics essential for evaluating MAS. After that, we introduce optimization algorithms and techniques such as Ant Colony Optimization (ACO), Genetic Algorithms, Particle Swarm Optimization (PSO), among others. The following section will address environmental factors that influence the effectiveness of the foraging process. Finally, we will review the most widely used simulation techniques in the MAS field. By the end of this study, we will have gained a clearer understanding of the subject, identified potential future research areas, and outlined the next steps in our research.

1.2 Background

1.2.1 Introduction to Multi-Agent Systems (MAS)

Multi agent systems are systems of multiple interacting computing elements, known as agents. Agents are computer systems with two important capabilities. First they are at least to some extent capable of *autonomous action* of deciding by *themselves* what they need to do in order to satisfy their design objectives. Second they are capable of interacting with other agents - not simply by exchanging data, but by engaging in analogues of the kind social activity that we all engage in every day of our lives: cooperation, coordination, negotiation and the like [87]. MAS can manifest self-organization and complex behaviors even when the individual strategies of all their agents are simple.Summarizes the theory,method and the key technologies of agent and multi-agent system.The main topics are as follows: the properties,structure and reasoning of intelligent agent;the architecture and communication method of multi-agent system;current status and developing trends of agent oriented programming [10].

Definition and characteristics

MAS are described as macro-systems consisting of multiple agents, with each agent considered a micro-system. The organization of multiple agents within an environment gives rise to MAS. The key characteristics of MAS include [77]:

- Each agent in a MAS has a subjective view, limited by incomplete information due to restrictions in its viewpoint;
- No global control is applied in a MAS; each agent maintains its own state inaccessible to other participants, leading to individual state changes based on behavior rules influenced by the environment;
- Data in a MAS is fully decentralized and distributed among participating agents and the environment;
- Agents in a MAS are designed to be concurrent, operating independently from one another. However, actual concurrency may not always be guaranteed by the underlying implementation;
- MAS integrate a specific organization of the environment in which agents evolve, imposed by the model or the physical layer.

Historical development and evolution

The historical development and evolution of Multi-Agent Systems (MAS) have been quite fascinating. The concept of MAS originated from the field of Distributed Artificial Intelligence (DAI) in the 1980s as a novel and promising technology[29]. The roots of Artificial Intelligence, the broader field from which MAS emerged, can be traced back to the 1940s. A significant milestone was the publication of Isaac Asimov's short story "Runaround" in 1942, which introduced the concept of intelligent machines[29].

In 1950, Alan Turing published an article that describes creating intelligent machines and testing their intelligence, known as the Turing test, which is still considered a benchmark for artificial systems1. The term "Artificial Intelligence" was officially coined in 1956 as a result of the Dartmouth Summer Research Project on Artificial Intelligence[29].

MAS technology has shown rapid growth due to its intelligence and flexibility when solving complex distributed problems. The technology comprises multiple decision-making agents that exist in an environment to achieve common or conflicting goals. It has marvelous features such as flexibility and intelligence that are very useful when solving complex distributed problems[29].

The present applications of MAS are diverse, ranging from self-driving cars and smart speakers to image recognition techniques. The future trends in MAS technology are also promising, with ongoing research and development aimed at enhancing the foundations or key principles of MAS such as agent taxonomy, agent communication approaches, and MAS development frameworks[29].

1.2.2 Overview of bio-inspired models in MAS

Nature has always inspired human race in solving various problems. Bio-inspired models in Multi-Agent Systems (MAS) draw inspiration from biological processes and structures to enhance the design, coordination, and functionality of agent-based systems. These models are particularly focused on emulating the collective intelligence and decentralized decision-making found in nature, such as in insect colonies, animal herds, and cellular systems[54]. The various categories in which this has been vital include:

- **Swarm Intelligence:** Swarm intelligence is a principal bio-inspired model that is derived from the behavior of social insects like ants, bees, and termites. It includes algorithms like Ant Colony Optimization (ACO) and Particle Swarm Optimization

(PSO), which are used for solving optimization problems by mimicking the natural behavior of swarms[54].

- **Evolutionary Computation:** This model includes genetic algorithms and evolutionary strategies that simulate biological evolution. Agents in MAS can use these algorithms to evolve their strategies and adapt to changing environments[57]
- **Artificial Immune Systems:** Inspired by the human immune system, these models help MAS identify and respond to threats or changes in the environment. They can be used for anomaly detection and maintaining system integrity[23].
- **Artificial Neural Networks(ANN):** Neural networks, inspired by the human brain, enable MAS to learn from experience. Deep learning models can be applied within MAS for complex problem-solving and pattern recognition tasks[76].
- **Bio-inspired Robotics:** This involves the design of robotic agents that mimic the physical form and function of biological organisms, leading to more adaptable and efficient MAS in physical environments[54].

The application of these bio-inspired models in MAS aims to achieve enhanced problem-solving capabilities, robustness, and scalability. By leveraging the decentralized control, information sharing, and emergent behaviors observed in natural systems, MAS can tackle a wide range of complex tasks across various domains, from autonomous robotics to distributed computing and logistics[54].

How important are biological inspiration in MAS ?

Biological inspired models are relevant in Multi-Agent Systems (MAS) in the sense that it gives the unique ability to take advantage of the efficiency, adaptability, and robustness of natural systems for solving complex computational problems. Biological systems have evolved over millions of years to become highly optimized and resilient. By mimicking these systems, MAS can achieve similar levels of performance and flexibility. This can be particularly noticable in areas such as

- **Efficient Problem-Solving:** Biological systems such as ant colonies and bee hives demonstrate efficient ways to find food and resources. Algorithms inspired by these behaviors, like Ant Colony Optimization, help MAS to solve routing, scheduling, and optimization problems with similar efficiency[57].

- **Adaptability:** Just as organisms adapt to their environment, bio-inspired MAS can adjust to dynamic conditions and uncertainties in real-time, making them suitable for applications like robotics and manufacturing systems[54].
- **Robustness and fault tolerance**: The decentralized nature of biological systems provides resilience against failures. Similarly, MAS that use bio-inspired models can continue functioning even if some agents fail or behave unexpectedly[54].
- **Self-Organization:** Many biological systems self-organize without central control. This characteristic is useful in MAS for tasks like pattern formation, area coverage, and collective decision-making without the need for a central coordinator[70].
- **Scalability:** Biological systems can scale up or down efficiently. MAS that incorporate bio-inspired principles can manage increasing numbers of agents and tasks without a significant drop in performance[54].
- **Innovation:** Studying biological systems can lead to novel approaches and solutions in MAS that might not be discovered through traditional engineering methods[57].

In these days where we face challenges of a growing complexity, the biological inspiration in MAS is crucial as it provides a framework for developing systems that are not just intelligent but also capable of complex interactions and behaviors that are otherwise difficult to achieve with conventional algorithms. This is an interdisciplinary approach that keep pushing the boundaries of what is possible in *artificial intelligence* and *distributed computing*.

Different bio-inspired models

Bio-inspired models of multi-agent systems (MAS) leverage principles and mechanisms observed in nature to enhance computational problem-solving capabilities. These models draw inspiration from various biological phenomena, including the social behaviors of animals, evolutionary processes, and ecological interactions. The application of these models spans numerous fields, including robotics, optimization, and resource management. One prominent bio-inspired model is the Ant Colony Optimization (ACO) algorithm, which mimics the foraging behavior of ants. ACO utilizes the concept of stigmergy, where agents communicate indirectly through the environment, marking paths with pheromones that guide other agents towards optimal solutions. This approach has been effectively applied in various optimization problems, demonstrating its utility in multi-agent systems[90]. The adaptability of ACO allows it to be integrated into MAS frameworks, facilitating dynamic problem-solving in complex environments. Another significant category of bio-inspired algorithms includes Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). PSO is inspired by the social behavior of birds and fish, where individuals adjust their positions based on their own experience and that of their neighbors. This model has been widely adopted in MAS for tasks such as path planning and resource allocation, showcasing its effectiveness in optimizing multi-dimensional problems[75]. Similarly, GAs, which simulate the process of natural selection, have been employed in MAS to evolve solutions over generations, making them suitable for complex optimization tasks[22].

The Firefly Algorithm (FA) is another notable bio-inspired approach that utilizes the flashing behavior of fireflies to attract others, facilitating the exploration of solution spaces. This algorithm has been successfully applied in multi-objective optimization scenarios, particularly in engineering design, where it helps balance competing objectives[49]. The integration of FA within MAS frameworks allows for enhanced collaboration among agents, leading to improved solution quality. Moreover, the application of bio-inspired models extends to mobile robotics, where algorithms such as the Dragonfly Algorithm and bio-inspired neural networks are employed for real-time path planning and task assignment. These models utilize biological principles to enhance the efficiency and adaptability of robotic systems in dynamic environments[59][60][94]. The incorporation of such algorithms into MAS not only improves operational efficiency but also enables robots to navigate complex terrains and avoid obstacles effectively. Bio-inspired models of multi-agent systems harness the collective intelligence and adaptive behaviors observed in nature to solve complex optimization problems. By employing algorithms such as ACO, PSO, GA, and FA, these models facilitate enhanced collaboration and problem-solving capabilities among agents, making them invaluable in various applications, from engineering design to mobile robotics.

1.3 The Foraging Ants Model

1.3.1 Introduction

The foraging ants model is a prominent example of bio-inspired algorithms that leverage the collective behavior of ants to solve complex optimization problems. This model is primarily encapsulated in the Ant Colony Optimization (ACO) algorithm, which simulates the way ants find the shortest paths to food sources by laying down pheromones.

The pheromone trails serve as a form of indirect communication among ants, guiding others towards optimal routes while also allowing for the exploration of new paths[21] [93]. This decentralized approach is a hallmark of swarm intelligence, where individual agents follow simple rules that lead to complex, emergent behaviors at the group level[93]. The ACO algorithm has been widely applied in various fields, including routing, scheduling, and resource allocation, demonstrating its versatility and effectiveness in solving NP-hard problems. The foundational principles of ACO are rooted in the natural foraging behavior of ants, where they exhibit a balance between exploration and exploitation. Ants tend to explore new food sources while also returning to previously successful locations, a behavior that can be modeled mathematically to enhance algorithm performance[19][92]. This balance is crucial for ensuring that the algorithm can adapt to dynamic environments, where the availability of resources may fluctuate over time[89]. Research has shown that the foraging behavior of ants can be further optimized by integrating additional mechanisms, such as hybrid approaches that combine ACO with other algorithms like Genetic Algorithms (GA) or Particle Swarm Optimization (PSO). These hybrid models can enhance the exploration capabilities of the ACO by introducing genetic operators or swarm behaviors that allow for a more diverse search of the solution space[21][80]. For instance, the integration of a crossover operator in a hybrid artificial bee colony algorithm has been shown to improve numerical optimization outcomes, indicating that combining different bio-inspired strategies can yield superior results[89]. Moreover, the study of ant foraging behavior has also led to insights into task allocation within multi-agent systems. By modeling the roles of foragers, transporters, and followers within an ant colony, researchers have developed algorithms that can effectively allocate tasks among agents in a way that mimics these natural processes[92][5]. This approach not only enhances the efficiency of the agents but also allows for more robust performance in uncertain environments, where agents must adapt to changing conditions[3].

The foraging ants model exemplifies the power of bio-inspired algorithms in addressing complex optimization challenges. By mimicking the natural behaviors of ants and integrating these principles into computational frameworks, researchers have developed effective solutions that leverage the strengths of collective intelligence. The ongoing exploration of hybrid models and task allocation strategies continues to expand the applicability of ACO and similar algorithms across various domains.

Biological Background

Ant foraging has a *social structure*, The ant colonies typically consist of a queen, male ants, and numerous sterile female worker ants. The workers are responsible for various tasks, including foraging, and they are highly organized in their roles[26] and they have that capacity to **communicate** with each other using chemical signals called **pheromones**. Foraging ants leave a pheromone trail on their way back to the nest after finding food, which other ants can follow to the food source[6].

Another interesting fact about the foraging process is their ability to share work also known as division of labor whereby, within the colony, there are often specialized groups of ants, such as scouts and foragers. Scouts search for food sources, while foragers are responsible for collecting the food and bringing it back to the colony[26].

Foraging process can involve hunting, often described as **Predatory and Scavenging Behavior**, it stipulate that Ants can be predators, scavengers, or both. They prey on various insect species and also feed on plant exudates and secretions produced by other insects[26]

Ants have successfully colonized almost every landmass on Earth, demonstrating a remarkable ability to **adapt** to diverse **environments**. Their foraging strategies vary significantly based on food availability and the presence of predators or competitors[26]. This flexibility in behavior allows ants to thrive in a wide range of ecological niches, showcasing their exceptional adaptive capabilities. An interesting aspect of foraging strategies is the use mass recruitment for foraging in some ants species, where many ants are recruited to a food source once it is discovered. Others use a more solitary approach, with individual ants foraging alone[85]

1.3.2 Pheromone evaporation model

The pheromone evaporation process can be mathematically modeled using an exponential decay function:[72]

$$P(t) = P_0 e^{-\lambda t} \tag{1.1}$$

where:

- $P(t)$ is the pheromone intensity at time t,
- P_0 is the initial pheromone intensity,

- λ is the evaporation rate constant ($\lambda > 0$),
- t is the time elapsed.

Alternatively, in a discrete-time simulation, the pheromone level is updated iteratively as follows:

$$P_{t+1} = (1-\rho)P_t \tag{1.2}$$

where:

- P_t is the pheromone level at time step t,
- ρ is the evaporation rate ($0 < \rho < 1$), representing the fraction of pheromone that evaporates at each time step.

This formulation ensures that the pheromone concentration gradually decreases over time, preventing excessive accumulation and allowing ants to dynamically adapt their foraging paths[72].

Mechanism of foraging behaviour in ants

The foraging behavior of ants is a complex and fascinating phenomenon that has garnered significant attention in ecological and biological research. This behavior is characterized by a series of interactions and mechanisms that enable ants to efficiently locate and exploit food resources. Central to this process are several key factors, including communication through pheromones, social interactions, and the division of labor among colony members.One of the primary mechanisms underlying ant foraging is the use of pheromones, which serve as chemical signals that guide foragers to food sources. When an ant discovers food, it lays down a pheromone trail back to the nest, which other ants can detect and follow. This process not only helps in the recruitment of additional foragers but also allows for the establishment of a network of trails that can be reinforced or altered based on the availability of food[68] [53]. The intensity of the pheromone trail diminishes over time, leading to a dynamic system where ants continuously evaluate the quality of food sources and adjust their foraging efforts accordingly[69][2]. The mechanisms of foraging behavior in ants are a complex interplay of individual and collective actions driven by biological instincts and environmental interactions. The article *“The ethology of foraging in ants: revisiting Tinbergen’s four questions”* helps us to break it down in these stages [85]:

- Pheromone Communication;

- Path Integration;
- Tandem Running;
- Use of Multiple Sensory Modalities;
- Division of Labor;
- Collective Decision-Making.

Social interactions among ants play a crucial role in regulating foraging activity. For instance, studies have shown that interactions among foragers can increase their availability and activity levels, thereby enhancing the overall foraging efficiency of the colony[68][67]. These interactions can include direct contact between ants, which may stimulate foraging behavior, or indirect effects where the presence of returning foragers influences the departure rates of those still in the nest[30][31]. This feedback mechanism allows colonies to adapt their foraging strategies in response to environmental changes, such as food availability and predating risks[71] [30]. The division of labor is another critical aspect of ant foraging behavior. Different castes within an ant colony may specialize in various tasks, including foraging, nest maintenance, and brood care. This specialization allows for a more efficient allocation of resources and efforts, as certain ants may be better suited for specific roles based on their physical traits or behavioral tendencies [71][61]. For example, polymorphic worker ants can exhibit different foraging strategies, which can enhance the colony's ability to exploit diverse food resources effectively[71][63].

On another hand, the regulation of foraging activity can be influenced by environmental factors and the colony's internal state. Research indicates that variations in foraging behavior can persist over time, reflecting the colony's adaptability and resilience[69] [30]. For instance, colonies may adjust their foraging intensity based on external conditions, such as humidity and temperature, which can affect food availability and the risk of desiccation during foraging trips[30][31]. The foraging behavior of ants is a multifaceted process that involves the interplay of pheromone communication, social interactions, and labor division. These mechanisms enable ant colonies to efficiently locate and exploit food resources while adapting to changing environmental conditions. Understanding these behaviors not only provides insights into the ecological roles of ants but also inspires the development of bio-inspired algorithms in fields such as optimization and robotics.

1.3.3 Application of the foraging ants to MAS

The foraging behavior of ants is a valuable model for multi-agent systems (MAS). In MAS, decentralized and self-organized strategies inspired by ants can improve efficiency in tasks such as resource allocation, network routing, and collaborative problem-solving. For example, ant colony optimization algorithms leverage the principles of ant foraging to find optimal paths and distribute tasks among agents, enhancing performance in dynamic and complex environments. This biologically-inspired approach leads to robust, scalable, and adaptive solutions in various fields, from robotics to logistics.

Ants utilize a variety of cues and interactions to regulate their foraging activities. For instance, harvester ants (Pogonomyrmex) adjust their foraging rates based on the success of returning foragers, creating a feedback loop that optimizes resource collection[64][67]. This closed-loop system can inform MAS design by incorporating similar feedback mechanisms, allowing agents to adapt their behavior based on the success of their peers. Such adaptive mechanisms can enhance the efficiency of resource allocation in robotic swarms or distributed sensor networks.

Moreover, the use of chemical cues, such as pheromones, plays a crucial role in ant foraging. For example, patrollers in harvester ant colonies deposit secretions that guide foragers toward food sources[33]. This behavior can be translated into MAS through the implementation of virtual pheromones, where agents leave behind digital markers that influence the movement of other agents towards resources. This approach has been successfully modeled in robotic path planning, where ant foraging behavior is mimicked to navigate complex environments[65].

The flexibility of foraging strategies among ants also highlights the importance of adaptability in MAS. Research indicates that ants can switch between foraging strategies based on environmental conditions and resource availability[20][14]. For instance, leaf-cutting ants exhibit different foraging behaviors depending on traffic flow on trails, allowing them to efficiently allocate foragers to either ephemeral or stable resources[20][18]. Implementing similar adaptive strategies in MAS can enhance their robustness in dynamic environments, allowing agents to respond effectively to changing conditions.

The intricate foraging behaviors of ants provide valuable insights for the development of multi-agent systems. By leveraging feedback mechanisms, chemical communication, adaptability, and collective dynamics observed in ant colonies, researchers can design more efficient and resilient algorithms for various applications, from robotics to distributed computing.

1.3.4 Features and benefits of using the foraging ants in MAS

The use of foraging ants in Multi-Agent Systems (MAS) offers several features and benefits that enhance the efficiency and adaptability of these systems. By leveraging the collective behaviors observed in ant colonies, researchers can develop algorithms and models that mimic these natural processes, leading to improved performance in various applications.

Features of Foraging Ants in MAS

The model of the foraging ant exhibits the following features:

- Collective Decision-Making: Foraging ants exhibit sophisticated collective decision-making processes that allow them to efficiently locate and exploit food sources. This is often achieved through pheromone communication, where ants deposit pheromones on successful paths, guiding others toward food[73]. In MAS, similar mechanisms can be implemented to facilitate decentralized decision-making among agents, allowing them to adapt to changing environments and optimize resource allocation;
- Adaptability to Environmental Changes: Ants can adjust their foraging strategies based on environmental conditions and resource availability. For instance, studies have shown that the foraging behavior of ants is influenced by interactions with returning foragers, which can inform potential foragers about the quality of food sources[14]. This adaptability can be modeled in MAS to enhance the system's responsiveness to dynamic conditions;
- Emergent Behavior: The simple rules followed by individual ants lead to complex and organized behaviors at the colony level, a phenomenon known as emergent behavior. This characteristic can be harnessed in MAS to create systems that exhibit robust performance without centralized control. For example, the emergent regulation of foraging frequency through simple movement rules has been demonstrated in ant models[9];
- Stigmergy: Ant foraging behavior is a prime example of stigmergy, where agents communicate indirectly through modifications to their environment (e.g., pheromone trails). This principle can be applied in MAS to enhance cooperation and coordination among agents, allowing them to work together effectively without direct communication[9].

Benefits of Using Foraging Ants in MAS

As a nature-inspired model of Multi-Agent Systems (MAS), foraging ants offer several benefits, including:

- Efficiency in Resource Utilization: The foraging strategies of ants are optimized for efficiency, allowing them to maximize food intake while minimizing energy expenditure. By modeling these strategies in MAS, systems can achieve similar efficiencies in resource allocation and task execution. For instance, the use of pheromone trails in MAS can lead to more efficient routing in network optimization problems[11];

- Robustness to Failures: Ant colonies demonstrate resilience to individual failures, as the collective behavior can compensate for the loss of individual ants. This robustness can be beneficial in MAS, where individual agents may fail or become less effective. The ability of the system to continue functioning despite such failures is a significant advantage in critical applications[31];

- Scalability: The principles of ant foraging can be scaled to accommodate varying numbers of agents and changing environments. As the number of agents increases, the collective foraging efficiency can be maintained or even improved, making MAS suitable for large-scale applications. This scalability is particularly relevant in scenarios such as robotic swarms or distributed sensor networks[65];

- Enhanced Learning and Adaptation: ants demonstrate learning behaviors that allow them to optimize their foraging routes based on past experiences. This capability can be integrated into MAS to enable agents to learn from their interactions and adapt their strategies over time, leading to improved performance in dynamic environments[92];

- Modeling Complex Interactions: The study of ant foraging provides insights into complex interactions within populations, which can be modeled in MAS to understand and predict collective behaviors. For example, the interactions between foragers and returning ants can inform decision-making processes, enhancing the overall effectiveness of the system[68].

Incorporation of foraging ants into Multi-Agent Systems offers a range of features and benefits that enhance their efficiency, adaptability, and robustness. Leveraging the collective behaviors observed in ant colonies allows researchers to develop innovative algorithms and models that address complex challenges across various domains.

The foraging ant model demonstrates the power of nature-inspired solutions in optimizing and enhancing the efficiency of Multi-Agent Systems (MAS). By emulating the decentralized, self-organizing behaviors of ant colonies, this model showcases how simple agents can collectively solve complex problems, such as resource allocation and path finding, with remarkable adaptability and robustness. The inherent scalability and resilience of the foraging ant model make it a valuable approach in various domains, providing a strong foundation for the development of more advanced MAS algorithms.

As we move forward, it becomes essential to assess the effectiveness of these models in practical applications. Evaluating the performance of a MAS is critical for identifying strengths and weaknesses, optimizing its functionality, and ensuring it meets the demands of the task at hand. In the next section, we will introduce key metrics used to evaluate MAS, examining how these metrics influence the system's performance and providing insights into how they can guide further improvements.

1.4 Quantitative Evaluation Metrics for MAS

Quantitative evaluation metrics for Multi-Agent Systems (MAS) are essential for assessing their performance, efficiency, scalability, robustness, and other critical aspects

1.4.1 Importance of quantitative evaluation metrics

Quantitative metrics are crucial in evaluating Multi-Agent Systems (MAS) because they provide objective, measurable, and data-driven insights into the performance and effectiveness of the system.

It is important to establish standards and Performance Benchmarking well known Quantitative metrics facilitate the comparison of different Multi-Agent Systems (MAS) or various configurations of the same system. This enables benchmarking against established performance standards, providing a clear understanding of how well each system or configuration performs relative to benchmarks. This comparison is crucial for identifying the most efficient and effective solutions[47].

We can tweak the parameters and variables to achieve system optimization. Quantitative metrics are invaluable in identifying areas of strength and weakness within the MAS, guiding developers and researchers in optimizing agent behaviors and interactions. Focusing specific variables that need improvement, these metrics enable targeted enhancements, leading to more efficient and effective system performance[47]. Additionally, the metrics are important for the following aspects:

- Decision Making: Quantitative metrics support informed decision-making by providing clear evidence of how well the MAS is performing in various scenarios[47];
- Validation and Verification: Quantitative metrics serve as essential tools for validating and verifying the MAS against its design objectives and requirements. They provide a means to ensure that the system meets its intended goals and operates as expected, highlighting any discrepancies and areas needing adjustment[17];
- Research and Development: In the field of MAS research, quantitative metrics are essential for evaluating the impact of new theories, models, or algorithms. They provide a clear and objective measure of how changes we brought in affected system performance, enabling researchers to assess improvements and validate their work[47].

1.4.2 Commonly used metrics in MAS evaluation

In the evaluation of Multi-Agent Systems (MAS), several quantitative metrics are commonly used to assess their performance and effectiveness. These metrics provide a means to measure various aspects of MAS, such as efficiency, reliability, and scalability[16].

Key performance metrics for a Multi-Agent System (MAS) include **system throughput**, which measures the number of tasks completed within a given time frame; **success rate**, evaluating the percentage of tasks successfully completed by agents; and **resource utilization**, assessing how effectively computational resources like memory and processing power are used. Additionally, **response time** gauges how quickly the MAS responds to requests or completes tasks, while **scalability** reflects its ability to manage increasing numbers of tasks or agents. **Robustness** measures the system's capacity to maintain functionality despite errors or environmental changes, and adaptability evaluates how well it adjusts to new, changing, or uncertain conditions. **Efficiency** also commonly used is defined as the ratio of output gained to input provided, offering a holistic view of the MAS's overall performance[47]. And finally, **fault tolerance** reflects the MAS's ability to continue functioning properly even when some of its components fail[17].

These metrics are essential for understanding the capabilities and limitations of a MAS and for guiding future improvements.

1.4.3 Metrics for evaluating the foraging ants

Evaluating the foraging ants model in Multi-Agent Systems (MAS) involves using specific metrics that can measure the efficiency, effectiveness, and adaptability of the agents' behaviors.

- Path Efficiency: Measures how effectively the ants find the shortest path to food sources. This can be quantified by comparing the length of the path taken by the ants to the shortest possible path[41].
- Pheromone Utilization: Assesses the effectiveness of pheromone communication by measuring how quickly and accurately other ants follow the pheromone trails to find food[39].
- Foraging Success Rate: The percentage of foraging ants that successfully find food and return it to the nest[36].
- Search Time: The time it takes for ants to locate a food source. This metric is important for understanding the efficiency of the exploration strategies used by the ants[41].
- Food Return Rate: The rate at which food is returned to the nest, which reflects the overall productivity of the foraging process[39].
- Adaptability to Changes: The ability of the ant agents to adapt their foraging strategies in response to changes in the environment, such as the introduction of new food sources or obstacles[41].
- Robustness: The resilience of the foraging process in the face of individual ant failures or disruptions in the pheromone trails[36].
- Scalability: How well the foraging process scales with an increasing number of ants or food sources[41].

These metrics help in understanding the dynamics of the foraging ants model and in making comparisons with other models or algorithms[36].

Quantitative evaluation metrics play a crucial role in assessing the performance of Multi-Agent Systems (MAS). These metrics provide a structured way to measure various aspects of MAS, such as efficiency, scalability, adaptability, and robustness. By applying these metrics, researchers can identify strengths and weaknesses, ensuring that the system performs optimally under different conditions. Metrics also serve as a benchmark for comparing different MAS implementations, guiding improvements and fostering the development of more efficient systems.

With a solid understanding of evaluation metrics, the next step is to explore optimization techniques that further enhance MAS performance. In the upcoming section, we will introduce several key optimization algorithms, such as Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO). We will discuss how these techniques are applied in MAS to solve complex problems, improve decision-making processes, and increase overall system efficiency.

1.5 Environmental Factors Influencing MAS Performance

Environmental factors play a crucial role in influencing the performance of Multi-Agent Systems (MAS), particularly in the context of foraging behaviors observed in ant colonies. The interplay between environmental conditions and ant foraging strategies can provide insights into how these systems adapt and optimize their collective behavior. This synthesis examines various studies that highlight the impact of environmental factors on ant foraging performance, drawing from the provided references.

1.5.1 Description of various environmental factors

One significant environmental factor affecting ant foraging activity is climate, particularly temperature and humidity. Research indicates that harvester ants (Messor andrei) exhibit increased recruitment to food sources under humid conditions, suggesting that moisture levels directly influence foraging behavior[30]. Additionally, the regulation of foraging activity in response to humidity is critical for managing water loss, which is particularly important in arid environments [31]. This adaptability to environmental moisture levels illustrates how external conditions can shape the collective foraging strategies of ant colonies.

Another critical aspect is the influence of geographical variation on foraging behavior. A study by highlights how climate and net primary productivity drive geographical differences in ant foraging activity and resource use[46]. This suggests that the availability of resources, influenced by environmental factors, can dictate the foraging patterns and efficiency of ant colonies across different regions. Such findings emphasize the importance of understanding local environmental conditions when studying the foraging dynamics of ant species.

The structural characteristics of the environment also play a significant role in shaping foraging behavior. For example, the presence of obstacles and the asymmetry of trails can affect how ants choose their paths during foraging[24]. Environmental disruptions, such as obstacles on pheromone trails, can lead to changes in exploration patterns, which in turn influence the overall foraging efficiency of the colony. This highlights the need for MAS to adapt their strategies based on the physical characteristics of their environment. Furthermore, the interactions among foraging ants are influenced by environmental conditions. found that the daily patterns of foraging activity in harvester ants are affected by temperature and humidity, which in turn influence the rate of interactions necessary for ants to leave the nest [14]. This suggests that environmental factors not only affect individual foraging decisions but also the collective dynamics of the colony, as ants adjust their behavior based on the prevailing conditions.

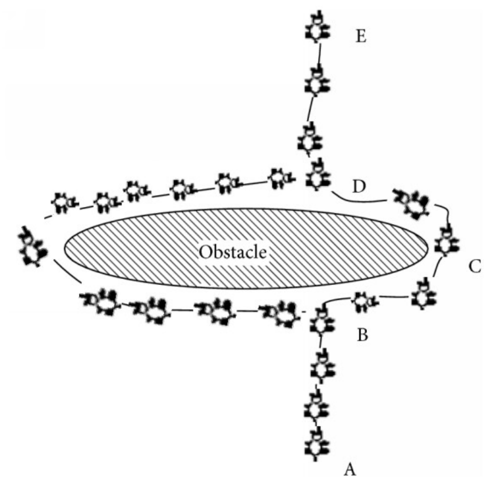

Figure 1.1: An obstacle appears and the ants try to avoid it

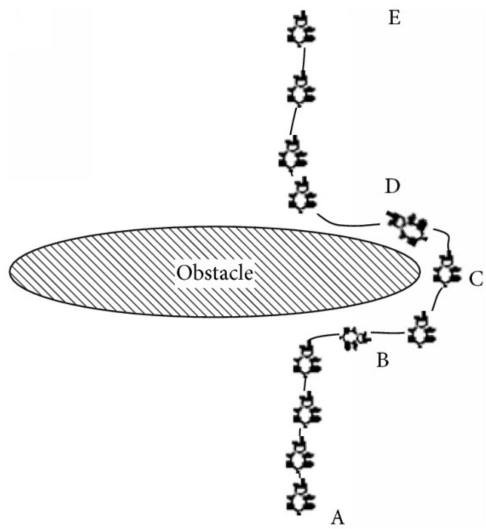

Figure 1.2: Ants select the shorter route

Figure 1.3: Ant Behavior Around Obstacles

Environmental factors such as climate, geographical variation, and structural characteristics significantly influence the performance of Multi-Agent Systems in foraging scenarios. The adaptability of ant colonies to these factors underscores the importance of considering environmental conditions when studying collective behavior and optimizing foraging strategies in MAS. Future research should continue to explore these interactions to enhance our understanding of how environmental dynamics shape the performance of foraging agents.

1.6 Simulation Techniques for MAS

Simulation techniques for Multi-Agent Systems (MAS) are essential for understanding and optimizing the collective behaviors of agents, particularly in contexts such as foraging. This chapter explores several key simulation techniques relevant to MAS, particularly in the context of foraging behaviors inspired by ant colonies.

One prominent technique is the use of agent-based modeling (ABM), which allows for the simulation of individual agents and their interactions within a defined environment. This approach is particularly effective in studying the foraging behaviors of ants, as it enables researchers to model the actions of individual ants while considering their interactions with both the environment and other agents [64]. For example, highlight the importance of closed-loop excitable systems in regulating foraging activity, demonstrating how ABM can track individual behaviors and collective dynamics[64]. This level of detail is crucial for understanding how environmental factors influence foraging efficiency and decision-making processes.

Another significant simulation technique is the application of Ant Colony Optimization (ACO) algorithms, which are inspired by the foraging behavior of real ants. ACO algorithms utilize pheromone trails to guide agents toward optimal solutions in various optimization problems, such as routing and scheduling[38]. The simulation of ACO can be enhanced by incorporating dynamic environmental factors, allowing for the adaptation of foraging strategies in response to changing conditions. For instance, describe a variable sampling ACO algorithm that improves performance in continuous optimization scenarios, showcasing the versatility of ACO in different contexts [38].

1.6.1 Importance of simulation in MAS research

Hybrid approaches that combine ACO with other optimization techniques, such as genetic algorithms, can further enhance the performance of MAS. present a hybridized ant colony algorithm for solving the Multi Compartment Vehicle Routing Problem, demonstrating how integrating multiple methodologies can lead to improved outcomes in complex optimization tasks[1]. Such hybrid models can leverage the strengths of different algorithms to optimize foraging strategies effectively.

Simulation techniques can also be applied to study the impact of environmental factors on foraging behavior. For example, investigate how climate and body size influence the foraging performance of seed-eating ants, revealing the importance of environmental conditions in shaping foraging strategies[78]. By simulating these interactions, researchers can gain insights into how ants adapt their foraging behaviors based on resource availability and competition.

Simulation techniques for Multi-Agent Systems, particularly in the context of foraging behaviors, are vital for understanding and optimizing collective decision-making processes. By employing agent-based modeling, Ant Colony Optimization algorithms, and hybrid approaches, researchers can effectively simulate the complex interactions between agents and their environments. These techniques not only enhance our understanding of natural foraging behaviors but also provide valuable insights for developing efficient algorithms applicable to various optimization problems.

1.6.2 Overview of automated simulation techniques

Multi-Agent Systems (MAS) have gained significant attention in recent years due to their ability to model complex systems composed of interacting autonomous agents. Automated simulation techniques for MAS are essential for understanding the dynamics of these systems, particularly in applications such as traffic management, resource allocation, and social behavior modeling.

One of the foundational techniques in MAS simulation is Agent-Based Modeling and Simulation (ABMS). ABMS allows for the representation of individual agents with distinct behaviors and decision-making processes, enabling researchers to study how these agents interact within a given environment. emphasize that ABMS is particularly useful for modeling complex adaptive systems, where agents can self-organize and exhibit emergent behaviors[50]. This approach facilitates the exploration of individual-level interactions and their collective outcomes, making it a powerful tool for simulating MAS.

In addition to traditional ABMS, hybrid approaches that incorporate data mining techniques can enhance simulation studies. highlight how data mining can improve the analysis of emergent behaviors in agent-based simulations, allowing for a deeper understanding of the underlying phenomena[74]. By integrating data mining with ABMS, researchers can uncover hidden patterns and relationships within the simulation data, leading to more informed decision-making and model refinement.

Another significant aspect of MAS simulation is the use of specialized frameworks and tools designed for specific applications. For instance, the FLAME GPU framework enables mesoscopic and microscopic simulations, allowing for the modeling of vehicle interactions in traffic systems[37]. This framework supports the development of detailed simulations that can capture the complexities of real-world scenarios, making it particularly valuable for traffic management and urban planning.

Additionally, the application of ABMS in various fields demonstrates its versatility. For example, developed an agent-based simulation model for dynamic real-time traffic signal control, showcasing how MAS can be employed to address challenges in heterogeneous environments[4]. Similarly, the use of ABMS in studying the spread of radicalism and extremism illustrates its applicability in social and political contexts[62].

Validation techniques are also crucial in ensuring the reliability of MAS simulations. discuss the importance of validation methods for agent-based models, particularly in geospatial simulations, where accuracy is paramount[13]. By employing robust validation strategies, researchers can enhance the credibility of their simulations and ensure that they accurately reflect real-world dynamics.

Automated simulation techniques for Multi-Agent Systems are essential for modeling and analyzing complex interactions among agents. The integration of Agent-Based Modeling and Simulation, data mining techniques, specialized frameworks, and validation methods provides a comprehensive approach to understanding the dynamics of MAS. As research continues to evolve, these simulation techniques will play a critical role in advancing our knowledge of complex systems across various domains.

Agent-based modeling

Agent-based modeling (ABM) is a powerful simulation technique that allows researchers to model complex systems composed of autonomous agents that interact with one another and their environment. This approach is particularly useful in various fields, including economics, ecology, and social sciences, as it enables the exploration of emergent behaviors resulting from individual interactions.

One of the primary advantages of agent-based modeling is its ability to capture the heterogeneity of agents and their behaviors. Each agent in an ABM can have distinct characteristics, decision-making processes, and interaction rules, allowing for a more nuanced representation of real-world systems. For instance, demonstrate the application of a multi-agent-based model for distributed fault diagnosis systems, highlighting how such models can be integrated into existing automation architectures without significant modifications[88]. This flexibility makes ABM suitable for a wide range of applications, from industrial processes to social dynamics.

Moreover, ABM facilitates the study of complex adaptive systems where agents adapt their behaviors based on their interactions and environmental conditions. This adaptability is crucial for understanding how systems evolve over time. For example, review the use of multi-agent systems to simulate environmental pollution issues, emphasizing how ABM can forecast the impact of human activities on ecosystems[28]. By simulating various scenarios, researchers can gain insights into how changes in agent behavior or environmental conditions can lead to different outcomes.

The integration of ABM with other methodologies can enhance its effectiveness. For instance, the combination of ABM with data mining techniques can improve the analysis of emergent behaviors within simulations. This hybrid approach allows for the extraction of meaningful patterns from simulation data, leading to better understanding and decision-making.

Furthermore, ABM can be utilized to explore the influence of environmental factors on agent behavior and system performance. For example, the study by discusses the need to identify variables that reflect the environmental dimension of performance in utility sectors, which can be modeled using ABM to simulate how agents respond to different environmental conditions[52]. This capability is particularly relevant in contexts where environmental dynamics significantly impact agent interactions and system outcomes.

Validation of agent-based models is also a critical aspect of ensuring their reliability and applicability. emphasize the importance of validation techniques in agent-based modeling, particularly in geospatial simulations, where accuracy is essential for effective decision-making. By employing robust validation methods, researchers can enhance the credibility of their models and ensure that they accurately reflect real-world dynamics.

Real-time simulation frameworks

Real-time simulation frameworks are essential for modeling complex systems where timely responses to dynamic changes are crucial. These frameworks enable the integration of real-time data into simulations, allowing for more accurate predictions and decision-making.

One prominent application of real-time simulation frameworks is in transportation systems. present an agent-based modeling and simulation approach for real-time collision handling in railway transport networks. Their framework, implemented in the JADE environment, demonstrates how real-time datasets can be utilized to enhance the safety and efficiency of train operations by simulating various scenarios involving trains, stations, and junctions[12]. This highlights the potential of real-time simulations to improve operational decision-making in critical infrastructure.

Another significant area of application is in the calibration of agent-based models (ABMs) using real-time data assimilation techniques. discuss the dynamic calibration of ABMs, emphasizing the importance of integrating real-time data to track changes in system dynamics accurately. This approach allows for a better understanding and prediction of human behavior, ultimately leading to more reliable decision-making in various contexts[86]. The ability to adapt models based on real-time data is crucial for applications in fields such as urban planning and emergency response.

Real-time simulation frameworks are also employed in scientific visualization. explore interactive steerable scientific visualization techniques for free surface flow, demonstrating how real-time simulations can enhance the understanding of fluid dynamics in avionics[48]. By providing real-time visual feedback, such frameworks can improve performance in safety-critical tasks, making them valuable tools in engineering and design.

In the context of energy management, frameworks that support real-time simulations are vital for optimizing the operation of microgrids. discuss a real-time simulation technique for microgrid models, emphasizing the advantages of hardware-in-the-loop simulations. This approach allows for the testing and validation of electrical properties of power system devices in real-time, facilitating the integration of renewable energy sources and enhancing grid reliability[55]. Such frameworks are essential for managing the complexities of modern energy systems.

Real-time simulation frameworks play a pivotal role in enhancing the understanding and management of complex systems across various domains. By integrating real-time data, these frameworks enable more accurate modeling, dynamic calibration, and improved decision-making. As technology advances, the development of more sophisticated real-time simulation techniques will continue to enhance their applicability in critical fields such as transportation, energy management, and scientific research.

1.6.3 Simulation tools for the foraging ants model

UPPAAL

UPPAAL is an integrated tool environment designed for modeling, validation, and verification of real-time systems. It uses networks of timed automata extended with data types like bounded integers and arrays[83]. Developed collaboratively by Uppsala University in Sweden and Aalborg University in Denmark, UPPAAL is particularly useful for systems that can be modeled as a collection of non-deterministic processes with finite control structures and real-valued clocks[84]. The latest version, UPPAAL 5.0 (Figure

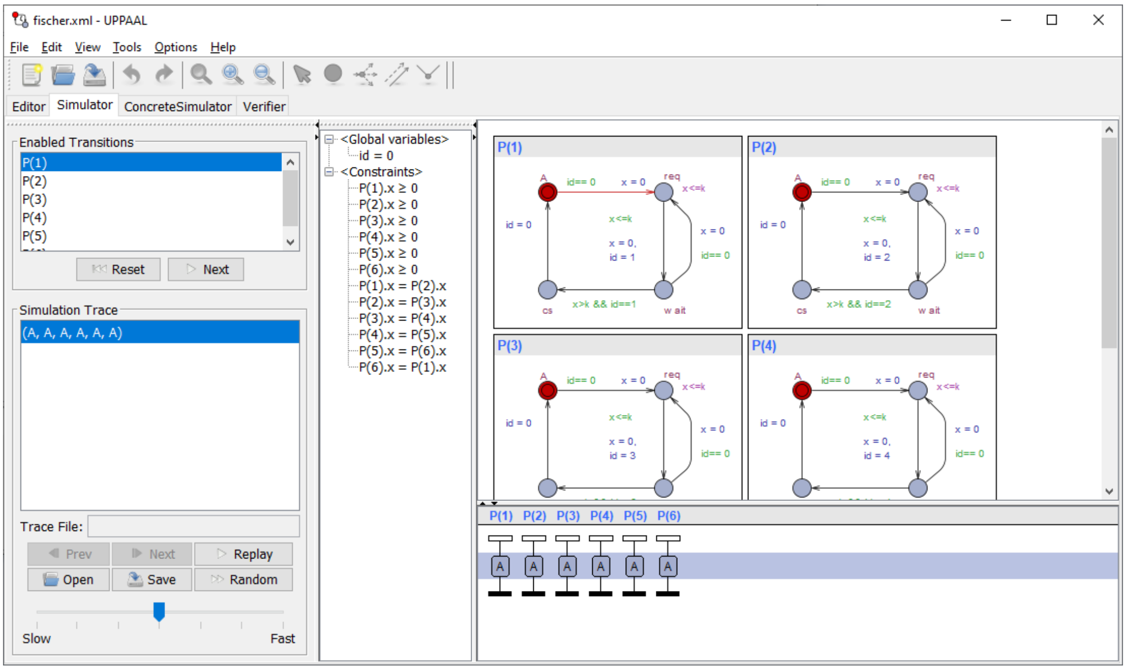

Figure 1.4: Simulation with UPPAL

1.4), includes enhanced features from tools like TIGA and Stratego1. It's available for free for academic use, but commercial use requires a license[83]. Using UPPAAL for Multi-Agent System (MAS) simulation is a powerful approach to model and analyze the interactions between multiple agents in a system. In UPPAAL, agents are modeled as timed automata, where each agent is defined by a set of states, transitions, and clocks to capture time-dependent behavior. Communication between agents is facilitated through the use of channels and shared variables, enabling interaction and data exchange. To ensure coordinated behavior among agents, synchronization primitives are employed, allowing actions to be aligned across different components. UPPAAL is designed with scalability in mind, making it suitable for modeling complex systems composed of multiple interacting agents[84]. Here is a compiled list of steps needed to simulate MAS in UPPAAL:

1. Define Agents: Create individual models for each agent, specifying their behavior and interactions.

2. Set Up Communication: Use UPPAAL's channels to model message passing between agents.

3. Run Simulations: Execute the simulation to observe how agents interact over time. You can set breakpoints, inspect variables, and step through the execution.

4. Analyze Behavior: Use UPPAAL's verification tools to check properties such as deadlock-freedom, reachability, and timing constraints.

Example Scenario To simulate a traffic system with multiple autonomous vehicles (agents). Each vehicle can be modeled as an agent with states like "moving," "waiting," and "stopped." You can use channels to model traffic signals and shared variables to represent road conditions.

SimPy

SimPy is a process-based discrete-event simulation framework written in Python. It's designed to model real-world processes and systems, such as customer service operations, traffic systems, and manufacturing processes[79].

In Python, you can use the simpy framework for event simulation. First, take a quick look at how a simulated process would run in Python. Below is a code snippet from a simulation of a security checkpoint system. The following three lines of code set up the environment, pass all necessary functions, and run the simulation[91]: as a very

Python

```python
# Set up the environment
env = simpy.Environment()

# Assume you've defined checkpoint_run() beforehand
env.process(checkpoint_run(env, num_booths, check_time, passenger_arrival))

# Let's go!
env.run(until=10)
```

Figure 1.5: Simulation with Simpy

powerfull simulation tool, Simpy has amazing features making the process seamless among which the most popular are:

1. Process-Based Simulation: SimPy models dynamic systems as processes, where each process represents a component of the system. Processes are functions or generators that yield at specific points in time, representing events like resource requests, waits, or releases.

2. Discrete-Event Simulation: SimPy operates on the principle of discrete-event simulation, where system state changes occur at discrete points in time, triggered by events such as arrivals, departures, or completions.

3. Resources Management: SimPy provides tools for managing resources like servers, machines, or workers. These resources can be requested, used, and released by processes. The framework supports a variety of resource-sharing mechanisms to simulate real-world systems effectively. These include general shared **resources** with limited capacity, **containers** that store quantities like fuel or materials, and **stores** used to model storage systems such as warehouses or inventories. It also offers **Priority Resources**, which handle requests based on priority levels. Additionally, SimPy enables **flexible event scheduling**, allowing users to manage timelines by scheduling future events and letting processes wait for specific durations or conditions, thus capturing dynamic time-based behaviors.

4. Inter-Process Communication: Processes can interact with each other using signals and shared resources. SimPy provides event constructs to synchronize and coordinate these interactions.

5. Ease of Use: Written entirely in Python, SimPy is easy to integrate with other Python libraries. Its syntax is simple and intuitive, making it accessible for both beginners and experienced developers.

6. Extensibility: SimPy is highly extensible and can be customized or extended to fit various simulation needs, including more complex systems such as multi-agent systems, supply chains, or business processes.

7. Real-Time Simulation: Although SimPy is mainly for discrete-event simulation, it can simulate real-time processes or be combined with real-time systems.

SimPy's flexibility and ease of use make it a powerful tool for creating simulations in a wide variety of fields, including operations research, engineering, and computer science[79].

Anylogic

AnyLogic is a powerful simulation modeling software used for business applications across various industries. It supports multiple simulation methodologies, including agent-based, discrete event, and system dynamics[7]. the key features of anylogic include

- Multimethod Modeling: Combine different simulation methods in one model to capture complex system behaviors[7].
- Visualization: Create 2D and 3D animations to visualize your models and make them more understandable[8].
- Industry-Specific Libraries: Utilize libraries tailored for specific industries, such as logistics, healthcare, and manufacturing[8].
- Integration: Connect with GIS maps, databases, and other external systems to enhance your models[8].
- Cloud Support: Run simulations in the cloud for scalability and collaboration[8].

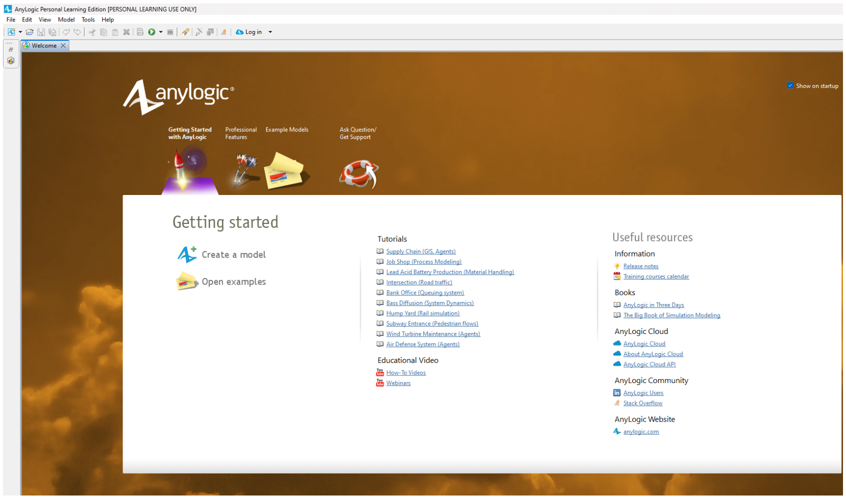

Figure 1.6: Anylogic simulation software

Using AnyLogic for Multi-Agent System (MAS) simulation is a powerful way to model and analyze the interactions between multiple autonomous agents. AnyLogic supports agent-based modeling, which is ideal for MAS due to its flexibility and ability to capture complex behaviors and interactions.

1.6.4 Evaluation of simulation accuracy and efficiency

Evaluating the accuracy and efficiency of Multi-Agent Systems (MAS) simulations, particularly those inspired by foraging ants, is crucial for understanding their performance in real-world applications. This evaluation can be approached through various dimensions, including the fidelity of the models, the robustness of the algorithms, and the adaptability of the systems to dynamic environments.

Accuracy of MAS Simulations

- **Model Fidelity**: The accuracy of MAS simulations largely depends on how well the model represents the underlying biological processes. emphasize that the adaptive value of tandem communication in ants is highly dependent on environmental factors and colony composition, suggesting that simulations must accurately capture these dynamics to yield valid results[32]. Accurate parameterization and validation against empirical data are essential for ensuring that the simulation reflects real-world behaviors.
- **Parameter Sensitivity**: The performance of ant foraging models can be sensitive to specific parameters, such as pheromone decay rates and foraging rates. demonstrate that variations in these parameters can significantly influence the foraging activity of ant colonies, highlighting the need for careful calibration in simulations[69]. Sensitivity analysis can help identify critical parameters that affect model outcomes and improve the accuracy of predictions.
- **Behavioral Representation**: Accurate representation of individual ant behaviors, such as decision-making processes and communication strategies, is vital for simulation fidelity. show that factors like bifurcation angles can affect collective decision-making in ants, suggesting that simulations should incorporate such behavioral nuances to achieve realistic outcomes[35]. This level of detail enhances the model's ability to predict how ants will respond to various environmental conditions.

Efficiency of MAS Simulations

- **Computational Efficiency**: The efficiency of MAS simulations is often measured by the computational resources required to run the models. discuss how foraging distance influences the search strategies of ants, indicating that efficient algorithms can lead to faster convergence in simulations[56]. Optimizing algorithms, such as Ant Colony Optimization (ACO), can significantly reduce computation time while maintaining solution quality.
- **Scalability**: The ability of a simulation to scale with the number of agents is a critical factor in its efficiency. 's work on chemotaxis approaches highlights how effective modeling of trail-laying and foraging behavior can lead to scalable solutions that adapt to larger populations of agents[5]. Efficient algorithms must maintain performance as the number of agents increases, ensuring that simulations remain practical for large-scale applications.

- **Dynamic Adaptability**: The adaptability of MAS simulations to changing environments is a key aspect of their efficiency. propose an active inference framework that allows for dynamic adjustments in foraging behavior based on environmental feedback[25]. This adaptability can enhance the efficiency of simulations by allowing agents to optimize their strategies in real-time, reducing the need for extensive recalibration.
- **Algorithmic Improvements**: The development of improved algorithms, such as the enhanced ACO proposed by , can significantly boost the efficiency of simulations by optimizing search processes and reducing computational overhead[38]. These advancements can lead to faster convergence and better solution quality, making simulations more effective for practical applications.

The evaluation of MAS simulation accuracy and efficiency, particularly in the context of foraging ants, is essential for developing robust models that can be applied to real-world problems. By focusing on model fidelity, parameter sensitivity, behavioral representation, computational efficiency, scalability, dynamic adaptability, and algorithmic improvements, researchers can enhance the performance of MAS simulations. The integration of empirical data and advanced algorithms will continue to drive improvements in the accuracy and efficiency of these systems.

1.7 Conclusion

As demonstrated by the reviewed literature sources, the quantitative evaluation of multi-agent systems using foraging ant models and automated simulation techniques involves a multifaceted approach that incorporates various performance metrics, communication strategies, and adaptive behaviors. By leveraging these methodologies, researchers can gain valuable insights into the effectiveness and efficiency of MAS in achieving complex tasks. This approach represents a promising area of research with significant implications for various applications. While substantial progress has been made, addressing the identified knowledge gaps and pursuing the suggested future research directions will be crucial for advancing the field and enhancing the effectiveness of multi-agent systems.

In the upcoming phase of our research, we plan to utilize the SimPy simulation framework to conduct a series of controlled experiments involving the various approaches and tools previously discussed. This simulation environment will enable us to model complex system behaviors and evaluate the performance of each method under diverse conditions. Through iterative testing and comparative analysis, we aim to identify the most effective solutions tailored to the specific requirements of our tasks. The outcomes of these simulations will serve as foundational results, providing critical insights that can be further analyzed, validated, and expanded upon in subsequent stages of the research. This process will not only guide our methodological choices but also contribute to the development of a robust and adaptable framework for future applications.

2 Methodology

2.1 Introduction

Social insects, such as ants, demonstrate remarkably efficient collective behaviors despite their limited individual capabilities. One of the most fascinating examples of such type of cooperative behavior is foraging, the process by which ants locate, retrieve, and distribute food resources across their colony. A decentralized coordination of their activities needed to achieve the overall goal emerges from simple local interactions, including pheromone-based communication and response to environmental challenges. Studying these mechanisms not only deepens our understanding of biological systems but also inspires algorithms in robotics, optimization, and distributed artificial intelligence.

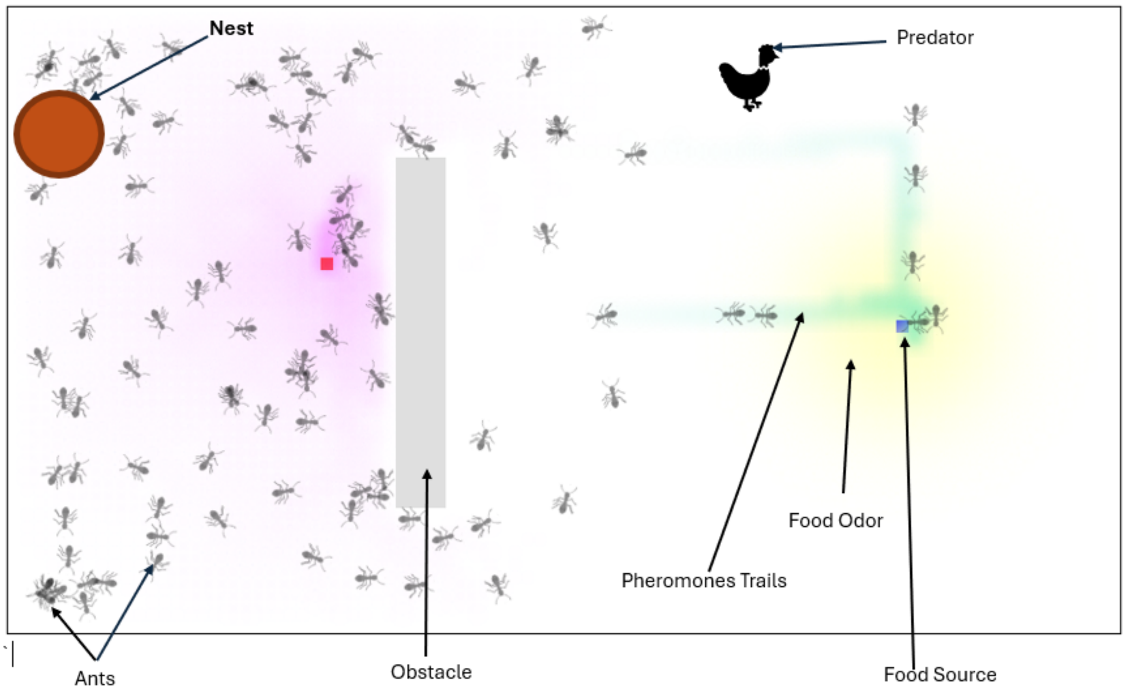

Figure 2.1: Foraging in the nature

Ant foraging behavior depicted in Figure (2.1) is a complex and highly organized activity driven by both individual actions and collective intelligence. **Ants** leave their **nest** in search of food, navigating through diverse and often challenging terrains that may include physical **obstacles** such as rocks, vegetation, and uneven ground. The presence of **predators** adds further complexity, as foraging ants must balance exploration with caution to ensure **survival**. A key element in this process is the use of **pheromones**, chemical signals that ants deposit along their paths to communicate with nestmates. When a **food source** is discovered, ants often detect it initially through food odor, which helps them orient toward the source. Upon returning to the nest, they reinforce the trail with pheromones, enabling others to follow the established route. Over time, as more ants use and strengthen this path, a highly efficient foraging trail emerges.

The environment itself, including temperature, humidity, and terrain structure, influences pheromone longevity and detection, thereby affecting the success of trail formation. The nest serves as the central hub of activity, where resources are collected and distributed. Ant foraging demonstrate an adaptive, decentralized system relying on local information and simple **behavioral rules** to achieve collective goals in a dynamic and often hostile world.

2.1.1 Goals and objectives

The primary objective of this study is to develop a computational framework for simulating ant foraging behavior using **agent-based modeling (ABM)** and **discrete-event simulation (DES)** with SimPy, a Python-based simulation tool. By modeling ants as autonomous agents interacting within a structured environment, we investigate how simple individual rules such as pheromone deposition, predator avoidance, and food retrieval—give rise to emergent colony-level behavior leading to efficent food foraging. Furthermore, we conduct a quantitative analysis of key system parameters such as pheromone decay rates, predator probability, food distribution to evaluate their impact on foraging outcomes. This simulation not only demonstrates the principles of self-organization in biological systems but also serves as a versatile foundation for studying swarm intelligence in applied contexts such as robotics, AI or optimization algorithms.

Our objective in this study is to use a well-known Python-based simulation tool to create a foraging ant simulation framework and make a quantitative study of properties which could having a major impact on the outcome of the process. This simulation models the emergent foraging behavior of ant colonies using agent-based modeling and discrete-event simulation with SimPy. The implementation demonstrates how simple individual behaviors can produce complex collective intelligence through environmental interactions.

2.1.2 Plan of work

To provide a clear roadmap for this study, the rest of this document will be structured as follows. In the next section, we present the architecture of the simulation framework, detailing the main components, core classes, and the interaction dynamics that govern agent behavior within the environment. After that, we will present the methodology used to conduct the experiments, including the established metrics to evaluate ant's communication efficiency, the simulation parameters, and the different scenarios implemented to evaluate the system's performance. Once this is done, we will present, analyze and discuss various results obtained from these experiments, focusing on key aspects such as food collection, agent efficiency, survival rates, and discovery dynamics, along with a discussion of the findings and any notable anomalies. Finally, we concludes the quantitative evaluation by summarizing the main contributions and suggesting directions for future research.

2.2 Simulation Framework structure

2.2.1 Simulation Overview

This study presents an **agent-based model** of ant foraging behavior, implemented in Python using the **SimPy** discrete-event simulation framework. The simulation captures key aspects of colony-level foraging dynamics, including **pheromone-mediated communication**, **predator avoidance**, and **adaptive resource collection** in a variable environment. By modeling individual ants as autonomous agents interacting within a 2D grid-based ecosystem, we investigate how simple behavioral rules give rise to emergent collective intelligence. The framework allows systematic testing of environmental parameters—such as predator density, food distribution, and pheromone decay rates—to quantify their impact on foraging efficiency and colony resilience.

2.2.2 Core components of the simulation

The simulation is designed using a class diagram (Figure 2.2) to model the interactions between key entities: the **Grid**, **Ants**, **Predator**, **Pheromones**, and **Food Source**. The environment is represented as a 2D square grid of fixed dimensions, where autonomous agents (ants) navigate in search of food while avoiding obstacles and predators. The grid may contain one or more food sources, which are modeled as SimPy containers with a finite capacity. Upon locating food, an ant releases pheromones—chemical signals that guide other ants toward the discovered resource. Additionally, the grid includes predators that pose threats by hunting and eliminating ants. The core components of the simulation are defined as follows:

1. **The Grid**: a 2D plane serving as the spatial foundation for the simulation;
2. **Ant**: an agent that explores the grid, collects food, and avoids hazards;
3. **Predator**: a dynamic hazard capable of attacking ants;
4. **Pheromone**: a chemical marker deposited by ants to influence colony behavior;
5. **Food Source**: a resource node that dispenses a unit of food when accessed by an ant.

These elements collectively form the basis of the foraging simulation, enabling the study of emergent colony behavior under varying environmental conditions.

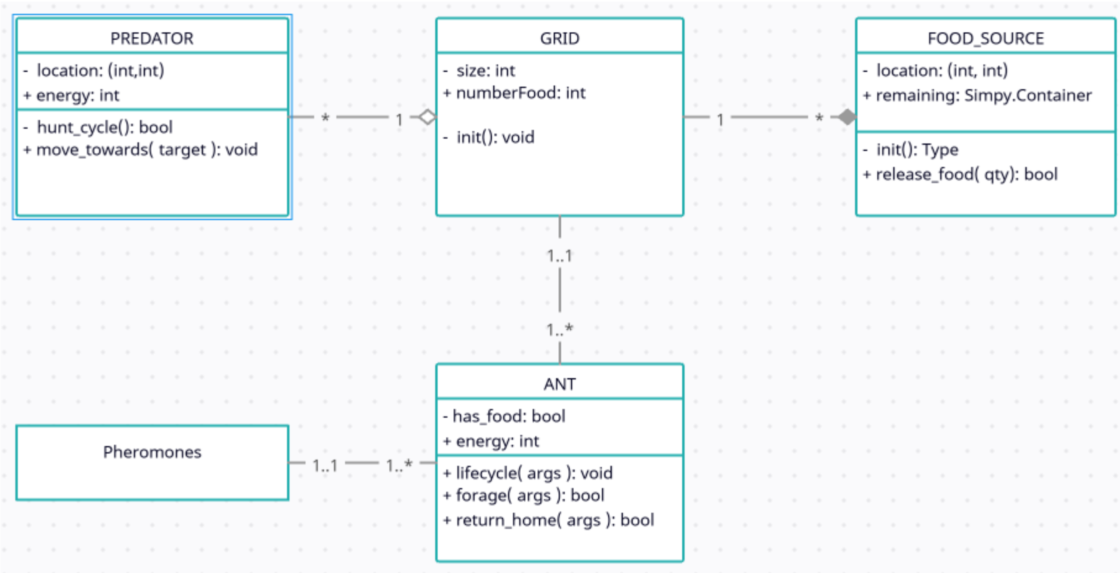

Figure 2.2: System class diagram

2.2.3 Description of core classes

FoodSource class

Role : The class represents food patches in an environment with limited capacity. An ant can come close to the food source and trigger an event that decreases the amount of food available. FoodSource is modeled as a Simpy shared resource of type Container. Using this type, Simpy allows one to model the production and consumption of a homogeneous, undifferentiated bulk. It may be either continuous (like water) or discrete (like apples).

```python
class FoodSource:
    def __init__(self, env, location):
        self.env = env
        self.location = location
        self.remaining = simpy.Container(env, FOOD_CAPACITY, init=FOOD_CAPACITY)
        self.discovery_time = None

```

Figure 2.3: FoodSource class in python

Attributes: Table 2.1 below presents the list of class attributes

Key Methods: The **FoodSource** class has a constructor which initializes a food source object within the simulation environment. It takes two parameters: *env*, which represents the SimPy simulation environment, and *location*, a coordinate or identifier indicating where the food source is placed in the simulation space. The *self.env* and *self.location*

Attribute	Type	Description
location	(int, int)	Grid coordinates (x, y)
remaining	simpy.Container	Track available food units (capacity: 30)
discovery_time	float	Simulation step when first discovered (None if untouched)

Table 2.1: FoodSource class attributes

attributes store these values for later use. The *self.remaining* attribute is defined as a simpy.Container, which models the quantity of food available at the source. It is initialized with a fixed capacity (FOOD_CAPACITY) and a starting amount equal to that capacity, simulating a full food source at the beginning. Finally, self.discovery_time is initialized as None and will be used later in the simulation to record the time at which the food source is first discovered by an ant. This setup enables dynamic tracking of food consumption and discovery events throughout the simulation.

Predator Class

Role : The class simulates predators that hunt ants based on pheromone trails and proximity. The predator can detect and kill ants within its attack range if the ants do not demonstrate or execute an escape behavior.

Attributes: Table 2.2 below lists the key attributes of a predator, their types and description.

Attribute	Type	Description
id	int	Predator's unique identifier
location	(int, int)	Grid coordinates (x, y)
energy	int	Energy level (decreases with every movement on the grid)
kill_count	int	Tracks the number of successful kills
hunting_range	int	Detection radius (the number of cells)

Table 2.2: Predator class attributes

Key Methods: The **Predator** class initializes a predator agent in the simulation. It takes five parameters: *env* (the SimPy simulation environment), *grid* (the environment in which movement and interactions occur), *pheromone_grid* (a grid storing pheromone levels left by ants), ants (a reference to the list of ant agents), and *predator_id* (a unique identifier for the predator). The constructor assigns these to corresponding instance attributes. The predator's initial position is randomly determined by calling the *_random_location()* method and stored in self.location. It starts with an initial energy level (*self.energy*) and a *kill_count* of 0 to track how many ants it has captured. Finally, *self.env.process(self.hunt_cycle())* initiates the predator's autonomous behavior in the simulation by launching its *hunt_cycle* process, which allows it to act over simulated time, moving through the grid, interacting with pheromones, and attempting to capture ants.

Ant Class

Role : The class simulates individual ant agents with foraging, navigation, and energy management. Ants can move in the grid either randomly or following the pheromone trails that they can perceive from the grid.

Attributes: Table 2.3 below represents the list of key attributes of the ant's class.

Attribute (%)	Type	Description
id	int	ant unique identifier
has_food	bool	Whether carrying food to nest
energy	int	Current energy (depletes at 1.5/step when carrying food)
pheromone_grid	np.ndarray(numpy)	Shared environment pheromone map
location	(int, int)	ant's position in the nest

Table 2.3: Ant class attributes

Key Methods: The Ant class sets up an individual ant agent within the simulation environment. It accepts six parameters: *env* (the Simpy simulation environment), *grid* (the spatial grid in which the ant moves), *pheromone_grid* (used to track and update pheromone trails), *nest_loc* (the fixed location of the nest), *ant_id* (a unique identifier for the ant), and *food_sources* (a list or collection of available food sources in the environment). The constructor initializes the position of the ant in the nest by assigning *self.location = nest_loc*, and stores its ID and the list of food sources. The boolean *attribute self.has_food* tracks whether the ant is currently carrying food, and *self.survived* indicates whether it is still alive. The initial energy of the ant is set using a predefined constant *ANT_ENERGY*. Finally, the behavior of the ant is launched by initiating the *life_cycle* process with *self.env.process(self.life_cycle())*, which allows the ant to begin autonomous actions based on time such as exploring, foraging, depositing pheromones, and reacting to environmental events throughout the simulation.

Python code snippet: The definition of *life_cycle* is shown in Figure 2.4.

foraging-simulation.py > analyze_results

```python
class Ant:
    def life_cycle(self):
        while self.survived and self.energy > 0:
            if not self.has_food:
                yield from self._forage()
            else:
                yield from self._return_home()
            yield self.env.timeout(1)
```

Figure 2.4: Ant life-cycle function

Explanation The *life_cycle* method defines the main behavioral loop of an ant throughout the simulation. It is implemented as a **generator function** using *yield*, allowing it to interact with the SimPy event-based simulation timeline. The loop continues to execute as long as the ant is **alive** (*self.survived is True*) and has **energy remaining** (*self.energy* $\geq$ *0*). Within each cycle, the ant checks whether it is currently carrying food:

- If **it's not carrying food** (i.e. *not self.has_food*), it calls the *_forage()* method, which contains logic for searching for food, moving through the environment, and possibly reacting to pheromones or encountering obstacles;
- If the ant **has food**, it calls the *_return_home()* method, which handles navigation back to the nest, possibly reinforcing the pheromone trail and finally, depositing food in the nest.

After each of these actions, the ant waits for **one time unit** using *yield self.env.timeout(1)*, simulating the passage of time in the environment before starting the next decision cycle. This design models realistic, time-stepped behavior and allows the ant to act independently within the broader multi-agent simulation.

ForagingSimulation Class

Role : The class serves as the central controller and environment manager for the entire ant foraging simulation. Its primary role is to encapsulate the high-level orchestration of the simulation, coordinating the environment, agent initialization, and time progression, making it the backbone of the simulation framework. The class attributes are listed in Table 2.4.

Attribute	Type	Description
grid	np.ndarray	20×20 grid tracking objects ('empty', 'obstacle'...)
pheromone_grid	np.ndarray	Matrix of pheromone concentrations
data	list	Records simulation metrics per timestep, which will be used at the end

Table 2.4: ForagingSimulation class attributes

Key Methods: *ForagingSimulation* class initializes the entire foraging simulation environment. It begins by creating a new SimPy *Environment* instance (*self.env*), which serves as the central scheduler for all time-based events in the simulation. The simulation space is represented by *self.grid*, a 2D NumPy array of size GRID_SIZE × GRID_SIZE, initialized with the value 'empty' to indicate that all cells are unoccupied at the start. In parallel, *self.pheromone_grid* is created as a 2D array of the same size, initialized with zeros to represent the absence of pheromone trails at the beginning. The attribute *self.data* is initialized as an empty list to store data collected during the simulation, such as performance metrics or event logs. Finally, the *_setup()* method is called to populate the environment with initial elements such as ants, predators, food sources, and the nest, effectively preparing the simulation for execution.

Foraging behavior implementation: Advanced *Pheromone* class

Role : The class introduces different types of pheromones such as *food*, *danger* or *explore* needed for advanced communication between ants. Improve pheromones communication to inform the colony of the presence of hazards.

Attributes : The list of class attributes is presented in Table 2.5

Attribute	Type	Description
Type	PheromoneType	$'food'$, $'danger'$ or $'explore'$
Strength	int	the strength of the pheromones which evaporates over time

Table 2.5: Pheromone class

2.2.4 Interaction Dynamics

This section outlines the key interaction mechanisms between *ants*, the *grid environment*, and *predators* in our agent-based simulation. Ants interact dynamically with their surroundings by navigating the grid, sensing pheromone levels, avoiding obstacles, and responding to food odors. Their movement decisions are influenced by the local concentration of pheromones, which guide them toward food sources and back to the nest. Interactions between ants are indirect but critical and occur primarily through pheromone trails that serve as a shared medium of communication. The grid environment itself acts as a spatial framework that constrains movement, stores environmental features including food sources and nest, but also supports pheromone diffusion and decay. Predators introduce a layer of threat, actively hunting ants based on proximity and possibly reacting to ant density or pheromone presence. These predator-ant interactions create a survival pressure that forces ants to adapt their paths and behaviors. Collectively, these interaction dynamics contribute to the emergent behavior of the colony, balancing exploration, exploitation, and survival within a shared and dynamic environment.

Ant-Ant Interaction Mechanisms

Pheromone Communication In the simulation, pheromone signaling plays a central role in coordinating ant behavior and allowing efficient foraging. **Trail laying** occurs when ants find food; they deposit pheromones along their return path, forming chemical gradients that help guide other ants to the discovered resource. In addition to foraging signals, ants also engage in **danger signaling** by releasing alarm pheromones near predators, causing nearby ants to initiate evasive behavior and avoid threat. The **pheromone dynamics** are governed by both deposition and evaporation processes. When ants return to the nest, they deposit pheromones based on context: **+40 units** if they carry food, to reinforce successful paths, and +20 units if they return empty handed, which can still indicate explored terrain. To ensure adaptability and prevent the persistence of outdated information, pheromone trails **evaporate at a rate of 7% per time step**, managed through the SimPy's timing mechanism (Figure 2.5). In addition, pheromone levels are **capped at a maximum of 100 units** to avoid over-saturation, maintaining balance in the chemical signaling system. This dynamic interplay ensures that the colony collectively adapts to changing environmental conditions and resource availability.

```python
def _evaporation_process(self):
    while True:
        self.pheromone_grid *= (1 - BASE_EVAPORATION)
        self.pheromone_grid[self.pheromone_grid < 0.1] = 0
        yield self.env.timeout(1)
```

Figure 2.5: Pheromone evaporation in python

Local Coordination Ants probabilistically alter paths in high-density areas (crowd avoidance) to reduce congestion. Also, direct encounters (grid cell overlaps) between ants(food recruitment) can amplify food retrieval efforts.

Conflict Resolution Ants may engage in brief contention for food access(resource competition), this is modeled via SimPy priority queues. And regarding *trail conflicts*, overlapping trails from multiple food sources are resolved through pheromone strength comparisons.

Ant-Grid Interactions

Ants navigate a discrete 2D grid, perceiving local cell states (pheromones, food, predators). The grid enforces spatial constraints (e.g., obstacle avoidance) and mediates pheromone diffusion/evaporation. Food sources are grid-located containers; ants reduce their capacity upon retrieval.

Ant-Predator Interactions

Predators patrol the grid, stochastically attacking ants within their detection radius. Ants employ evasion strategies like pheromone-based danger signals or random directional changes. Predator success rates modulate colony foraging risk/reward trade-offs.

Energy Model

Represents the energy consumption of an ant's activity during the foraging process and specify how energy is restored upon returning to the nest. Ants spend energy as they move through the environment, with **1 energy unit consumed per time step while exploring** and a higher rate of **1.5 units per step when carrying food** (Figure 2.6), this represents the added effort of transport. This consumption creates a natural limit on how long ants can remain active in the field and introduces a strategic trade-off between exploration and return efficiency. When ants return to the **nest**, their energy is **fully replenished** to a maximum of 250 units(simulation parameter), regardless of whether they are carrying food. This simple yet effective energy mechanism ensures that foraging behavior is controlled by physiological limits, adding realism to the simulation and encouraging timely returns to the nest to sustain individual survival and colony productivity.

```python
def _forage(self):
    moves = self._valid_moves()
    if not moves:
        return

    weights = []
    for pos in moves:
        pheromone_weight = 0.7 * (self.pheromone_grid[pos]**2 / (MAX_PHEROMONE**2))
        exploration_weight = 0.3 / len(moves)
        weights.append(pheromone_weight + exploration_weight)

    self.location = random.choices(moves, weights=weights)[0]
    self.energy -= 1
```

(a) 1 unit of energy consumed when foraging (without food)

```python
def _return_home(self):
    """Yield control while returning to nest with food"""
    while self.has_food and self.location != self.nest_location:
        moves = self._valid_moves()
        if not moves:  # No valid moves available
            yield self.env.timeout(1)  # Wait a step
            continue

        # Weight moves toward nest
        weights = [1/(self._distance(pos, self.nest_location)+0.1 )
                for pos in moves]
        self.location = random.choices(moves, weights=weights)[0]
        self.energy -= 1.5  # Higher cost when carrying food
```

(b) 1.5 units of energy consumed when returning home(with food)

Figure 2.6: Ant metabolic cost with and without food

2.3 Methodology and Experiments

The simulation is initialized with standardized baseline parameters to establish control conditions. During execution, we systematically measure and analyze primarily pheromone utilization efficiency in food localization, and secondly, the navigation effectiveness during nest return journeys. This dual-metric approach allows for quantitative assessment of the collective foraging performance of the colony.

2.3.1 Metrics for Assessing Pheromone Communication Efficiency

To assess the effectiveness of pheromone-based communication in guiding ant behavior, several quantitative metrics are employed in the simulation. The **Time to First Food Discovery** measures how quickly the first ant locates a food source, indicating the responsiveness of the colony's initial exploration efforts. The **Time to Collect All Food** captures the overall foraging efficiency by recording how long it takes for the colony to fully deplete all available food sources. The **Pheromone Trail Strength** is evaluated as the average concentration of pheromones along the shortest successful paths to food, reflecting how well pheromone signals guide movement. **Exploration Efficiency** is calculated as the percentage of ants that choose to follow existing pheromone trails compared to those engaging in random, unguided movement, providing insight into the colony's reliance on chemical communication. Lastly, **Redundancy in Paths** measures the number of unique routes ants take to reach food, highlighting the degree of path optimization or dispersion within the colony. Collectively, these metrics provide a robust framework for evaluating the performance and adaptability of the pheromone communication system under varying environmental and experimental conditions.

1. **Time to First Food Discovery**: How quickly the first ant finds food;
2. **Time to Collect All Food**: Total time for all food to be collected;

3. **Pheromone Trail Strength**: Pheromone concentration along the shortest path to food;
4. **Exploration Efficiency**: Percentage of ants following trails vs. random exploration;
5. **Redundancy in Paths**: Number of unique paths ants take to reach food.

2.3.2 Simulation Parameters

The model's behavior is governed by a set of configurable parameters that define the environment, agent properties and system dynamics. These variables control the simulation's scale, complexity, and emergent behaviors, allowing for systematic experimentation with different foraging scenarios. Below, we describe each parameter and its role in shaping the ant colony's collective behavior, independent of their specific assigned values.

This setup enables controlled testing of the efficiency of the pheromone communication, predator-prey interactions, and resource allocation strategies in a reproducible simulation framework.

2.3.3 Simulation Setup (Baseline)

The baseline configuration establishes standard experimental conditions for investigating ant foraging dynamics. This parameter set will serves as a reference point for comparative studies, ensuring reproducibility while capturing essential aspects of collective food retrieval behavior.

Specifically, the system parameters are set to the following vales:

- GRID_SIZE = 100;
- NUM_ANTS = 200;
- NUM_PREDATORS = 1;
- NUM_FOOD_SOURCES = 5;
- FOOD_CAPACITY = 300;
- BASE_EVAPORATION = 0.07;
- PHEROMONE_DEPOSIT = 20;
- MAX_PHEROMONE = 100;

Table 2.6: Simulation Parameters and Their Descriptions

Parameter	Description
GRID_SIZE	Defines the edge length (in cells) of the square 2D simulation environment. Governs spatial resolution and computational scale.
NUM_ANTS	Specifies the initial population of autonomous ant agents in the colony. Determines foraging density and collective behavior complexity.
NUM_PREDATORS	Sets the count of predator agents that actively hunt ants. Influences environmental risk and adaptive colony strategies.
NUM_FOOD_SOURCES	Controls the number of discrete resource locations available in the environment. Affects foraging competition and trail network topology.
FOOD_CAPACITY	Represents the total units of food each source can provide before depletion. Impacts simulation duration and resource scarcity dynamics.
BASE_EVAPORATION	Determines the rate at which pheromones decay per simulation step. Regulates trail persistence and information freshness.
PHEROMONE_DEPOSIT	Specifies the pheromone units deposited by ants when traversing paths. Affects trail strength and gradient steepness.
MAX_PHEROMONE	Imposes an upper bound on pheromone concentration per grid cell. Prevents unbounded signal accumulation.
ANT_ENERGY	Defines the lifespan metric for individual ants (in steps or energy units). Introduces mortality constraints.
PREDATOR_SENSE_RANGE	Sets the radial distance (in cells) at which predators detect ants. Governs predation pressure spatial extent.
OBSTACLE_DENSITY	Specifies the probability of grid cells being impassable obstacles. Modifies navigation complexity.
SIM_TIME	Determines the total duration (in simulation steps) for experimental runs. Ensures standardized temporal bounds.
NUM_TRIALS	Determines the total number of times we run the simulation with a fixed number of parameters.

- ANT_ENERGY = 250;
- PREDATOR_SENSE_RANGE = 4;
- OBSTACLE_DENSITY = 0.1 (10% of grid cells);
- SIM_TIME = 500;

- NUM_TRIALS = 100;
- Metrics:
  - **Foraging Efficiency**: Food collected per ant per time unit;
  - **Survival Rate**: % ants surviving simulation;
  - **Discovery Time**: Time to find food sources.

2.3.4 Experimental Scenarios

To systematically evaluate emerging foraging behaviors, we designed multiple experimental conditions that modify key parameters from the baseline configuration. These scenarios isolate specific aspects of colony-environment interactions while maintaining controlled comparison conditions. The simulation explores five distinct experimental scenarios, each targeting a specific aspect of environmental or behavioral complexity.

Scenario 1: The variation of Pheromone Dynamics tests how changes in pheromone evaporation rates and deposition amounts affect communication efficiency and path optimization. **Scenario 2: Predator Pressure Gradients** introduces varying levels and distributions of predator presence to observe how ants adapt their movement patterns and survival strategies under threat. **Scenario 3: Resource Distribution** Tests manipulates the spatial placement and quantity of food sources - clustered, uniform, or sparse - to assess how the layout of the resources influences the discovery time and collection efficiency. **Scenario 4: Population scaling** investigates the effects of colony size by varying the number of ants, providing information on scalability, congestion, and cooperation under different population densities. Finally, **Scenario 5: Extreme Conditions** simulates high-stress environments, such as rapid pheromone decay, high predator density, or minimal resources, to test the limits of the system's resilience and behavioral flexibility. Together, these scenarios offer a comprehensive view of the multi-agent system's performance across diverse and realistic challenges.

Scenario 1: Pheromone Dynamics Variation

Pheromone Dynamics Variation investigates how different pheromone behaviors influence the efficiency and structure of foraging. This scenario explores the effects of varying **evaporation rates** (low, medium, or high) alongside changes in **pheromone deposit quantities**, ranging from weak to strong. These modifications directly affect the persistence and strength of chemical trails laid by ants. By analyzing how these variations influence key metrics such as **trail network stability**, **latency in discovering food**, and the **formation of redundant or inefficient paths**, the experiment aims to understand the trade-offs between responsiveness and consistency in pheromone-based communication. Lower evaporation may lead to more stable but potentially outdated trails, while higher rates encourage fresher paths at the cost of faster information decay. Similarly, stronger deposits can accelerate convergence on productive routes but may also increase the risk of overcommitment. This scenario provides insight into how finely tuned pheromone dynamics affect colony-wide coordination. In particular, it

- Tests trail persistence effects by modifying: Evaporation rates (low/medium/high) and Deposit quantities (weak/standard/strong).
- Measures impact on: Trail network stability, Food discovery latency and Redundant path formation.

Table 2.7 below outlines the attributes to be adjusted along with their corresponding values.

Condition	BASE_EVAPORATION	PHEROMONE_DEPOSIT
Weak/Transient Trails	0.03	10
Baseline	0.07	20
Strong/Persistent Trails	0.01	40
No Pheromones	N/A	0

Table 2.7: Pheromone Trail Parameters

Scenario 2: Predator Pressure Gradients

Predator Pressure Gradients examines the colony's behavioral adaptations in response to varying levels of external threat by systematically varying the **number of predators**, **hunting rang**e, and **attack frequency**. These parameters simulate different intensities of predation risk within the environment. The experiment focuses on how ants assess and respond to danger, evaluating their **risk assessment strategies**, such as rerouting or hesitating near high-risk areas. It also investigates **trail abandonment patterns**, where previously active foraging paths may be deserted due to predator presence, affecting the overall efficiency of food retrieval. Futhermore, this scenario analyzes the mortality-recovery tradeoffs, considering how the loss of foragers impacts the colony's ability to maintain productivity and re-establish efficient trails. By observing these dynamics, the scenario provides insights into the resilience of decentralized coordination under hostile conditions and how predator pressure reshapes foraging behavior. In particular, it performs the following actions:

- Examines threat response by varying: Predator population density, Hunting range parameters and Attack frequency.
- Evaluates colony: Risk assessment strategies, Trail abandonment patterns and Mortality-recovery tradeoffs.

Table 2.8 below outlines the attributes to be adjusted along with their corresponding values.

Condition	NUM_PREDATORS	PREDATOR_SENSE_RANGE
No Predators	0	0
Baseline	1	4
High Predation	3	6
Wide Detection	1	8

Table 2.8: Predator Parameters

Scenario 3: Resource Distribution Tests

Resource Distribution Tests explores how different spatial configurations and availability patterns of food sources affect collective foraging behavior. This scenario introduces variations such as clustered versus dispersed food placement, heterogeneous source capacities, and dynamic replenishment patterns to simulate a range of environmental resource distributions. The experiment aims to analyze how these factors influence the colony's **division of labor**, particularly how ants self-organize to cover multiple sources efficiently. It also examines the **balance of exploration and exploitation**, assessing how ants allocate effort between searching for new resources and exploiting known ones. In addition, it evaluates the efficiency of load distribution, determining whether food retrieval is evenly shared among foragers or concentrated along a few dominant paths. By observing how the colony adapts to varying resource landscapes, this scenario provides valuable insights into the flexibility and scalability of foraging strategies under realistic environmental heterogeneity. In particular, it performs the following actions:

- Investigates spatial allocation through: Clustered vs dispersed food placement, Variable source capacities and Dynamic replenishment patterns.
- Analyzes effects on: Division of labor, Exploration-exploitation balance and Load distribution efficiency.

Table 2.9 below outlines the attributes to be adjusted along with their corresponding values.

Condition	NUM_FOOD_SOURCES	FOOD_CAPACITY
Centralized	1	1500
Baseline	5	300
Distributed	10	150
High Obstruction	N/A	N/A

Table 2.9: Food Source and Obstruction Parameters

Scenario 4: Population Scaling

Population Scaling tests the robustness of the simulation by progressively increasing colony size and adjusting **ant-to-predator** ratios to capture density-dependent effects. As the number of agents rises, the scenario tracks emergent **traffic patterns** on shared trails, shifts in **communication efficiency** as pheromone signals potentially saturate or dissipate, and any **scaling limitations** that appear—such as congestion around the nest, heightened predator impact at lower ant-to-predator ratios, or diminishing returns in collective efficiency. By illuminating how foraging performance evolves with population growth and changing ecological pressures, this scenario reveals the thresholds at which the decentralized coordination of the colony begins to falter or adapt. In particular, it performs the following actions:

- Assesses system robustness via: Colony size increments, Ant-predator ratios and Density-dependent effects.
- Monitors emergent: Traffic patterns, Communication efficiency and Scaling limitations.

Table 2.10 below outlines the attributes to be adjusted along with their corresponding values.

Condition	NUM_ANTS	ANT_ENERGY
Sparse And Long-Lived	50	500
Baseline	200	250
Dense/Short-Lived	500	100
High Density	200	N/A

Table 2.10: Ant Colony Parameters

Scenario 5: Extreme Conditions

Extreme Conditions serves as a stress test to evaluate the resilience and adaptability of the ant colony under highly challenging environments. This scenario introduces compounded difficulties such as **accelerated pheromone evaporation**, **minimal food availability**, **high predator density**, and **unfavorable terrain layouts** to push the system to its limits. The aim is to observe how the colony's foraging strategies degrade or adapt under pressure, examining critical behaviors such as **survival prioritization**, **fallback mechanisms**, and **emergency trail formation**. By monitoring foraging success rates, mortality levels, and behavioral shifts, this scenario provides insight into the colony's capacity to maintain function and recover in adverse conditions. It highlights the strengths and potential failure points of the decentralized, pheromone-driven coordination model when subjected to extreme ecological stressors.

Table 2.11 below outlines the attributes to be adjusted along with their corresponding values.

Condition	Param 1	Param 2
HostileEnv	OBSTACLE_DENSITY = 0.3	NUM_PREDATORS = 3
Baseline	OBSTACLE_DENSITY = 0.3	NUM_PREDATORS = 3
Poor_comms	BAS_EVAPORATION = 0.2	PHEROMONE_DEPOSIT = 5
Scarce_resources	NUM_FOOD_SOURCES = 2	FOOD_CAPACITY = 100

Table 2.11: Stress Test Parameters

Operation mode Each scenario is executed by 100 simulation runs with randomized initial conditions to ensure statistical significance. Between scenarios, we maintain identical environmental parameters (grid size, obstacle layout) while systematically altering target variables. Performance metrics are normalized against baseline results to highlight differential effects.

3 Simulation results & Discussion

In this chapter, we aim to present and analyze the results of simulation experiments carried out in the five scenarios mentioned above. Each scenario highlights specific aspects of the foraging system, including pheromone communication, predator avoidance, resource allocation, population dynamics, and stress resilience. We will use a combination of tables and visual charts, to provide a clear and comparative overview of key performance metrics such as food **discovery time**, **trail stability**, **survival rates** and **path efficiency**. The results are then discussed in detail, with observations on behavioral trends, unexpected outcomes, and notable differences across experimental conditions. Through this analysis, we aim to draw meaningful conclusions about the strengths, limitations, and adaptability of the ant-based multi-agent system in varying environmental contexts.

For each unique configuration, we executed **100 simulation runs** to gather data. After each set of simulations, we leveraged the **Numpy Python** library to compute the **mean and standard deviation** of the collected metrics. This statistical analysis allowed us to understand both the central tendency and the variability of our results. Finally, to effectively communicate our findings and achieve our visualization objectives, we utilized the **Matplotlib** Python library to generate appropriate charts and diagrams.

3.1 Scenario 1: Pheromone Dynamics Variation

In this scenario, we have four distinct parameter configurations: **Baseline**, **No Pheromones**, **Strong and Persistent Trails**, and **Weak Transient Trails**.

3.1.1 Food discovery latency

Table 3.1 presents a compilation of the **total quantity of food collected** by the agents within the simulation environment for each of the tested configurations. This metric is crucial for understanding the efficiency and success of different parameter settings in terms of resource acquisition.

The results of Table 3.1 clearly demonstrate the impact of pheromone behavior on the colony's foraging performance, particularly in terms of the total amount of food collected. The **baseline** setup, which uses default pheromone parameters, produced the highest mean food collection **(67.83)** but also showed significant variability **(std = 106.43)**, suggesting occasional high-performance runs and frequent inconsistencies. The **No Pheromones** condition drastically reduced efficiency, with a low average of **15.22** food units collected and a much smaller standard deviation, indicating limited success and consistent under-performance due to the absence of guided trail formation. In contrast, both **Strong Persistent Trails** and **Weak Transient Trails** showed moderate improvements over the no-pheromone case, collecting means of **41.37** and **45.99** respectively (Figure 3.1), though both still fell short of the baseline. The higher standard deviations in these two cases suggest instability in trail formation, either due to excessive reinforcement (in Strong Persistent Trails) or rapid decay (in Weak Transient Trails). Overall, these results highlight the importance of balanced pheromone dynamics in allowing effective food discovery and consistent performance.

Table 3.1: Food Collected Across Different Scenarios

Parameters Configuration	**mean**	**std**
Baseline	67.83	106.428907
No Pheromones	15.22	24.078818
Strong and Persistent Trails	41.37	69.280232
Weak Transient Trails	45.99	87.723451

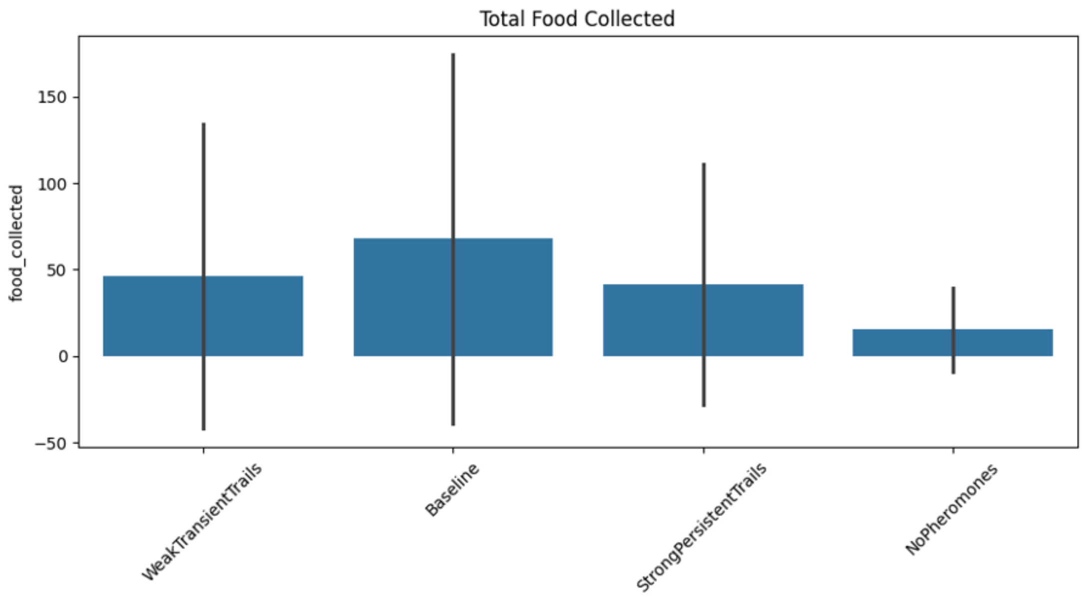

Figure 3.1: Food Collected Across Different Scenarios

3.1.2 Foraging Efficiency & Discovery time

Table 3.2 below presents a compilation of the **Foraging Efficiency** measurement within the simulation environment for each of the configurations.

Table 3.2: Foraging Efficiency Across Different Scenarios

Parameters Configuration	mean	std
Baseline	0.006783	0.010643
No Pheromones	0.001522	0.002408
Strong Persistent Trails	0.004137	0.006928
Weak Transient Trails	0.004599	0.008772

The results from Table 3.2 for **foraging efficiency** in Scenario 1 further support the critical role of pheromone signaling in optimizing collective behavior. The **baseline** configuration achieved the highest mean efficiency **(0.006783)**, indicating a relatively effective conversion of movement and effort into successful food collection. However, the high standard deviation **(0.010643)** once again reflects variability across runs. The **No Pheromones** setup, which lacks any trail guidance, performed the worst with a mean efficiency of only **0.001522**, showing how random exploration leads to poor coordination and minimal resource return. **Strong Persistent Trails** and **Weak Transient Trails** yielded moderate efficiencies (**0.004137** and **0.004599**, respectively), confirming that both overly persistent and rapidly evaporating trails can hinder optimal performance. The elevated standard deviations in these two cases suggest inconsistent exploitation of pheromone paths, possibly due to overcrowding or unstable trail dynamics. Overall, these findings emphasize that a well-balanced pheromone system not only boosts the amount of food collected but also improves the overall efficiency of foraging efforts.

The scatterplot (see Figure 3.1) likely exhibits a downward trend, indicating a negative correlation where longer discovery times correspond to lower foraging efficiency. **Baseline** appears in the top-left, representing optimal performance with high efficiency and minimal discovery time. In contrast, **NoPheromones** is positioned at the bottom-right, reflecting the poorest performance with the lowest efficiency and the longest discovery time. **StrongPersistentTrails** and **WeakTransientTrails** fall between these extremes, forming a gradient toward the bottom-right. While they demonstrate an improvement over the no-pheromone scenario, their performance remains inferior to Baseline.

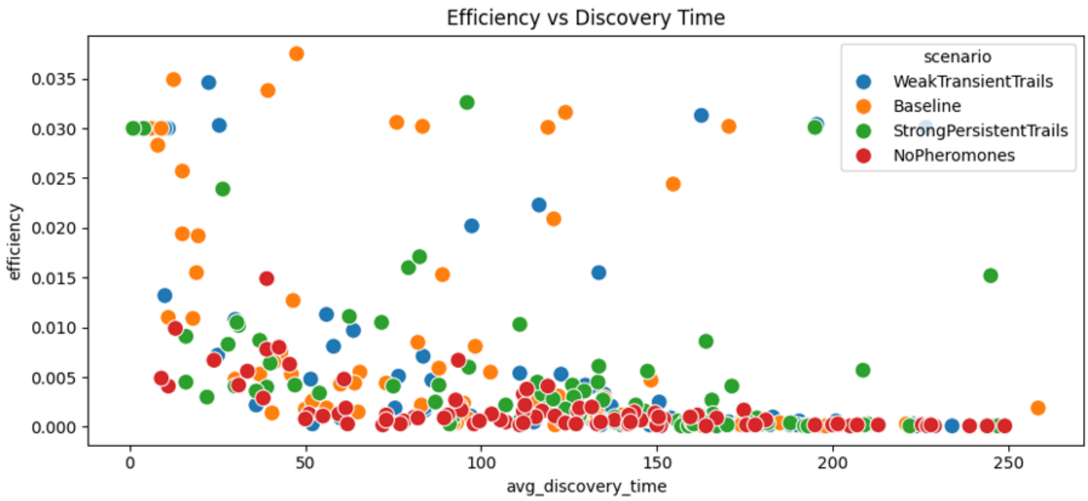

Figure 3.2: Foraging Efficiency vs average discovery time

3.2 Scenario 2: Predator Pressure Gradients

In this scenario, we have 4 different configurations: **Baseline**, **High Predation**, **No Predators**, and **Wide Detection**.

3.2.1 Mortality-recovery tradeoffs

Table 3.3 presents a compilation of the **number of ants that survived** after running the simulation with various environmental and behavioral parameters. The results from Table 3.3 clearly illustrate how varying levels of **predation pressure** and **detection capabilities** impact the survival rates of the ant colony. The results from Table 3.3 clearly

Table 3.3: Ants Survived Across Different Scenarios

Scenario	ants_survived
	Value
Baseline	191 ± 18.3
High Predation	156 ± 38.1
No Predators	200 ± 0.0
Wide Detection	184 ± 25.5

illustrate how varying levels of predation impact the survival of ants within the colony. The **No Predators** setup resulted in perfect survival, with an average of **200 ants** and **zero variability**, serving as a control benchmark. In contrast, the **High Predation** scenario showed the most severe impact, with an average of only **156 ants surviving** and a large standard deviation (±38.1), indicating frequent and significant mortality across runs. This suggests that high predator density introduces unpredictability and substantial risk to foraging ants.

The **baseline** condition, with standard predator parameters, achieved **191 survivors** on average **(±18.3)**, showing moderate threat and manageable losses. The **Wide Detection** scenario, where predators have a larger hunting range but not necessarily greater frequency, led to **184 survivors** with a slightly higher variance **(±25.5)**, indicating that broader threat zones do increase risk but may not be as deadly as sheer predator numbers. Overall, these results demonstrate that both predator density and detection range signif-

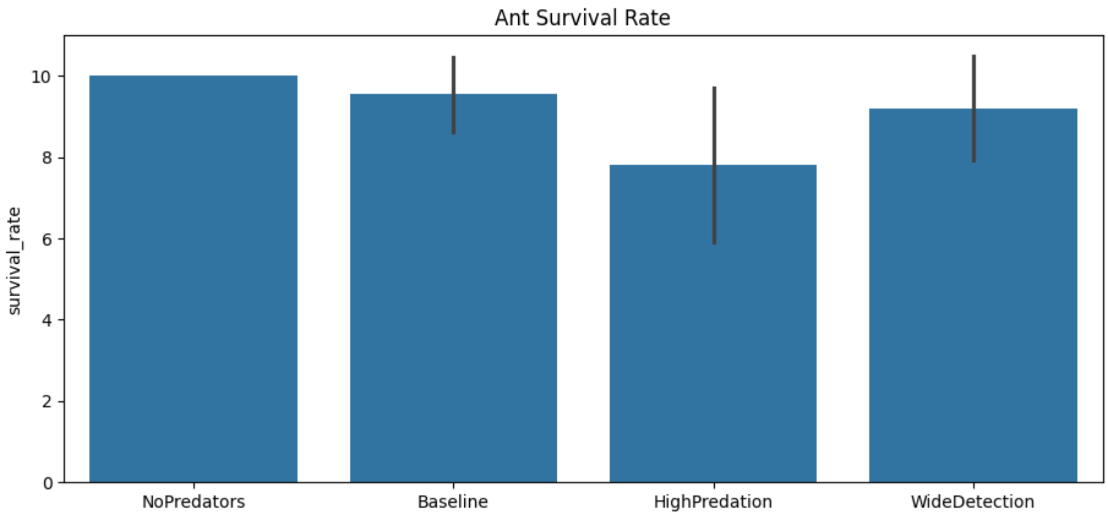

Figure 3.3: Ants Survived Across Different Scenarios

icantly influence colony survival, and that ants exhibit better resilience when threats are moderate or spatially limited. The increasing standard deviations across higher-risk scenarios also reflect the greater uncertainty and challenge posed by aggressive or wide-ranging predators.

3.2.2 Food collection dynamics with predator Gradient

Table 3.4 analyze the impact of varying the level of predation on the quantity of food agents can bring back to the nest.

The food collection results from Table 3.4 present an interesting perspective on how predation pressure influences foraging success. Surprisingly, the **No Predators** condition did not yield the highest food collection, although it achieved a relatively high mean **(49.04)** with a substantial standard deviation **(±87.38)**, indicating that while some runs were highly productive, others were not, possibly due to other limiting factors like **trail formation** or **food distribution**.

Table 3.4: Food Collected Across Predation Scenarios

Scenario	mean	std
Baseline	28.66	48.226155
High Predation	45.84	88.905207
No Predators	49.04	87.377611
Wide Detection	42.49	80.724059

Interestingly, **High Predation** led to a slightly lower mean **(45.84)** but a similarly high variability **(±88.91)**, suggesting that while more ants were lost (as seen in survival data), the survivors may have optimized their routes out of necessity, leading to concentrated and efficient exploitation of known food paths in some runs. This could reflect an adaptive tradeoff where reduced numbers force more efficient behavior among remaining ants.

The **Wide Detection** setup yielded slightly lower performance **(42.49, std ±80.72)**, likely due to increased spatial pressure from predators causing frequent trail disruptions and **evasive behavior**. Finally, the Baseline scenario produced the **lowest food collection (28.66)** with the **least variability (±48.23)**, indicating more consistent but modest performance under balanced threat levels.

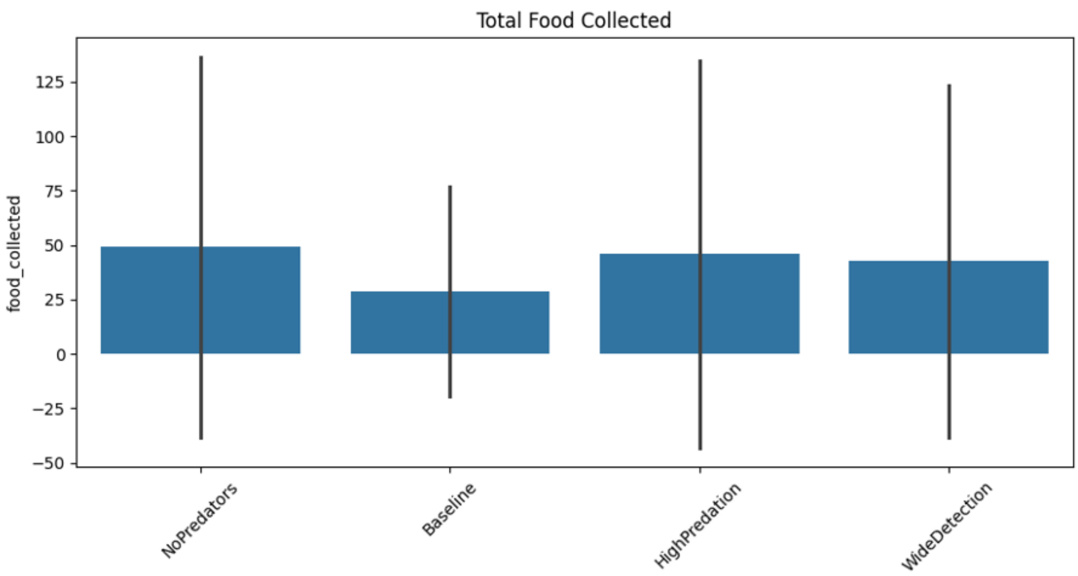

Figure 3.4: Food Collected Across Predation Scenarios

These results (Figure 3.4) suggest that moderate or even high predation does not necessarily suppress foraging output and in some cases, may pressure the system into more efficient behavior. However, high variability across all predator-exposed scenarios highlights the instability and unpredictability introduced by dynamic threats.

3.3 Scenario 3: Resource Distribution Tests

Under this specific scenario, we explore the influence of food placement strategies on simulation outcomes. We investigate four distinct configurations: **Centralized**, **Distributed** and **High Obstruction** with **Baseline** serving as a control for comparison.

3.3.1 Exploration-exploitation balance

We analyze pheromones deposit on the grid in the various configurations

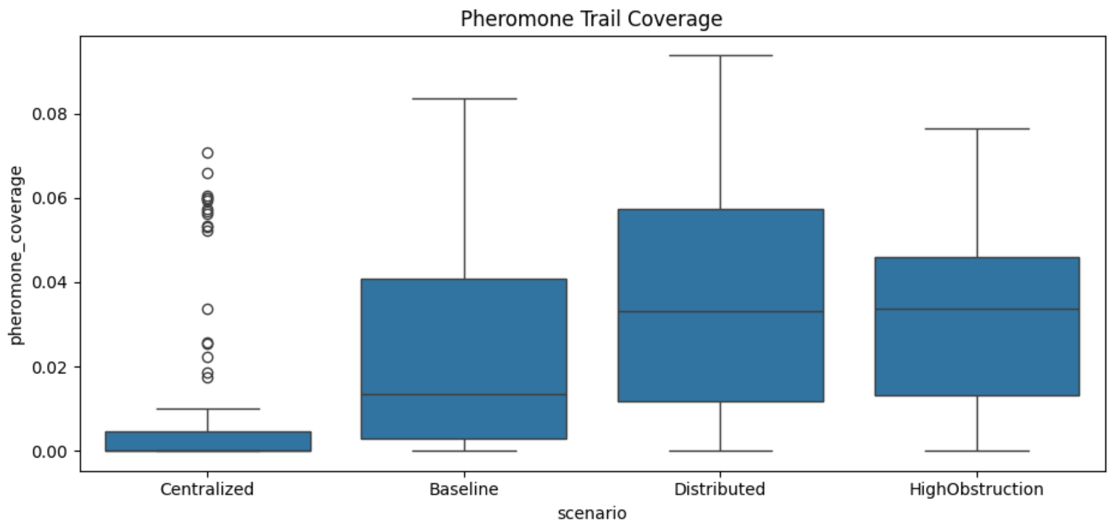

Figure 3.5: Pheromones coverage trail

3.3.2 Resource Distribution and Food Collected

Table 3.5 below compiles the quantity of food collected in the various configurations related to food distribution in the grid The results from Table 3.5 highlight how the spatial arrangement and accessibility of food sources influence foraging performance. The High Obstruction scenario achieved the highest mean food collection **(63.30)** with relatively low variability **(±64.22)**, indicating that despite obstacles in the environment, ants were still able to exploit food effectively—possibly due to improved trail reinforcement or localized specialization in certain accessible routes.

Table 3.5: Food Collected Across Different Food Placement Configurations

Scenario	mean	std
Baseline	45.60	95.514238
Centralized Food Sources	21.76	64.236325
Distributed Food Sources	58.31	72.135912
High Obstruction (obstacles)	63.30	64.224433

The **Distributed** setup followed closely with a mean of **58.31** and slightly higher variance **(±72.14)**, suggesting that when food is spread across the environment, ants can exploit multiple fronts, leading to more balanced load distribution and higher overall yield. However, the variability reflects occasional inefficiencies in coordinating across distant locations.

In contrast, the **Centralized** condition, where food is clustered in one region, resulted in the lowest mean **(21.776)** and lower variability **(±64.24)**, likely due to congestion, over reinforced paths, and limited exploitation opportunities when competition between ants is high or access is disrupted. The baseline scenario produced a moderate

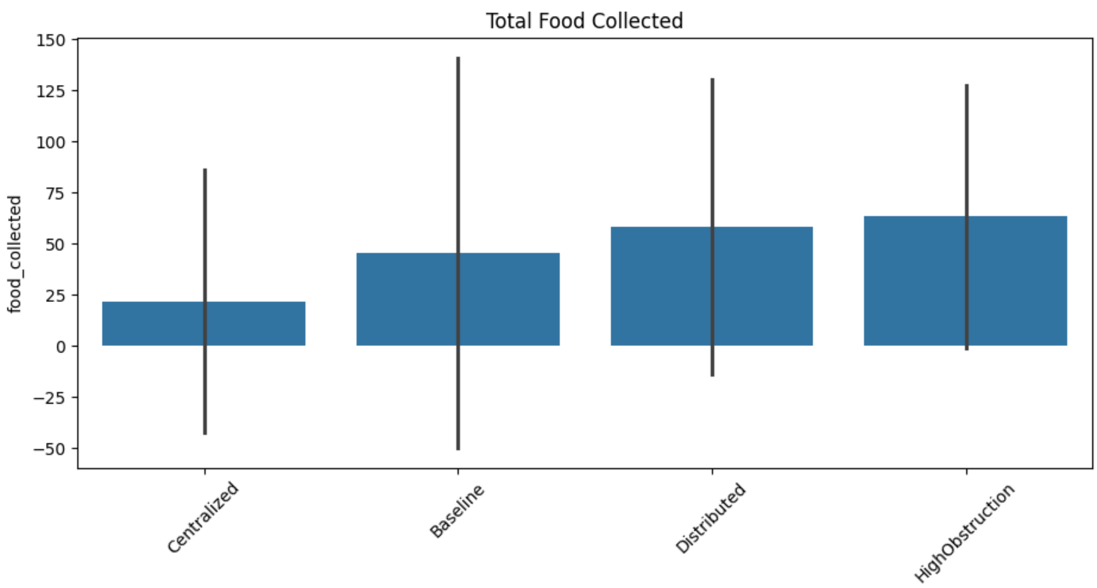

Figure 3.6: Food Collected Across Different Food Placement Configurations

mean of **45.60** with the highest variability **(±95.51)** (see Figure 3.6), serving as a control with standard food layout and conditions. In general, these findings suggest that **distributed or obstructed environments may actually enhance colony foraging** by encouraging exploration, decentralization, and more efficient division of labor, while centralized resources can lead to bottlenecks and reduced collection performance.

3.4 Scenario 4: Population Scaling

We gradually increase the number of our foraging agents. In this scenario, we have four main configurations: Baseline, Sparse and long-lived, Dense and short-lived and finally High Density.

3.4.1 Foraging Efficiency

3.4.2 Food collected

Table 3.6: Food Collected Across Different Ant Longevity and Density Configurations

Scenario	mean	std
Baseline	50.51	97.525184
Dense/Short-Lived	39.58	78.074580
High Density	98.66	121.250656
Sparse/Long-Lived	19.90	52.272885

The results from Table 3.6 demonstrate how variations in colony size and population density influence food collection efficiency. The **High Density** scenario clearly stands out, achieving the highest mean food collected **(98.66)** despite a high standard deviation **(±121.25)**. This suggests that larger colonies can harvest significantly more food, but their performance is more variable—possibly due to congestion effects or uneven trail formation under crowded conditions.

The **Baseline** scenario had a moderate performance **(50.51, std ±97.53)**, serving as the reference point for balanced colony size and density. Interestingly, the **Dense and Short-Lived** setup, which likely involves a high initial population with shorter lifespans or quicker exhaustion, underperformed relative to the baseline **(39.58, std ±78.07)**, suggesting that sheer numbers are insufficient if individuals lack endurance or sustainability.

On the opposite end, the **Sparse And Long-Lived** colony collected the least food **(19.90, std ±52.27)**, highlighting the limitations of having too few ants even if they persist longer. The lower variance here also suggests more consistent—but modest—performance due to the limited capacity for parallel exploration and exploitation.

Overall, the results (see Figure 3.7) indicate that scaling up population size can dramatically boost resource collection, but only up to a point. Beyond that, efficiency may suffer from coordination challenges, making population composition and trail dynamics critical to balancing performance and stability.

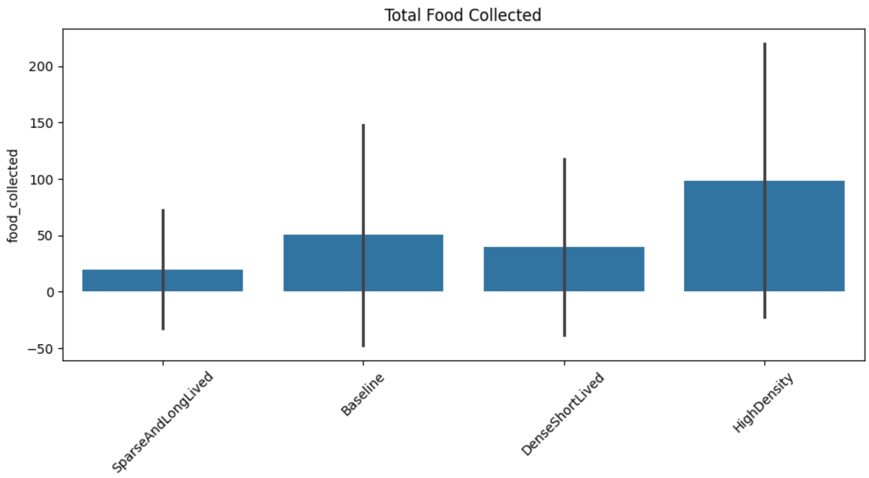

Figure 3.7: Food Collected Across Different Ant Longevity and Density Configurations

3.5 Scenario 5: Extreme Conditions

The results from **Scenario 5: Extreme Conditions** provide a compelling **stress test** of the colony's resilience under compounding or limiting environmental factors. Each condition isolates a major challenge to the foraging *system—communication* breakdown, *environmental hostility*, and *resource scarcity*.

3.5.1 Foraging efficiency

Extreme conditions impacts are clearly reflected in the **food_collected** metric. Table 3.7 compile the figures of foraging effciency The **Scarce_resources** scenario performed the

Table 3.7: Food Collected Across Different Environmental Configurations

Scenario	mean	std
Baseline Configuration	45.77	80.482115
Hostile Environment	52.75	93.815893
Poor ants communication (limited pheromones strength)	24.41	59.056743
Scarce resources	3.80	8.061631

worst by far, with **a mean food collection of only 3.80** and very low variance **(±8.06)**. This outcome is expected, as limited food availability sets a hard ceiling on what the colony can achieve, regardless of behavior or strategy. The consistency (low standard deviation) suggests that performance is uniformly poor, constrained more by environmental scarcity than by behavioral variability.

The **Poor_comms** scenario, where pheromone signaling is likely weakened or disabled, resulted in a **significant drop in performance (mean 24.41, std ±59.06)** compared to the **Baseline (mean 45.77)**. This highlights how essential effective communication is for guiding foragers and reinforcing productive trails.

Interestingly, the **HostileEnv** condition, which likely includes more obstacles or increased predator pressure, showed **slightly better performance than the Baseline (52.75 vs. 45.77)**, though with higher variability **(±93.81 vs. ±80.48)**. This may reflect adaptive behaviors such as trail rerouting or risk-based evasion that, under certain runs, lead to more efficient resource exploitation. It also suggests that a well-configured colony can maintain—and sometimes exceed—baseline performance even in dangerous environments, thanks to its decentralized flexibility and distributed redundancy.

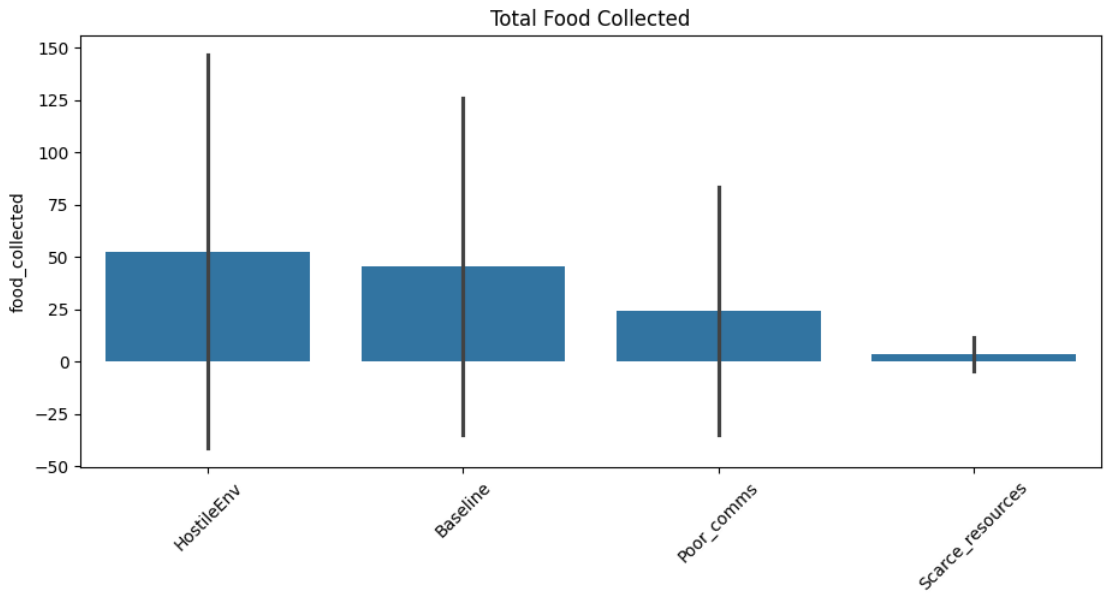

Figure 3.8: Food Collected Across Different Environmental Configurations

These results (see Figure 3.8) affirm that resource abundance and communication fidelity are critical drivers of foraging success, while environmental hostility, although challenging, can be mitigated through adaptive swarm behavior. The findings underscore the robustness of ant-inspired systems but also reveal their limits when core mechanisms like pheromone signaling or resource presence are impaired.

3.5.2 Ants Survival rates

These **ants_survived** results (Table 3.8) reflect how different extreme conditions affect colony survival

Table 3.8: Ants Survived Across Different Environmental Configurations

Scenario	ants_survived
	Value
Baseline	171 ± 35.7
Hostile Environment	167 ± 34.4
Poor_communication	156 ± 35.9
Scarce_resources	169 ± 34.1

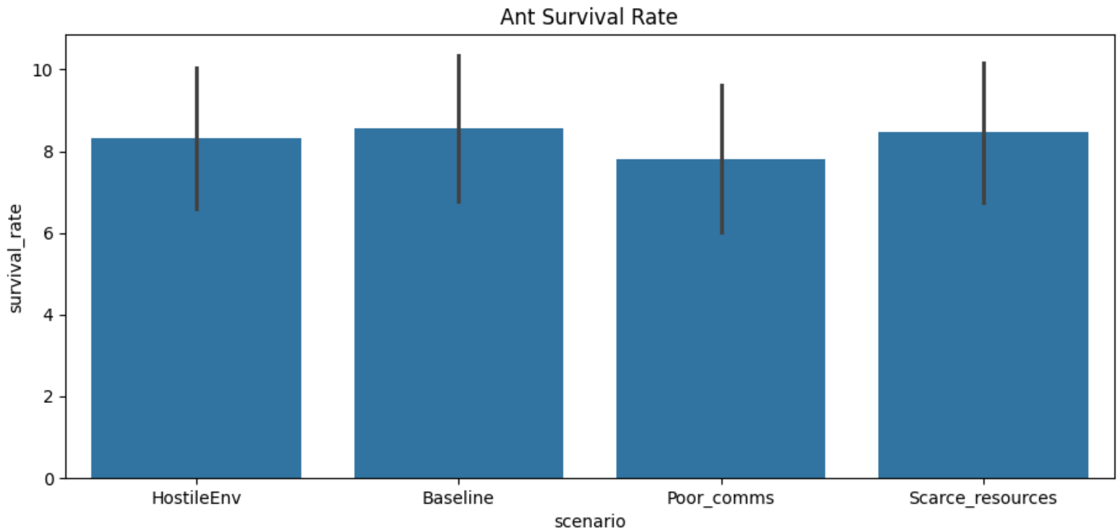

Figure 3.9: Ants Survived Across Different Environmental Configurations

- The **Baseline** scenario shows the highest average survival (171 ants), serving as a reference point for normal system behavior.
- **Hostile Environment** (i.e., with more obstacles or predators) results in only a slight decrease in survival (167), indicating that the colony exhibits **robust threat-avoidance mechanisms** and can maintain most of its members even in unsafe surroundings.
- **Scarce_resources** has a **surprisingly high survival rate (169)**, despite very poor food collection (from earlier data). This suggests that ants can survive long periods without successfully retrieving food, likely due to the simulation allowing survival without immediate energy needs being met or due to ants reaching the nest to recharge.
- **Poor_communication** leads to the lowest survival **(156 ants)**, highlighting that **communication breakdown has a tangible impact** not just on foraging efficiency, but also on survivability, likely due to inefficient exploration, increased

exposure to threats, or energy depletion.

Overall, **colony survival remains relatively high across all scenarios** (see Figure 3.9), indicating resilience in the face of various stressors. However, communication plays a more critical role in both performance and survival compared to resource scarcity or environmental hostility.

3.6 Conclusion

This study presented a quantitative evaluation of multi-agent systems through the lens of ant foraging behavior, implemented via an agent-based simulation framework using SimPy. By modeling realistic foraging dynamics—including pheromone-based communication, predator threats, energy constraints, and environmental variability—we were able to analyze how collective behavior emerges under diverse operational conditions. The simulation incorporated core biological principles such as pheromone deposition and evaporation, energy replenishment cycles, and dynamic interactions among agents, food sources, and threats.

Our experimental scenarios revealed key insights into the efficiency and adaptability of ant-inspired systems. Scenario 1 demonstrated that pheromone dynamics significantly influence trail stability and foraging efficiency, with balanced evaporation and deposition rates yielding optimal results. Scenario 2 highlighted the impact of predator presence, showing that ants can maintain performance under threat, but with increased variability in survival and food retrieval. In Scenario 3, distributed and obstructed resource layouts improved colony performance by promoting decentralized exploration and reducing bottlenecks. Scenario 4 emphasized that population scaling boosts food collection, though it introduces coordination challenges at higher densities. Finally, in Scenario 5 stress-testing under extreme conditions confirmed the colony's resilience, but also underscored performance degradation when multiple stressors coincide.

Overall, the simulation confirms that simple local rules and effective communication can generate robust and scalable foraging strategies. These findings have broader implications for the design of decentralized algorithms in robotics, logistics, and swarm intelligence. Future work may incorporate learning-based adaptations, more diverse agent roles, or real-world environmental data to further bridge biological inspiration and engineered systems.

Conclusions

This research presented a simulation-based analysis of multi-agent systems inspired by ant foraging behavior, using a custom agent-based model built with SimPy. By systematically varying environmental conditions, pheromone dynamics, predator threats, and colony characteristics, we quantitatively evaluated the efficiency and adaptability of decentralized foraging strategies. The model captured key interaction mechanisms such as pheromone communication, energy consumption, and predator avoidance while enabling the measurement of foraging performance through metrics like food collection, trail efficiency, and survival rates.

Several findings aligned with biological expectations. As an illustration of this, we found out that **strong pheromone trails significantly improved food collection**, confirming the critical role of stigmergic coordination in ant foraging. Similarly, **predator presence reduced survival and altered path usage**, validating the impact of external threats on colony behavior. Some results, however, were surprising. In scenarios with **scarce resources**, ants maintained relatively high survival despite collecting minimal food, suggesting that **colony endurance may not always correlate with resource intake**, a nuance that may be attributed to model assumptions such as energy replenishment rules. Furthermore, **communication breakdowns (Poor_comms)** had a more detrimental effect on both performance and survival than expected, highlighting the importance of efficient information flow in decentralized systems.

Despite its insights, the simulation has notable limitations. The **behavioral models are simplified abstractions**, lacking the full biological complexity of real ant colonies, such as learning, role specialization, or real-time adaptive behavior. Spatial and temporal resolutions are also constrained by computational resources and granularity of the model. Additionally, while stochasticity is introduced to mimic natural variability, the **lack of sensory noise or learning dynamics** may underestimate the complexity of natural exploration strategies.

Future work could focus on integrating adaptive behaviors, heterogeneous agent roles, or multiple nest competition to better mirror ecological dynamics. Nevertheless, this study demonstrates how agent-based simulations can provide valuable insights into collective behavior and the emergent properties of simple rule-based interactions.

Bibliography

[1] M. Abdulkader, Y. Gajpal, and T. ElMekkawy. "Hybridized ant colony algorithm for the multi compartment vehicle routing problem". In: *Applied Soft Computing* 37 (2015), pp. 196–203. DOI: 10.1016/j.asoc.2015.08.020.

[2] F. Adler and D. Gordon. "Optimization, conflict, and nonoverlapping foraging ranges in ants". In: *The American Naturalist* 162.5 (2003), pp. 529–543. DOI: 10.1086/378856.

[3] V. Agrawal, H. Sharma, and J. Bansal. "Bacterial foraging optimization: a survey". In: *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011)*. 2012, pp. 227–242. DOI: 10.1007/978-81-322-0487-9_23.

[4] A. Aljaafreh, M. Saleh, N. Al-Oudat, and M. Alaqtash. "An agent-based simulation model of dynamic real-time traffic signal controller". In: *International Journal of Advanced Computer Research* 10.46 (2020), pp. 1–11. DOI: 10.19101/ijacr.2019.940085.

[5] P. Amorim. "Modeling ant foraging: a chemotaxis approach with pheromones and trail formation". In: *Journal of Theoretical Biology* 385 (2015), pp. 160–173. DOI: 10.1016/j.jtbi.2015.08.026.

[6] AntWiki. *Foraging Theory*. Accessed: 2024-06-16. 2020. URL: https://www.antwiki.org/wiki/Foraging_Theory.

[7] *AnyLogic*. https://en.wikipedia.org/wiki/AnyLogic. Accessed: 2024-08-24.

[8] *AnyLogic*. https://www.anylogic.com/. Accessed: 2024-08-24.

[9] L. Baltiansky, G. Frankel, and O. Feinerman. "Emergent regulation of ant foraging frequency through a computationally inexpensive forager movement rule". In: *eLife* 12 (2023). DOI: 10.7554/elife.77659.

[10] L. Fu-cheng. "Research of Multi-Agent System Technology". In: *Computer Technology and Development* (2008). URL: https://api.semanticscholar.org/CorpusID:64299042.

[11] T. Czaczkes, B. Czaczkes, C. Iglhaut, and J. Heinze. "Composite collective decision-making". In: *Proceedings of the Royal Society B: Biological Sciences* 282.1809 (2015), p. 20142723. DOI: 10.1098/rspb.2014.2723.

[12] P. Dalapati, A. Padhy, B. Mishra, A. Dutta, and S. Bhattacharya. “Real-time collision handling in railway transport network: an agent-based modeling and simulation approach”. In: *Transportation Letters* 11.8 (2017), pp. 458–468. DOI: 10.1080/19427867.2017.1395983.

[13] M. Darvishi and G. Ahmadi. “Validation techniques of agent based modelling for geospatial simulations”. In: *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*. Vol. XL-2/W3. 2014, pp. 91–95. DOI: 10.5194/isprsarchives-xl-2-w3-91-2014.

[14] J. Davidson, R. Arauco-Aliaga, S. Crow, D. Gordon, and M. Goldman. “Effect of interactions between harvester ants on forager decisions”. In: *Frontiers in Ecology and Evolution* 4 (2016). DOI: 10.3389/fevo.2016.00115.

[15] C. Detrain and J.-L. Deneubourg. “Collective Decision-Making and Foraging Patterns in Ants and Honeybees”. In: *Advances in Insect Physiology*. Ed. by S. J. Simpson. Vol. 35. Academic Press, 2008, pp. 123–173. ISBN: 9780123743299.

[16] P. Di Bitonto, M. Laterza, T. Roselli, and V. Rossano. “Evaluation of Multi-Agent Systems: Proposal and Validation of a Metric Plan”. In: *Transactions on Computational Collective Intelligence VII*. Ed. by N. T. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 198–221. ISBN: 978-3-642-32066-8. DOI: 10.1007/978-3-642-32066-8_9. URL: https://doi.org/10.1007/978-3-642-32066-8_9.

[17] C. Dimou, F. Tzima, A. L. Symeonidis, and P. A. Mitkas. “Performance Evaluation of Agents and Multi-agent Systems Using Formal Specifications in Z Notation”. In: *Agents and Data Mining Interaction*. Ed. by L. Cao, Y. Zeng, B. An, A. L. Symeonidis, V. Gorodetsky, F. Coenen, and P. S. Yu. Cham: Springer International Publishing, 2015, pp. 64–78. ISBN: 978-3-319-20230-3.

[18] A. Dussutour, S. Beshers, J. Deneubourg, and V. Fourcassié. “Crowding increases foraging efficiency in the leaf-cutting ant Atta colombica”. In: *Insectes Sociaux* 54.2 (2007), pp. 158–165. DOI: 10.1007/s00040-007-0926-9.

[19] A. Dussutour, V. Fourcassié, D. Helbing, and J. Deneubourg. “Optimal traffic organization in ants under crowded conditions”. In: *Nature* 428.6978 (2004), pp. 70–73. DOI: 10.1038/nature02345.

[20] L. Elizalde and A. Farji-Brener. “To be or not to be faithful: flexible fidelity to foraging trails in the leaf-cutting ant Acromyrmex lobicornis”. In: *Ecological Entomology* 37.5 (2012), pp. 370–376. DOI: 10.1111/j.1365-2311.2012.01368.x.

[21] M. Faisal, H. Mathkour, and M. Alsulaiman. "Antstar: enhancing optimization problems by integrating an ant system and A* algorithm". In: *Scientific Programming* 2016 (2016), pp. 1–12. DOI: 10.1155/2016/5136327.

[22] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari. "Review and classification of bio-inspired algorithms and their applications". In: *Journal of Bionic Engineering* 17.3 (2020), pp. 611–631. DOI: 10.1007/s42235-020-0049-9.

[23] X. Feng. "A Bio-Inspired Multi-Agent System Framework for Real-Time Load Management in All-Electric Ship Power Systems". Available electronically from https://hdl.handle.net/1969.1/ETD-TAMU-2012-05-11031. PhD thesis. Texas A&M University, 2012.

[24] A. Forster, T. Czaczkes, E. Warner, T. Woodall, E. Martin, and F. Ratnieks. "Effect of trail bifurcation asymmetry and pheromone presence or absence on trail choice by Lasius niger ants". In: *Ethology* 120.8 (2014), pp. 768–775. DOI: 10.1111/eth.12248.

[25] D. A. Friedman, A. Tschantz, M. J. Ramstead, K. Friston, and A. Constant. "Active Inferants: an active inference framework for ant colony behavior". In: *Frontiers in behavioral neuroscience* 15 (2021), p. 647732.

[26] G. Gathalkar and A. Sen. "Foraging and Predatory Activities of Ants". In: *The Complex World of Ants*. Ed. by V. Shields. Rijeka: IntechOpen, 2018. Chap. 3. DOI: 10.5772/intechopen.78011. URL: https://doi.org/10.5772/intechopen.78011.

[27] F. Ge, Z. Wei, Y. Lu, Y. Tian, and L. Li. "Decentralized coordination of autonomous swarms inspired by chaotic behavior of ants". In: *Nonlinear Dynamics* 70 (2012), pp. 571–584.

[28] S. Ghazi, T. Khadir, and J. Dugdale. "Multi-agent based simulation of environmental pollution issues: a review". In: *Multi-Agent Systems for Environmental Environments*. 2014, pp. 13–21. DOI: 10.1007/978-3-319-07767-3_2.

[29] S. T. Goonatilleke and B. Hettige. "Past, Present and Future Trends in Multi-Agent System Technology". In: *Journal Europeen des Systemes Automatises* 55 (6 Dec. 2022), pp. 723–739. ISSN: 21167087. DOI: 10.18280/jesa.550604.

[30] D. Gordon, K. Dektar, and N. Pinter-Wollman. "Harvester ant colony variation in foraging activity and response to humidity". In: *PLOS ONE* 8.5 (2013), e63363. DOI: 10.1371/journal.pone.0063363.

[31] D. Gordon, A. Guetz, M. Greene, and S. Holmes. "Colony variation in the collective regulation of foraging by harvester ants". In: *Behavioral Ecology* 22.2 (2011), pp. 429–435. DOI: 10.1093/beheco/arq218.

[32] N. Goy, S. Glaser, and C. Grüter. “The adaptive value of tandem communication in ants: insights from an agent-based model”. In: *Journal of Theoretical Biology* 526 (2021), p. 110762. DOI: 10.1016/j.jtbi.2021.110762.

[33] M. Greene and D. Gordon. “How patrollers set foraging direction in harvester ants”. In: *The American Naturalist* 170.6 (2007), pp. 943–948. DOI: 10.1086/522843.

[34] C. A. Groenewald, G. Saha, G. Mann, B. Bhushan, E. Howard, and E. Groenewald. “Multi-Agent Systems in Robotics: Coordination and Communication using Machine Learning”. In: 28 (Mar. 2024), pp. 882–897.

[35] C. Grüter, R. Schürch, T. Czaczkes, K. Taylor, T. Durance, S. Jones, and F. Ratnieks. “Negative feedback enables fast and flexible collective decision-making in ants”. In: *PLOS ONE* 7.9 (2012), e44501. DOI: 10.1371/journal.pone.0044501.

[36] S. Hartman, S. D. Ryan, and B. R. Karamched. “Walk This Way: Modeling Foraging Ant Dynamics in Multiple Food Source Environments”. In: *bioRxiv* (2024). DOI: 10.1101/2024.01.20.576461. eprint: https://www.biorxiv.org/content/early/2024/01/23/2024.01.20.576461.full.pdf. URL: https://www.biorxiv.org/content/early/2024/01/23/2024.01.20.576461.

[37] P. Heywood, P. Richmond, and S. Maddock. “Road network simulation using FLAME GPU”. In: *Euro-Par 2015: Parallel Processing Workshops*. 2015, pp. 430–441. DOI: 10.1007/978-3-319-27308-2_35.

[38] X. Hu, M. Zhang, H. Chung, Y. Li, and O. Liu. “SamACO: Variable sampling ant colony optimization algorithm for continuous optimization”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40.6 (2010), pp. 1555–1566. DOI: 10.1109/tsmcb.2010.2043094.

[39] K. Imai and A. Okuyama. “Research on a Multi-agent System That Mimics Ant Foraging Behavior”. In: *Proceedings of Eighth International Congress on Information and Communication Technology*. Ed. by X.-S. Yang, R. S. Sherratt, N. Dey, and A. Joshi. Singapore: Springer Nature Singapore, 2024, pp. 193–203. ISBN: 978-981-99-3236-8.

[40] T. Islam, M. E. Islam, and M. R. Ruhin. “An analysis of foraging and echolocation behavior of swarm intelligence algorithms in optimization: ACO, BCO and BA”. In: *International Journal of Intelligence Science* 8.01 (2018), p. 1.

[41] A. Kasprzok, B. Ayalew, and C. Lau. “An ant-inspired model for multi-agent interaction networks without stigmergy”. In: *Swarm Intelligence* 12 (1 Mar. 2018), pp. 53–69. ISSN: 19353820. DOI: 10.1007/s11721-017-0147-4.

[42] D. W. King. *Emergent Behavior Development and Control in Multi-Agent Systems*. 2019.

[43] A. Krishnaveni, R. Shankar, and S. Duraisamy. "A survey on nature inspired computing (NIC): algorithms and challenges". In: *Global journal of computer science and technology: D Neural and Artificial Intelligence* 19.03 (2019), p. 1.

[44] L. Laibinis, E. Troubitsyna, Z. Graja, F. Migeon, and A. Hadj Kacem. "Formal Modelling and Verification of Cooperative Ant Behaviour in Event-B". In: *Software Engineering and Formal Methods. SEFM 2014*. Ed. by D. Giannakopoulou and G. Salaün. Vol. 8702. Lecture Notes in Computer Science. Springer, Cham, 2014, pp. 2–5.

[45] L. Laibinis, E. Troubitsyna, Z. Graja, F. Migeon, and A. H. Kacem. *LNCS 8702 - Formal Modelling and Verification of Cooperative Ant Behaviour in Event-B*.

[46] C. Lasmar, T. Bishop, C. Parr, A. Queiroz, F. Schmidt, and C. Ribas. "Geographical variation in ant foraging activity and resource use is driven by climate and net primary productivity". In: *Journal of Biogeography* 48.6 (2021), pp. 1448–1459. DOI: 10.1111/jbi.14089.

[47] R. N. Lass, E. A. Sultanik, and W. C. Regli. "Metrics for Multiagent Systems". In: *Performance Evaluation and Benchmarking of Intelligent Systems*. Ed. by R. Madhavan, E. Tunstel, and E. Messina. Boston, MA: Springer US, 2009, pp. 1–19. ISBN: 978-1-4419-0492-8. DOI: 10.1007/978-1-4419-0492-8_1. URL: https://doi.org/10.1007/978-1-4419-0492-8_1.

[48] G. Lingaraju, P. Nithiarasu, M. Kusuma, and R. Gowda. "Real-time interactive steerable scientific visualisation of free surface flow in the context of synthetic vision". In: *Defence Science Journal* 61.4 (2011), pp. 299–305. DOI: 10.14429/dsj.61.1115.

[49] F. Lobato and J. Steffen. "Multi-objective optimization firefly algorithm applied to (bio)chemical engineering system design". In: *American Journal of Applied Mathematics and Statistics* 1.6 (2013), pp. 110–116. DOI: 10.12691/ajams-1-6-1.

[50] C. Macal and M. North. "Tutorial on agent-based modelling and simulation". In: *Journal of Simulation* 4.3 (2010), pp. 151–162. DOI: 10.1057/jos.2010.3.

[51] U. Madhushani. "Learning Through Social Interactions and Learning to Socially Interact in Multi-Agent Learning". PhD thesis. Princeton University, 2023.

[52] F. Matteis, A. Tafuro, D. Preite, and G. Dammacco. "The role of the environmental sphere in factors affecting performance and in performance dimensions in the utility sector: a literature review". In: *International Journal of Business Administration* 13.1 (2022), p. 30. DOI: 10.5430/ijba.v13n1p30.

[53] M. Mavrovouniotis, S. Yang, M. Van, C. Li, and M. Polycarpou. "Ant colony optimization algorithms for dynamic optimization: a case study of the dynamic travelling salesperson problem [research frontier]". In: *IEEE Computational Intelligence Magazine* 15.1 (2020), pp. 52–63. DOI: 10.1109/mci.2019.2954644.

[54] MDPI. *Bio-Inspired Collective Intelligence in Multi-Agent Systems*. https://www.mdpi.com/journal/applsci/special_issues/4O6JJ39770. Accessed: 2024-06-11. 2023.

[55] K. Mentesidi, E. Rikos, and V. Kleftakis. "Real-time simulation technique of a microgrid model for DER penetration". In: *EAI Endorsed Transactions on Energy Web* 1.2 (2014), e2. DOI: 10.4108/ew.1.2.e2.

[56] T. Merkle and R. Wehner. "Desert ants use foraging distance to adapt the nest search to the uncertainty of the path integrator". In: *Behavioral Ecology* 21.2 (2010), pp. 349–355. DOI: 10.1093/beheco/arp197.

[57] G. Mitra and S. Bandyopadhyay. "A Study on Some Aspects of Biologically Inspired Multi-agent Systems". In: *Progress in Advanced Computing and Intelligent Engineering*. Ed. by K. Saeed, N. Chaki, B. Pati, S. Bakshi, and D. P. Mohapatra. Singapore: Springer Singapore, 2018, pp. 207–217. ISBN: 978-981-10-6875-1.

[58] H. M. Mustafa and F. B. Tourkia. "On Comparative Analysis of Two Ant Colony's Optimal Selectivity Decisions for Best Food Source Versus Migration to Well Protected Nest Site (Neural Networks' Approach)". In: *Unknown Journal/Conference* (2010).

[59] J. Ni, X. Wang, M. Tang, W. Cao, P. Shi, and S. Yang. "An improved real-time path planning method based on dragonfly algorithm for heterogeneous multi-robot system". In: *IEEE Access* 8 (2020), pp. 140558–140568. DOI: 10.1109/access.2020.3012886.

[60] J. Ni, L. Wu, P. Shi, and S. Yang. "A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles". In: *Computational Intelligence and Neuroscience* 2017 (2017), pp. 1–16. DOI: 10.1155/2017/9269742.

[61] T. Novgorodova. "Organization of honeydew collection by foragers of different species of ants (Hymenoptera: Formicidae): effect of colony size and species specificity". In: *European Journal of Entomology* 112.4 (2015), pp. 688–697. DOI: 10.14411/eje.2015.077.

[62] H. Nurdiansyah, H. Almubaroq, A. Risdhianto, and M. Mualim. "Evaluation of the spread of radicalism, extremism, and terrorism in Indonesia's defense using

agent-based simulations". In: *International Journal of Humanity Studies (IJHS)* 6.2 (2023), pp. 228–239. DOI: 10.24071/ijhs.v6i2.5382.

[63] L. Oudenhove, X. Cerdá, and C. Bernstein. "An evolutionary dynamics model adapted to eusocial insects". In: *PLOS ONE* 8.3 (2013), e55159. DOI: 10.1371/journal.pone.0055159.

[64] R. Pagliara, D. Gordon, and N. Leonard. "Regulation of harvester ant foraging as a closed-loop excitable system". In: *PLOS Computational Biology* 14.12 (2018), e1006200. DOI: 10.1371/journal.pcbi.1006200.

[65] J. Peng and Z. Bao. "An improved ant colony foraging behavior modeling in the application of robot path planning problem". In: 2016. DOI: 10.2991/icmemtc-16.2016.144.

[66] I. Pereverzeva, L. Laibinis, E. Troubitsyna, M. Holmberg, and M. Pöri. "Formal modelling of resilient data storage in cloud". In: *Formal Methods and Software Engineering: 15th International Conference on Formal Engineering Methods, ICFEM 2013, Queenstown, New Zealand, October 29–November 1, 2013, Proceedings 15*. Springer Berlin Heidelberg, 2013, pp. 363–379.

[67] N. Pinter-Wollman, A. Bala, A. Merrell, J. Queirolo, M. Stumpe, S. Holmes, and D. Gordon. "Harvester ants use interactions to regulate forager activation and availability". In: *Animal Behaviour* 86.1 (2013), pp. 197–207. DOI: 10.1016/j.anbehav.2013.05.012.

[68] E. Pless, J. Queirolo, N. Pinter-Wollman, S. Crow, K. Allen, M. Mathur, and D. Gordon. "Interactions increase forager availability and activity in harvester ants". In: *PLOS ONE* 10.11 (2015), e0141971. DOI: 10.1371/journal.pone.0141971.

[69] B. Prabhakar, K. Dektar, and D. Gordon. "The regulation of ant colony foraging activity without spatial information". In: *PLOS Computational Biology* 8.8 (2012), e1002670. DOI: 10.1371/journal.pcbi.1002670.

[70] B. Ranjbar-Sahraei, K. Tuyls, I. Caliskanelli, B. Broecker, D. Claes, S. Alers, and G. Weiss. "Bio-inspired multi-robot systems". In: *Biomimetic Technologies: Principles and Applications* (Dec. 2015), pp. 273–299. DOI: 10.1016/B978-0-08-100249-0.00013-6.

[71] J. Retana, X. Arnán, and X. Cerdá. "A multidimensional functional trait analysis of resource exploitation in European ants". In: *Ecology* 96.10 (2015), pp. 2781–2793. DOI: 10.1890/14-2326.1.

[72] E. J. H. Robinson, K. E. Green, E. A. Jenner, et al. "Decay rates of attractive and repellent pheromones in an ant foraging trail network". In: *Insect. Soc.* 55 (2008), pp. 246–251. DOI: 10.1007/s00040-008-0994-5.

[73] S. Ryan. “A model for collective dynamics in ant raids”. In: *Journal of Mathematical Biology* 72.6 (2015), pp. 1579–1606. DOI: 10.1007/s00285-015-0929-5.

[74] M. Saoud, A. Boubetra, and A. S. “How data mining techniques can improve simulation studies”. In: *International Journal of Computer Theory and Engineering* (2014), pp. 15–19. DOI: 10.7763/ijcte.2014.v6.828.

[75] O. Saraereh, A. Saraira, Q. Alsafasfeh, and A. Arfoa. “Bio-inspired algorithms applied on microstrip patch antennas: a review”. In: *International Journal on Communications Antenna and Propagation (IRECAP)* 6.6 (2016), p. 336. DOI: 10.15866/irecap.v6i6.9737.

[76] I. Satoh. “Bio-inspired Organization for Multi-agents on Distributed Systems”. In: *Biologically Inspired Approaches to Advanced Information Technology*. Ed. by A. J. Ijspeert, T. Masuzawa, and S. Kusumoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 355–362. ISBN: 978-3-540-32438-6.

[77] M. Schumacher. “Multi-agent systems”. In: *Objective Coordination in Multi-Agent System Engineering: Design and Implementation* (2001), pp. 9–32.

[78] U. Segev, K. Tielbörger, Y. Lubin, and J. Kigel. “Consequences of climate and body size on the foraging performance of seed-eating ants”. In: *Ecological Entomology* 39.4 (2014), pp. 427–435. DOI: 10.1111/een.12115.

[79] *SimPy Documentation*. https://simpy.readthedocs.io/en/latest/. Accessed: 2024-08-24.

[80] J. Tang, G. Liu, and Q. Pan. “A review on representative swarm intelligence algorithms for solving optimization problems: applications and trends”. In: *IEEE/CAA Journal of Automatica Sinica* 8.10 (2021), pp. 1627–1643. DOI: 10.1109/jas.2021.1004129.

[81] *The AnyLogic simulation framework*. Unknown.

[82] A. M. Uhrmacher and D. Weyns. *Multi-Agent systems: Simulation and applications*. Ed. by A. M. Uhrmacher and D. Weyns. CRC press, 2009.

[83] A. U. Uppsala Universitet. *UPPAAL*. https://uppaal.org/. Accessed: 2024-08-24.

[84] A. U. Uppsala Universitet. *UPPAAL Documentation*. https://docs.uppaal.org/. Accessed: 2024-08-24.

[85] M. E. L. Vieira, S. Chameron, and N. Châline. “The ethology of foraging in ants: revisiting Tinbergen’s four questions”. In: *Frontiers in Ethology* 3 (Mar. 2024). DOI: 10.3389/fetho.2024.1341120.

[86] J. Ward, A. Evans, and N. Malleson. "Dynamic calibration of agent-based models using data assimilation". In: *Royal Society Open Science* 3.4 (2016), p. 150703. DOI: 10.1098/rsos.150703.

[87] M. Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[88] L. Ya, H. Wang, and L. Wu. "Multi-agent based design for distributed fault diagnosis systems". In: *Advanced Materials Research* 179-180 (2011), pp. 1266–1271. DOI: 10.4028/www.scientific.net/amr.179-180.1266.

[89] X. Yan, Y. Zhu, H. Chen, and H. Zhang. "A novel hybrid artificial bee colony algorithm with crossover operator for numerical optimization". In: *Natural Computing* 14.1 (2013), pp. 169–184. DOI: 10.1007/s11047-013-9405-6.

[90] D. Zeghida, D. Meslati, and N. Bounour. "Bio-IR-M: a multi-paradigm modelling for bio-inspired multi-agent systems". In: *Informatica* 42.3 (2018). DOI: 10.31449/inf.v42i3.1516.

[91] J. Zhané. *SimPy: Simulating Real-World Processes With Python*. https://realpython.com/simpy-simulating-with-python/. Accessed: 2024-08-24.

[92] N. Zhang. "Dynamics, statistics, and task allocation of foraging ants". In: *Physical Review E* 108.5 (2023). DOI: 10.1103/physreve.108.054306.

[93] Y. Zhang, P. Agarwal, V. Bhatnagar, S. Balochian, and J. Yan. "Swarm intelligence and its applications". In: *The Scientific World Journal* 2013.1 (2013). DOI: 10.1155/2013/528069.

[94] D. Zhu, X. Cao, B. Sun, and C. Luo. "Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system". In: *IEEE Transactions on Cognitive and Developmental Systems* 10.2 (2018), pp. 304–313. DOI: 10.1109/tcds.2017.2727678.

List of Abbreviations

MAS	**M**ulti **A**gent **S**ystem
ACO	**A**nt **C**olony **O**ptimization
ANN	**A**rtificial **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
GAN	**G**enerative **A**dversarial **N**etwork
MSE	**M**ean **S**quare **E**rror
DAI	**D**istributed **A**rtificial **I**ntelligence
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
SSIM	**S**tructural **SIM**ilarity
FAO	**F**oraging **A**nt **O**ptimization
DCO	**D**istributed **C**onstraint **O**ptimization
GA	**G**enetic **A**lgorithm
FA	**F**irefly **A**lgorithm
PSO	**P**article **S**warm **O**ptimization
DCOP	**D**istributed **C**onstraint **OP**timization
ES	**E**volutionary **S**trategies
ABM	**A**gent **B**ased **M**odeling
DES	**D**iscrete **E**vent **S**imulation
FLAME GPU	**F**lexible **L**arge-scale **A**gent **M**odelling **E**nvironment for GPUs
GPU	**G**raphics **P**rocessing **U**nit
ABMS	**A**gent **B**ased **M**odeling and **S**imulation
RTHS	**R**eal **T**ime **H**ybrid **S**imulation
UPPAAL	**UPP**SALA UNIVERSITET **AAL**BORG UNIVERSITY