NTNU
Norwegian University of
Science and Technology

There and back again; Efficient foraging with heterogeneous Agents using Pheromones

Kyrre Laugerud Moe

Master of Science in Informatics
Submission date: December 2016
Supervisor: Anders Kofod-Petersen, IDI

Norwegian University of Science and Technology
Department of Computer Science

Kyrre Laugerud Moe

There and back again;
Efficient foraging with heterogeneous Agents using Pheromones

master project, autumn 2016

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

Abstract

Foraging is a canonical task for swarm intelligence systems. Traditionally this task has been preformed by a swarm of homogeneous agents, often with the help of pheromones for indirect communication. Some researcher have experimented with how heterogeneous agents can achieve this task. Yet little research exist, combining morphologically heterogeneous agents with pheromone based foraging algorithms. This is confirmed thought a structured literature review. C-SAF is an existing foraging algorithm employing homogeneous agents, communication via pheromones. In this thesis, morphologically heterogeneous agents are introduced and combined with the core mechanics of C-SAF to creating a new algorithm, dubbed H-CAF. H-CAF is compared with C-SAF across different scenarios, using well established performance metrics from the literature.

Sammendrag

Sanking av ressurser i ukjente miljøer er en populære oppgave i systemer basert på sverm-intelligens. Tradisjonelt har dette blitt utført av en sverm med homogene agenter. Disse benytter ofte indirekte kommunikasjon ved hjelp av feromoner. Noe forskning har eksperimentert med hvordan heterogene agenter kan utføre den samme oppgaven. Likevel eksisterer det lite forskning som kombinerer morfologisk heterogene agenter, med feromonbaserte algoritmer, for å sanke. Dette er bekreftet gjennom en strukturert analyse av relatert forskning. C-SAF er en eksisterende algoritme, som løser sanke oppgaven. Den gjør dette ved hjelp av en en sverm med homogene agenter, som kommuniserer ved hjelp av feromoner. I denne oppgaven presenteres en ny algoritme, kalt H-CAF. Den introduserer morfologisk heterogene agenter, og kombinerer disse med kjernefunksjonaliteten til C-SAF. H-CAF og C-SAF blir sammenlignet over forskjellige scenarier, basert på velkjente effektivitetsmål.

Preface

This thesis and its associated source code is written by Kyrre Laugerud Moe at Norwegian University of Science and Technology (NTNU). This research is done in association with the Artificial Intelligence Group at the Department of Computer and Information Science. The thesis was supervised by Anders Kofod-Petersen.

Acknowledgement

I would like to thank Anders Kofod-Petersen for guiding this thesis, through valuable feedback an advice. Thanks to friends and fellow students for helpful discussion. I am grateful to Grethe Laugerud for the extra care she gave me the last days before the deadline, as well as endless love and support. Special thanks to Martin Moe for taking the time to proofread this paper, in addition to fruitful discussion and constructive criticism.

Kyrre Laugerud Moe
Trondheim, December 31, 2016

Contents

Abbreviations

AI	Artificial intelligence
SI	Swarm intelligence
PSO	Particle swarm optimization
ABC	Artificial bee colony algorithm
ACO	Ant colony optimization
RQ	Research question
SLR	Structured literature review
APF	Artificial potential field
BFS	Breadth first search
IC	Inclusion criteria
QC	Quality criteria
SDK	Software development kit
JDK	Java development kit
JRE	Java runtime environment
APF	Artificial potential field
RFID	Radio frequency identification
GA	Genetic algorithm

List of Figures

List of Tables

Chapter 1

Introduction

Over the last decades distributed control structures have been introduced in different sectors of society. Transportation, short term vacation rental, and currency have all experienced the introduction of decentralized control. Uber, Airbnb and Bitcoin are all examples of peer-to-peer services. They are distributed with no central distributor.

It is true that the Airbnb and Uber communities could not function without their respective apps and web-pages. But they serve less as a central controllers, and more as a facilitator of their domain. They merely act as an environment in which people can interact directly, user to user.

Cryptocurrencies (e.g. Bitcoin) take this one step further. Here there are no central facilitator. The users them self are the facilitator. Each peer-to-peer transaction is validated by the rest of the users, through the use of blockchains [Ron and Shamir, 2013]. Creating a truly distributed system. Eliminating any need for a central control structure, an thus a single point of failure.

The same can be said for the Internet it self. It too is distributed. Their is no single point of control, delegating traffic throughout the system. The Internet is decentralized. Millions of nodes communicate independently to create a web of information.

Decentralized control structures have long existed in nature. Perhaps most famous of which is the societies of ants. Simple creatures unknowing of their greater goal, interact and behave based on novel input from local information. Yet they demonstrate an extraordinary ability to cooperate on complex tasks.

Ants achieve this coordination by depositing chemicals known as pheromones through the environment [Jackson and Ratnieks, 2006]. These pheromones are detectable to other agents, and act as a messaging system between ants. Thus an ant sensing pheromones can change its behavior without coming in direct contact with the depositing ant.

This emergent behavior has long been of interest in the field of Artificial Intelligence (AI). So much so that it has spawned its own field of research, Swarm Intelligence (SI).

Swarm intelligence has many potential applications. Applications include numerical optimization, micro robotics, medicine, construction, surveillance and exploration. Some are strictly algorithmic, others applied to the field of swarm robotics.

These systems employee relatively simple agents in larger numbers. Agents can be described as single entities capable of some rudimentary interaction. In other fields of AI, the goal can often be to make a single agent capable of complex reasoning and decision making. In SI the focused is switched to simpler agents, operating in larger numbers. These agents cooperate through local interaction. Through these interaction, intelligent behavior emerges [Kennedy et al., 2001], reminiscent of a colony ants.

Foraging

Foraging is often defined as the search for wild food resources. It is a technique seen in many species in nature. It often plays a vital roll in an individual, or a collection of individuals, ability to survive.

Bears, bees, squirrel and birds all forage food to survive. This process of foraging in the wild is comparable to many processes in modern society. For starters we too forage for wild food resources, like berries and mushroom. Fishing, mining, harvesting, search and rescue can all be considered foraging activities. Consequently, being able to deploy an autonomous swarm of robots to perform these tasks is a captivating one.

In this thesis we are interested in looking at at how swarm intelligence can preform a foraging task. In specific we want to investigate a system of morphologically heterogeneous agents, utilizing pheromones as means of communication.

The use of pheromones as a communication medium is a popular technique in foraging related algorithms. By depositing detectable pheromones in the environment, agents can signal intended actions to other agents. This effectively lets reactive agents use the environment as shared memory.

Another approach in SI is to employ a swarm of heterogeneous agents, in an effort to increase the swarm's abilities or productivity. These agents can ether be behaviorally or morphologically heterogeneous. Such swarms of heterogeneous agents have also been applied in foraging algorithms.

This thesis seeks to explore whether these two approaches can be combined in a way that can improve a swarm's ability to forage.

1.1 Background and Motivation

Realizing swarm foraging systems capable of real world application will open a host of possibilities. Toxic waste clean-up, de-mining, collection of terrain samples and collection of specimens in hazardous environmens are all examples of possible use cases [Campo and Dorigo, 2007; Winfield, 2009; Balch, 1999]. Search and rescue is an especially promising application, as it can reduces the need human involvement in hazardous environments (e.g. fire, unstable buildings) [Campo and Dorigo, 2007; Jennings et al., 1997; Kantor et al., 2003; Steele Jr and Thomas, 2007; Winfield, 2009]. Liemhetcharat et al. [2015] propose a system for autonomous fishing. Harvesting and planetary exploration is also of interest [Winfield, 2009]. Balch [1999]; Chattunyakit et al. [2013] suggest it be applied to mining operations.

One of the key advantages of employing swarm robotics in foraging scenarios, is its robustness. This robustness is obtained through redundancy. The systems are redundant in the sense that they can continue to operate even though individual robots break down [Dorigo and Roosevelt, 2004].

In a well functioning swarm these robots work in parallel, decreasing task completion time [Kennedy et al., 2001]. Having many simpler agents/robots, makes maintaining the system operational easier.

The use of pheromones in foraging algorithms is fairly common [Panait and Luke, 2004; Sugawara et al., 2004]. It is also common in foraging related algorithms. Exploration, area coverage, mapping, reconnaissance and surveillance systems all utilize pheromones [Masár, 2013; Calvo et al., 2011; De Rango et al., 2015; Fossum et al., 2014; Rodríguez et al., 2015]. Most of these task can be seen as sub tasks in a foraging scenario.

The large majority of work done on swarm intelligence in the scientific community, has been on homogeneous systems [Dorigo et al., 2013]. This is mainly due to early research drawing inspiration from self-organizing natural systems. Implementation of such systems often rely on high levels of abstraction, an often overlook the heterogeneity of these natural systems [Dorigo et al., 2013].

Some research has nevertheless been conducted on heterogeneous swarms solving foraging related tasks. The combination of ground based agents and air born agents is a popular approach [Ducatelle et al., 2010; Dorigo et al., 2013; Sauter et al., 2008; Liemhetcharat et al., 2015]. Momen and Sharkey [2009]; Rodríguez et al. [2015] explore the use of heterogeneous ground based foraging agents, through division of labour and information sharing. But on the whole, relatively little research seem to have been done on heterogeneous swarm systems in the context of a foraging task.

As a consequence, research on combining SI using pheromones to forage with heterogeneous agents, is also sparse. Yet heterogeneous agents can increase a swarms capabilities and provide flexibility [Ducatelle et al., 2011]. In this thesis we therefore explore the possibility of combining pheromone based foraging algorithms with the use of heterogeneous agents.

A promising pheromone based foraging algorithm is extended to work with heterogeneous agents [Zedadra et al., 2016]. A system of scouts and harvester agents is developed, and compared with the heterogeneous C-SAF foraging algorithm [Zedadra et al., 2016]. The systems are compared on performance metrics common in the related literature, to investigate if the inclusion of heterogeneous agents can have a positive effect on foraging capabilities.

1.2 Goals and Research Questions

The focus of this master thesis is to develop a swarm intelligence system capable of autonomous foraging. The swarm should be morphologically heterogeneous, and utilize a pheromone model as its primary agent to agent interaction.

Goal *Develop a morphologically heterogeneous swarm intelligence system, using a pheromone model for foraging.*

Based on this goal the following research questions are formulated:

Research question 1 (RQ1): *What is the state-of-the-art in morphologically heterogeneous swarm intelligence systems used in foraging?*

Answering this question will identify existing methods proposed in the literature. In addition it will serve as a theoretical foundation for this thesis.

Research question 2 (RQ2): *What is the state-of-the-art in swarm intelligence systems utilizing pheromones for foraging related tasks? (e.g. foraging, exploration, surveillance)*

This will also serve as a theoretical foundation as well as identify existing methods in the literature. Formulating RQ1 and RQ2 as separate research questions allow search in the existing literature to include papers focused on RQ1 or RQ2, as well as a combination of the two.

Research question 3 (RQ3): *Can swarm intelligence systems utilizing pheromones be combined with morphologically heterogeneous agents to create an autonomous foraging system, in a way that improves performance?*

For a heterogeneous swarm to be considered a viable strategy, it needs to outperform a homogeneous equivalent by some meaningful performance metric. This question aims to establish if a swarm system based on pheromones can improve its performance metrics by employing heterogeneous agents.

1.3 Research Method

In this thesis two research methods are used to answer the research questions described in 1.2. To answer RQ1 and RQ2 a analytical approach is taken. Through the use of a structured literature review (SLR) [Kofod-Petersen, 2012; Keele, 2007], described in section 2.5, existing solutions in the literature are explored. A wide document search is performed in order to capture literature possibly related to RQ1 and RQ2. By systematically pruning this document collection, state-of-the-art research relevant to both questions is reviled.

Like most topics in science there exist a wast amount of research related to SI based foraging algorithms. Thus it is difficult to get an overview of the existing literature. It is also a difficult process to figure out which of these documents constitutes state-of-the-art research on the topic.

SLR offers no guaranty of capturing all relevant research, nor that the resulting document collection is reduced to what is truly state-of-the-art. It does however offer a systematic and reproducible method of retrieving relevant literature [Kofod-Petersen, 2012]. This allows the research to be reproduced, increasing its scientific value.

RQ3 is answered through the design and implementation of the system proposed in chapter 3. A system is developed based on an existing swarm intelligence system, described in the literature [Zedadra et al., 2015a,b, 2016]. This existing solution employs a homogeneous swarm of agents, and a pheromone model, to solve a foraging task. In chapter 3, this solution is extended to a heterogeneous swarm solving the same foraging task. By directly comparing the two system through experimentation, RQ3 is answered.

1.3.1 Direct comparison of homogeneous and heterogeneous agents

As explained in section 2.6, it is difficult to compare existing solutions on foraging tasks. There exist a mirage of different approaches. Some focus on bridging the gap between swarm intelligence and swarm robotics [Ducatelle et al., 2010, 2011; Hrolenok et al., 2010; Hecker et al., 2012]. Others focus on improving foraging algorithms in simulation [Rodríguez et al., 2015; Momen and Sharkey, 2009; Liemhetcharat et al., 2015; Florea et al., 2015; Calvo et al., 2015; Letendre and Moses, 2013; Zedadra et al., 2015b,a, 2016].

But even here the approach varies. Often they solve different variations of a foraging task, or some other prerequisite differ (e.g. sensory capacity, communication, performance metrics). Therefore a need exists to directly compare homogeneous pheromone based swarm foraging algorithms, with heterogeneous versions.

Building on [Zedadra et al., 2015a,b, 2016] allows the two algorithms to be tested in the same simulation framework, with the same foraging scenarios and performance metrics. By comparing the two within this common frame of reference, performance can be compered. Thus advantages and disadvantages uncovered.

1.4 Thesis Structure

Chapter 2 explains the background theory needed to understand the content, and terminology, of this thesis. It starts by describes the core aspects of swarm intelligence systems, before describing a famous example of a emergent behavior with simple agents (section 2.2).

Optimization algorithms, inspired by foraging, are presented in section 2.3. This section also present popular mechanics throughout the literature. In section 2.4, the fundamentals of swarm robotics and foraging algorithms are explained. Here terminology, important for understanding the related work, and the proposed system, is established. In sections 2.5 and 2.6, the structured literature review, and related work is presented. The full SLR protocol is found in Appendix A, while the related work is discussed in section 2.6.3. Section 2.7, describes the motivation for developing a new control algorithm. this motivation is primarily grounded in the current state of the related work, and the subsequent discussion.

Chapter 3 presents the proposed system. This chapter starts by introducing some core mechanics important for understanding the system, as well as the problem definition (section 3.4). The proposed system is named H-CAF, and is based on both the S-MASA and C-SAF algorithms. As such, S-MASA is presented in section 3.5, and C-SAF in section 3.6. H-CAF is then presented in section 3.8.

Chapter 4 describes the experiments performed, design to compare C-SAF and H-CAF across different scenarios. The experimental plan, setup, and experiments them self are described in sections 4.1 - 4.3. The results are presented in section 4.4.

Chapter 5 presents an evaluation of both the state of related literature, and the results found in section 4.4. A discussion of the proposed system is presented in 5.1.In section 5.2, the thesis contributions are presented. Finlay, future work is described in section 5.3.

Chapter 2

Background Theory and Motivation

Swarm intelligence (SI) often take their inspiration from nature. As such swarm behavior can be found in numerous species throughout the world. Flocking birds, schools of fish, ant and bee colonies are all examples of natural swarm intelligence.

The primary mechanic of any swarm, natural or not, is the principle of local interaction. Observing a school of fish or a flock of birds one can easily be fooled into thinking it is controlled by some centralized mechanism. Perhaps a leading individual dictating the swarms behavior, or some other hierarchical structure. In reality there is no leader. Instead every individual in the swarm follow a set of rules for interacting with there local environment.

Birds and fish interact by correcting there speed and direction according to the speed and direction of neighbouring individuals. Through this simple local interaction we get what is known as emergent behavior [Cucker and Smale, 2007]. This emergent behavior is what we observe when we see a flock of 10 000 starling dancing in the sky, or a colony of termites building impressive nests.

The absence of any hierarchical structure means that it has no single point of failure. If we were to shoot half the birds out of the sky. It would not impact the behavior of the flock. The remanding individuals would simply follow the rules they always have, and the swarms functionality would remain. This decentralized structure is the source of a swarm's robustness. The loss of no single individual will halt the swarm. As long as there are functioning individuals, the swarm will continue to function.

2.1 Agents and core principles

When creating a swarm intelligence system the individuals in a swarm are referred to as agents. These often react to the environment, with limited internal memory. Their behavior is usually decided by relatively simple interactions, rather than complex decision making. These interactions give rise to complex behavior.

The following can be said to be core principals of any swarm intelligence system:

- Decentralized control
- Redundancy
- Robustness
- Local interaction
- Emergence

2.2 Boids

Perhaps the most famous example of a swarm intelligence system is the artificial life program "Boids" [Reynolds, 1987]. Boids simulate the flocking behavior of birds. An agent in this system is called a boid, corresponding to a shortened version of "bird-oid object".

By giving these boids simple rules to follow Reynolds was able simulate emergent behavior, similar to that seen in flock of birds. In the simples version boids follow three rules, as seen in figure 2.1. Combining the vertices generated by these behavior rules will result in a steering vector. This vector corresponds to the direction in which the boids will travel.

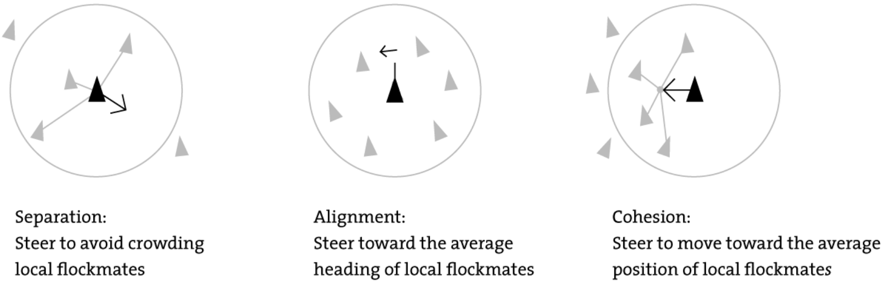


Figure 2.1: Boids behavior rules, with vectors. Figure borrowed from [Unknown, 2013]

2.3 Optimization algorithms

Some SI algorithms such as the Particle swarm optimization (PSO), Artificial bee colony algorithm (ABC) and Ant colony optimization (ACO) are metaheuristic optimization algorithms. Metaheuristic algorithms are useful for finding sufficient solutions to optimization problems. They do not guarantee optimal solutions. Rather the goal is to efficiently explore the search space to find a near optimal solution. This is especially useful with incomplete information or limited computational power [Karaboga, 2005].

2.3.1 Bee inspired optimization

A number of different optimization algorithms draw their inspiration form the behavior of bee colonies. Bee colonies are known to search large areas in search of nectar form flowers. Flowers yielding nectar are often found in patches. In a bee colony a small fraction of the bees will constantly be searching the environment looking for new flower patches. These are known as scout bees.

These scouts move randomly in the area surrounding the hive searching for flower patches. Once a flower patch is locate, the bee will evaluate the profitability of this food source. In essence this is done by estimates net energy yield, taking into account the amount of nectar, and the distance to the hive [Tereshko and Loengarov, 2005].

After a flower patch is located, and its quality established the bee will collect nectar and return to the hive. Here it land in a recruitment are, described as the "dance floor". Once landed onlooker bees will observe the scout bee as it preforms a waggle dance [Von Frisch, 1967].

Through the waggle dance the scout bee communicates the location, and the quality of the flower patch. Onlooker bees will make a stochastic decision on whether to pursue the food source. Higher profitability translate to a higher chance of recruiting onlookers

If an onlooker bee is recruited it switches role to a forager. It will then follow the direction given by the scout to the food source. After preforming the waggle dance the scout will join the foraging process. Both the scout and the foragers will continue to forage the flower patch as long as it is evaluated as profitable.

When returning to the nest both scouts and foragers may repeat the waggle dance, increasing the recruitment for highly rewarding flower patch. Because of this mechanic bees can dynamically adapt their efforts to optimize foraging results.

Artificial bee colony algorithm

The artificial bee colony algorithm (ACO), introduced by Karaboga [2005], is a population based optimization algorithm heavily inspired by the foraging behavior of natural bees. Like in nature ACO relies on employed bees (foragers), onlookers and scout.

In ACO the position of a food source correspond to a possible solution. The amount of nectar represent the solutions fitness (quality). At the beginning a randomly distributed population of solutions is generated. The number of employed bees is equal to the number of solutions.

After initialization the system is subject to a repeating cycle. At the beginning of every cycle each employed bee stores its solution, before modifies its position. If the fitness of the modified solution is higher than the fitness of the original, the modified solution replaces the original.

After this the employed bees preforms a waggle dance on the "dance floor". Onlooker bees do a stochastic evaluation based on the fitness of the employer bees solutions. Thus better solutions have a higher chance of recruiting more onlookers.

Once an onlooker is recruited it will create its own modified solution, taking on the role as an employed bee. Artificial scouts determine which solutions are to be abandoned, and randomly produce new once.

2.3.2 Ant colony optimization

The ant colony optimization algorithm (ACO) was first purposed by Colorni et al. [1991]. It is a metaheuristic algorithm for solving computational problems which can be reduced to finding good paths through graphs. The algorithm takes its inspiration from the behavior of ants. Ants find food through the process of foraging. When finding food they seek paths connecting the food source with the nest. Through the use of pheromones ant have found a way of optimizing these path. It is this optimization process that form the bases of ACO.

In most spices of ant, ants wander randomly around the world looking for food. When an ant finds a food source, it leaves a trail of pheromones on the ground as it tracks home to the nest [Jackson et al., 2004]. Other ants finding such a trail have a high probability of following the path it creates. If an ant starts following a trail it to deposits pheromones, reinforcing the trail.

The pheromones evaporates over time, and the higher the concentration of pheromones, the higher the probability of other ants joining the path. A shorter path will be traversed faster than a longer one. As such the concentration of pheromones will be higher. This will result in ants having a higher probability of joining the shorter path, further reinforcing it. This will lead to the ants converging on the shorter path.

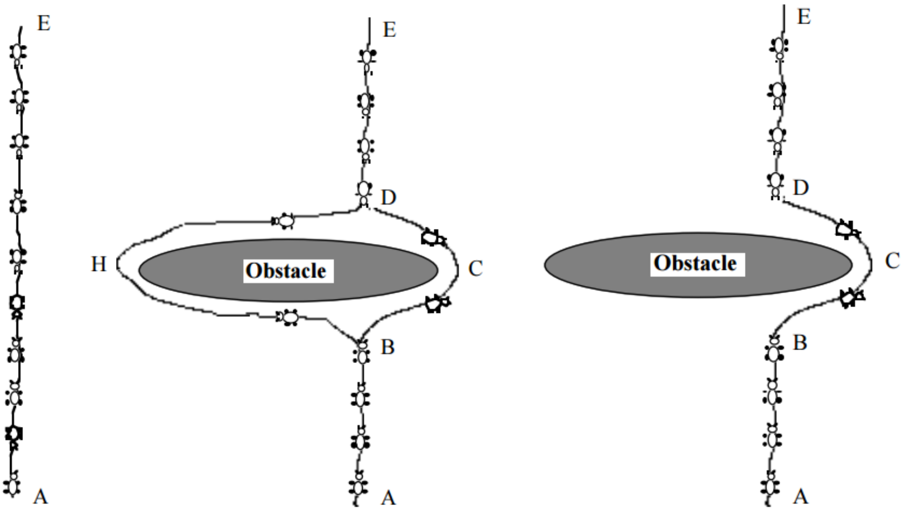


Figure 2.2: Ants adapting path around an obstacle [Colorni et al., 1991]

Through this mechanic ants also show the ability to adapt to dynamic changes in the environment. If an obstacle is placed on an existing path, the ants will reroute the path converging on the shorter way around it.

This can be seen in figure 2.2, where an obstacle is placed on an existing path creating a short and a long way around it. The first ants to encounter this new obstacle will have an equal probability of turning left or right. As the first ants navigate around the obstacle they will deposit pheromones. Since the shorter way around the obstacle will be traversed faster, the concentration of pheromones will be higher. As a result new ants encountering the obstacle have a higher probability of choosing this path. Over time the longer trail will evaporate and the shorter path is established.

ACO is closely based on this idea found in nature. In ACO, artificial ants (simple reactive agents) traverse a graph looking for good paths. Before any established paths have been created, the ants move at random. This randomness ensures width in the initial search.

Just like real ants, these artificial ants will deposit pheromones once paths through the graph are discovered. As pheromone trails become more distinct randomness will decrees. After each iteration an amount of pheromone will be evaporated. This helps the system from converging on a local optima. Eventually most of the ants follow the same path, thus a solution is found.

2.4 Swarm robotics and Foraging algorithms

Solving foraging tasks is one of the canonical applications of swarm robotics [Zedadra et al., 2016; Winfield, 2009; Sugawara et al., 2004]. It is a complex problem requiring coordination both in exploration, harvesting, homing and depositing. Because of this it serves as a benchmark problem in swarm robotics [Winfield, 2009]. Many variations exist, but they all share a common goal.

Foraging can be described as follows: Explore an environment, known or unknown, looking for resources. Upon finding a recourse transport this recourse back to the starting location. The starting location, at which agents start and return to, is typically described as a nest.

Thus the foraging task can be split into two parts. First, perform a area search for resources. Second, collect and transport these recourse back to a central location [Campo and Dorigo, 2007].

Although the foraging scenario, and the approach taken vary (see 2.6), some common elements can be identified:

- Cooperative exploratiom/search.
- Discovery of a resource location.
- Establishment of good paths back to the nest.
- Transportation resources back to the nest.

2.4.1 Heterogeneous robots/agents and Morphology

The majority of swarm robotics research is done using homogeneous robots [Dorigo et al., 2013]. In a homogeneous swarm all the robots are constructed to be identical. This means that they have the same sensory capabilities, speed, capacity, and other abilities.

In addition their behavioral rules are typically identical. This means that if two robots are placed in an identical environment, and there are no stochastic behavioral mechanics, each robot would behave and preform identical.

Such robots are in the literature commonly refereed to as homogeneous robots. Although this is true, they are in fact both morphologically homogeneous and behaviorally homogeneous. Morphological because they are psychical identical, and behavioral because they behave identical.

This is important because robots that are morphologically homogeneous, can still be behaviorally heterogeneous. This is the case when robots are constructed identical but behaviour differ. Robots are then typically assigned a role in the swarm [Momen and Sharkey, 2009]. This role defines its task, that might be a subset of the task required to achieve the goal. These roles can either be predetermined, or robots can change roles dynamically [Rodríguez et al., 2015].

Alternatively the robots can be morphologically heterogeneous. If this is the case, the robots are usually also behaviorally heterogeneous. This implies that robots are given different capabilities. Either by varying equipment (e.g. sensors, communications, memory), or the fundamental structure of the robot.

The difference in robot functionality in a morphologically heterogeneous swarms vary. Perhaps the most common morphologically heterogeneous robot composition is the combination of aerial drones and land based robots [Chaimowicz and Kumar, 2004; Ducatelle et al., 2010, 2011; Dorigo et al., 2013; Liemhetcharat et al., 2015].

Lastly robots can in theory be combined to create a morphologically heterogeneous, but behaviorally homogeneous. This is however rarely seen in practice, as the rational for changing morphology is to specialize robots for different tasks [Parker, 2003].

The use of terminology vary throughout the literature. Some papers make due with describing their system as heterogeneous, letting the morphology of the agents be revealed by the context or description. In this theses the terminology descried above will be used unless it is considered explicit. Thus we are left with the following swarm compositions:

1. Morphologically homogeneous, behaviorally homogeneous.
2. Morphologically homogeneous, behaviorally heterogeneous.
3. Morphologically heterogeneous, behaviorally heterogeneous.
4. *Morphologically heterogeneous, behaviorally homogeneous.*

In this section different swarm compositions have been described using the term robots. However, a lot of the research done on foraging algorithms, meant for robots, is strictly done in simulation. In such cases one can still describe agents as being morphologically heterogeneous. This is the case if agents are given different capabilities in simulation (e.g. capacity, energy use, speed, senses).

2.4.2 Pheromone foraging algorithms

Swarm intelligence is based on implicit cooperation between agents. This communication can be direct, where agents communicate peer to peer. Alternatively indirect communication can be used. In the literature, much of this indirect communications draws is inspired by the pheromone communication seen in nature [Panait and Luke, 2004].

Repulsive pheromones

In systems where the goal requires exploration (e.g. exploration, surveillance, mapping, reconnaissance and foraging), repulsive pheromones are often employed [Fossum et al., 2014; Calvo et al., 2015; Kuiper and Nadjm-Tehrani, 2006; Pearce et al., 2006]. By depositing repulsive pheromones, and avoiding pheromones deposited by other agents, the swarm can explorer an area in a highly efficient manner.

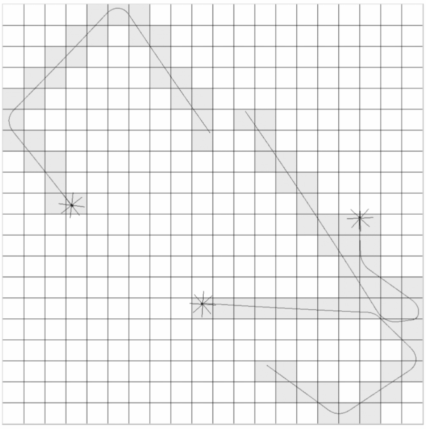


Figure 2.3: An illustration of agents leaving repellent pheromone as they explore an environment. Borrowed from [Fossum et al., 2014]

As agents are repelled from areas already visited, agents gravitate towards unexplored areas. If the task is strictly to explore, or map an area the pheromones are often persistent. If the task is surveillance, an evaporating pheromone can be used [Calvo et al., 2015], encouraging agents to revisit areas that have not been visited in a while.

These repulsive pheromones can be combined with other types of pheromone to improve a foraging system [Calvo et al., 2015]. By introducing path pheromones, agents can efficiently and cooperatively navigate back to the nest [Zedadra et al., 2016].

2.4.3 Implementing pheromones in swarm robotics

As we know ants communicate by depositing pheromones in their environment (section 2.3.2). But how does one implement this in a swarm of robots? Some attempt to solve this by depositing chemicals such as ethanol in the environment [Fujisawa et al., 2008; Liu, 2008]. Few researchers do however consider chemicals to be a viable strategy for implementing pheromones [Hayes et al., 2002].

Another alternative is to utilize virtual pheromones [Sugawara et al., 2004]. How this is best implemented is still an open question. Some researchers have turned their attention towards deployable electronic markers. Either in the form of RFID chips [Johansson and Saffiotti, 2009; Sakakibara et al., 2007], or deployable beacons [Hrolenok et al., 2010].

Another alternative is to let each agent maintain a virtual model of pheromones dispersed throughout the environment. Keeping these models in sync then becomes an issue. Some have proposed propagating these changes as agents meet locally [Rodríguez et al., 2015]. Others suggest inter-agent communications through ad-hoc wireless networks [Howard et al., 2006].

2.5 Structured Literature Review

A structured literature review (SLR) is as the name implies a way of reviewing related literature in a structured manner. Its a formal way of synthesising available related literature down to a set of research questions [Kofod-Petersen, 2012]. As mentioned in 1.3 a SLR is no guarantee of capturing all related literature, but it aims to give a fair evaluation of the related work [Keele, 2007].

Each step of the procedure is documented thoroughly. This rigorous procedure give rise to a trustworthy and auditable methodology. Every step of the structured literature review should be reproducible. This allows the SLR as a whole to be reproduced independently. This increases the value of the results, as they can be confirmed and validated by others.

There are many reasons to perform a structured literature review. For instance a SLR can be performed to investigate if empirical evidence in the existing literature supports an existing hypothesis, or even as a bases for formulating a new hypothesis [Keele, 2007]. In this paper the SLR primary function is to uncover what is state-of-the-art within heterogeneous swarm used for foraging, pheromone based swarms used for foraging, and a combination of the two. This is done both as a source of inspiration for developing new approaches, but also to uncover any gaps in current research. Finding such gaps can suggest that more research in this areas is needed. Ideally this will revile what is state-of-the-art, and possible directions to take this research in in order to broaden our understanding of it.

2.5.1 Structured literature review protocol

In this thesis a structure literature review was performed to answer research questions RQ1 and RQ2, see section 1.2. To achieve this a structured literature review protocol was developed. It is a detailed description of the steps taken to preform the SLR. In addition it gives an overview of the document collection at each step, and a quality assessment of the final papers. The SLR protocol can be found in Appendix A.

2.5.2 SLR results

Initially the SLR returned a collection of 1805 document, out of which 633 were duplicates. This gave a total of 1172 unique documents. Through an iterative filtration process the document collection was reduced to 27 documents. These documents underwent a detailed quality assessment, seen in section A.7. A threshold based on quality criteria was set, excluding papers that achieved a unsatisfactory score based on these criteria. After this exclusion, 14 papers remained (See table 6, Appendix A). These 14 papers constitute the core papers of the SLR, and are presented in section 2.6.

2.6 Related Work

Throughout this section food and resource will be used interchangeably to describe the unit which agents forage, e.i. search for and transport back to the nest. The word "nest" is used to describe the location at which agents drop of resources. This is typically the starting location of agents.

Most of the work cited an described in this section is part of the core studies retrieved from the SLR. Other papers are occasionally referenced to put the core studies in relation to the rest of the literature. A more complete overview of the selection can be found in Appendix A, table 6 and 8.

The core papers presented in this section vary both in their problem definition, goals and approach to such an extent that presenting them only in context of each other would poorly describe their function and place in the literature. As such each paper is presented in its own sub-sections, detailing their core functions and the problem they try to solve. Parallels between papers are highlighted where present.

2.6.1 Heterogeneous swarm

An Ant-like Task Allocation Model for a Swarm of Heterogeneous Robots

An important issue arises when the use of a heterogeneous swarm is considered. How does one decide division of labour? Momen and Sharkey [2009] proposes a threshold based approach for division of labor in a heterogeneous swarm. The idea being that agents can adapt to dynamic changes in the environment on demand.

A change in the environment may result in the demand for one task increasing, and the demand for another decreasing. Agents can then switch task accordingly, adjusting to demand.

Momen and Sharkey describe a system of three types of behaviorally heterogeneous agent: Foragers, Larva, and Brood-Carriers. Larva stay in the ”nest” and consume food (the resource). Foragers forage for food, and deposits the food in a depot area. Brood-Carriers move food from the depot and feed it to the Larvae in the nest.

The system’s threshold functions can be divided into two. The first being threshold functions for deciding whether to perform their respective task, e.g. feed larva or forage.

The second type of threshold allows Brood-Carriers to switch to foraging if the amount of food in the depot is to low. By doing so the swarm adapts to the increased demeaned for food, which in turn improves the efficiency of the swarm. Brodd-Carriers can then switch back to feeding larva if the food level in the depot recovers.

Foraging-inspired Self-organisation for Terrain Exploration with Failure-prone Agents

Rodríguez et al. [2015] explores an interesting approach around information sharing. Instead of direct transfer via the environment, like chemical pheromones [Liu, 2008] or beacons [Hrolenok et al., 2010], they employ virtual pheromones stored in agent memory. Agents communication via Trophallaxis. That is whenever an agent senses a neighbouring agent in its vicinity, they exchange local information.

This tries to solve the problem of propagating changes to a virtual pheromone map throughout a swarm. which is a canonical issue in virtual pheromone systems [Payton et al., 2004].

To solve this issue Rodríguez et al. [2015] employ the use of a system consisting of Seeker and Carrier agents for exploration and search. Carriers look for uncharted terrain and hence pursues locations with less pheromones. Seekers look for other agents and hence explores locations with more pheromones.

Agents have a social status, that indicates their role, e.g. Seeker or Carrier. An agent a_0 receiving information from an agent a_1 can have two outcomes. Either a_0 can receive new information, changing its internal pheromone model. Alternatively no new information is received, leaving the pheromone model unchanged. If the pheromone model is changed a_0 becomes a Seeker. If pheromone model remain unchanged a_0 becomes a Carrier.

The idea is to have some agents focus on exploration and data collection (Carries), while other agents (Seekers) focus on recording and distributing this data. This ensures that the swarm both explores the environment, and that data is distributed within the swarm. This in turn helps avoid scenarios where agents with a lot of unique data fail before sharing it, thereby losing data.

If an agent receives new information it has likely fallen behind the exploration front. Interestingly Rodríguez et al. discovered through experimentation that the Seeker-Carrier system did not perform better than a hybrid approach, which relies agents to meet passively. The main problem being that Seekers tend to increase encounters between agents with similar data.

Continuous Foraging and Information Gathering in a Multi-Agent Team

Liemhetcharat et al. [2015] proposes a system of continuous foraging using heterogeneous agents. The system consist of reconnaissance agent and foraging agents. The agents have widely different capabilities and are thus considered morphologically heterogeneous.

The swarm is comprised of foraging agents, that transport resources back to the nest. And reconnaissance agents that gather and shares information with the rest of the swarm. Liemhetcharat et al. [2015] considers resource distribution adhering to the Bernoulli, Poisson and Logistic distribution.

Liemhetcharat et al. [2015] foraging scenario does however contain some simplifications. These are done in accordance with the motivating scenario, where foraging agents are fishing boats and reconnaissance agents are uav. First it assumes that the location of resources are known. This is justified by the possibility of using satellite photos. It also assumes that the change in resources at a location is Markovian, i.e. the number of resources at a time step T_n only depends on $T_n - 1$.

Hence the role of the reconnaissance agent is not what one would initially expect, namely that of locating resources. Rather it is to observe resource locations firsthand, in order to determine the amount of resources present. The reconnaissance agent then broadcast these observations to the foraging agents.

The foraging agents can only estimate the resource at a given location, and firsthand observations are useful to correct these estimates. It is also assumed that the reconnaissance agents movement is instantaneous, but limited to a subset of all resource locations at each timestep.

Liemhetcharat et al. [2015] considers three foraging algorithms: Greedy rate, Adaptive sleep and Adaptive sleep with target change. These algorithms are primarily intended for different scenarios.

Greedy rate chooses the resource with the highest expected rate of foraging that is not earmarked by another agent. In Adaptive sleep agents sleep until a resource is deemed "worth" foraging. This is primarily to make the swarm as energy efficient as possible. Adaptive sleep with target change allows agents to change targets dynamically.

Cooperative Self-Organization in a Heterogeneous Swarm Robotic System, and Self-organized cooperation between robotic swarms

Ducatelle et al. [2011, 2010] explores a physical system with morphologically heterogeneous agents. The system consist of two types of robots, foot-bots and eye-boots. The eye-boots are small drones capable of flying and attaching to a ceiling. Here they observe foot-bots and other objects via cameras. Foot-bots are small robots moving on the ground using treels. They have two cameras, one for spotting other foot-bots, and one for spotting eye-boots.

Both robot types communicates via powerful LED lights in 256 colors. In addition they can communicate via infrared. This system is implemented physically, demonstrating its functionality. In addition they have created a three dimensional simulation allowing for more complex testing.

The key component of [Ducatelle et al., 2011, 2010] is a process of mutual adaptation. Foot-bots execute instructions given by eye-bots, and eye-bots observe the behavior of foot-bots to adapt the instructions they give.

The system performs a simple foraging task, by locating a target area and moving back and forth between the target area and the source. The system is capable of converging on the shortest path, and spreading out in case of congestion.

Dealing with heterogeneous swarm opens up new possibilities, but also adds complexity. As such Ducatelle et al. [2011, 2010] tires to reduce the complexity by looking at different parts of the system separately. For instance the problem of coordinating, moving and deciding locations for the eye-boots can be seen as independent of coordinating the foot-bots, using eye-boots in position. Therefore Ducatelle et al. [2010] ignores the later problem, focusing on the behavior of foot-bots.

This is done by distributing enough eye-bots throughout the environment to achieve 100% coverage, before the foot-bots are deployed. The eye-bots task is to guide foot-bots to the location of the target. Because the topology of the terrain is different for the foot-bots and the eye-bots, the eye-bots can not simply guide the foot-bots according to their own sensor information.

Rather the system relies on what the authors call cooperative self-organization. The emergent behavior of the swarm comes from communications between eye-bots and foot-bots, rather than foot-bot to foot-bot communication.

Initially eye-bots give foot-bots random instructions. Foot-bots in turn give feedback about their behaviour and experience. This allows the eye-bots to adapt the instructions they give. Ducatelle et al. [2011] improves on the solution by considering the positioning of the eye-bots. The idea is that eye-bots move in the direction they observe a lot of foot-bots. The rationale being that the foot-bots gravitate toward areas that are navigable.

Allowing eye-bots to adaptively learn the best location and relocate to this position is more flexible. Furthermore it allows for task completion with fewer eye-bots.

These mechanics result in a swarm capable of adapting to changes in the environment, converging on the shortest path to a target (e.g. a resource) and avoiding obstacles.

2.6.2 Pheromones based swarm

Collaborative foraging using beacons

One of the problems with using pheromone in swarm robotics is how one implement it in a physical system. As long as the system is simulated one can simply overlook this problem, and implement the pheromone model as shared data between the agents. However, once a system enters the realm of the physical, and thus transitions from swarm intelligence to swarm robotics, this problem must be considered.

Relatively little work has been done on real robot foraging with pheromone controlled behaviours due to the technical difficulties using chemical materials [Liu, 2008]. Meng and Gan [2008] tries to solve the problem by mimicking pheromones with phosphorescent glowing paint. Others simply let the pheromone model be stored on a central server, letting agents update and retrieve data from it [Hecker et al., 2012; Ando et al., 2006]. This does however violate one of the core principles of swarm intelligence, namely that it should be decentralized.

Although such systems can work, they introduce a single point of failure, thus dramatically lowering the system's robustness.

Hrolenok et al. [2010] tries to solve this problem with the use of stigmergic markers. They circumvent the physical difficulty of implementing actual pheromones, by proposing a sparse representation of the pheromones using movable beacons. There is no communication between the beacons to propagate pheromones; instead, robots make movement and update decisions based entirely on local pheromone values.

The swarm utilizes three types of pheromones: foraging pheromone, ferrying pheromone, and a wandering pheromone. The foraging pheromone is used to build a gradient to a food source. The ferrying pheromone is used to build a gradient to the nest. The wandering pheromone is used to indicate how often a beacon has been visited.

From the perspective of the robots, the beacons are a graph of states with utility values. Each beacon stores one value per pheromone employed by the swarm. An agent will associate itself with the nearest beacon, this will be its current beacon. Agents can read and update pheromone data from its current beacon. In addition agents can sense if food or the nest is within range.

In order to increase robustness Hrolenok et al. [2010] introduces some randomness. The idea is that by giving agents a small chance of entering a "wandering state", as opposed to following pheromones, agents will be pushed into areas not well explored.

The system can be deployed in two modes. The first being a scenario where beacons have already been placed in the foraging environment. In this setup the system resembles other systems relying on dense pheromone grids. In the second mode the agents themselves both deploy and move the beacons. By deploying beacons as the agents explore the environment, the system becomes more realistic and flexible, approaching a usable real world application.

The goal is to place beacons in uncrowded regions, that are still within range of the current beacon. By allowing agents to move beacons, the swarm can improve suboptimal trails (e.i. improving the current path to a food source). The goal here is to move beacons between neighbors that are likely to be on the trail. In addition to this agent can also remove beacons. This is important because there exist a finite number of beacons.

Through experimentation Hrolenok et al. [2010] show that their system is capable of dealing with different environments and dynamic changes in these. They show that there system can adopt to moving obstacles, moving food sources, losing agents and beacons.

Formica ex Machina: Ant Swarm Foraging from Physical to Virtual and Back Again, and Synergy in Ant Foraging Strategies: Memory and Communication Alone and In Combination

Hecker et al. [2012] and Letendre and Moses [2013] are both written on related research out of the University of New Mexico. Their approach is largely inspired by the foraging of the Pogonomyrmex desert seed-harvester ants. These ants typically leave their colony's nest, travel in a relatively straight line to some location on their territory. Here they switch to a correlated random walk in search of seeds.

Drawing inspiration from this Hecker et al.; Letendre and Moses suggest a system that combines pheromone trails with individual memory. Initially all agents move away from the nest in a straight line and a random direction. Arriving at their location agents perform a correlated random walk, looking for food.

Upon finding a food source, agents count the food found in the Moore neighbourhood. Agents then create a pheromone trail from the food source to the nest.

After depositing food at the nest agents make a choice. Based on GA balancing of parameter, agents can either choose to follow existing pheromone trials or use their internal memory of sites visited, so called site fidelity, to continue foraging.

If a food source is sparse, the most efficient strategy is typically to let an agent use site fidelity to harvest the remaining food. If the food source is rich, it may be more beneficial to recruit additional agents (using pheromones). By combining site fidelity with pheromone models in a well balanced manner, Hecker et al.; Letendre and Moses show that such a system can outperform either approach on their own.

A multi-pheromone stigmergic distributed robot coordination strategy for fast surveillance task execution in unknown environments

Calvo et al. [2015] employs an ant based system utilizing three types of pheromones. Their system is design for surveillance of an area, rather than foraging. As such they have not implemented a pheromone for guiding agents carrying food back to the nest, like the ferrying pheromone employed in Hrolenok et al. [2010].

Instead they introduce vertex pheromones. The idea being that agents can mark strategic locations with this pheromone, effectively creating a vertex. This will allow agents to better navigate environments with complex obstacles such as tight corridors or a building with many rooms.

This is combined with a repulsive pheromone. By having the repulsive pheromone evaporate over time, agents are encouraged to explore areas that are either unexplored, or have not been visited recently.

The third pheromone, named path pheromone, takes advantage of the vertices created using vertex pheromones. By depositing path pheromones between vertices, agents connect them. This creates a path that agents can follow to navigate the environment efficiently. Marking strategic locations will in essence expand this path.

By avoid repulsive pheromones agents can efficiently surveil the surrounding area. By introducing these new types of pheromone Calvo et al. [2015] is able to outperform the existing System-Based Surveillance System [Calvo et al., 2011].

Pheromone Averaging Exploration Algorithm

Momen and Sharkey [2009]; Rodríguez et al. [2015]; Zedadra et al. [2015a,b, 2016]; Fortino et al. [2014] have all implemented their algorithms on a discrete grid model. This is also the case with Florea et al. [2015]. But unlike the above mentioned, Florea et al. have chosen to model their algorithm in a 8-connected grid, meaning that all diagonals are both observable and visitable.

This is in contrast to the usual 4-connected grid model found in the other studies. Florea et al. [2015] seeks to improve the performance of pheromone based exploration algorithms by changing the pheromone update rule. Their approach is to incorporate the information from all neighbouring vertices of the previously visited vertex into the pheromone update rule.

When an agent visits a vertex, pheromone values essentially propagates from the neighborhood of the previously visited vertex to the current vertex. This aims to create a pheromone barrier so that agents avoids re-visiting that area. The design of the update rule means that agents propagates the pheromone trail from the frequently visited areas towards the exploration frontier. The greater the distance from the frequently explored areas, the less pheromones it deposits.

Fortino et al. and Zedadra et al.

Fortino et al. [2014]; Zedadra et al. [2015a,b, 2016] try to improve upon the c-marking algorithm of [Simonin et al., 2014]. This in turn is a multi-agent distributed and asynchronous version of Barraquand et al. [1992] canonical work on robot path planning using numerical potential field.

By creating an artificial potential field (APF) across a grid representation of the world, agents can follow the gradient of the APF to achieve the shortest path from the starting location to any location.

In order for this to work the system must create an optimal APF. The c-marking algorithm converges to an optimal APF, but this takes considerable time as agents often re-write values multiple time.

Fortino et al. [2014] present an extension of the agents model seen in the c-marking algorithm, based on resources affluence. This is designed to change the behavior of robots to enhance the results. By doing so Fortino et al. [2014] is able to reduce the foraging time by a significant amount.

To avoid the huge amount of time the c-marking algorithm takes to converge on an optimal APF, Zedadra et al. [2015a,b, 2016] employes the S-MASA exploration algorithm [Zedadra et al., 2014]. In short this involves agents synchronously exploring in a spiraling fashion away from the center, e.i. the nest. This allows agents to constantly walk along the exploration front and expanding it synchronously.

Stigmergy in the work of Ducatelle et al.

An interesting aspect of Ducatelle et al. work on heterogeneous swarm using foot-bots and eye-bots (presented in 2.6.1), is that it can also be seen as a stigmergic system.

Ducatelle et al. [2010, 2011] have primarily described the system from the perspective of the eye-bots, dictating the movement of foot-bots. But as Ducatelle et al. [2010] points out, by looking at the system from the view of the foot-bots one gets a different picture.

In this view, foot-bots try different paths between the two locations, and the role of eye-bots is to store past foot-bot experiences and communicate them to other foot-bots. Seen this way, eye-bots play the role of stigmergic communication points for foot-bots in the environment.

2.6.3 Discussion

Heterogeneous agents

[Liemhetcharat et al., 2015] show that a morphologically heterogeneous swarm can be useful in a foraging like scenario. However, their simplifications and assumptions means that a these advantages may not presist in a physical system.

Ducatelle et al. [2010, 2011] explores an interesting approach to foraging, using a morphologically heterogeneous swarm. Their system pushes swarm robotics forward in more than one area. In particular it shows that morphologically heterogeneous robots can efficiently cooperate on a foraging task. It is however not a complete foraging system.

In Ducatelle et al. [2010, 2011] the issue of foraging multiple targets is not considered. Nor do they offer any direct solutions to how such a system can function without magnetic attachments to a ceiling. The eye-boots are also limited by line of sight. Meaning that objects that are highly plausible in a foraging environment may disrupt communication. In an indoor environment a large table can allow foot-bots to move under it, breaking line of sight.

Pheromones

Hecker et al. [2012]; Letendre and Moses [2013] combine pheromone maps and site fidelity to create a system that outperforms either approach on their own. However for this to be the case a particular distribution of food is needed. Namely a distribution where some of the food is found in large piles, and some of the food is found scattered in smaller piles or separately.

Florea et al. [2015] does extensive performance testing on their algorithm. In their experimentation, they show that the system outperforms all of is comparison. However the algorithms they compare with are old. Consequently it can be argued that these exploration algorithms can not be considered good performance references.

A problem to consider when repellent pheromone are employed, is that agents can get stuck in local maxima/minima. This is a known problem in the literature [Arkin, 1998]. Although this is avoidable [Barraquand et al., 1992], it is still an issue in the system proposed by Rodríguez et al. [2015]. In the case of pheromone maps this will translate to an agent being trapped in a "cluster" of pheromones, trying to avoid all neighbouring areas because of their high value.

Peer to peer information sharing

Rodríguez et al. [2015] explores a system of Seeker and Carrier agents in an attempt to improve information exchange within a swarm with limited communication range. The idea is that allowing some agents to focus on information gathering can improve the swarm's efficiency and robustness.

Through simulation they discovered that Seekers did not improve performance over passive interaction. The main problem being that the Seekers generally stay in the same area, thus interacting with agents that contain much of the same information.

The approach is nevertheless intriguing. One can for instance imagine a mechanism where Seeker agents are repelled from agents they have already interacted with. This would encourage Seekers to approach "new" agents.

Using mobile agents as information propagators is an interesting one, and may prove useful in solving the problem of distributing changes to a pheromone model through a swarm.

A priori knowlage

Liemhetcharat et al. [2015] assumption that the location of resources is known in advance begs the question if the scenario can really be called a foraging problem. One can argue that since the amount of a resource at a location can only be approximated, it can still be considered foraging.

However, no matter the definition it seems self evident that it is a simplification of the foraging task. In most foraging system agents must also locate the resources [Ducatelle et al., 2011; Zedadra et al., 2016; Rodríguez et al., 2015; Hecker et al., 2012].

Food/Resource distributions and models

The way food/resources are modeled and distributed vary. This is understandable as all these models and distributions correlate to different use cases.

Consequently it is difficult to say which approach is better, as they have different advantages. This inconsistency does however make the research difficult to compare.

In environments with discreet locations, food can be clustered so that a large number of neighbouring locations all contain food [Hecker et al., 2012; Letendre and Moses, 2013]. Food can also be scattered thought the environment [Liemhetcharat et al., 2015; Fortino et al., 2014; Zedadra et al., 2016].

Hecker et al. [2012]; Letendre and Moses [2013] use a specific food distribution in their experiments, clustering some of the food an scattering the rest. Liemhetcharat et al. [2015] use a different set of predefine models, while Ducatelle et al. [2010, 2011] use a single source.

There is also the question of how much food should be present at each location. Hecker et al. [2012]; Letendre and Moses [2013]; Liemhetcharat et al. [2015] employ a model where food is distributed in some fashion across a large number of locations. Fortino et al. [2014]; Zedadra et al. [2016]; Ducatelle et al. [2010, 2011] opt for a model with fewer locations, where each location contain a higher quantity of a recourse.

Discrete vs continiouse enviornments

Most systems that simulated agent behavior use a discrete environment. This means that simulations are run in an incremental cycles. In these systems the world is typically represented as a grid. Florea et al. [2015] lets each cell on this grid be connected with up to 8 neighbours.

This 8-connected grid allows agents to move in ways that better represent movement in a physical system. A 4-connected grid is however more popular [Momen and Sharkey, 2009; Rodríguez et al., 2015; Zedadra et al., 2015a,b, 2016; Fortino et al., 2014]. If energy consumption or distance is considered, this has the advantage of giving equal distance between all connected cells.

Alternatively systems utilizes a continuous environment. If the system is physical, this must necessarily be the case at some level [Ducatelle et al., 2010, 2011; Hrolenok et al., 2010]. Some simulations are also based on discreet environments [Hecker et al., 2012; Letendre and Moses, 2013].

From simulation to physical implementation

Hrolenok et al. [2010] demonstrate how swarm intelligence systems relying on pheromone maps can be realised in real world systems. Their work serves as a stepping stone from simulation to real world application.

However, in the current setup the paths will likely never be fully optimized. This is a consequence of the present beacon deployment and beacon movement rules being overly conservative. This is because the beacon movement rule at all cost tries to avoid breaking a chain of beacons. The beacon deployment rule will refuse to deploy beacons that can optimize the path if it considers the area to be overcrowded.

The system shows impressive robustness in dealing with dynamic events, like moving obstacles (destroying agents and beacons) and relocation of food sources. However, is deployable beacons a viable strategy in the real world?

The system will more than likely perform well in a laboratory environment, as it is specifically designed with this in mind. But how will the system fare in a truly dynamic environment? One in which the chain of beacons can be broken, either by damage, malfunction or even theft.

Is it feasible, both economically and otherwise, to deploy large amount of beacons in large scale scenarios, located in open environments. By effectively deploying infrastructure for communication in every scenario, as large amounts of hardware is both needed and exposed.

Long distance ad-hoc agent-to-agent communication is likely to become both cheaper and better in the coming years [Sugawara et al., 2004]. Considering this, is Hrolenok et al. taking swarm intelligence in a direction that will benefit the most from technological improvements?

Regardless of this, Hrolenok et al. [2010] show that it is possible to realising a physical swarm intelligence system relying on pheromones for communication. They show that such a system can also adapt to dynamic changes in the environment.

Ducatelle et al. [2010, 2011] clearly demonstrates that this can also be achieved with a swarm of morphologically heterogeneous robots. They do however achieve this without the use of pheromones.

In addition Ducatelle et al. [2010, 2011] bridges the gap between robotics and simulation, by accurately simulating their robotic system. This seems a likely direction to take in future years, as both practises have significant advantages.

Simulation vs Robotics

When studying the approach taken by different research in the core literature, a difference in research focus is observed. This is perhaps best illustrated in the divide between simulation and physical implementation.

Although the common goal is a physical swarm of robots, capable of autonomous foraging, the method for achieving this differ. Ducatelle et al. [2010, 2011]; Hrolenok et al. [2010]; Hecker et al. [2012] demonstrate that a foraging task can be performed with a swarm of robots, although it is clearly at an experimental stage.

These are important contributions to the research field. However, this kind of research is resource demanding, and inevitably focused on the robotic elements. This removes focus from the foraging algorithms behind it. Thus it is not surprising that the rest of the research is done in simulation [Momen and Sharkey, 2009; Rodríguez et al., 2015; Zedadra et al., 2016; Fortino et al., 2014; Florea et al., 2015].

2.7 Motivation

Although most foraging swarm intelligence systems are at an experimentation phase today, the end goal of most research is a swarm robotics system capable of real world application [Winfield, 2009]. Many systems are therefore implemented in robotics labs, to demonstrate and improve there capabilities in physical systems [Ducatelle et al., 2011; Hrolenok et al., 2010].

Transitioning from swarm intelligence to swarm robotics causes complications. Sensor inaccuracy, mechanical failure, positional accuracy, communication and cost must all be taken in to consideration [Winfield, 2009]. Because of this, few large scale physical implementations have been made [Seyfried et al., 2004]. These are problems that have to be solved if swarm intelligence systems are to ever be applied in real world foraging applications.

However, implementing swarm intelligence systems in simplified simulations offer some clear advantages. Yes, it diverges from any immediate real world application, but in return it offers a controlled and testable environment, free of uncontrollable factors.

Dorigo and Roosevelt [2004] point out that simulation is essential to speed up development of control algorithms. Simulation allows research to focus on investigation and discovery of the foraging algorithms them self. As such the future of foraging swarm robots lie along to axis. Realizing foraging robots capable of real world application, and continued research on foraging algorithms [Winfield, 2009].

2.7.1 Morphologically heterogeneous agents

The rational behind developing swarms of morphologically heterogeneous agents is two-fold. Firstly, giving agents different capabilities can increase a swarms flexibility [Parker, 2003; Ducatelle et al., 2011]. It can also increase the ability to dynamic respond to changing scenarios [Momen and Sharkey, 2009]. In addition it can improve a swarms capability to preform complex tasks [Ducatelle et al., 2011].

The second rational is that specializing agent capabilities allow different agents to perform different tasks with different efficiency. By leveraging this, morphologically heterogeneous agent can improve swarm efficiency [Dorigo et al., 2013].

One can for instance imagine small scouting agents fitted with sensory equipment capable of searching for resources. Because of their size and sensory equipment these agents would have limited carrying capacity. Other larger agents could be specialized to collect resources. These harvester agents would not need the same sensory equipment, and could be tailored to maximise carrying capacity in an energy efficient manner.

This can be observed throughout society in the way we transport goods. As technology has evolved their has been a shift towards specialized machines for transportation. Boats and trucks provide higher transport capacity at a lower energy/weight ratio, compared to smaller machines with sensory equipment.

A concrete example of morphological agents with different abilities is the system presented in Ducatelle et al. [2010, 2011]. Here the eye-bots are mobile and have greater sensory capabilities. This comes at a cost of being less energy efficient, and unable to transport objects.

2.7.2 Combining morphologically heterogeneous agents with pheromones

As described in section 2.7.1, morphologically heterogeneous robots seem to be able to improve performance in a foraging swarm. At the same time pheromone inspired algorithms are popularly employed in foraging scenarios [Panait and Luke, 2004; Sugawara et al., 2004; Rodríguez et al., 2015].

Yet to the best of the authors knowledge, little work has been done on combining these to approaches in a foraging task. Through a detailed literature review (section 2.5) no research addressing this combination was found.

Thus it seems likely that research is needed, both to successfully combine these approaches, and investigating their efficiency in different scenarios. It is also of interest to see if these approaches can be combine in a way that maintain their appealing attributes, as described in section 2.7.1

Chapter 3

Architecture/Model

Throughout this chapter food and resource will be used interchangeably to describe the unit which agents forage. Agents and robots will also be used interchangeably. Nest is used to refer to the starting location of agents, and their drop-of point for resources.

In this chapter the proposed foraging algorithm is presented. To avoid confusion with other algorithms presented in this chapter, the proposed algorithm has been named Heterogeneous Cooperative Algorithm for Foraging (H-CAF). The system is run in a discrete environment, on a 4-connected grid world. The world is unknown, meaning that agents have no a prior knowledge of its properties (e.g. obstacles, food source).

H-CAF can be considered an extension of the Cooperative Switching Algorithm for Foraging (C-SAF) [Zedadra et al., 2016]. H-CAF borrows the core mechanics of C-SAF, and extends them to work in unison with a new agent type (harvester). This makes H-CAF a control algorithm for a swarm of morphologically heterogeneous agents.

Additionally elements of the C-CMFA algorithm are incorporated [Zedadra et al., 2015b]. It can be considered a predecessor of C-SAF. These two algorithms employ homogeneous agents, communicating via pheromones or artificial potential fields (APF). Both are designed to forage resources spread throughout the environment.

In addition to serve as a foundation for H-CAF, C-SAF act as a homogeneous reference algorithm. Allowing H-CAF to be directly compered with a homogeneous algorithm, within the same simulation environment, using the same performance metrics.

Both C-SAF and H-CAF employs the S-MASA algorithm for exploration of the environment [Zedadra et al., 2014]. All agents in S-MASA, C-SAF and H-CAF are behaviorally deterministic.

3.1 Development environment and source code

Source code can be found on github:
https://github.com/kyrrelm/MasterThesis/

H-CAF and the systems it is based on, are all algorithms based on agent based modeling. Different core libraries and systems exist to facilitate the development of such systems. Mason [Luke et al., 2004], Player/Stage [Vaughan, 2008] and NetLogo [Tisue and Wilensky, 2004] are popular options, to name a few. These systems and others like them offer different advantages.

The C-SAF algorithm (section 3.6) was originally implemented in NetLogo in [Zedadra et al., 2016]. Mason is a java based simulation core, centered on discrete event multiagent simulation, with support of 2d visualization. Taking this into account both NetLogo and Mason was seriously considered as bases for developing H-CAF and comparing it with C-SAF.

After considering these alternatives a decision was made to build the simulation environment, and the agent models from scratch. The rational being that a discreet event based system with a 4-connected grid environment running in discreet time steps (detailed in 3.4), would not be too labour intensive to implement. Further, building the system from the ground up allows for a much greater degree of control and flexibility.

3.1.1 Programming language

The system is implemented in Java 8, using the java 1.8 SDK. The system is currently run as a JavaFX application. As of Java 8, JavaFX is integrated with the java JRE/JDK, thus JavaFX 8 is utilized.

Although the system is currently constructed as a JavaFX application, it is not primarily intended as a system with a graphical user interface or visualization. The inclusion of JavaFX is primarily to allow for simple visualization during development.

Therefore the simulation in its entirety (e.g. environment, agent models, etc..) contained within a Simulation class. Thus simulations can be run thought JavaFX, with and without visualization, or as a separate java application.

3.2 Wavefront algorithms and Artificial potential field

A popular approach to robot/agent path planing is through the use of a wavefront algorithm. The strategy is to let agents explore the environment in a series of wavefront expansions [Barraquand et al., 1992]. Each expansion constitutes an incremental increase in distance from a goal or source. Through this expansion agents create what is commonly refereed to as a artificial potential field (APF).

The classical APF approach consist of adding repulsive potential fields to obstacles and an attractive field to the goal [Arkin, 1998; Khatib, 1986]. This way agents seeking the goal can follow the negative gradient of the composite potential, much like a ball rolling down a hill. This approach is however prone to creating local minima where robots can be trapped [Zhu et al., 2006]. Barraquand et al. [1992] avoid this issue by computing one field from the goal. Implemented on a grid world this results in a BFS-like algorithm [Simonin et al., 2014]. Each cell is then given an integer value corresponding to its distance from the goal. Cells with the same integer value are in the same wavefront [Zedadra et al., 2015b].

Simonin et al. [2014]; Zedadra et al. [2015b] flip the APF in Barraquand et al. [1992] on its head. Instead of creating an APF from the goal, an APF is created from the nest. This way each new wavefront constitute a step away from the nest. Thus any agent returning with resources can follow the negative gradient back to the nest.

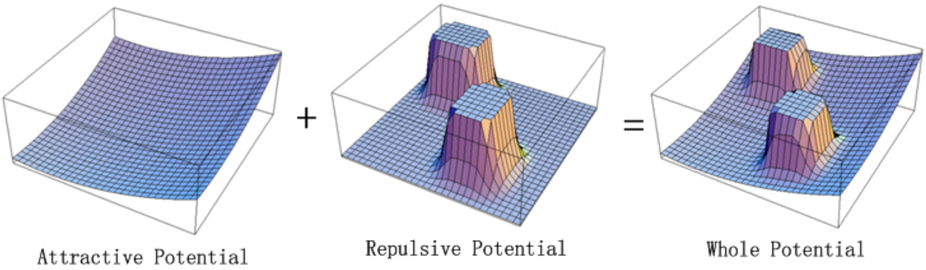


Figure 3.1: A graphical representation of an APF [Wang, 2012]

3.3 APF vs Pheromones

If pheromone based algorithms are to be realized in physical systems, a mechanism for deposition and sensing pheromones in the environment must exist. As presented in section 2.4.3, an implementation of pheromones can either be physical (e.g. chemical), or virtual (e.g. beacons, peer-to-peer network). Although research is still needed in this area, virtual pheromones seems the more promising solution.

If a physical implementation is used, a likely constraint is that pheromones will have to evaporate over time. A virtual system is however more flexible. Depositing an evaporating pheromones then becomes a matter of storing floating points associating with locations, and running them through an evaporation function at each time step.

In such a system non-evaporating values can easily be stored. This would allow for the creation of an APF. In addition it would be way more computationally efficient as values would only be written once, not overwritten through an evaporation function at each time step. S-MASA, C-SAF and H-CAF can all function with both an APF and evaporating pheromones.

3.4 Problem definition

In this section the problem that both the C-SAF and H-CAF algorithm tries to solve is defined. In addition the mechanics of the environment and the agents operating within it is described, as well as their constraints. These mechanics and constraints are the same for S-MASA, C-SAF and H-CAF. However, S-MASA only solves one of the tasks needed to solve the whole problem, namely exploration/search.

Environment

The simulation runs with discrete time increments denoted T_n, starting at T_0. The environment is unknown, meaning that agents have no knowledge about the location of any objects at T_0 (e.g. obstacle, food source).

The world consists of a finite set of 4-connected cells, denoted C. These cells make up a bounded 2D grid of size $N \times M$. All cells are neighboured by at most 4 other cells, and at least 2 other cells. The cells boarded by less then 4 cells constitutes the outer perimeter of the grid, e.i. sides, and corners. These cells are bordered by special border cells, that are sensible by agents, but are neither part of C nor reachable. Thus each cell $c = (x, y) \in C$ has a maximum of four neighbors $(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$.

All cells in C are either a food source, an obstacle, the nest or a blank cell. All cells are visitable except from obstacles. Thus all cells in C can be divided into one of two subsets, C_{Free} or $C_{Obstacle}$. This means that $C = C_{Free} \cup C_{Obstacle}$. $C_{Reachable}$ denotes the subset of C_{Free} containing all cells that are reachable, e.i. visitable cells that are not enclosed by obstacles and borders. $C_{Obstacles}$ denotes the subset of C that are obstacles.

Let c_0 be the nest and the starting cell for all agents. c_0 has the coordinates (c_{0x}, c_{0y}), where $0 < c_{0x} < N$ and $0 < c_{0y} < M$. $C_{Visited}$ contains all cells in $C_{Reachable}$ visited by at least one agent, meaning $C_{Visited} \subseteq C_{Reachable}$. $C_{NotVisited}$ is the subset of $C_{Reachable}$ that has not been visited by any agent. At $T_0; C_{Visited} = \{c_0\}$ while $C_{NotVisited}$ contains all cells in $C_{Reachable}$ except c_0. Thus $C_{NotVisited} \subset C_{Reachable}$ and $C_{Reachable} = C_{Visited} \cup C_{NotVisited}$.

Agents

All agents are located at any cell $c \in C_{Reachable}$. At T_0 all agents are located at c_0, e.i. the nest. At each time step T_n an agent a can sense all neighbouring cells of its current location, denoted $C_{Neighbours}$. Note that $C_{Neighbours}$ can contain cells that are not in C, namely the boarder cells. In the same time step T_n, a can move to any cell $c \in C_{Neighbours} \cap C_{Reachable}$. Note that their is no constraint on the number of agents located at the same cell, see section 3.7.

At each time step an agent can perform many actions, but only move once. Each move expends energy. The permitted actions are:

- Deposit food.
- Pick up food.
- Deposit pheromones in current cell, or any cell $c \in C_{Neighbours} \cap C_{Reachable}$.
- Remove pheromones in current cell, or any cell $c \in C_{Neighbours} \cap C_{Reachable}$.

As multiple actions can be executed in the same time step, agents can do the following example in one time step:

1. Sense cells in $C_{Neighbours}$.
2. Move to front cell $c_{front} \in C_{Neighbours} \cap C_{Reachable}$, containing food.
3. Pick up food.
4. Remove pheromone in current cell
5. Deposit pheromone in current cell
6. Deposit pheromone in left cell $c_{left} \in C_{Neighbours} \cap C_{Reachable}$.

Goal

C_{Food} denotes the set of all cells containing a food source, where $C_{food} \subset C_{Reachable}$. At T_0 all cells in C_{Food} typically contain substantially more food then the carrying capacity of an agent. Let F_c be the total amount of food in all C_{Food}, and F_n be the total amount of food at the nest (c_0). At T_0, F_n is equal to 0 and F_c some finite integer p. Then the goal is reached when F_n is equal to p, meaning that all the food has been deposited at the nest. Additionally the goal should be reached in the lowest amount of time using the least amount of energy, this will serves as performance metrics (section 4.2.3).

3.5 Stigmergic Multi-Ant Search Area (S-MASA)

S-MASA, first described in [Zedadra et al., 2014], is distributed exploration algorithm for homogeneous agents. It allows agents to collaboratively explore an unknown environment, in search of targets. Through stigmergic communication [Elliot, 2006], agents explore the environment, with minimal revisiting of cells. This is in contrast to the c-marking algorithm, where agent have to revisit cells and rewrite values, to eventually converge on an optimal APF.

In S-MASA, agents explore the environment in a vortex-like pattern outwards from a central location. Upon visiting a cell an agent marks this cell as visited. An agents exploration movement is dictated by a simple behavioral rule. If its right cell is marked as visited it will move to the cell in front, marking this as visited. It will continue to do so until it encounters a right cell not yet visited. The agent then makes a right turn, visiting the unexplored cell. Then it continues straight until another unvisited right cell is discovered.

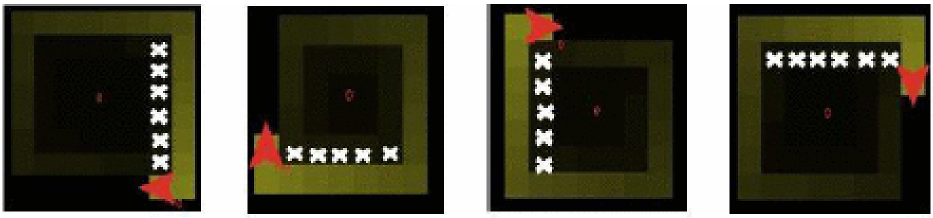


Figure 3.2: S-MASA coordination principle, where white crosses represent already visited cells [Zedadra et al., 2014]

This movement is well suited for parallel exploration by multiple agents. Since all agents move in a clockwise fashion around a center point, agents will converge to a pattern of exploring a row or column one cell further away from the centre compared to the previous agent. This results in a simultaneous expansion of visited cells, where agents rarely visits cells that have already been visited, as seen in figure 3.3.

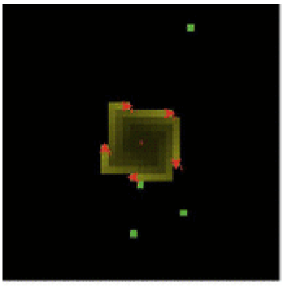


Figure 3.3: S-MASA: Agents explore the environment in a coordinated vortex patern. [Zedadra et al., 2014]

As the agents move about, they can either deposit pheromones that evaporate over time [Zedadra et al., 2014, 2015a, 2016], or a static integer values [Zedadra et al., 2015b]. If a static integer value is used an APF is created. In an obstacle free environment this will be an optimal APF without the need to rewrite any values, see [Zedadra et al., 2015b] for prof. Agents can then return to the nest, following the negative gradient. In an obstacle free environment, this is also constitutes the shortest path. The same principle can be applied if the agents use evaporating pheromones. Agents simply moves to the neighbouring cell with the lowest amount of pheromone. The return path will differ slightly from the return path generated by an APF, but in an obstacle free environment it two will result in a shortest path.

3.5.1 Obstacle Avoidance

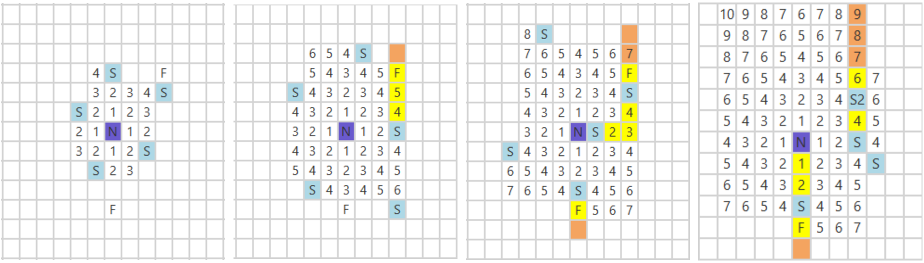
If an agent senses an obstacle, the strategy is to get around the obstacle in the direction of already explored cells. Upon sensing an obstacle in its front cell c_{front} ($c_{front} \in C_{Obstacles}$), an agent a does the following. (i) It rotates in its right direction until $c_{front} \in C_{Reachable}$. (ii) a will then move to c_{front} while $c_{front} \in C_{Reachable}$. It will do this until either $c_{front} \in C_{Obstacles}$, or its front, left or right cell is in $C_{NotVisited}$. (iii) If one of these three cells are in $C_{NotVisited}$, a will move into this cell, and return to its normal behavior. If $c_{front} \in C_{Obstacles}$, a will preform the same procedure, starting at step i.

3.6 Cooperative Switching Algorithm for Foraging (C-SAF)

The C-SAF, described in [Zedadra et al., 2016], extends S-MASA from an exploration/search algorithm to a foraging algorithm. In the C-SAF algorithm the swarm consists of homogeneous agents, with a set capacity for carrying food. Each agents starts at the nest. Initially all agents will search for food sources. Whenever an agent is searching for food it utilize the S-MASA exploration algorithm.

Upon discovering a food source at cell c_f an agent a_0 does three things. As agents can only sense food in there current cell, a_0 is already in the cell containing food. First it deposits a recruitment pheromone in its left cell c_b(referred to as brown pheromone), creating the start of a recruitment trail. Second it picks up available food at c_f up to its carrying capacity. After this it follows the negative gradient, of either the pheromones deposited or the APF created by the S-MASA, back to the nest.

At each visited cell whilst tracking home to the nest, a_0 deposits a trail pheromone (yellow). Upon arriving at the nest a_0 deposits the food, before climbing the trail following the positive gradient and the yellow pheromone back to c_f. After reaching the c_f, a_0 picks up more food an follows the existing trail, using the negative gradient and the yellow pheromone, back to the nest.



(a) (b) (c) (d)

Figure 3.4: C-SAF: S=agent, N=nest, F=food. Integers represent APF values. (a): S-MASA behavior before food is discovered. (b): Food discovered, brown pheromone dispersed. (c): A second agent has extended the brow recruitment trail, and joined the food transportation. (d): Trail is removed by the agent depleting the food source.

Because of the vortex pattern, the next agent a_1 approaching c_f will be exploring the column or row of cells directly to the left of the row or cell explored by a_0. Thus as a_1 approach c_f it will move into cell c_b containing brown recruitment pheromone, seen in figure 3.4d.

After a_1 moves to c_b it deposits brown pheromone in its left cell, extending the recruitment trail. This behavior applies to all approaching agents. Thus every new agent exploring a row or column one cell further out from the previous agent extends the recruitment trail and allows for the recruitment of more agents.

After an agent a_n deposits brown pheromone, it follows the trail of brown pheromones until it reaches the food source. Here it joins the foraging procedure started by the a_0. This mechanic of depositing brown recruitment pheromones allows agents to cooperate on foraging the same food source. It constitute the behavioral difference between C-SAF and non-cooperatice C-SAF [Zedadra et al., 2016].

Eventually a food source is depleted. When this happens the agent depleting the food source is responsible for removing the trail. When agents pick up food they store the amount of food left at the source, denoted f. Upon depositing food at the nest agents check this value. If an agent a has an f equal to 0, it was the one depleting the food source. After depositing the food, a climbs the nest as normal, only it removes the yellow trail pheromone as it climbs. Upon reaching the depleted food source it follows and removes the brown recruitment trail before it returns to exploring using the S-MASA algorithm.

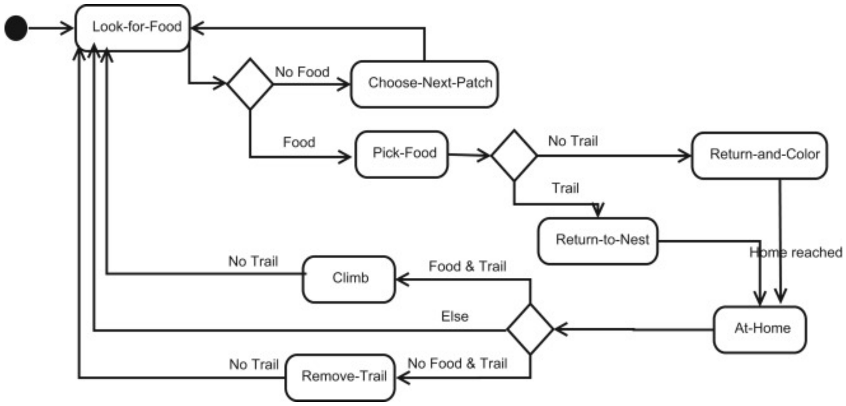


Figure 3.5: State diagram showing the possible states of a foraging agent in C-SAF and NC-SAF algorithms. Black circle is the initial state, white diamonds are decision points and rectangles are states. [Zedadra et al., 2016]

3.7 Adaption, assumptions and simplifications

In this sections adaptations, assumptions and simplifications are elaborated. These are in regards to the implementation of C-SAF as a base and reference algorithm, and the implementation of H-CAF.

Cell occupation and agent collisions

C-SAF [Zedadra et al., 2016] does not clearly state whether multiple agents can occupy the same cell at any time step. In this paper we argue that this should be allowed in simulation.

If the system is to be implemented on robotic agents, some mechanism for collision detection amongst agents need to be in place. In addition an effective method is needed for letting agents pass each other as they track along a trail in opposite direction. This thesis considers these to be mostly robotics engineering problems, and outside the scope of this research. As such multiple agents can inhabit the same cell at any time step.

Border cells

The environment described in C-SAF is a finite set of 4-connected cells, represented as a grid. As the number of cells in this grid is finite, it must have and outside border/edge.

Zedadra et al. [2016] does not clearly state the behavior of agents if they reach this border. In this implementation of both C-SAF and H-CAF this is solved by letting each cell along the edge of the board, border special border cells. This is detailed in section 3.4.

These cells are not part of any set of cells, reachable or not. Thus they are not part of the set of all cells C. They are however still observable to agents. Upon detecting a front cell c_{front} that is a border cell, agents will simply rotate to the right until $c_{front} \in C$.

3.8 Heterogeneous Cooperative Algorithm for Foraging (H-CAF)

H-CAF expands the capabilities of C-SAF by allowing it to work with a set of heterogeneous agents. It does this through the introduction of specialized "harvester" agents. To avoid confusion agents following the behavioral rules of the C-SAF algorithm are renamed "scouts".

Harvesters are agents that initially reside in the nest. Once a scout return to the nest with food from a source (following C-SAF behavior), they make an attempt at recruiting harvesters. If harvesters are successfully recruited, they follow the trail established by the scout, and join the foraging effort. This mechanic is inspired by the recruitment behavior seen in both natural bee colonies and ACO (section 2.3.1).

H-CAF is intended to be used on a set of morphologically heterogeneous agents. Meaning that scouts and harvester have different capabilities, as well as behavior. This means that in physical implementation scouts and harvesters will have a different construction. In simulation this can be achieved by giving the agents different sensory capabilities, carrying capacity and energy consumption.

Note that the system can function with morphologically homogeneous but behaviorally heterogeneous agents, as explored in section 4.3.3. This can be achieved by giving harvesters the same carrying capacity and energy consumption.

In addition to the inclusion of harvester agents, H-CAF introduces an optional behavioral rule for scout agents. This "left turn" behavior is introduced in section 3.9.1.

3.9 Scouts

Scouts are essentially agents following the behavioral rules of the C-SAF algorithm with some extensions. These extensions include the option to use the "left turn" behavior. In addition scout have the option of depositing APF values instead of evaporating pheromones. Note that in obstacle free environment, pheromones and APF have been shown to preform identical. The rest of the extensions relate to interaction with harvesters. Specifically the recruitment of harvester to help forage discovered food sources.

All these extensions are made in a manner that allow a swarm consisting only of scout agents to behave and preform identical to the C-SAF algorithm. This allows the heterogeneous H-CAF algorithm to be directly compared to C-SAF in an identical environment. This is simply achieved by running H-CAF with a population of only scouts. This is explored in chapter 4, where the two algorithms are compared.

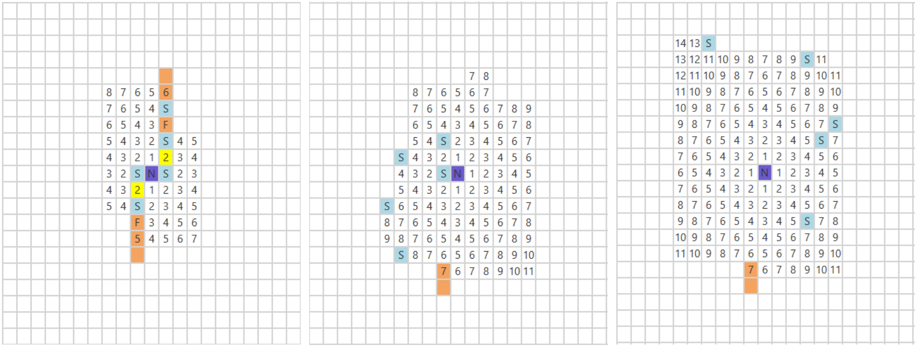
3.9.1 Left turn behavior

When S-MASA is deployed in an obstacle free environment, scouts will constantly expand $C_{Visited}$, once the vortex pattern is established. This means that at each time step all scouts at the wavefront will move into a cell in $C_{NotVisited}$ (as long as the border is not reached).

However, once a food source is discovered. the vortex pattern is somewhat disturbed. The recruitment of other scouts creates an asymmetry in the vortex.

If the number of scouts is high, and multiple food sources are foraged in the same time space, this can disrupt the pattern.

C-SAF solves this by restricting agents/scouts in the Choose-Next-Patch state (figure 3.5) to moving straight or right. Consequently if a scout moves along the wavefront and senses a front cell in $C_{Visited}$ it will continue straight until it senses a cell in $C_{NotVisited}$, either to its right. This translates to scouts moving thought the field of $C_{Visited}$ cells until it emerges at the wavefront on the other side, seen in figure 3.6.



(a) (b) (c)

Figure 3.6: (a): Two food sources are forage at the same time. (b): The vortex pattern is deformed, causing scouts/agents to explore cells in $C_{Visited}$. (c): Over time the pattern deformations are reduced.

To avoid this sub-optimal behavior where some scouts use valuable time restepping cells in $C_{Visited}$, H-CAF introduces an optional ”left turn behavior”. If active it allows scouts to turn left in certain situations. Namely if there is a ”left turn” in the wavefront, seen for the perspective of the scout.

Default scout behavior is to move straight if possible (e.i. no obstacle or border), but turn right if the right cell is not part of $C_{Visited}$. If left turn behavior is allowed the scouts do the following before the above mentioned behavior is executed: If both the front and right cells are in $C_{Visited}$, and the left cell is in $C_{NotVisited}$, turn left. This allows scouts to trace along the wavefront, even if the vortex pattern is disrupted.

3.10 Harvesters

In a homogeneous swarm of only scouts, scouts have to primary functions. Firstly they function as explorers looking for food. Upon finding food some agents change behavior to facilitate the need to transport food back to the nest.

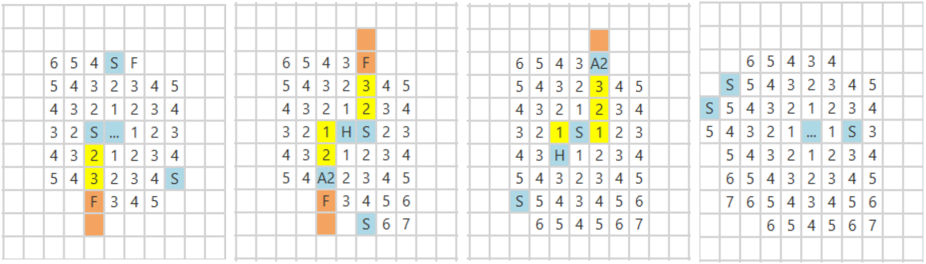
By depositing brown pheromones this process is improved. However, the population of scouts is now split between exploration and food transport. Lowering the swarms total ability to explore. By creating specialized harvester agents, that aid scouts once food has been found, they can effectively increase food transportation capabilities. Doing so allows more scouts to preform search/exploration.

As detailed in section 2.7.1, creating morphologically heterogeneous agents can also increase agents efficiency at preforming specific task. This opens the possibility of having harvester agents with a higher carrying capacity and/or a better capacity/energy ratio, than scouting agents.

3.10.1 Behavior

Harvesters can not function without the aid of scouts. At the beginning of a foraging scenario, harvesters are placed idly at the nest. They remain idle, while a sub-swarm of scouts begin there search for food according to C-SAF behavior.

After finding a food source a scout will begin transporting food home, creating a food trail back to the nest. Upon reaching the nest, scouts initiate a recruitment procedure, described in 3.11. Regardless of how many harvesters are recruited (if any), the scout continues according to C-SAF behavior. This is key to allowing a H-CAF swarm of only scouts to preform identical to C-SAF.



(a) (b) (c) (d)

Figure 3.7: H-CAF: S=scout, H=harvester, A=scout(s) and harveser(s). (a): Scout returns to the nest. It will deposit food an recruit harvester(s). (b): A second scout approaches from another source, it to will recruit. (c): Harvester removes trail after depleting food source. (d): Harvester go idle, and scouts return to the wavefront.

Harvesters recruited by a scout, will start climbing the same food trail as the scout recruiting them. Upon reaching the source they load food and return to the nest depositing food, much like the scouts do. As with scouts, harvester stores the amount of food left at the source. Generally (see 3.11.1) harvesters continue this process until the food source is depleted. At which point they return to the nest, assuming an idle state until recruited again.

3.10.2 Removing food trails

As mention harvesters return to the nest once the food source is depleted. However, there will always be one agent that deplete the food source.

In the case of a homogeneous swarm of scouts the agent that deplete the food source will first returns to the nest and deposit the food. Then it will climb the trail while removing it, and any brown pheromone trail linked to it. Other scouts finding an empty source, before these trails are removed, resume search behavior.

When harvesters are introduced, it introduces the possibility that a harvester is the one to deplete the food source. The issue is to make sure that scout behavior remains unchanged, while harvesters function as intended. Thus the role of removing trail falls on the harvester.

Therefore, after depleting a food source a harvester does the following. Instead of returning to the nest immediately, it follows the brow trail away from the depleted food source. It removes the brow trail, while at the same time extending the food trail leading back to the nest. Once all the brown pheromones are removed. The harvester returns to the nest, removing the trail as it descends it, see figure 3.7c.

3.10.3 Harvesters losing trail

Harvester may end up in a situation where they lose their trail. This can happen if they climb a trail to a depleted food source passing the scout that depleted it on their way. This will result in the scout removing the trail while the harvester is still away from the nest. A similar scenario occurs when another harvester removes the trail. To work around this harvesters losing their trail will follow the negative gradient back to the nest.

3.10.4 Brown pheromones with H-CAF

As described in [Zedadra et al., 2016], brown recruitment pheromones dramatically improves the foraging speed of C-SAF. In fact if scouts/agents are not allowed to deposit brown pheromones, the system is preforming non-cooperative C-SAF (NC-SAF). Zedadra et al. [2016] shows how C-SAF consistently outperform NC-SAF.

This performance increase comes as a result of scouts helping each other transport food, increasing transportation throughput. This will however temporarily lower the swarms exploration speed, as less scouts are available to explore.

In a swarm with harvester agents, this trade-off has already happened. Substituting some of the scouts for harvesters, increases transportation capabilities at the cost of exploration capabilities. Thus it is not certain that use of recruitment pheromone amongst scouts will increase performance in all heterogeneous swarms. The rationale being that the need for exploration trumps the need for transportation.

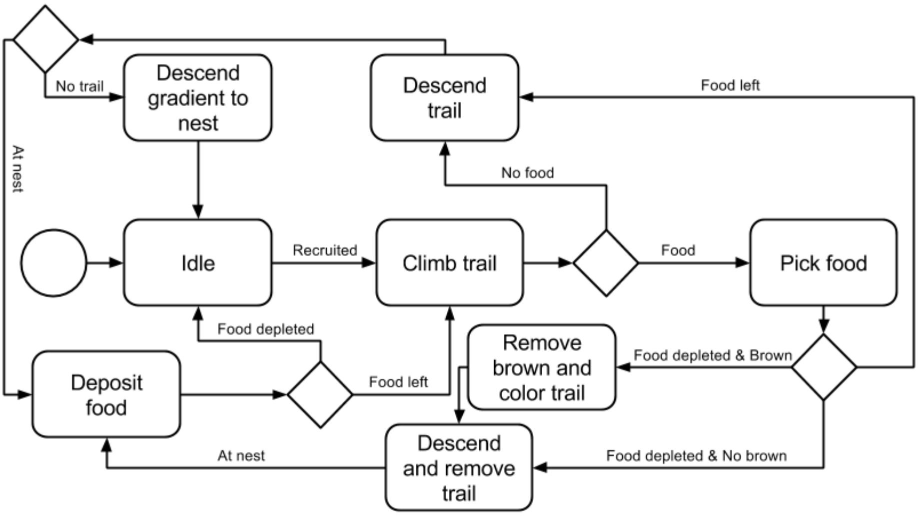


Figure 3.8: H-CAF Harvester state diagram

3.11 Recruitment

Fundamentally the recruitment of harvesters work by letting scouts recruit one or multiple harvesters as they deposit food at the nest. This is done by sharing the food trail with harvesters, thereby allowing them to forage the same food source. This is reminiscent of recruitment behavior seen in colonies of bees (section 2.3.1).

Scouting for food is preformed with a high level of parallelism. This can result in multiple food sources being foraged within the same time frame. As a consequence scouts returning from different food sources may wish to recruit harvesters within the same interval. Recruiting an unnecessary amount of harvesters will therefore reduce the number of harvesters available to other scouts. Additionally it can increase energy consumption, as some harvesters may be forced to return to the nest without a full load. Hence the performance of H-CAF is greatly affected by the recruitment strategy employed by the swarm.

After a food source s is discovered at time T_d, there will be a finite number of agents committed to depleting s. This set of agents, denoted A_s, contains at least one scout but can also contain harvesters. Every agent has a carrying capacity. Let c be the combined carrying capacity of A_s. Let s_f be the total amount of food at s at any time step.

Agents depleting a food source can be in one of two states (excluding: depositing, and loading). Either they are climbing the trail in search of the food source, or they are descending the trail to deposit their load at the nest. Thus we can divide A_s into two subsets. A_{Climb} containing all agents climbing the trail at the beginning of any time step $T_d + n$. A_{Desc} containing all agents returning to the nest at the beginning of $T_d + n$. Depositing and loading happens within a time step, thus $A_{Climb} \cup A_{Desc} = A_s$.

Let c_{Climb} be the combined carrying capacity of A_{Climb} at any time step and δ be the capacity mismatch of c_{Climb}. Then $c_{Climb} + \delta = s_f$, and $|\delta|$ should be as small as possible, ideally 0, at each time step.

Informally this means that the agents climbing the trail ideally should have a combined carrying capacity equal to the amount of food left at the source. However, achieving this without breaking the core principles of swarm intelligence, is not trivial. How does one achieve a small δ, in a fault tolerant way, only through local interaction?

3.11.1 Recruitment strategies

H-CAF propose two methods for recruiting harvesters, static and dynamic. Static recruitment works by having scouts recruit a constant predefined number of harvesters when returning to the nest. This parameter is named "recruitment size". Static recruitment can be executed continuously or only by the first scout (single).

Single static recruitment

With single static recruitment, only one recruitment attempt is preformed for each source. This is done by the scout discovering the source, as it returns to the nest after coloring the trail. The problem with this behavior is that it is highly dependent on the amount of idle harvesters at the time step the scout arrives at the nest. In a worst case scenario this can result in a scout a_s foraging a source s, arriving at the nest at T_n with 0 harvesters idle and eligible for recruitment. Then at $T_n + 1$ a large number of harvesters arrive at the nest and return to an idle state. As a_s has already attempted its recruitment these harvesters will stay idle (unless recruited by another scout form a different source) for the duration s is depleted. This wasts valuable resources.

Continuous static recruitment

To combat the issues with single recruitment, continuous recruitment was introduced. When continuous recruitment is enabled, every scout returning to the nest attempts to recruit harvesters. This include scouts returning from their second round trip who have already recruited harvesters to the same source at an earlier stage.

This proves a viable strategy. In combination with static harvesting, this dramatically decreases the combined idle time for harvesters. If "recruitment size" is set to a relatively small fraction of the total harvester population, one ensures that harvesters are better distributed on multiple sources. Largely avoiding scenarios where all the harvesters are committed to foraging a food source with less food than their capacity, while other sources are available.

Continuous static recruitment results in a pattern where δ usually starts as relatively large positive integer an gradually decreases, sometimes transitioning into a negative integer. Although this is not ideal it ensures a relatively good δ while harvesters are distributed among multiple resources, when these are discovered within the same time frame.

Dynamic recruitment

Dynamic recruitment strives to reduce $|\delta|$ over time by dynamically recruit and decommission harvesters as needed. To achieve this a mechanism to estimate c_{Climb} is needed. Dynamic recruitment solves this by having the nest store an integer value associated with each food trail currently existing. This integer value, denoted s_h, represents the number of harvesters in A_c.

Dynamic recruitment is always preformed continuously, meaning that each time a scout visits the nest it makes an attempt at recruiting harvesters (as long as the source is not depleted). As described in section 3.6, scouts maintain a value f corresponding to an estimate of the amount of food left at the source. Every time a scout deposit food it calculates the combined capacity of the subset of A_s that are harvester, using s_h. This capacity is denoted $c_{Harvesters}$. If h is individual harvester capacity, then $c_{Harvesters} = s_h \times h$.

Scouts also have a capacity, denoted s_s. A scout returning to the nest has no knowledge of the number of other scouts foraging the same source, only that it itself is foraging it. Therefore scouts attempt to recruit harvesters if the following is true: $f > c_{Harvesters} + s_s$. If this is true, then n harvesters are recruited, where $n = (f - s_s)/h$ rounded up to the nearest integer. If n idle harvesters are not available, n is equal to the number of idle harvesters. If harvesters are recruited, s_h is overwritten so that $s_h = s_h + n$.

Like scouts harvesters store a value f corresponding to an estimate of the amount of food left at the source. When a harvester returns to the nest, after depositing food it checks if $f < c_{Harvesters} + s_s$. If this is the case, more harvesters than needed are active, and the harvester decommissions it self by decrementing s_h and going idle.

Dynamic recruitment thus work by making sure that $c_{Harvesters} + s_s$ is as close to f as possible. $c_{Harvesters} + s_s$ is however not equal to c_{Climb}. $c_{Harvesters} + s_s$ is typically closer to c as $c_{Harvesters} + s_s$ is equal to c subtracting the capacity of other scouts foraging the same source. Consequently if recruitment pheromones are deactivated $c_{Harvesters} + s_s = c$. This means that if enough harvesters are present so that $n = (f - s_s)/h$, δ will be a positive integer.

Chapter 4

Experiments and Results

4.1 Experimental Plan

To help answer research RQ1 (section 1.2) the performance of the H-CAF (section 3.8) algorithm will be directly compared with that of the C-SAF (section 3.6) algorithm. As described in section 3.8, running H-CAF with a population of only scouts is equivalent to running C-SAF. Thus when experimenting with different ratios of scouts and harvesters, a population of only scouts constitutes the performance of C-SAF. This means that throughout this chapter, experiments run with only scouts serve as a performance reference, and as the primary point of comparison.

As detailed in section 3.8, an optional left turn behavior was introduced to improve scout performance. This is a behavioral change to the original homogeneous system (C-SAF), and thus will not alter harvester behavior. As such some initial testing of H-CAF, with and without this behavior, and in combination with other parameters, will be conducted.

After the homogeneous reference system has been tested using different parameters to establish best performance, the C-SAF and H-CAF can be directly compared. Both systems will be tested on maps of different size and with different food distribution and density. This should give valuable insight into the performance of the systems in different foraging scenarios.

There are no stochastic behavior rules for scouts nor harvesters. All maps and there corresponding food distribution (including coordinates) remain unchanged throughout the experiments. Furthermore all parameters are of a deterministic nature. Therefore the system as a whole is deterministic, an each scenario needs only be simulated once, to obtain comparable metrics.

4.2 Experimental Setup

4.2.1 Maps

The purpose of these experiments is to evaluate the performance of the C-SAF algorithm against the performance of H-CAF in different foraging scenarios. This will primarily be done using a collection of 8 obstacle free maps. These maps have different food distribution and density, allowing the systems to be tested in different scenarios. Three of the maps are of size 100x100 and the remanding five are of size 1000x1000.

Table 4.1 presents the maps as well as a short description of their food distribution. All maps have one nest at approximately the center of the map. In this table each map is also given an id. This id will be referenced as data is presented.

On maps containing more than one food source, coordinates of food sources were generated at random before experimentation commenced. On maps containing only one food source, this was placed at the cell were the highest equal Manhattan distance from nest to source and from source to the closest boarder was achieved. This to avoid scenarios where the only food source is right next to the boarder or nest, severely reducing the need for either search or transportation. All food source coordinates remained unchanged throughout the experiments.

id	Name	Food distribution
a	NoObs100x100_Food1(1000)	Food sources: 1 Food amount: 1,000 units Total food: 1,000 units Food density: 0.1 Size: 10,000 cells
b	NoObs100x100_Food10	Food sources: 10 Food amount: 100 units Total food: 1,000 units Food density: 0.1 Size: 10,000 cells
c	NoObs100x100_Food10(300)	Food sources: 10 Food amount: 300 units Total food: 3,000 units Food density: 0.3 Size: 10,000 cells
d	NoObs1000x1000_Food1(1000)	Food sources: 1 Food amount: 1,000 units Total food: 1,000 units Food density: 0.001 Size: 1,000,000 cells
e	NoObs1000x1000_Food10	Food amount: 100 units Total food: 1,000 units Food density: 0.001 Size: 1,000,000 cells
f	NoObs1000x1000_Food10(1000)	Food sources: 10 Food amount: 1,000 units Total food: 10,000 units Food density: 0.01 Size: 1,000,000 cells
g	NoObs1000x1000_Food100	Food sources: 100 Food amount: 100 units Total food: 10,000 units Food density: 0.01 Size: 1,000,000 cells
h	NoObs1000x1000_Food100(1000)	Food sources: 100 Food amount: 1,000 units Total food: 100,000 units Food density: 0.1 Size: 1,000,000 cells

Table 4.1: Obstacle free maps: Denoted with id and there food distribution. "Food sources" represents the number of cells containing food, and "Food amount" represent the number of units in each of these cells at T_0."Food density" is total food divided by the total number of cells.

4.2.2 Parameters

The system can be run with a host of different parameters. Some of these have a default value that are used throughout the experiments, unless specified otherwise. Table 4.2 gives an overview of these parameters, and a description of their effect on the system.

Parameters	Description	Default
Agents	The total number of agents deployed.	100
Scouts	The total number of scouts. If this is equal to the total number of agent, the swarm is homogeneous, and H-CAF is reduced to C-SAF.	
Harvesters	The total number of harvester. Harvesters can not function without any scouts, thus the number of harvesters is always less then the total amount of agents.	
Brown/No brown	Dictates whether scouts can diffuse brown recruitment pheromone to attract other scouts to a food source.	
Left	Controls whether scouts can rotate left upon encountering a ”left turn” in the wavefront.	true
Scout capacity	Max units of food a scout can carry.	1
Harvester capacity	Max units of food a harvester can carry.	5
Scout energy	Units of energy a scout expends by moving from one cell to another.	1
Harvester energy	Units of energy a harvester expends by moving from one cell to another.	1
Recruitment	Strategy used by scouts to recruit harvesters in a foraging task.	dynamic

Table 4.2: The parameters that are used and modified in the experiments

4.2.3 Performance metrics

Throughout most of the experiments two primary performance metrics will be used, foraging speed and energy use. All experiments are performed by having a population of agents forage units of food. This is done through simulation in discreet time steps. At each time step an agent can move at most once. If it chooses to do so it will expend units of energy. As a base, agents expend one unit of energy per move. In different scenarios agent may expend more energy.

Foraging speed will be measured as time of completion. As defined in section 3.4, time of completion is the time step at which the last unit of food is deposited at the nest. Energy efficiency will be measured in energy to food ratio (Energy/Food). Meaning how many units of energy are consumed by the entire swarm for each unit of food foraged (e.i deposited at the nest).

4.3 Experiments

This section detail the different stages of the experimentation, varying the parameters and maps described in sections 4.2.1 and 4.2.2. The experiments are presented in chronological order. Some experiments build upon the results found in earlier stages. Meaning they will be utilizing well functioning parameters found in earlier stages.

The experiments will be divided into three main stages. First H-CAF with a homogeneous swarm of only scouts will be tested with different parameters to establish what yields best performance.

Second a a broad comparison of C-SAF (using these parameters) and H-CAF using different ratios of scouts and harvester will be conducted. This is done to investigate how different ratios of heterogeneous agents affect the swarms performance. In addition it will establish well performing and robust agent ratio.

Third the systems will be compared by varying harvester energy consumption and carrying capacity. This will shed light on what carrying capacity and energy consumption a physical implementation of H-CAF must achieve to outperform a C-SAF.

4.3.1 Left turn behavior and brown recruitment pheromones

This experiment aims to investigate the impact of the left turn behavior. As this only affects the exploration behavior of scouts, a homogeneous swarm consisting of only scouts is employed throughout these simulation.

In the original C-SAF algorithm (3.6), performance is severely reduced when agents (i.e. scouts) can not recruit each other by dispersing brown recruitment pheromones. To see if this still holds true when left turn behavior is introduced the system is tested varying two parameters. These are the inclusion or denial of the left turn behavior and brown recruitment pheromones.

The remaining parameters are set to default, meaning the swarm will consist of a population of 100 scouts. This gives a total of 4 parameter permutations that are tested across different maps.

4.3.2 Heterogeneous agent ratio

When one employs a swarm of heterogeneous agents the complexity of the swarm increases. One of key aspect of this complexity concerns the ratio between the different types of agents in the swarm. Different ratios will necessarily change the performance characteristic of a swarm. This experiment seeks to explore how different ratios of scouts and harvesters affect the performance of the swarm. It also seeks to explore if any performance trade-offs occurs as the ratio is changed. Different maps will be tested to see how size and food distribution affect performance.

Finally all these scenarios will be tested with and without brown pheromones. In the C-SAF algorithm the use of brown pheromones is highly advantageous as it allows scouts to help each other transport food. This is at the cost of exploration speed. Thus it is interesting to explore if this continues to be an advantageous strategy when harvesters contribute to food transport.

Brown	Ratio (Scouts:Harvesters)									
true	100:0	90:10	80:20	70:30	60:40	50:50	40:60	30:70	20:80	10:90
false	100:0	90:10	80:20	70:30	60:40	50:50	40:60	30:70	20:80	10:90

Table 4.3: The different parameters used to test heterogeneous ratios.

Simulations will be run on all obstacle free maps seen in table 4.1. On each map the system will be run with 10 different ratios. First with the use of brown recruitment pheromones, and then without the use of recruitment pheromones.

All runs will be performed with 100 agents. The first run will be performed with 100 scouts, constituting the C-SAF reference. Runs will then be performed in intervals, where each successive interval will have a reduction of 10 scouts and an increase of 10 harvester. This change in ratio will continue until a population of 10 scouts and 90 harvesters is reached. The system can not function without any scouts, as such a run with only harvesters is unnecessary. Simulation will thus be performed on each map using the parameters seen in table 4.3.

4.3.3 Harvester capacity and energy consumption

The performance of H-CAF turns out to be highly dependent on the carrying capacity of harvesters. As discussed in section 2.7.1, the incentive to create H-CAF is based on the notion that morphologically different agents can perform different tasks at different efficiency.

However as this thesis explores algorithmic control structures for a morphologically heterogeneous swarm through simulation, this efficiency can be manipulated freely. Consequently there will always exist some parameter settings, relating to energy efficiency or capacity, that will favour a heterogeneous ratio, and vice versa.

In section 4.3.2, these parameters (e.i. scout energy and capacity, harvester energy and capacity) are set to what the author considers a fair reference point. However these parameters can not truly be justified without a physical implementation. Therefore this section will explore foraging performance with different parameter settings.

Scout parameters will remain at their default values as a base reference, while harvester parameters are varied. This means that experiments in this section will modify harvester capacity and energy consumption. We will explore how this variation affects the swarm at different heterogeneous ratios, and how they compare to C-SAF.

While foraging time is only dependent on carrying capacity for any given ratio, energy efficiency is affected by both carrying capacity and energy consumption.

Morphological homogeneity

When H-CAF is executed with different carrying capacity and/or energy consumption for scouts and harvesters, it simulates morphologically heterogeneous agents. When the carrying capacity and energy consumption is equal, the swarm can be considered morphologically heterogeneous. This is because harvester agents can now be created by changing the behavior of scouts. Thus the swarm is behaviorally heterogeneous.

An experiment will be performed where harvester capacity and energy consumption is set to 1, second column table 4.4. H-CAF will be simulated on all maps seen in table 4.1, using agent ratios described in 4.3 with brown pheromones.

Exploring harvester capacity with proportional energy use

Keeping the 1:1 relationship between harvester capacity and energy use, higher parameter settings will be explored. This will investigate foraging performance as capacity is increased, while energy consumption is also increased dramatically. Using the ratios described in 4.3 with brown pheromones, the parameters seen in table 4.4 will be simulated on all maps in table 4.1.

Capacity	1	5	10	50	100
Energy	1	5	10	50	100

Table 4.4: Different harvester capacities and energy consumptions, keeping a 1:1 ratio.

Exploring other harvester capacities and energy consumptions

Finally performance is explored at different proportion of harvester capacity to energy consumption. This is done to explore how different proportion affect energy efficiency. The parameters seen in table 4.5 will be used.

Capacity	1	5	5	5	10
Energy	1	1	2	4	4

Table 4.5: Different harvester capacities and energy consumptions

4.4 Results

This is a presentation of a subset of the results. All the results can be found as text files on github:
https://github.com/kyrrelm/MasterThesis/tree/master/Thing/savedResults

4.4.1 Left turn behavior and brown recruitment pheromones

Table 4.6 shows the performance of a homorogeneous swarm consisting of 100 scouts when the inclusion of left turn behavior and brown pheromones are either allowed or denied. As seen in the table performance is dramatically reduce when recruitment is not allowed, both with and without left turn. By allowing agents to turn left upon sensing a "left turn" in the wavefront the performance is either improved or remain unchanged. This was the case for all scenarios using these parameters. Therefore the rest of the test cases were performed with the inclusion of the left rule.

Map	No brown, No left	No brown, Left	Brown, No left	Brown, Left
a?	19372	16242	1750	1676
b?	182340	151600	56368	16688
c?	1002250	1002250	13896	13896
d?	1495656	1465600	102126	102126

Table 4.6: Forage completion time with 100 scouts. With and without brown recruitment pheromone and left rule. Maps are as follows, a: 100x100, food 10(100); b: 1000x1000, food 10(100); c: 1000x1000, food 1(1000); d: 1000x1000, food 10(1000)

The performance of a homogeneous swarm of scouts is severely reduce if recruitment pheromones are not allowed. Because of this, brown recruitment pheromones were enabled if the swarm consisted of only scouts in all subsequent experiments. This means that even when the performance of heterogeneous swarms are tested without recruitment pheromones, the homogeneous reference still uses recruitment pheromones (as in table 4.7). This is done so that the reference performers identical to C-SAF, offering fairer comparison.

4.4.2 Default parameters

Foraging time, with and without recruitment pheromones (brown)

Using the default parameters and varying the ratio of homogeneous agents yielded interesting results. Across all maps seen in table 4.1, some ratio of heterogeneous agents outperformed a homogeneous swarm of scouts (C-SAF) on both primary performance metrics, e.i. foraging speed and energy use.

Map	Brown	No Brown	Density
a	40:60	30:70	0.1
b	50:50	30:70	0.1
c	30:70	20:80	0.3
d	60:40	50:50	0.001
e	90:10	70:30	0.001
f	30:70	30:70	0.01
g	50:50	40:60	0.01
h	20:80	10:90	0.1
Avg.	46:54	35:65	

Table 4.7: The best ratio of scouts to harvester (scouts:harvester) on various maps, with and without brown recruitment pheromones. As well as food density on the respective maps.

When different ratios are compared using "time of completion", the best performing ratio vary from map to map. This can be seen in table 4.7. By analysing this data a trend emerges. Higher food density seem to favour harvester heavy ratios, while lower food density tend to pull the best ratio in the direction of more scouts.

This can be seen as map *d* and *e* are the only maps with a favoured ratio of more scouts than harvesters. The highest ratio of scouts is observed on map *e*, which has the same amount of food, the same food density and size as *d*. However *e* has more food sources than *d* by one order of magnitude.

Brown/Maps	a	b	c	d	e	f	g	h	Avg.
true	9/9	8/9	8/9	7/9	4/9	9/9	8/9	9/9	7.75
false	9/9	8/9	8/9	6/9	4/9	8/9	8/9	9/9	7.5

Table 4.8: Number of heterogeneous ratios that outperformed a homogeneous swarm of scout (C-SAF), based on time of completion.

On average 7.75 out of 9 heterogeneous ratios outperformed a population of only scouts, when recruitment pheromones were used. When recruitment pheromones were disabled this changed to 7.5 out of 9. This can be seen in table 4.8.

Using the same ratios as described in 4.7 an average improvement in foraging speed of around 50% is achieved, seen in table 4.9. As can be seen the removal of recruitment pheromones improve best achieved foraging speed in some scenarios, and worsens it in others. Overall removal of recruitment pheromones averages out to a slight improvement in foraging speed, using these scenarios and parameters.

Map	Scouts	Brown	No Brown
a	1246	498 (60.1%)	460 (63.1%)
b	1676	802 (52.2%)	692 (58.7%)
c	4240	1492 (64.9%)	1546 (63.5%)
d	13896	9040 (35.0%)	9750 (29.8%)
e	16688	15036 (9.90%)	14204 (14.9%)
f	102126	44320 (56.6%)	42104 (58.8%)
g	89522	43288 (50.0%)	32428 (63.8%)
h	803698	212386 (74.6%)	188820 (76.5%)
Avg.		50.3%	53.6%

Table 4.9: Time of completion and percentage improvement on various maps using: A heterogeneous swarm (only scouts). Using the best ratio of heterogeneous agents, with and without brown recruitment pheromones (as seen in table 4.7).

Energy efficiency and distance

Using the same scenarios and parameters used to compare foraging time, we can compare energy efficiency. This is measured in unites of energy consumed per unit of food deposited at the nest. Using the default carrying capacity and energy consumption all combinations of heterogeneous ratios, with and without recruitment pheromones, H-CAF outperformed C-SAF (with left turn).

In addition, with only one exception, all scenarios achieved better energy efficiency as the ratio of harvesters was increased. This can be seen in figure 3. As the number of harvester increase the energy to food ratio decreases. Proportional results were observed in all scenarios. On map *c* (with pheromones) a slight energy increase was seen from 100:0 to 90:10 before energy efficiency followed the described pattern seen in figure 3.

Of course these results are highly dependent on the carrying capacity and energy consumption of harvester. Consequently varying these parameters is examined in section 4.4.4.

However, using an energy consumption of 1 unit for each move operation performed by both scouts and harvesters has an advantage. In addition to serve as a reference point for energy consumption it doubles as a measurement of the combined distance traveled by the swarm.

Thus we can conclude that with a carrying capacity of 5 units per harvester (1 for scouts), the combined distance traveled by the swarm decreases as the harvester ratio increase. This is true regardless energy consumptions, with the exception described above.

4.4.3 Foraging time at different capacities

When measuring foraging time with respect to capacity, energy use can be ignored.

Figure 4.1 shows time of completion on map *g*, using different harvester capacities. Here we can see that as harvester capacity increase time of completion decreases.

At a capacity of 1, little change to performance is seen. Time of completion rises slowly, but increases as the harvester ratio becomes more extreme. "Plot 5" in figure 4.1 is the same as as plot *g* in figure 2.

At capacities larger than 1, foraging time is significantly reduced. Higher capacities result in faster foraging, as can be expected. As the harvester ratio becomes more extreme, foraging time increases. This trend is observed on all maps.

On all map seen in table 4.1, swarms with 10 to 60 harvesters and a capacity of 5 or higher achieve a foraging time considerable lower than the homogeneous reference (100 scouts).

However the performance of the more extreme ratios (70-90 harvesters) differ. Some achieve a foraging time higher than the homogeneous reference, others archive a lower time. Never the less, average foraging time across all maps was significantly reduced for all heterogeneous ratios with a capacity of 5 or higher. This can be seen in figure 4.2

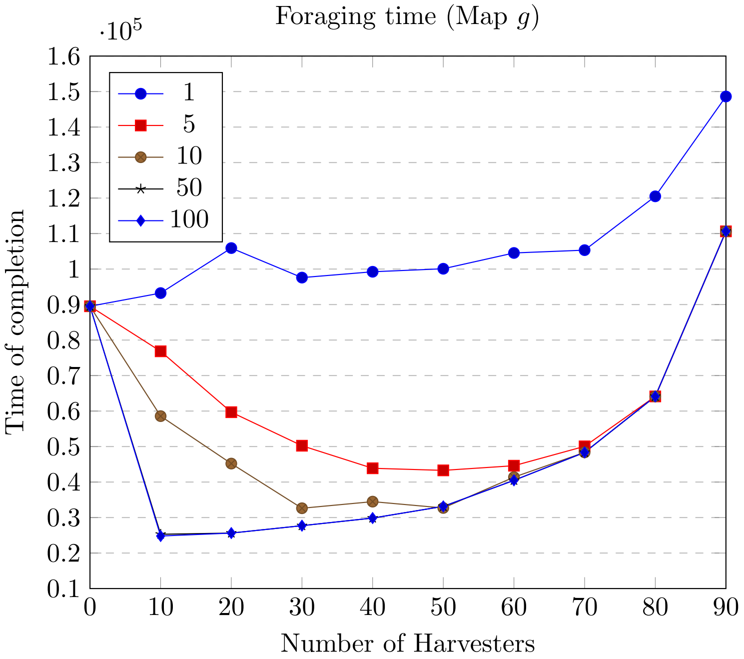


Figure 4.1: Time of completion: Using different harvester capacity on map g

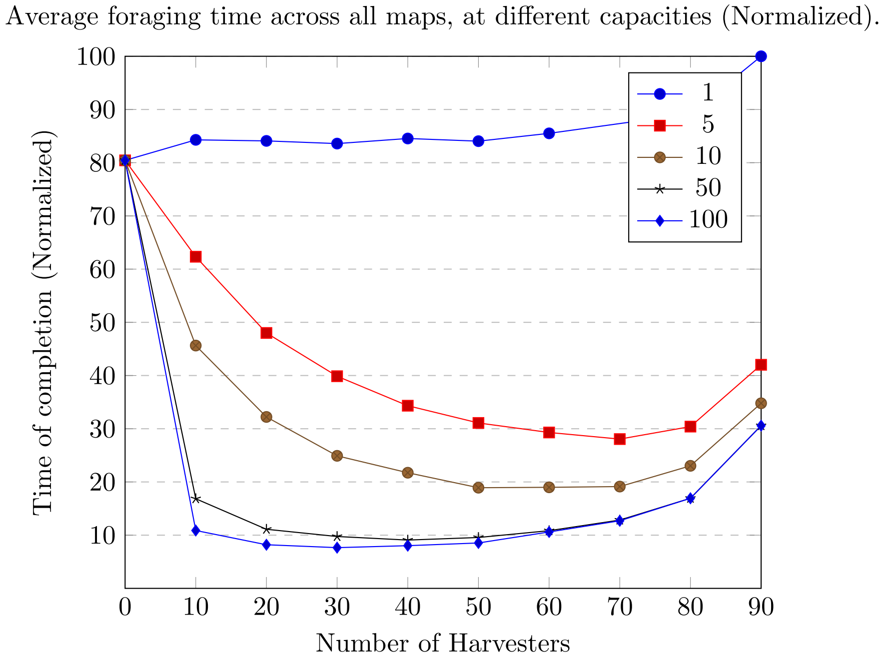


Figure 4.2: Average foraging time across all maps, at different capacities (Normalized)

4.4.4 Energy efficiency at different harvester capacities and energy consumptions

Energy efficiency at varying capacity, maintaining a 1 to 1 relationship between energy consumption and capacity

Figure 4.3 shows how energy efficiency evolves as harvester capacity and energy consumption is increased proportionally to maintain a 1 to 1 ratio. The specific ratios are presented in section 4.3.3.

Although energy use differ from map to map, increased capacity does not seem to have a large effect on energy efficiency when the "energy:capacity" ratio is maintained. This tight correlation between different versions of the same ratio, seen in figure 4.3, is observe on all maps in table 4.1.

Across all maps the same pattern of an increase in energy consumption for the lower ratio of harvester (10-30) can be seen. After this spike, energy consumption moves towards the consumption of a homogeneous swarm of scouts.

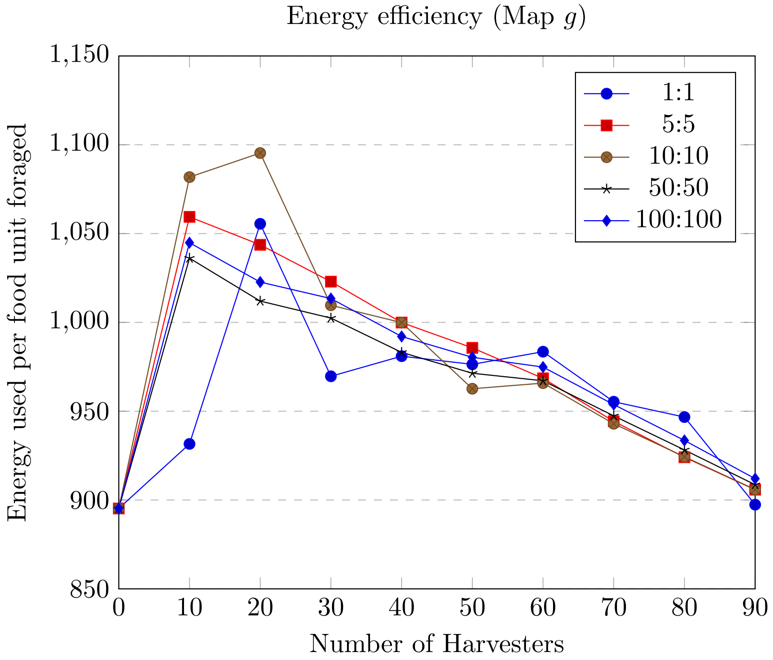


Figure 4.3: Energy/food ratio: Using different harvester capacity/energy settings on Map g, with a 1 to 1 ratio. (capacity:energy)

Energy efficiency at different proportions of "harvester capacity/energy consumption"

Figure 4.4 illustrates how energy efficiency is affected by different proportions of "harvester capacity/energy consumption". Here we can see that in general, as harvester capacity becomes proportionally higher than energy consummation, the swarm is more energy efficient. If capacity is higher than energy consumption, any ratio of scouts/harvesters outperforms C-SAF, with one exception. In general we can also see that energy efficiency improves as the ratio of harvesters is increased.

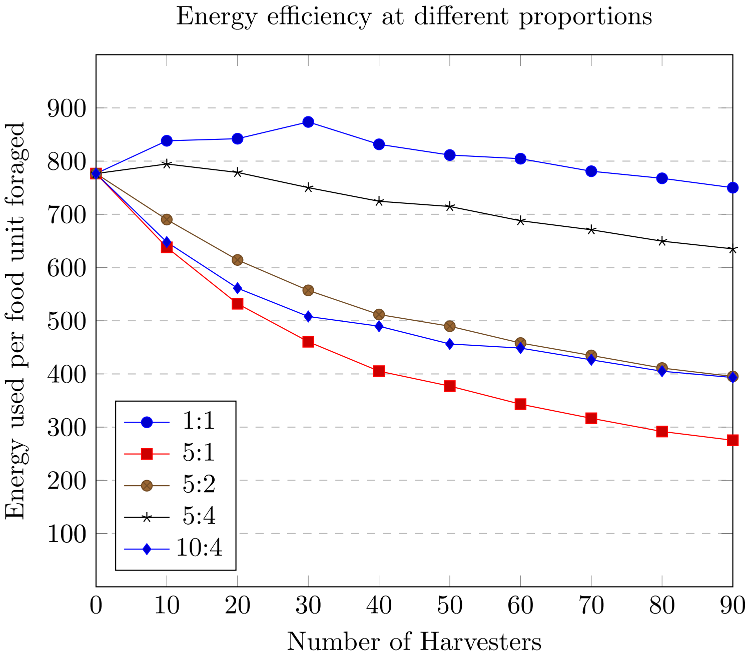


Figure 4.4: Average Energy/food ratio across all maps: Using different proportions of harvester capacity/energy consumption (capacity:energy).

Chapter 5

Evaluation and Conclusion

5.1 Discussion

In this thesis, C-SAF has been compared with H-CAF across 8 different maps. These maps have been of different size with different food distributions. This has been done in an effort to compare the two systems in different foraging scenarios.

5.1.1 Left turn behavior

Before testing the system with heterogeneous agents, H-CAF was tested with a population of only scouts. This was to see how the inclusion of a left turn behavior would affect performance. As seen in table 4.6, the inclusion of left turn behavior either improved performance or it remained unchanged.

As this is a change to the original C-SAF behavior, this is best seen as an improvement in homogeneous performance. As such C-SAF with the new left turn behavior was used as comparison for H-CAF. The rational being that if improvement to scout performance was only included in H-CAF, this could be confused with an increase in performance due to a heterogeneous population.

Although the left rule behavior improves performance in obstacle free environment it has not been tested on maps. The inclusion of this behavior doe not break the system on maps with obstacles. However, it is unclear if the same performance increase can be seen in scenarios with obstacles.

Mangler

5.1.2 Performance comparison of the C-SAF and H-CAF with default parameters

When comparing H-CAF with C-SAF using default parameters, most harvester/scout ratios improved both energy efficiency and time of completion across all maps. This can be seen in table 4.8, where on average over 7/9 heterogeneous ratios outperformed C-SAF based on time of completion.

However, knowing which ratio is optimal before deploying the system is not possible without any a priori knowledge about the environment. This can be seen in table 4.7, where the best performing ratios differ considerably. Although most ratios perform better than C-SAF, without any a prior knowledge one can not be certain that the ratio chosen will preform better.

Figure 4.2 presents the average foraging time across all maps, using different harvester capacities. By examining the performance when the capacity is 5 (default), we can study the average performance of different ratios of scout/harvester, with respect to time.

Here we can see that all ratios of harvesters/scouts on average perform better than C-SAF. On average a population of 30 scout and 70 harvesters had the lowest time of completion, being 75% faster than C-SAF. Thus it seems likely that H-CAF will be faster than C-SAF in most scenarios, if harvester capacity is 5 times greater than scout capacity.

As seen in figure 4.4, the average energy efficiency also improved as the proportion of harvesters increased.

The effect of disabling recruitment pheromones, on heterogeneous population

Recruitment pheromones were originally introduced in C-SAF to let scout help each other forage a resource. When harvesters assume much of this responsibility it is interesting to see if recruitment pheromones still improve performance.

The underlying hypothesis was that H-CAF will perform better without the use of recruitment pheromones, when the proportion of harvesters is high. The idea being that the remaining scout are needed more to explorer the environment.

Table 4.7 presents the improvement in foraging time with and without recruitment pheromones. Here we can see that the best agent/harvester ratio with recruitment pheromones, had an average improvement of 50.3%. The best agent/harvester ratio without recruitment pheromones improved foraging time by 53.6%.

However as can be seen in table 4.8, on average 7.5/9 heterogeneous ratios performed better than C-SAF, when recruitment pheromones were disables. When recruitment pheromones were enabled an average of 7.75/9 outperformed C-SAF.

This seem to indicate that disabling recruitment pheromones in H-CAF can improve peak performance at the cost of narrower range the range of advantageous ratios.

The difference in performance is however pretty small. One possible reason for the lack of performance improvement might be due to the current recruitment mechanic. In H-CAF only scouts can recruit harvesters. When recruitment pheromones are disabled only one scout will forage each source. Consequently there will only be one "recruiter" per source.

In contrast, multiple scouts will forage the same source if recruitment pheromones are activated. This will result in more recruitment attempts lowering the idle time of harvesters. Therefore it would be interesting to see how improved recruitment would affect the results.

Performance at a 1:1 ratio of harvester capacity and energy consumption

When comparing H-CAF and C-CAF at a harvester capacity of 1 and a energy capacity of 1 we see that C-SAF performs slightly better that H-CAF, with respect to foraging time. When compared on energy efficiency the same can be said. This is expected as this constitutes running H-CAF with morphologically homogeneous, but behaviorally heterogeneous agents. This is not what H-CAF is designed for, while C-SAF is designed to take advantage of morphologically homogeneous agents. However, the performance difference is small.

Form figure 4.3 we can see that energy efficiency is conserved at higher energy consumptions, as long as capacity is increased to maintain a 1 to 1 ratio. As can be seen in figure 4.2, higher harvester capacity dramatically decrees average foraging time. This shows that H-CAF will likely consistently outperform C-SAF with regards to time of completion, if harvester have a higher capacity.

Performance at different ratios

For figure 4.4 we can see that consistently outperforms C-SAF at different energy/capacity ratios, with regards to energy efficiency. This is true as long as harvester capacity is higher than energy consumption. This is not unlikely as discussed in section 2.7.1.

H-CAF implemented in swarm robotics

H-CAF faces the same challenges as C-SAF, if it is to be implemented on robots. Such an implementation is far away, but one can still discuss the parameters used.

Given that H-CAF has a better average energy efficiency even at a capacity of 5 and a energy consumption of 4, it seems likely that H-CAF can improve foraging in physically implemented systems.

Furthermore even at a 1:1 ratio of harvester capacity to energy consumption, H-CAF achieved a energy efficiency comparable to C-SAF. However, H-CAF has a much lower time off completion when capacity is increased. Thus even at a 1:1 ratio, one can a argue that H-CAF outperforms C-SAF, as energy consumption is almost the same, while time of completion is dramatically improved.

Obstacles

Although H-CAF has not been formally tested on maps containing obstacles, there is no reason to believe that this would change performance when compared with C-SAF. This is because obstacle avoidance in H-CAF is identical to obstacle avoidance in C-SAF.

As Zedadra et al. [2016] points out, the current obstacle avoidance is somewhat crude, and one of the weaker aspects of the algorithm. Hence it seems likely that better obstacle avoidance would be beneficial, both for C-SAF and H-CAF. This is discussed in section 5.3.

However, as obstacle avoidance is strictly related to the behavior of scouts, one can argue that it has nothing to with the introduction of harvesters. Thus it can be considered outside the scope of comparing H-CAF with C-SAF.

Dynamic recruitment is dependent on central storage in the nest. This makes the nest a central point of failure, but it is that be anyway. Dynamic recruitment: Bees indicate food quality by their waggle dance

5.1.3 State-of-the-art

Part of the goal of this thesis is explore what existing solutions exist to the foraging task. This is answered by investigating state-of-the-art swarm intelligence foraging systems. Thw research questions were formulated , RQ1 and RQ2. They were formulated as follows:

Research question 1 (RQ1): What is the state-of-the-art in morphologically heterogeneous swarm intelligence systems used in foraging?

Research question 2 (RQ2): What is the state-of-the-art in swarm intelligence systems utilizing pheromones for foraging related tasks? (e.g. foraging, exploration, surveillance)

To answer these questions a Structured literature review was conducted 2.5. Thought this a document collection of "core studies" was retrieved. These studies were analysed and discussed in sections 2.6 and 2.6.3.

This process revealed that little research to date, combines morphologically heterogeneous agents with pheromone foraging algorithms. This helped motivate the development of a new swarm control algorithm (see 2.7), to answer research question RQ3.

The SLR revealed a mirage of different approaches to foraging or foraging related tasks. One of the issues with these solutions is that they solve different variations of foraging relate scenarios.

These variations are not unjustified, as researchers are motivated by different applications. Thus the requirements of real world applications, motivate foraging related research, vary. This is reflected in how researchers define the foraging task they aim to solve. As a result the foraging scenarios used for experimentation diverge, both in their setup and constraints. Consequently it is difficult to compare research quantitatively.

5.2 Contributions

In this thesis a new foraging algorithm is presented, called H-CAF. H-CAF serves as a control algorithm for a swarm of heterogeneous agents solving a foraging task. H-CAF combines elements of foraging algorithms seen in the literature. This is done in a way that, to the best of the authors knowledge, has not been done before.

H-CAF introduces harvester agents, their recruitment strategy is inspired by the recruitment behavior in bee colonies. It combines this with the core mechanics of C-SAF [Zedadra et al., 2016], in a way that does not disrupt the behavior of agents following the original behavior of C-SAF.

H-CAF achieves promising results when compared with C-SAF, even with conservative parameters. As such it seems likely that it can function well as a control mechanism for morphologically heterogeneous agents.

This thesis demonstrate that it is possible to combine pheromone based foraging with morphologically heterogeneous agents in a way that improves performance.

In addition it has introduced a behavioral rule improving the performance of C-SAF in obstacle free environment, dubbed ”Left turn behavior”. This behavioral rule is also compatible with the H-CAF algorithm.

5.3 Future Work

In this section, ideas related to future development of the H-CAF algorithm is presented. For instance, would the inclusion of speed as a parameter be interesting.

5.3.1 A framework for comparing foraging algorithms

By reviewing the related literature it has become apparent that there is a lack of comparable experimentation. Different research groups solve different variations of a foraging task. Granted, some of this research is not comparable by the same performance metrics, as they aim to solve engineering challenges [Hrolenok et al., 2010; Ducatelle et al., 2011].

However, research conducted on control algorithms for foraging tasks, should strive to be comparable. As these systems are often motivated by different applications their performance in different scenarios will differ.

If a collection of benchmark scenarios is established, systems can compared in a common framework. This can be accomplished by standardizing a large set of scenarios, varying: recourse distribution, agent failure, environment size, obstacles, constraints, etc.

A new system need not be capable of solving all these scenarios, but by demonstrating what scenarios it can solve, it strengths and weaknesses are revealed. Furthermore, systems that solve an overlapping set of scenarios, can be directly compared, using a standard set of performance metrics.

5.3.2 H-CAF continuations

Parameter optimization through genetic algorithms (GA)

As can be seen in table 4.2, H-CAF has a high dimension of adjustable parameters. When combined with different maps (table 4.1), this results in a high number of testable permutation.

Consequently it is difficult to determine what parameters, and in particulate what agent ratio, perform best across different scenarios. Hence the use of parameter optimization through GA is an interesting idea.This is easily adaptable, as performance metrics across all scenarios can serve as and individuals fitness. Parameters serve as the genome.

Obstacles

In the development of H-CAF, much effort was put into improving the obstacle avoidance present in the C-SAF algorithm (section 3.5.1). Unfortunately this was very time consuming. Although H-CAF would benefit from improved obstacle avoidance, it not important when comparing H-CAF with C-SAF. In fact, having the two system handle obstacles identically, is of greater importance. This isolates differences in performance to the introduction of heterogeneous agents.

As obstacle avoidance is not directly related to any of the research questions, development of this was put aside. However, some interesting ideas are worth considering for future work. Upon encountering an obstacle, instead of tracking around the obstacle in the direction of visited cells, scouts can do the following:

- Track around the obstacle in the direction of unvisited cells, until encountering a visited cell.
- While tracking around an obstacle, write APF values as new cells are explored, but mark these cells as unsafe, using a new type of pheromone.
- After encountering a visited cell, retrace over the unsafe cells, and remove the "unsafe pheromone". Rewrite the APF value of the current cell $apf_{current}$, if the APF value of previous cell $apf_{prev} < apf_{current} + 1$.

This will encapsulate any obstacle in the APF field, and place them behind the exploration front. If scouts follow the "left turn behavior" (section 3.9.1), they will no longer encounter the obstacle. Furthermore, since APF values are rewritten as the "unsafe pheromones" are removed, the APF should be optimal.

Recruitment

As explained in section 3.11, the recruitment mechanic is suboptimal. Although it is not clear how optimal recruitment can be achieve, the system can benefit from improving this mechanic.

In the current system only scouts can recruit harvesters. Allowing harvesters to recruit each other might improve performance. Building on this idea, it would be interesting to see the effect of letting recruited harvesters "depart" from the nest at different time steps. This would have the benefit of distributing recruitment attempts, done by harvesters, across different time step.

Another interesting idea is to let distance to a food source be a factor in recruitment. This idea is inspired by the recruitment behavior of bees, where the fitness of a food source is determined by its quality and distance to the nest (2.3.1).

Bibliography

Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., and Honiden, S. (2006). Performance of pheromone model for predicting traffic congestion. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 73–80. ACM.

Arkin, R. C. (1998). *Behavior-based robotics.* MIT press.

Balch, T. (1999). The impact of diversity on performance in multi-robot foraging. In *Proceedings of the third annual conference on Autonomous Agents*, pages 92–99. ACM.

Barraquand, J., Langlois, B., and Latombe, J.-C. (1992). Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241.

Calvo, R., Constantino, A. A., and Figueiredo, M. (2015). A multi-pheromone stigmergic distributed robot coordination strategy for fast surveillance task execution in unknown environments. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Calvo, R., de Oliveira, J. R., Figueiredo, M., and Romero, R. A. F. (2011). A distributed, bio-inspired coordination strategy for multiple agent systems applied to surveillance tasks in unknown environments. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 3248–3255. IEEE.

Campo, A. and Dorigo, M. (2007). Efficient multi-foraging in swarm robotics. In *European Conference on Artificial Life*, pages 696–705. Springer Berlin Heidelberg.

Chaimowicz, L. and Kumar, V. (2004). Aerial shepherds: Coordination among uavs and swarms of robots. In *In Proc. of DARS'04*. Citeseer.

Chattunyakit, S., Kondo, T., Nilkhamhang, I., Phatrapornnant, T., and Kumazawa, I. (2013). Two foraging algorithms for a limited number of swarm robots. In *SICE Annual Conference (SICE), 2013 Proceedings of*, pages 1056–1061. IEEE.

Colorni, A., Dorigo, M., Maniezzo, V., et al. (1991). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France.

Cucker, F. and Smale, S. (2007). Emergent behavior in flocks. *IEEE Transactions on automatic control*, 52(5):852–862.

De Rango, F., Palmieri, N., Yang, X. S., and Marano, S. (2015). Bio-inspired exploring and recruiting tasks in a team of distributed robots over mined regions. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2015 International Symposium on*, pages 1–8. IEEE.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., et al. (2013). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71.

Dorigo, M. and Roosevelt, A. F. (2004). Swarm robotics. In *Special Issue, Autonomous Robots*. Citeseer.

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

Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. (2011). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73–96.

Elliot, M. (2006). Stigmergic collaboration: The evolution of group work. *m/c journal*, 9(2).

Florea, B.-F., Grigore, O., and Datcu, M. (2015). Pheromone averaging exploration algorithm. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 617–622. IEEE Xplore.

Fortino, G., Zedadra, O., Jouandeau, N., and Seridi, H. (2014). A decentralized ant colony foraging model using only stigmergic communication. In *Proceedings of XV Workshop Dagli Oggetti agli Agenti (WOA 2014)*, volume 1260. Citeseer.

Fossum, F., Montanier, J.-M., and Haddow, P. C. (2014). Repellent pheromones for effective swarm robot search in unknown environments. In *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, pages 1–8. IEEE.

Fujisawa, R., Imamura, H., Hashimoto, T., and Matsuno, F. (2008). Communication using pheromone field for multiple robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1391–1396. IEEE.

Hayes, A. T., Martinoli, A., and Goodman, R. M. (2002). Distributed odor source localization. *IEEE Sensors Journal*, 2(3):260–271.

Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex machina: Ant swarm foraging from physical to virtual and back again. In *International Conference on Swarm Intelligence*, pages 252–259. Springer.

Howard, A., Parker, L. E., and Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6):431–447.

Hrolenok, B., Luke, S., Sullivan, K., and Vo, C. (2010). Collaborative foraging using beacons. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 3-Volume 3*, pages 1197–1204. International Foundation for Autonomous Agents and Multiagent Systems.

Jackson, D. E., Holcombe, M., and Ratnieks, F. L. (2004). Trail geometry gives polarity to ant foraging networks. *Nature*, 432(7019):907–909.

Jackson, D. E. and Ratnieks, F. L. (2006). Communication in ants. *Current biology*, 16(15):R570–R574.

Jennings, J. S., Whelan, G., and Evans, W. F. (1997). Cooperative search and rescue with a team of mobile robots. In *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on*, pages 193–200. IEEE.

Johansson, R. and Saffiotti, A. (2009). Navigating by stigmergy: A realization on an rfid floor for minimalistic robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 245–252. IEEE.

Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V., Pereira, G., and Spletzer, J. (2003). Distributed search and rescue with robot and sensor teams. In *Field and Service Robotics*, pages 529–538. Springer.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE.*

Kennedy, J., Kennedy, J. F., Eberhart, R. C., and Shi, Y. (2001). *Swarm intelligence.* Morgan Kaufmann.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98.

Kofod-Petersen, A. (2012). How to do a structured literature review in computer science. *Document released as a guide to performing a Structured Literature Review at NTNU. Retrieved December*, 19:2014.

Kuiper, E. and Nadjm-Tehrani, S. (2006). Mobility models for uav group reconnaissance applications. In *2006 International Conference on Wireless and Mobile Communications (ICWMC'06)*, pages 33–33. IEEE.

Letendre, K. and Moses, M. E. (2013). Synergy in ant foraging strategies: memory and communication alone and in combination. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 41–48. ACM.

Liemhetcharat, S., Yan, R., and Tee, K. P. (2015). Continuous foraging and information gathering in a multi-agent team. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1325–1333. International Foundation for Autonomous Agents and Multiagent Systems.

Liu, W. (2008). *Design and modelling of adaptive foraging in swarm robotic systems.* PhD thesis, Citeseer.

Luke, S., Cioffi-Revilla, C., Panait, L., and Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 swarmfest workshop*, volume 8, page 44.

Masár, M. (2013). A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In *2013 IEEE 17th International Conference on Intelligent Engineering Systems (INES).*

Meng, Y. and Gan, J. (2008). A distributed swarm intelligence based algorithm for a cooperative multi-robot construction task. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–6. IEEE.

Momen, S. and Sharkey, A. (2009). An ant-like task allocation model for a swarm of heterogeneous robots. In *Proceedings of the 2nd swarm intelligence algorithms and applications symposium (SIAAS)*, pages 31–38.

Panait, L. and Luke, S. (2004). A pheromone-based utility model for collaborative foraging. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 36–43. IEEE Computer Society.

Parker, L. E. (2003). The effect of heterogeneity in teams of 100+ mobile robots. *Multi-Robot Systems*, 2:205–215.

Payton, D., Estkowski, R., and Howard, M. (2004). Pheromone robotics and the logic of virtual pheromones. In *International Workshop on Swarm Robotics*, pages 45–57. Springer.

Pearce, J. L., Powers, B., Hess, C., Rybski, P. E., Stoeter, S. A., and Papanikolopoulos, N. (2006). Using virtual pheromones and cameras for dispersing a team of multiple miniature robots. *Journal of Intelligent and Robotic Systems*, 45(4):307–321.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34.

Rodríguez, A., Gómez, J., and Diaconescu, A. (2015). Foraging-inspired self-organisation for terrain exploration with failure-prone agents. In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*, pages 121–130. IEEE.

Ron, D. and Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer.

Sakakibara, T., Kurabayashi, D., et al. (2007). Artificial pheromone system using rfid for navigation of autonomous robots. *Journal of Bionic Engineering*, 4(4):245–253.

Sauter, J. A., Mathews, R. S., Yinger, A., Robinson, J. S., Moody, J., and Riddle, S. (2008). Distributed pheromone-based swarming control of unmanned air and ground vehicles for rsta. In *SPIE defense and security symposium*, pages 69620C–69620C. International Society for Optics and Photonics.

Seyfried, J., Szymanski, M., Bender, N., Estana, R., Thiel, M., and Wörn, H. (2004). The i-swarm project: Intelligent small world autonomous robots for micro-manipulation. In *International Workshop on Swarm Robotics*, pages 70–83. Springer.

Simonin, O., Charpillet, F., and Thierry, E. (2014). Revisiting wavefront construction with collective agents: an approach to foraging. *Swarm Intelligence*, 8(2):113–138.

Steele Jr, F. and Thomas, G. (2007). Directed stigmergy-based control for multi-robot systems. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 223–230. ACM.

Sugawara, K., Kazama, T., and Watanabe, T. (2004). Foraging behavior of interacting robots with virtual pheromone. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 3074–3079. IEEE.

Tereshko, V. and Loengarov, A. (2005). Collective decision making in honey-bee foraging dynamics. *Computing and Information Systems*, 9(3):1.

Tisue, S. and Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA.

Unknown (2013). The three rules of the boids. [Online; accessed 2013-11-29].

Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence*, 2(2-4):189–208.

Von Frisch, K. (1967). The dance language and orientation of bees.

Wang, T. (2012). Artificial potential field. [Online; accessed 2016-12-02].

Winfield, A. F. (2009). Foraging robots. In *Encyclopedia of complexity and systems science*, pages 3682–3700. Springer.

Zedadra, O., Seridi, H., Jouandeau, N., and Fortino, G. (2014). S-masa: A stigmergy based algorithm for multi-target search. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1477–1485. IEEE.

Zedadra, O., Seridi, H., Jouandeau, N., and Fortino, G. (2015a). Design and analysis of cooperative and non cooperative stigmergy-based models for foraging. In *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, pages 85–90. IEEE.

Zedadra, O., Seridi, H., Jouandeau, N., and Fortino, G. (2015b). A distributed foraging algorithm based on artificial potential field. In *Programming and Systems (ISPS), 2015 12th International Symposium on*, pages 1–6. IEEE.

Zedadra, O., Seridi, H., Jouandeau, N., and Fortino, G. (2016). A cooperative switching algorithm for multi-agent foraging. *Engineering Applications of Artificial Intelligence*, 50:302–319.

Zhu, Q., Yan, Y., and Xing, Z. (2006). Robot path planning based on artificial potential field approach with simulated annealing. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 622–627. IEEE.

Appendices

Appendix A

A SLR Protocol

This is the structure literature protocol. It describes the process after which the core studies of the SLR were found. The protocol is divided into sections. Each section describes a new step in the protocol, in chronological order.

A.1 Problem definition

Swarm robotics show promise as a a foraging system that can potentially be deployed in a host of different applications. However, lots of research is still required before swarm foraging systems can become a reality. As such there exist a need to explore ways to improve and expand existing solutions, in order to move closer to such a realization. In particular it seems that little effort has been put into combining heterogeneous agents with well known foraging strategies, such as pheromone based algorithms.

In order for a structured literature review to have a function, there needs to exist a problem (P). Problems can often be solved in different fashions. Consequently the system solving P needs some constraints, methods and/or approaches (C).

P: Improve performance and flexibility in swarm intelligence foraging algorithms.

C: In an attempt to achieve this, a pheromone based swarm foraging algorithm will be combine with the use of heterogeneous agents. This idea came about as a suspicion that little research had been done on this area. Through an unstructured review of the literature little relevant literature was found, reinforcing this suspicion. In addition prominent research indicated that the use of heterogeneous agents is and important step in advancing swarm robotics, as this can improve a swarms flexibility [Ducatelle et al., 2011].

For reasons mentioned above related literature needs to be explored if this is to be achieve. As the suspicion is that there exist little literature combining pheromone based foraging with heterogeneous agents, an SLR focused around this might return little information about the state of swarm intelligence foraging systems. Because of this the protocol is designed to retrieve literature related to one of the two approaches, in addition to literature combining the two. This allows for a broader review of both heterogeneous foraging, and pheromone based foraging. The idea being that there might exist a research gap in combining the two. Thus the problem and constraints are drawn to the following research questions:

RQ1: What is the state-of-the-art in morphologically heterogeneous swarm intelligence systems used in foraging?

RQ2: What is the state-of-the-art in swarm intelligence systems utilizing pheromones for foraging like tasks? (e.g. foraging, exploration, surveillance)

A.2 Choice of digital libraries

In order to perform a SLR that is manageable the search in the literature must be reduced to a sub-set available sources. It is important that the sources picked have a high likelihood of containing relevant literature. Consequently sources chosen should include the most popular journals for publishing work related to computer science and in particular swarm intelligence. As this thesis is written at the Norwegian University of Science and Technology (NTNU), a requirement is that literature from a candidate source is available under NTNU's licensing. Another requirement is that the digital libraries support boolean search. This is allows for the division of search terms into groups of key terms as explained in section A.3. With these factors in mind the digital libraries seen in table 1 were chosen.

Source	URL
ACM Digital libary	http://dl.acm.org/
Engineering village	https://www.engineeringvillage.com/
Scopus	https://www.scopus.com/
ScienceDirect	http://www.sciencedirect.com/
IEEE Xplore	http://ieeexplore.ieee.org/
Web of Science	https://webofknowledge.com/

Table 1: Digital libraries and databases used in the SLR.

A.3 Relevant studies

After a collection of sources has been chosen the search terms can be defined. This is done by grouping terms that are either synonyms, different forms of the same word, or terms that have similar or related semantic meaning within the domain. Each group should represent a concept of the problem or one of its constraints. The goal is to retrieve documents that contain at least one of the search terms from each of the groups. In a boolean search string this can be done by linking search terms within a group with the OR operator, and the AND operator between groups. This search string is then used to preformed a search on all the digital libraries selected, see table 1. The resulting sets of documents are merge to create the initial document collection. Moste of these document will be unrelated, so a filtration process is needed, described in sections A.5, A.6, A.7.

Group 1	Group 2	Group 3
Heterogeneous	Foraging	Swarm robotics
Heterogeneity	Surveillance	Multi-agent
Morphology	Exploration	Swarm intelligence
Morphologically		Artificial bee colony
Morphological		Ant colony optimization
Pheromone		Particle swarm optimization
Stigmergy		ABC
Stigmergic		ACO
		PSO
		Artificial Intelligence
		AI

Table 2: Division of terms in to search groups.

If the goal of the SLR was to retrieve research on heterogeneous agents combined with pheromones, these two concepts would have to be in separate groups. This would ensure that only documents addressing both these concepts were returned. As we are interested in literature addressing at leased one of these concepts, terms related to any of the two are put in the same group. This will result in the inclusion of literature relating to one of these concepts, but also literature that combines the two. Hence the first group of terms contain words that are ether related to heterogeneity or pheromones. The second group consist of terms related to foraging, as this is the problem that should be solved. Here terms that represent problems that are closely related to foraging are included. The third group is made up of terms that relate to swarm robotics, swarm intelligence or AI. Some of the more popular swarm intelligence algorithms, as well as their abbreviations, are included in case a paper references it but fails to mention the field of research. This division results in the grouping of terms seen in table 2.

A.4 Complete search strings

Each digital library has a different system for entering search queries. In all cases the advanced search mode, or something equivalent, was chosen. This allowed for search strings to be entered in free text. Although the logical meaning of these queries are equivalent, the syntax is different. In this subsection the exact search strings used for each digital library are presented. Each string is logically equivalent to the following expression (with the exception of some flags regarding language, date range, etc...):

(”Heterogeneous” OR ”Heterogeneity” OR ”Morphology” OR ”Morphologically” OR ”Morphological” OR ”Pheromone” OR ”Stigmergy” OR ”Stigmergic”) AND (”Foraging” OR ”Surveillance” OR ”Exploration”) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony” OR ”Ant colony optimization” OR ”Particle swarm optimization” OR ”ABC” OR ”ACO” OR ”PSO” OR ”Artificial intelligence” OR AI)

Restrictions and Differences

Different settings were available in different libraries. The settings chosen are either detailed in the notes relating to the search string, or part of the search string it self. All libraries supported limiting search by date of publishing. Consequently papers published before 2006 were filtered out. The rational being that state-of-the-art research should have been published within the last 10 years. Filtration of non-english literature was filtered where possible. Some libraries allowed for search within fields of research. Where this was possible categories were chosen as detailed.

ACM Digital libary

The search was preformed in advanced search on the ”ACM Full-Text Collection”. The option of searching ”from 2006” was set manually.

+(Heterogeneous Heterogeneity Morphology morphologically morphological Pheromone stigmergy stigmergic) +(Foraging Surveillance Exploration) +(”Swarm robotics” ”multi-agent” ”Swarm intelligence” ”Artificial bee colony” ”Ant colony optimization” ”Particle swarm optimization” ABC ACO PSO ”Artificial intelligence” AI)

Engineering Village

The search was preformed in expert search. The search was preformed on title, abstract and keywords. Articles in press and non-english were excluded. The following date range was selected: 2006-p.p

(((((((Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony” OR ”Ant colony optimization” OR ”Particle swarm optimization” OR ABC OR ACO OR PSO OR ”Artificial intelligence” OR AI)) WN KY))) NOT (ip WN DT)) AND (english WN LA))

Scopus

The search was preformed in advanced search.

TITLE-ABS-KEY((Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony” OR ”Ant colony optimization” OR ”Particle swarm optimization” OR ABC OR ACO OR PSO OR ”Artificial intelligence” OR AI)) AND SUBJAREA(COMP OR ENGI) AND PUBYEAR ¿ 2006

Science Direct

The search was preformed in advanced search on the following entries: computer science and engineering. The following date range was selected: 2006-p.p

tak((Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony” OR ”Ant colony optimization” OR ”Particle swarm optimization” OR ABC OR ACO OR PSO OR ”Artificial intelligence” OR AI))

IEEE Xplorer

The search was preformed in advanced search on the setting: ”Metadata only”. The following date range was selected: 2006-p.p

At the time of the search IEEE Xplorer limited their search string to no more than 15 terms. This protocol contains a total of 22 terms. To work around this issue, the search string is split into three. By splitting group 3 in 3 parts and performing a search for each of them in the place of group 3, we get 3 result sets. By taking the union of these sets, we get the same set of documents as we would were we to use the entire search string.

Part1:

(Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony”)

Part2:

(Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Ant colony optimization” OR ”Particle swarm optimization” OR ABC) Results: 98

Part3:

(Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (ACO OR PSO OR ”Artificial intelligence” OR AI) Results: 165

Web of Science (previously ISI web of knowledge)

The search was preformed in advanced search on title, abstract and keywords. The following date range was selected: 2006-2016

TS=((Heterogeneous OR Heterogeneity OR Morphology OR morphologically OR morphological OR Pheromone OR stigmergy OR stigmergic) AND (Foraging OR Surveillance OR Exploration) AND (”Swarm robotics” OR ”multi-agent” OR ”Swarm intelligence” OR ”Artificial bee colony” OR ”Ant colony optimization” OR ”Particle swarm optimization” OR ABC OR ACO OR PSO OR ”Artificial intelligence” OR AI)) AND SU=(Computer Science OR Engineering)

Results

After a boolean search with the terms found in table 2 were performed, the initial document collection was established. The initial collection contained 1805 documents. 633 of thes were duplicates. After removing these the collection was reduced to 1172 documents.

Source	Number of studies
ACM Digital libary	148
Engineering village	675
Scopus	334
ScienceDirect	36
IEEE Xplore	425
Web of Science	187
Total	1805

Table 3: Document collection with duplicates

Source	Number of studies
ACM Digital libary	147
Engineering village	317
Scopus	150
ScienceDirect	12
IEEE Xplore	415
Web of Science	131
Total	1172

Table 4: Document collection without duplicates

A.5 Study quality assessment

After the removal of duplicates the document collection must undergo a filtration process. The aim of this filtration is to drastically reduce its size, keeping the most relevant papers. To achieve this the collection is first filtered through a set of text screenings, detailed in section A.6. Then the remaining collection is ranked according to quality criteria (QC), detailed in section A.7.

A.6 Document screening

To filter out studies that are not thematically relevant they are judged according to a set of inclusion criteria (IC). These criteria can be seen in table 5. In addition to the inclusion criteria, two quality criterion are added as questions for completeness.

IC1	The studies main concern is that of foraging, surveillance or exploration of physical space, or simplified simulations of such.
IC2	The study is a primary study, presenting empirical results.
IC3	The study utilizes swarm intelligence in the form of either a morphologically heterogeneous swarm or a swarm utilizing pheromones as a means of communication, or a combination of the two.
IC4	The studies main concern is that of developing methods for achieving emergent behavior in a swarm of agents. And not technical problems related to implementing such a method on a physical system (e.g. error handling, accurate positioning, noisy sensor readings).
IC5	The study describes a clear method for achieving its goal.
QC1	There is a clear statement of the aim of the research.
QC2	The study is put into context of other studies and research.

Table 5: Inclusion Criteria

The inclusion criteria are applied in two iterations: abstract and full text. After this a full text quality screening is performed. Before the collection was filtered on inclusion criteria, papers that were obviously irrelevant were filtered out baste on title. This gave the following process:

1. Irrelevant title screening
2. Abstract inclusion criteria screening
3. Full text inclusion criteria screening
4. Full text quality screening

Irrelevant title screening

A large portion of the document collection were papers from other fields of research that where in no way related to foraging. These papers where filtered base on title or a fast glance at the abstract. This Reduced the collection from 1172 documents down to 226 documents.

Abstract inclusion criteria screening

At this stage the collection was reduced to 53 papers.

Full text inclusion criteria screening

At this stage it proved difficult to filter out documents. In accordance with IC4, research that focused more on physical implementation and its complications, rather than algorithmic advances were removed. Papers that were concerned with physical implementation, but still made contributions to algorithmic development were kept. After filtering based on this, and the other criteria, the collection was reduced to 34 documents.

Full text quality screening

At this stage the collection was reduced to 27 documents.

A.7 Quality assessment

Now that the collection is reduced to a handful of documents. They can go through a detailed quality assessment. This is done by ranking the remaining documents according to quality criteria (QC).

Most of these QC are measurements of scientific quality. Due to the nature of this SLR, and the decision to include papers that are related to foraging, some QC are measurements of relevance. These are included in ordered to weight studies in favour of those closely relating to the problem and its constraints, see A.1.

As a consequence if two papers, one written on surveillance using pheromones, the other on foraging using pheromones, is deemed to have the same scientific quality. The paper on foraging is ranked slightly higher.

16 quality criteria were defined as follows:

1. Is there a clear statement of the aim of the research?
2. Is the study put into context of other studies and research?
3. Are system or algorithmic design decisions justified?
4. Is the test data set reproducible?
5. Is the study algorithm reproducible?
6. Is the experimental procedure throughly explained and reproducible?
7. Is it clearly stated in the study which other algorithms the studies algorithm(s) have been compared with?
8. Are the performance metrics used in the study explained and justified?
9. Are the test results thoroughly analysed?

10. Does the test evidence support the findings presented?
11. Is the studies focus on foraging?
12. Does the study include the use of pheromones?
13. Does the study include the use of a heterogeneous swarm?
14. Is the studies primary focus combining heterogeneous swarm with pheromones?
15. Does the system handle obstacles?
16. Does the study shows some level of generic application? Meaning that it can work in different scenarios.

Scoring

Each of the papers should be scored on all quality criteria. Papers can be given a score of 1, 0.5 or 0 on each criterion. This corresponds to yes(1), partially(0.5) and no(0). The full scoring of papers can be seen in table 8. The average score was approximately 10.37. A threshold was set at 10.5, meaning that papers achieving a lower score would be excluded. 14 papers achieved a score above the threshold. These papers constitute the core literature of the SLR, see table 6.

A.8 Data collection

NEEDED?

A.9 Data analysis

NEEDED?

Name	Author
Self-organized cooperation between robotic swarms	Ducatelle et al.
Cooperative self-organization in a heterogeneous swarm robotic system	Ducatelle et al.
Continuous Foraging and Information Gathering in a Multi-Agent Team	Liemhetcharat et al.
Collaborative Foraging using Beacons	Hrolenok et al.
A Decentralized Ant Colony Foraging Model Using Only Stigmergic Communication	Fortino et al.
A Distributed Foraging Algorithm Based on Artificial Potential Field	Zedadra et al.
A Cooperative Switching Algorithm for Multi-Agent Foraging	Zedadra et al.
Pheromone Averaging Exploration Algorithm	Florea et al.
A multi-pheromone stigmergic distributed robot coordination strategy for fast surveillance task execution in unknown environments	Calvo et al.
Foraging-inspired Self-organisation for Terrain Exploration with Failure-prone Agents	Rodríguez et al.
Formica ex machina: Ant swarm foraging from physical to virtual and back again	Hecker et al.
An Ant-like Task Allocation Model for a Swarm of Heterogeneous Robots	Momen and Sharkey
Synergy in ant foraging strategies: memory and communication alone and in combination	Letendre and Moses
Design and Analysis of Cooperative and Non Cooperative Stigmergy-based Models for Foraging	Zedadra et al.

Table 6: SLR core studies with their respective authors

Quality criteria	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Score
Self-organized cooperation between robotic s...	1	1	1	1	1	1	1	1	1	1	0.5	0.5	1	0.5	1	1	14.5
Cooperative self-organization in a heteroge...	1	1	1	1	1	1	1	1	1	1	0.5	0.5	1	0.5	1	1	14.5
Continuous Foraging and Information Gathe...	1	1	1	1	1	1	0.5	0.5	1	1	1	1	1	0	0	1	13
Collaborative Foraging using Beacons	1	1	1	1	1	1	1	1	1	1	1	0.5	0	0	1	1	13.5
A Decentralized Ant Colony Foraging Model...	1	1	0.5	1	1	0.5	1	1	1	1	1	1	0	0	1	1	13
A Distributed Foraging Algorithm Based on...	1	0.5	1	1	1	1	1	1	1	1	1	1	0	0	1	0.5	13
A Cooperative Switching Algorithm for Mult...	1	0.5	1	1	1	1	1	1	1	1	1	1	0	0	1	0.5	13
Pheromone Averaging Exploration Algorithm	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	13
A multi-pheromone stigmergic distributed rob...	1	1	1	1	1	1	1	1	1	0.5	0	1	0	0	1	1	12.5
Foraging-inspired Self-organisation for Terra...	1	0.5	0.5	1	1	1	0.5	1	1	1	1	1	1	0	0.5	0	12
Formica ex machina: Ant swarm foraging fro...	1	1	1	1	1	1	0	0.5	1	1	1	1	0	0	0	1	11.5
An Ant-like Task Allocation Model for a Swa...	1	0.5	1	1	0.5	1	0.5	1	1	1	1	1	1	0	0	1	12.5
Synergy in ant foraging strategies: memory an...	1	0.5	1	1	1	1	0.5	0.5	1	1	1	1	0	0	0	0.5	11
Design and Analysis of Cooperative and Non...	0.5	0.5	1	1	0.5	0.5	1	0.5	1	1	1	1	0	0	1	0.5	11
A swarm-based robot team coordination proto...	1	1	0.5	1	0.5	1	0.5	0.5	0.5	1	0	1	0	0	0.5	0.5	9.5
Synthesis and analysis of control laws for swar...	1	0.5	0	1	0.5	1	0	1	0.5	1	1	1	0	0	0	0.5	9
Keeping diversity when exploring dynamic en...	1	0.5	0.5	0.5	1	1	0	0	1	0.5	0	1	1	1	0	0	9
Distributed online patrolling with multi-agent...	0.5	1	0.5	1	0.5	1	0	1	1	1	0	0	1	0	0	0.5	9
Collaborative multi agent physical search with...	1	1	0.5	1	1	1	0	1	1	1	0	0	0.5	0	0	0	9
Repellent pheromones for effective swarm robo...	0.5	0.5	1	0	0.5	1	0	1	1	1	0	1	0	0	1	0.5	9
Continuous swarm surveillance via distributed...	0.5	0.5	0.5	0.5	0.5	1	0.5	0.5	1	1	0	1	0	1	0	0.5	9
Putting Simple Hierarchy into Ant Foraging...	1	0.5	0	0.5	1	0.5	0	0.5	0.5	1	1	1	0	0	0	0	7.5
Coordination mechanisms for tracking and sur...	1	0.5	0.5	0.5	0	0.5	0	1	0.5	1	0	0	1	0	1	0	7.5
Distributed, heterogeneous, multi-agent social...	0.5	0.5	1	0.5	0.5	0.5	0	0.5	0	0	1	0	1	0.5	0	0.5	7
CyberRescue: A pheromone approach to multi...	1	0.5	0	0	1	0	0	0	0	0	0.5	1	0	0	1	0.5	5.50
Discrete firefly algorithm for recruiting ta...	0.5	1	0.5	1	0.5	0.5	0	0	0	0.5	0	1	0	0	0	0	5.5
Implementing pheromone-based, negotiating fo...	0.5	0.5	1	0	0	0	0	0	0.5	0.5	1	1	0.5	0	0	0	5.5
Average score:																	10.4

Table 8: Papers ranked on quality criteria. Sorted after score, and split at threshold.

Appendix B

B Data

B.1 Capacity and Energy Consumption

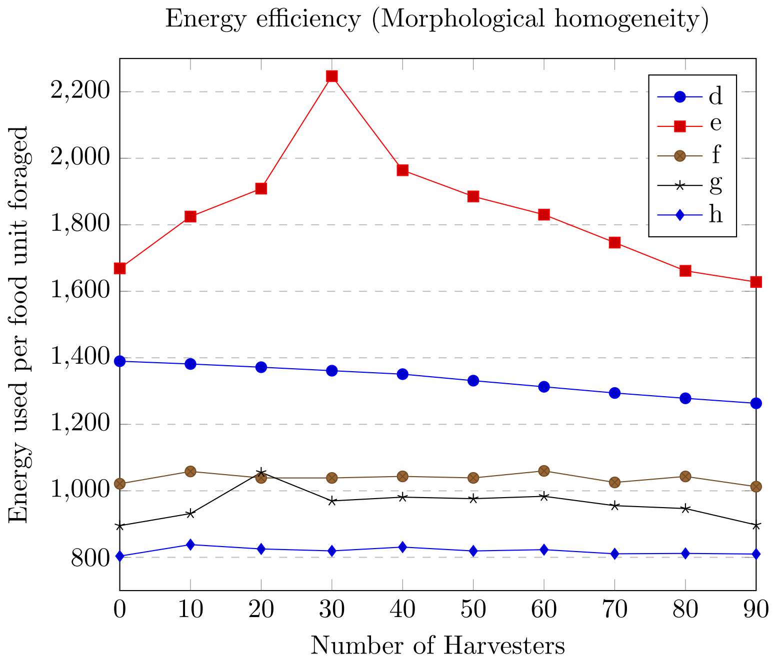


Figure 1: Energy/food ratio: Using harvesters with identical carrying capacity and energy consumption as scouts (1 and 1).

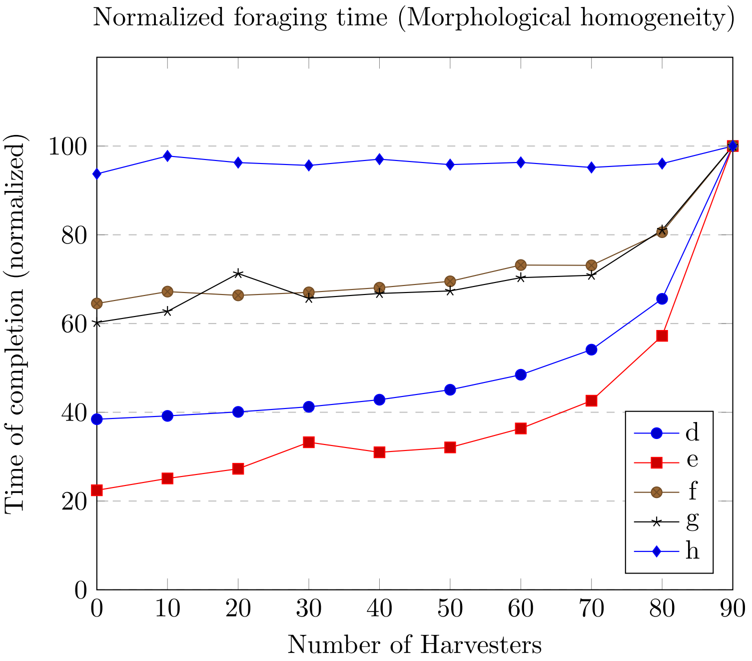


Figure 2: Time of completion: Using harvesters with identical carrying capacity and energy consumption as scouts (1 and 1). Each map is individually normalized to a range of 0-100.

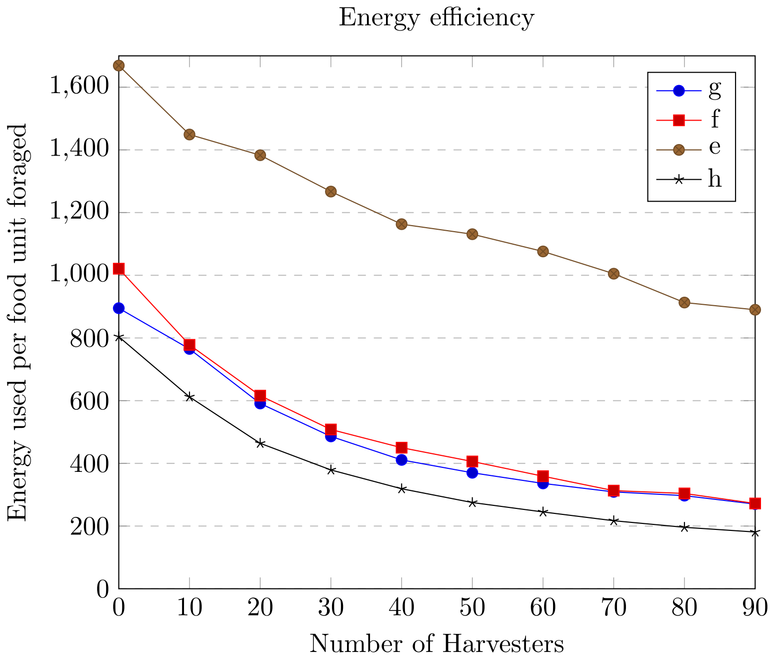


Figure 3: Energy/food ratio: Energy efficiency at different Scout to harvester ratios. X-axis show how many out of a total of 100 agents are harvesters. The rest are scouts.