Exploring communication network structures for the adaptability of a team of robots working in dynamical foraging environments

Juan Manuel Nogales Viedman

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2018

Juan Manuel Nogales Viedman

Exploring communication network structures for the adaptability of a team of robots working in dynamical foraging environments

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Bio-inspired robotics

Advisor: Gina Maira Barbosa de Oliveira

Uberlândia
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

N774e
2018

Nogales Viedman, Juan Manuel, 1987-
Exploring communication network structures for the adaptability of a team of robots working in dynamical foraging environments [recurso eletrônico] / Juan Manuel Nogales Viedman. - 2018.

Orientadora: Gina Maira Barbosa de Oliveira.
Tese (Doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://dx.doi.org/10.14393/ufu.te.2018.816
Inclui bibliografia.
Inclui ilustrações.

1. Computação. 2. Robótica. 3. Robôs - Movimento. I. Oliveira, Gina Maira Barbosa de (Orient.) II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU: 681.3

Maria Salete de Freitas Pinheiro - CRB6/1262

Este trabalho é dedicado a minha avó e minha mãe,
mulheres fortes e guerreiras.

Agradecimentos

Quero iniciar agradeçendo a minha orientadora, a profa Gina, por me ter tanta paciência, esforçando-se em entender meu portunhol e aceitar o desafio de dirigir um trabalho fora da area de dominio dela. Ela é pesquisadora de referência em Automatas Celulares e para lá quis me puxar muitas vezes.

Aos professores membros da banca pelo seu tempo e disposição para avaliar este trabalho. De forma especial ao professor Luiz Chaimowicz e ao professor Carlos Lopes por ter nos direcionado e assessorado no dia da qualificação e na prévia da defesa. Também, aos outros professores que me ajudaram durante minha estadia no Brazil: Benice Naves, Faruk Fonthal, Jean da mecânica, João Gama, Jorge Finke, Márcia Aparecida e Maurício Escarpinati. E de formal especial, ao professor Vladimir por me ensinar russo e me oferecer moradia quando a bolsa e minhas reservas estavam prestes a terminar.

Ao secretario e amigo, Erisvaldo, quem me ajudou demais em todos os processos e documentos que devem ser entregues ao longo do doutorado. Que caos se forma na pos da FACOM quando ele sai de ferias.

Aos colegas de laboratorio: Reslley, Claudiney, Danielli, Gabriel e Candido. E aos brasileiros que me acolheram como mais um deles: Alison, Alvaro, Bruno, Denize, Dyany, Emilia, Franciny, João, Joyce, Juliete, Mariana, Myllene, Raniele, Walter e outros que passei por alto.

Às pessoas que me apoiaram com diferentes ferramentas para implementar parcialmente este trabalho: Alex, Caio, Jhonson, Lucas, Marco, Micael e Vinicius. E aos que me apoiaram com seu tempo nas revisões dos textos: Celso Pacola, Charles Sousa, Gustavo Siqueira, Paulo Henrique, Rafael Dias e Taffarel Brantt.

Aos irmãos da Igreja de Deus Ministerial que me estenderam seu auxilio, não somente através de ajudas monetarias senão também através de suas orações.

Ao grupo de hispano-falantes, aos de Africa, aos de Ucrania e aos de Timor que fizeram mais divertida minha estadia através dos convites aos churrascos, festas e jogos de futebol.

A minha familia pelo constante apoio e por torcerem por mim, mesmo na distancia estiveram muito perto de mim neste periodo.

Não poderia deixar de agradecer ao filosofo e amante da tecnologia Juan G Diaz pelos seus muitos conselhos da vida e as incontáveis ajudas que ele me ofereceu. Uma conclusão das nossas conversas é que tudo tem um tempo e, às vezes, quando chega não é como queremos, contudo devemos esperar que o tempo tudo esclarece.

À OEA por gestionar as conexões entre o Brazil e estudantes estrangeiros através do concurso PAEC OEA/GCUB 2012. Onde tive a fortuna de ser escolhido entre muitos concorrentes para, através desta bolsa, representar meu país no Brazil enquanto desenvolvia meus estudos de doutorado. E à FACOM por abrir as portas para estudos superiores a estrangeiros.

Ao maior de America latina, Brazil estou grato pela oportunidade de conhecê-lo, estamos esperando muito de vocês.

Finalmente, agradeço ao Senhor, criador dos céus e da terra, quem faz tudo possível, foi Ele quem me instou a continuar, quem colocou em mim essa paixão pela robótica... ainda não sei o motivo.

"It's not about the cards you're dealt, but how you play the hand."
(Randy Pausch, the Last lecture)

Resumo

Na natureza, a tarefa de forrageamento tem uma importância vital para diversos organismos que exibem comportamento coletivo. Por esse motivo, uma das tarefas mais investigadas na robótica de enxame é o forrageamento. Neste trabalho se considera um time de robôs forrageiros no qual cada membro repete a tarefa de pegar um objeto e transportá-lo até um depósito. A navegação desses robôs depende de indicações visuais e locais, pois, nem odometria nem orientações globais foram utilizadas. O controlador é uma máquina de estados hierárquica, na qual subjazem os comportamentos dos robôs. Eles podem aprender qual comportamento se ajusta melhor às condições ambientais, que são dinâmicas. Para lidar com as mudanças ambientais, desenvolvemos estratégias que permitem aos robôs se adaptarem. Tais estratégias são avaliadas em ambientes distribuídos e aproveitam dois tipos de conexões: robô-ambiente e robô-robô. Em particular, essas conexões permitem o fluxo, a combinação das informações e, como consequência, o aprendizado. Visto que a informação influencia os comportamentos de cada robô e o desempenho do time, nossas estratégias foram avaliadas em dois cenários distintos: i) alocação de tarefas e ii) particionamento de tarefas. No ambiente de alocação de tarefas, os robôs interagem em três regiões do ambiente. Cada região conta com interfaces que geram objetos a diferentes taxas e compartilham informações entre elas e com os robôs trabalhando na região. A estratégia proposta alcançou uma solução próxima do ótimo teórico quando os robôs incorporaram essas informações. No outro ambiente, os robôs empregam interfaces para transferir objetos entre duas regiões ou os levam através de um caminho alternativo. Nesse cenário, as interfaces mudam a velocidade de transferência dos objetos e os robôs aprendem a partir de informações compartilhadas entre si. As estratégias propostas permitem que os robôs ajustem suas conexões para adaptar seu aprendizado e, assim, tomar decisões apropriadas à demanda pelas interfaces.

Palavras-chave: forrageamento, sistemas multi-robô, tomada de decisão probabilística, particionamento de tarefas, alocação de tarefas, aprendizado adaptativo.

Juan Manuel Nogales Viedman

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
FACULDADE DE COMPUTAÇÃO – FACOM
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO – PPGCO

The undersigned hereby certify they have read and recommend to the PPGCO for acceptance the dissertation entitled **Exploring communication network structures for the adaptability of a team of robots working in dynamical foraging environments** submitted by **Juan Manuel Nogales Viedman** as part of the requirements for obtaining the **Phd's degree in Computer Science**.

Uberlândia, 22 de September de 2017

Advisor: ______________________________________
Prof. Dr. Gina Maira Barbosa de Oliveira
Universidade Federal de Uberlândia

Examining Committee Members:

____
Prof. Dr. Carlos Roberto Lopes
Universidade Federal de Uberlândia

____
Prof. Dr. Jefferson Rodrigo de Souza
Universidade Federal de Uberlândia

____
Prof. Dr. Luiz Chaimowicz
Universidade Federal de Minas Gerais

____
Prof. Dra. Roseli Aparecida Romero
Universidade de São Paulo

Abstract

In nature, foraging is a fundamental task for several organisms that exhibit collective behaviors. For this reason, one of the most researched tasks in swarm robotics is foraging. In this work, we consider a team of foraging robots where each member repeats the same task: to pick up an object and transport it to a nest. Robot navigation depends completely on local visual clues. Neither dead reckoning nor global orientation helped them. The controller is a hierarchical state machine, which underlies robot behaviors. They learn which behavior fits better with the current condition. Since the environment changes, we developed strategies that enabled robots to achieve a good adaptation. These strategies dealt with distributed environments and exploited: robot-environment and robot-robot connections. In particular, such set of connections allowed the information to flow at a good rate and, as a consequence, their social learning. Since information influences robot behaviors and team performance, the proposed strategies were evaluated in two scenarios: *i*) task allocation and *ii*) task partitioning. In the task-allocation environment, robots interact within an environment, which consists of three regions where robots can forage. Each region includes a set of interfaces generating objects at different rates. These interfaces share information among them and with the subgroup of robots working inside each region. The proposed strategy reached a near-optimal solution because robots incorporated that information to make their decisions. In the task-partitioning environment, robots employ a set of interfaces to transfer objects between both regions or they carry the objects through an alternative path. In this scenario, those interfaces change their speed for transferring objects and robots only share information with other robots. The proposed solution allowed robots to adjust their connections to adapt their learning rate. As a consequence, they make decisions that fitted better to the changes in the interfaces.

Keywords: foraging, multi-robot system, probabilistic decision-making, task-partitioning, task-allocation, adaptive learning.

List of Figures

List of Tables

Glossary

Borda count: is a mechanism employed for elections where the candidates can be ranked by order of preference. It offers a proportional sum of the points the candidates got in the voting poll.

Communication structure: represents the paths or channels through which the messages are shared between the members. Over this structure, robots form neighborhoods, that is, subgroups of robots that are connected one another. A robot could emit a message to its neighbors and only them would have access to the message information.

Potential field: is a method to help robot navigation. The concept is inspired by the concept of electrical charges. Then, obstacles have the same charge of the robot creating a repulsive force, while a target has a contrary charge such that robots are attracted to it.

Hub: is a node that has a greater number of connections in the communication structure.

Information sharing: is the exchange of data between several units. It requires a communication structure where messages have a path to reach its receivers.

Law of diminishing returns: is a concept from economics that states that in a productive process, increasing the amount of a resource while holding constant the others. Then, after adding one unit more, the output of the process will increase in smaller increments. There is a point, where the output decreases.

Simplex: is a hyper-region of n dimensions whose boundaries limit the possible points that a set of variables may take.

Task allocation: is a technique to distribute a team of robots into a set of tasks such that they can achieve an optimal performance. Generally, tasks have different requirements and constraints demanding different amounts of robots.

Task partitioning: is a technique to divide the main task into subtasks such that robots can perform them sequentially or in parallel. Robots can coordinate between them, in an easier way, when they have to perform subtasks that require multiple individuals.

Contents

I hereby certify that I have obtained all legal permissions from the owner(s) of each third-party copyrighted matter included in my thesis, and that their permissions allow availability such as being deposited in public digital libraries.

Juan Manuel Nogales Viedman

Chapter 1

Introduction

Since the 80s, the community of robotics has been using teams or swarms of robots to imitate behaviors of social insects (SICILIANO, KHATIB, 2008). Roboticists try to mimic such behaviors because they provide flexible and robust strategies to solve complex tasks. Şahin (2005) stated the following criteria to consider whether a proposal belongs to the swarm robotics field:

- **Autonomy:** robots have a body and can interact physically and autonomously with the environment.
- **Group size:** the strategy of coordination should consider scalability to a large number of robots; but a lower bound is difficult to justify.
- **Homogeneity:** it is preferable to have homogeneous robots. The more heterogeneous, the less "swarm robotics" is.
- **Simplicity:** robots need to be relatively incapable or inefficient in relation to the task they perform.
- **Local sensing and communication:** these guarantee scalability and distributed coordination.

In general, the desired collective behavior emerges from local interactions among robots and between robots and their environment (BRAMBILLA et al., 2013). Current research in robotics follows the archetype of social insects. It employs several simple and cheap robots to replicate insect strategies (BAYINDIR, 2015). By working with multi-robot systems, researchers have gotten some benefits such as: reductions in the required time to complete a task (FERRANTE et al., 2013), increases in the team performance (PINI et al., 2013), robustness while dealing with error-prone and noisy environments (BUCHANAN, POMFRET, TIMMIS, 2016).

In several scenarios, cooperative multi-robot systems have shown that they can complete a task more quickly than a single complex robot (MOHAN, PONNAMBALAM, 2009). Notwithstanding, since each robot makes its own decisions, the team requires a good communication strategy to improve its performance (BALCH, ARKIN, 1994). Social insects provided roboticists with two ways of sharing information: explicit (MARCO, FARINA, 2001) and implicit (RATNIEKS, ANDERSON, 1999). On one hand, robots working with implicit communication can employ stigmergy and share information through clues they leave in the environment (e.g., chemicals (DENEUBOURG et al., 1991) or crumbs (DROGOUL, FERBER, 1993)). On the other hand, robots working with explicit communication employ robot-robot messages (e.g., visual signs (BALCH, ARKIN, 1994) or data strings (FONG, NOURBAKHSH, DAUTENHAHN, 2003)).

Both kinds of communication have helped to improve team performance even further (MOHAN, PONNAMBALAM, 2009). Those sharing-information strategies have worked well in foraging tasks, where robots need to find objects and transport them to a deposit (DROGOUL, FERBER, 1993). Some of the works employed implicit communication for navigation purposes as (CAMPO et al., 2010; GARNIER et al., 2007; RANJBAR-SAHRAEI, WEISS; NAKISAEE, 2012). In these works, robots modify the environment by leaving pheromone-like clues. The pheromone is some (virtual) chemical robots drop in the ground. Its rate of evaporation is essential in order that the group might deliver a good performance. When pheromone clues remain for long periods, the group could stagnate in a single option. Robots would have a high chance to reinforce former clues and never explore other alternatives. Fast rates of evaporation could vanish the benefits of communication. Each robot would be working individually because they could not benefit from the experiences of others. As a communication strategy, pheromone benefits the group if it has a suitable rate of evaporation (DENEUBOURG et al., 1991).

On the other hand, explicit communication allowed robots to share information between them such as: their position, food location, internal state, or goal, among others (BALCH, ARKIN, 1994; BALCH, HYBINETTE, 2000; PINI et al., 2013; PITONAKOVA, CROWDER; BULLOCK, 2016a). In these works, robots employ encoded messages that the receiver may decode. Robots can store information about the location of their targets and find the shortest path by integrating the shared information (MILETITCH et al., 2013). Also, it is common to find a North-star in the environment or to endow them with some hardware (as an embedded compass) for self-location (MILETITCH et al., 2013; PINI et al., 2013; PITONAKOVA, CROWDER; BULLOCK, 2016a). Robots locate themselves through this environmental information, which, sometimes, becomes virtual forces that support robot navigation. Generally, techniques like potential fields (SIMONIN, CHARPILLET, THIERRY, 2007) or social potential fields (BALCH, HYBINETTE, 2000) help transform the information into virtual forces.

Notwithstanding, besides navigation, robots execute more complex activities to complete foraging tasks. Robots have to find objects and transport them to a deposit while they deal with environmental conflicts (DROGOUL, FERBER, 1993; PINI et al., 2013).

According to (PITONAKOVA, CROWDER; BULLOCK, 2016a), mere implicit communication, as pheromone trails, could be insufficient to forage when environmental conditions change. In real scenarios, food decomposes, sources deplete and new ones appear in different locations. Consequently, robots would need a more complex communication strategy than those for static conditions (WEI, HINDRIKS, JONKER, 2014). Some authors proposed interesting strategies inspired by bees (MARCO, FARINA, 2001; SCHMICKL, THENIUS; CRAILSHEIM, 2012; PITONAKOVA, CROWDER; BULLOCK, 2016a). De Marco and Farina (2001) modeled how these insects could change their decisions when foragers share information about source quality. Once a forager shares information about a source that has better quality than the current ones, recruits that listen to it would disregard both messages about lower quality sources and their own information about their current source. Thus, when compared to ants, bees are better at establishing new food sources, because ants depend on pheromone evaporation, which may take a while.

Following with communication strategies inspired by bees, the proposals in (SCHMICKL, THENIUS; CRAILSHEIM, 2012; PITONAKOVA, CROWDER; BULLOCK, 2016a) took advantage of information-sharing and communication resources to increase team adaptability. Schmickl, Thenius and Crailsheim (2012) showed how a simulated colony of bees might adapt to new conditions. For this, the colony divides its team as follows: some foragers, few receivers, and many recruits. Foragers use short-range communication to share information about the quality of a source with the receivers, which help to recruit many other bees by sharing incoming information through long-range communication. Thus, by combining long and short range in their communication strategy, the information spreads faster and the colony can adapt to better sources. Pitonakova, Crowder and Bullock (2016a) implemented an information-sharing strategy based on the recruitment of bees proposed by (SCHMICKL, THENIUS; CRAILSHEIM, 2012) into a foraging testbed with robots. Their results show that when robots have long ranges of communication, the team could deliver better performance in static environments. However, long ranges of communication tend to stagnate the team in a single option if environmental conditions change. Their exploration rates are very low due to the broad range of communication. The authors indicated that if environmental conditions resemble real scenarios, lower ranges of communication would deliver better performance and increase the team adaptability.

In this document, we tried to resemble real conditions by working in dynamic environments. Our communication strategies regulate the benefits of sharing explicit information about the environmental conditions. We did not consider messages with navigation clues. In particular, we explored network structures for explicit information-sharing that could improve the adaptability of a team of robots working in environments that have controlled changes while they forage. In particular, we provided two environments where the team of robots can forage. Each environment is divided into regions such that it favors a different kind of strategy to forage as a team. One environment requires a task allocation solution, while the second is solved with a task partitioning strategy. Both ideas offer parallelism, efficiency, and flexibility to improve team performance, but in different ways (FERRANTE et al., 2013). Therefore, it is necessary to differentiate between them.

In particular, task allocation distributes the workforce into multiple tasks, i.e., how workers are assigned to each task (GERKEY, MATARIć, 2004). For instance, in (ZAHADAT et al., 2015), the environment consists of regions whose size is proportional to the demand of the task. The total number of robots increased (or decreased) with the *birth* (or *death*) of them. Since robots serve these regions by following a task allocation strategy, they find a solution when the number of robots in each region is proportional to the size of the region. Solutions following task allocation strategies depend on communication more than those following task partitioning strategies, because robots would need to coordinate which of them are going to serve each task (GERKEY, MATARIć, 2004).

Task partitioning organizes the task, i.e., how the task is divided into subtasks to be executed in parallel or sequentially (GERKEY, MATARIć, 2004). For instance, in (PINI et al., 2014), the environment is a long path where robots have to forage for objects from one extreme to deposit them on the other side. The authors allowed robots to transfer the object from one to another until the object reaches the nest. Thus, robots could complete a transportation by traveling smaller distances. They found a solution by adjusting the traveled distance for each of them, which depends on robot capacities (e.g., speed and vision). Task partitioning solutions emerge from local interactions, need little communication, and are difficult, if not impossible, to predict (GERKEY, MATARIć, 2004).

Our proposal considers the findings of Tambe (1997) for human teams. He explored the communication structures inside companies and concluded that a good communication strategy could improve the performance of cooperative team even further. In particular, we worked with explicit communication and showed, with a simulated team of robots, that performance can increase by allowing optimal structures for information sharing. In other words, we looked for communication structures that deliver an improvement in the team performance and a reduction of cost in the communication. Here, robot-robot connections build the communication structure where messages can be shared between robots. Since the environment changes, each of our robots can adjust its local structure of communication (connections) whenever it considers it is necessary. Those adjustments are based on its own experiences and current environmental conditions. According to (SUNG, AYANIAN, RUS, 2013), reaching an optimal solution with online and decentralized adjustments is (almost) impossible, but near-optimal solutions can be reached. We tested task allocation and task partitioning strategies because they require different levels of communication. Note that explicit inter-robot communication needs to regulate the following: What is it shared? Who is listening? How many messages is a robot sharing? To our knowledge, this is the first work considering data-streaming techniques, the law of diminishing returns, and concepts of complex networks to improve team performance in dynamical environments. Moreover, this improvement is achieved through the adjustment of the local communication structure of the robots.

Among our contributions, there is a navigation strategy inspired by shark hunting behaviors, which helped robots to navigate in both environments. For each task, besides the environments, we designed two controllers whose behaviors and transitions are in a hierarchical state machine adjusted accordingly. In both scenarios, robots employ a set of messages that include information about the tasks. They do not share any information about navigation. In particular, for the task allocation scenario, the environment included some aids to share information with robots, i.e., robot-environment connections. In this scenario, we adapted two base decision-making strategies and proposed a new strategy. The communication between robots worked over three different structures in those three strategies. For the task-partitioning scenario, we proposed some improvements on two base models. In this scenario, robots do not have environmental aids for information sharing. The communication worked over three different structures, which did not deliver clear results. However, we proposed an adaptive structure, which emerged from each robot adjustments. In other words, robots could adjust their own connections according to its experiences and measurements of the environmental conditions. This adaptive structure delivered promising results, which can be extended to other tasks.

1.1 Motivation

The goal of this proposal is to increase performance in dynamical foraging environments through adjustments in the (local) communication structure. For that, we explored static networks and evaluated their effects on team performance. Besides, we allowed robots to disable and enable their connections to adjust the communication structure according to changes in environmental conditions. We found some works where the authors improved the performance by considering the parameters of the structure of communication in their strategies for another kind of tasks (VISSCHER, SEELEY, PASSINO, 2006; HSU-SHAN et al., 2013; COUCEIRO et al., 2014). Furthermore, we noted that some authors increased the swarm adaptability exploiting explicit communication (HOFF, WOOD, NAGPAL, 2013; SARKER, DAHL, 2011; WEI, HINDRIKS, JONKER, 2014; PITONAKOVA, CROWDER; BULLOCK, 2016a). Increasing the swarm adaptability should increase the performance or, at least, reduce the effects of environmental changes.

Since communication supports the interaction between individuals such that team behavior may emerge, it opens the opportunity to analyze the effects of information sharing through complex networks. In complex networks, each node represents an acting unit or robot and a link between two nodes correspond to the interrelationship between them (NEWMAN, 2003). In other words, we can evaluate the effects of robot-robot and robot-environment interactions abstracting the team as a complex network. Thus, we took concepts from complex networks and communication strategies from recent approaches in multi-robot systems to propose a new way to improve swarm adaptability.

The following chapter details some of those recent approaches that improve the performance of multi-robot systems depending on team communication skills. From those works, we could infer that robots should share messages with more useful content (not only navigation clues). Moreover, short-range communication delivers two benefits: avoids over-commitment yielded by excessive information and exploits the experiences of other robots to improve team performance. Therefore, we should focus on message content and range of communication, particularly, when the environment changes. In other words, we have to design a communication strategy that can deal with dynamical environments, which is still an open challenge.

Our proposals explore predefined structures for information sharing such that we can find an optimal one over which robots may interact, which if it is necessary, they could adjust by changing their connections. Additionally, all the implemented strategies employed explicit communication, which is more suitable when robots must share content that is more complex, that is, the proposed messages were more complex than navigation clues. Note that we consider both message content and structures to improve team performance and adaptability while robots forage in dynamical environments. Thus, through their communication structure, robots should be able to regulate incoming information. Beyond the communication structure, it is clear that the information-sharing strategy may affect team performance. As slow pheromone evaporation rates can lock a team, quick and excessive information can deteriorate its adaptability. We tested several information-sharing strategies in two dynamical environments, where robots need to avoid over-commitment to or stagnation in the best source. In other words, our robots had to be able to balance between the benefits of little (short-range) and excessive (long-range) communication because environmental conditions change.

1.2 Hypothesis

We believe there is a way to improve the team adaptability through local modifications in the robot connections. As several authors have shown, communication is critical for robot coordination, and it seems that communication opens an alternative to influence robot behaviors positively.

1.3 Objectives

In this work, we investigate new communication strategies to allow robots to adapt to dynamic environments. Such strategies helped multi-robot systems working in two foraging tasks where their performance was measured through the number of delivered objects in the nests. In particular, our proposal provides robots with an information-sharing structure to deal with changes in the environment, where a connection means that two robots can share information between them. Thus, all connections generate the information-sharing structure. From cited works, we infer that probably there is not a universal structure that improves group performance in dynamic environments. However, if we cannot find one, maybe robots can find it by adjusting their own connections. Thus, our robots have to achieve the following objectives:

1.3.1 General objective

The broad goal of this work is to show evidence that a robot can adapt its behavior to improve the team adaptability by modifying its structure of communication (a.k.a. local connections) while dealing with dynamical environments.

1.3.2 Specific objectives

- To identify and implement some models of foraging robots where communication was relevant to the fulfillment of robot tasks.
- To select different communication structures based on topologies investigated in complex networks.
- To evaluate the performance of the team of robots in the implemented foraging tasks while they share information over the selected communication structures (static topologies).
- To identify a pattern between the structural parameters of the communication network and the performance of the team of robots.
- To discover which environmental events require robots to adjust their connections while they execute the implemented foraging tasks in dynamic environments.
- To elaborate local mechanisms of perception for the robots be able to identify the events that demand adjustments in their communication structure.
- To improve the adaptability of the team of robots while foraging in the implemented environments by embedding the mechanism of (local) structural adaptations.

We expect that the team adaptability will reflect in the performances when robots forage. Better adaptability should lead to faster adjustments to deal with the environmental changes, which should translate into more objects transported.

1.4 Thesis Organization

This document has the following organization: the first chapter introduced the problem, objectives, and the motivation of this research. The second chapter includes related works and tools that we employed for our simulations. The third chapter describes the problems we dealt with, while the proposals to solve them are in chapter four. All simulation details and results are in the fifth chapter. Since we introduced the problem of foraging and two ways of tackling this task, all the following chapters have two sections. Firstly, we describe the foraging task configuration that should be tackled through a task-allocation strategy. We proposed this strategy for multi-robot systems and published it in (NOGALES, OLIVEIRA, 2018b). This model is an adaptation of the deterministic solution proposed for multi-agent systems found in (NOGALES, FINKE, 2013). Since that is an extension of a deterministic solution, we only explored few structural adjustments because events are known beforehand. The second strategy to deal with foraging is based on a task-partitioning strategy. In particular, we explored network structures with a task partitioning solution based on the works of (PINI et al., 2011b; PINI et al., 2013). The results of this second part are published in (NOGALES, OLIVEIRA, 2018a) and other is still in submission and revision processes (NOGALES, VARGAS, OLIVEIRA, 2018). In this solution, robot distribution emerges from local interactions making the prediction of critical events a difficult, if not impossible, goal (GERKEY, MATARIć, 2004). Therefore, we implemented a special mechanism for detecting the events when robots need to adjust its connections (NOGALES, VARGAS, OLIVEIRA, 2018). Robot navigation in both scenarios follows a strategy proposed in (NOGALES, ESCARPINATI; OLIVEIRA, 2017). The final chapter presents the respective conclusions.

Furthermore, we included one appendix to provide more details about our research. Appendix A provides a short review of the environmental aids we found interesting to help robots that have limited resources (like ours). Some of them were included in the simulated environments, which were implemented in Webots simulator (MICHEL, 1998).

Chapter 2

Related works and tools

2.1 Related works

Research in multi-robot systems showed that diverse applications might achieve some benefits by exploiting a swarm of robots to solve complex problems. Due to parallelism, the swarm of robots could: *i*) Enable individual specialization (ZAHADAT et al., 2015), *ii*) Decrease energy consumption (CASTELLO et al., 2016), *iii*) Decrease odometry error and conflicts while traveling (BUCHANAN, POMFRET, TIMMIS, 2016; PINI et al., 2014), or *iv*) Increase the number of transported objects (PINI et al., 2011b; PINI et al., 2013). However, the designer responsibility is to propose coordination strategies that increase team performance and solve the emerging conflicts. Such strategies should regulate robot interactions by adapting themselves to environmental changes.

We have found several approaches coming from Entomology investigations (PITONAKOVA, CROWDER; BULLOCK, 2016a) to theory from Operational Research (YAN, JOUANDEAU, ALI-CHéRIF, 2012) passing through Combinatorial Optimizations (AKBARIMAJD, SIMZAN, 2014) that engineers and computer scientists applied in Robotics. These approaches have formed entire fields of research and developed complex strategies that can deal with real robot conditions and restrictions. For instance, according to Cao, Fukunaga and Kahng (1997), some of the multi-robot strategies can be traced back from Distributed Artificial Intelligence. In this approach, autonomous learning nodes processing and analyzing large amounts of data could solve complex problems. Afterward, it branched and brought forth multi-agent systems, which considers agents that also exhibit autonomy and work in a virtual world (CAO, FUKUNAGA, KAHNG, 1997). Likewise, similar approaches from different fields underlie some of the strategies that are applied in Robotics. Such transformations are due to multi-robot systems deal with individuals that can move in the physical world and must interact with each other beyond generic solutions that abstract inherent complexities of these systems.

Modern applications with swarms of robots range from containing wildfires (PHAN, LIU, 2008; PESSIN et al., 2010), detecting intruders (RATY, 2010; KHAN et al., 2016), exploring unknown areas (HUNTSBERGER et al., 2003; ANTOUN et al., 2016), and foraging for objects (BRUTSCHY et al., 2014; PITONAKOVA, CROWDER; BULLOCK, 2016b). In those scenarios, robots should be able to sense environmental information and make their own decisions (a.k.a. divisional autonomy). As insects, robot behaviors need regulation, preferably, through distributed control rules (a.k.a. decision-making strategies). Thus, they can coordinate between them to look after the group benefit. In other words, decision-making strategies should allow divisional autonomy and provide control rules that establish the boundaries of robot autonomy (ŞAHIN, 2005).

Among the possible task to tackle with a multi-robot system, foraging is a canonical problem for the study of robot-robot cooperation. One of the first works that define foraging as we used here is (DROGOUL, FERBER, 1993). They defined foraging as a task where individuals collect items from the environment to store them in a deposit or nest. In their experiments, robots create a trail of crumbs while going to the nest to attract more robots to the place where they found the object. Robots can choose a trail of crumbs randomly. The trails (dis-)appear as the robots pick or drop the crumbs. The authors measured the efficiency of the behaviors in foraging robots through the amount of collected items. As an implicit communication strategy, the trail helped robots to reach the desired behavior if the crumbs (dis-)appear at a middle slow rate.

We note in the extensive surveys (BAYINDIR, 2015; RAMAN, KRESS-GAZIT, 2013; MOHAN, PONNAMBALAM, 2009) that several authors investigated foraging models tackling different aspects: energy consumption, traveled distance, conflicts, time to deliver some objects, the number of delivered objects to the storage location. In this work, we try to increase the number of delivered objects with two kinds of strategies to deal with multi-robot systems: task partitioning and task allocation. The aim of multi-robot task allocation is to assign R robots to J jobs (GERKEY, MATARIć, 2004). The other strategy, a multi-robot task-partitioning solution, focus on the division of the task into consecutive or parallel subtasks (BRUTSCHY et al., 2014). The following sections detail about the related works in each kind of strategy. We avoid chains of robot formation because we want all robots foraging.

2.1.1 Related works in task allocation

Gerkey and Matarić (2004) provide a complete review of multi-robot systems where task allocation was the core strategy. The authors also highlighted two methods to achieve cooperation between robots: emergent and intentional. The first method allows for robot interactions without explicit communication during the robot allocation to tasks. However, these solutions make the prediction analysis a difficult, if not impossible, goal. On the other hand, intentional coordination requires robots to communicate and negotiate with one another to decide on which task each would work. Theory from operations research and combinatorial optimizations underlies several of these approaches of task allocation. Some of those approaches borrowed concepts like auctions (VIGURIA, MAZA, OLLERO, 2007), market-based processes (AKBARIMAJD, SIMZAN, 2014), game theory payoffs (ABDALLAH, LESSER, 2006; MARDEN, ARSLAN, SHAMMA, 2009), utility functions (LERMAN, GALSTYAN, 2002), and the like, to help robots in coordination.

One of the former works with negotiations through auction processes is (BERTSEKAS, 1990). The author implemented an auction strategy in a distributed fashion to assign jobs (objects) to individuals. The algorithm begins with a pre-assignment of the values of the jobs. Then, it runs a bidding phase where individuals offer a value for the job they prefer. Finally, in the assignment phase, the highest bidder receives the job. There are extensions of this auction process with iterative reassignment and local bidding (SUNG, AYANIAN, RUS, 2013; SCHNEIDER et al., 2015), but the idea is the same: to assign the jobs (or objects) to the best bidder.

Generally, auction-based approaches consider one or several auctioneers, which control the negotiation process. In this scenario, is common that centralized approaches may run faster than distributed ones (PARKER, 1998). However, their communication overhead could be higher because all member should share information with a central unit (VIGURIA, MAZA, OLLERO, 2007). Parker (1998) suggests centralized solutions as a more likely option for middle-scale systems and distributed implementations, in which there is no central controller, for larger scales. The works available in (SANDHLOM, LESSER, 1997; LUO, CHAKRABORTY, SYCARA, 2013; SCHNEIDER et al., 2015) confirm that distributed solutions fit better for larger scales. Moreover, distributed algorithms become more suitable and scalable having a bounded bandwidth to transmit messages. In some cases, a temporal (semi-)central auctioneer takes control of a local bidding process with its closest bidders. When the auction ends, another auctioneer rises in another subgroup (SANDHLOM, LESSER, 1997).

Authors also achieved task allocation solutions with auction-based and market-based approaches in transportation tasks (KUBE, BONABEAU, 2000; CHAIMOWICZ, CAMPOS, KUMAR, 2002; VIGURIA, MAZA, OLLERO, 2007; COSTA et al., 2011). Notwithstanding, the problem continues in terms of jobs and workers that must maximize the overall performance. Generally, the impact of robot selection is only known after the robot takes action according to its job or task decision (DASGUPTA, 2011). This means that further adjustments could take place if they do not coordinate well. Here, our focus is on object transportation. Despite search and rescue missions could be related to this kind of task, we did not deepen into them (AHUJA et al., 2002; LIU, NEJAT, 2013).

One of the former and most cited works in object transportation with task allocation is (KUBE, BONABEAU, 2000). The authors tested MURDOCH, an auction-based task allocation system in a box-pushing experiment with a heterogeneous team of robots. After this seminal work, several variations of auctions strategies appeared in multi-robot systems, e.g., the role assignment strategy, which also allows repetitive reassignments according to environmental conditions. Chaimowicz, Campos and Kumar (2002) introduced roles as actions robots must perform to complete a task. This idea of roles came from robot-soccer domain, where each robot calculates its utility for each role and broadcasts these values to its teammates. Thus, teammates coordinate among them who would take each role (STONE, VELOSO, 1999). Next, in the transportation task of (CHAIMOWICZ, CAMPOS, KUMAR, 2002), when a robot finds an object, if it needs help, it shares information about the available role and its utility. However, behind role assignment, the bidding processes of auctions continue. The robot that needs help offers the role as a job and makes a call of helpers. Robots interested in that role have to send a message indicating the value of their interest in that role. The robot that called for helpers chooses the best qualified among those that considered its offer.

Viguria, Maza and Ollero (2007) introduced another variation of market-based algorithms for recursive allocation of services called S+T. The basic idea of this algorithm is that a robot can ask others for services when it cannot execute a task by itself. They introduce the concept of services, which robots can complete with the help of another robot when they are unable to execute a task. They only consider independent and loosely coupled tasks, i.e., tasks where robots need few or no information about the state of other tasks. Costa et al. (2011) employed this idea to move large objects by contracting other robots for pushing services whenever a robot cannot move the object alone. Then, those robots that accepted the service can contract other robots if they cannot move the object, i.e., it is a recursive call of workers.

Utility functions help to propose another kind of strategies. In particular, some task allocation strategies to solve foraging problems were based on utilities removing the bidding process. For instance, Lerman and Galstyan (2002) examined two scenarios: one where robots only collect items and other where robots must deliver collected items to a nest. In the latter, their experiments show a decreasing average return effect, also known as the law of diminishing returns. Loosely speaking, this law states that each additional robot working on a task would increase the performance, but the size of its improvement is gradually lower until the group reaches a number where its performance declines (FÄRE, 1980). The authors explain that robots begin to interfere with one another while exploring or going after an object because they are foraging in a bounded space. This impedes the progress of the foraging task. When the group increases in number, robots begin to spend more time in avoiding actions than in the gathering of items. Lerman and Galstyan (2002) found the optimal quantity of robots in a particular region through the utility function of that region. Beyond that number, the benefits of parallelism that the task allocation solution could deliver begin to disappear.

We also found alternative solutions following heuristics based on cost functions (YAN, JOUANDEAU, ALI-CHéRIF, 2012) and task conditions (AKBARIMAJD, SIMZAN, 2014). In (YAN, JOUANDEAU, ALI-CHéRIF, 2012), the robots could reduce the total transportation time while keeping a low energy consumption on each of them. Robots have previous knowledge of the object positions and can employ path-planning strategies to avoid collisions in their collective transportation. The environment includes a place of constant production of goods, but their rate of production is unknown. The authors implemented a heuristic that helps robots to estimate the rate of production and define their idle periods (when their energy consumption is low). Finally, they compared their heuristic solution to the centralized *replanner* from (WAWERLA, VAUGHAN, 2010) and the results show that: *i*) their strategy was faster and *ii*) the cost of the required energy is only a few more than the required by the centralized *replanner* solution.

Akbarimajd and Simzan (2014) mention some alternative heuristics to pick the first task to forage: *closest, most starved, most starved and most complete, most proximal.* The *closest task* relates to the physical distance between the robot and the task, while the *most proximal task* considers the location of other neighboring robots. A *starved task* means some robots are lacking there, while the *most starved and most complete tasks* are the ones almost complete but lacking robots to finish them. For each of the heuristics, the process to find the optimal allocation resembles genetic algorithms where possible solutions are calculated and the best ones survive. However, with these heuristic rules, tasks enter a ranking process and the best ones win workers. Their strategy limits the decisions of the self-interested agents by a superior goal, the one of the society.

Finally, the most accepted taxonomy of the classification of MRTA problems was found in (GERKEY, MATARIć, 2004), which divides problems as follows:

- **Single (ST) vs Multiple Tasks (MT):** refers to the number of tasks a robot can carry out simultaneously
- **Single (SR) vs Multiple Robots (MR):** refers to the number of robots needed to fulfill a task
- **Instantaneous (IA) vs. Time-extended Assignment (TA):** refers to the available information for planning future allocations

Although our robots reallocate themselves dynamically due to dynamic environmental conditions, we can consider that our proposal belongs to the group of ST-MR-IA, because a robot can handle one object at a time and several robots have to forage for objects to complete the task without planning (loosely coupled tasks). In particular, we included the idea found in (YAN, JOUANDEAU, ALI-CHéRIF, 2012), regions in the proposed environment have unknown rates of object production. Thus, the number of robots in each region would depend on these rates. As in (LERMAN, GALSTYAN, 2002), we found the optimal number, but their initial distribution was different from the optimal one. In other words, we have the optimal number of robots for the environment, but the strategies must find the optimal distribution according to the rates of the three regions.

We associate each region to mathematical functions that satisfy the law of diminishing returns to regulate robots autonomy. Here, robots have to choose the region where they get the best utility return, which avoids a bidding process. Therefore, our robots did not employ auction processes nor heuristic solutions. Their decisions are based on this law to reach a near-optimal distribution in the three regions.

Furthermore, we opted for a solution following an intentional coordination fashion, that is, robots used explicit communication and possibly outdated information to distribute themselves between regions according to their estimations. If a robot informs that its region is providing more objects, most robots would end up arriving at that region generating a massive movement that could increase congestions and deteriorate team performance. We tried to regulate movements by providing local rules. Besides, different structures of communication were explored in the strategies to find an optimal one that balances between inhibition and stimulation yielded by shared information. These rules and communication structures should decrease the congestions and increase team performance because robots forage in a bounded space with limited resources.

2.1.2 Related works in task partitioning

In this scenario, most of the works employ adaptive solutions that are based on probability functions. One of the former foraging mathematical models we found is Deneubourg et al. (1991). The authors replicated the conditions where insects may pick up and sort items of different kinds in a parallel manner. Their environment is a grid of points where agents may move and leave traces of odor in four directions (north, south, east, and west). It is important to recall that agents work with more capacities and fewer complexities than robots, in this case, leaving odor traces with a discrete range of movements. In particular, the team depends on these environmental clues to indicate other members where they can make clusters of objects by kind. Agents decide between the clusters of objects through probability functions. Thus, the bigger the cluster, the more probability to drop a similar item there. For instance, a cluster with much food becomes more attractive for agents to drop more food there. Deneubourg et al. (1991) proved that their agents could complete the task through those probability functions.

When this kind of solutions includes task partitioning options, the robots can increase the group performance by organizing the task because the task is divided into subtasks that robots can execute in parallel or sequentially. Particularly, we found that task partitioning strategies in foraging with sequential division afford two options for study: using direct (ANDERSON, RATNIEKS, 1999; BUCHANAN, POMFRET, TIMMIS, 2016) and indirect transferences (PINI et al., 2011b; PINI et al., 2013). Direct transferences tend to increase the required time to complete a subtask because both robots have to gather in the same area and transfer the object from one to another. On the other hand, robots partitioning with indirect transferences drop the object in a common area where a partner would pick it up later. Nonetheless, the kind of transference depends greatly on the robotic architecture that researchers employ.

We found some of the first models that included the possibility of task partitioning with adaptability in (BONABEAU et al., 1997; THERAULAZ, BONABEAU; DENUEBOURG, 1998). These works considered local adaptive solutions to endow the group with flexibility. Notwithstanding, probability functions are behind robot decisions. In (BONABEAU et al., 1997), the group could deal with foraging through probability thresholds. Such thresholds depend on the ratios of workers performing an activity. Each worker has an activity-associated stimulus. The emission of pheromone of other members influences the stimulus of a worker. Based on this stimulus, probability functions indicate workers when they should switch from active to inactive state and back. In other words, workers can abandon an activity or continue struggling according to their stimuli. In particular, when the number of workers exceeds the demand of an activity, their stimuli trespass a threshold and push some of them to abandon that activity. Note that if the coordination through these stimuli is wrongfully handled, robots might keep switching from activity to activity, or worse, they might remain in a particular activity and overload it. The authors could adjust these stimuli by including local and global information: the number of workers in a particular activity and the number of individuals in the colony. Thus, each worker could estimate how many workers should attend that activity and employ its probability function to decide whether to stay or abandon it.

Theraulaz, Bonabeau and Denuebourg (1998) introduced specialization of workers through the adjustment of individual thresholds. Each worker can adjust its threshold as time passes. If the performed activity is progressing, the threshold decreases, else, it increases. Since the authors defined the initial thresholds randomly, workers tend to specialize in different activities. In particular, workers with an initial low threshold for an activity will have more probability of doing that activity. Each worker decreases its threshold of an activity after it performs that activity. Otherwise, if the worker fails to complete that activity, it would increase that activity threshold. Eventually, workers specialize in activities for which they are good. Notwithstanding, workers may abandon any activity with a fixed probability, but if the stimulus continues strong, they could re-engage in it. Note that, in both former models, the authors provided probability functions and assumed that individuals can have access to special task-related information. In distributed multi-robot systems, this kind of information is commonly unavailable, e.g., robots do not know how many of them are working nor do they measure the activity progress.

Some recent approaches (PINI et al., 2011b; PINI et al., 2013; PINI et al., 2014) analyze the advantages and bounds of task partitioning in foraging by using only local task-related information. In these works, the authors partitioned the environment into smaller regions to allow transfer the objects. This also facilitated the search process, which is one of the most time-consuming processes of foraging. Through sensory and pattern-based search algorithms, robots explored the unknown environment, which includes a North Star guidance and a set of light-colored booths. Brutschy et al. (2010) proposed the light-colored booths and called them TAMs (Task Abstraction Modules). TAMs work as a stoplight: showing through colors if there is an available object or room to store one object. Robots that lack handling-objects hardware may enter into them and wait, emulating to pick or drop a virtual object. Besides, TAMs can store and share information with robots. Those booths could avoid depositing information in the environment, as pheromone trails (HOLLAND, MELHUISH, 2000), crumbs (DROGOUL, FERBER, 1993), or clusters (DENEUBOURG et al., 1991).

TAMs were successfully employed in (PINI et al., 2011b; PINI et al., 2013) to test their strategies for coordination. Pini et al. (2011b) and Pini et al. (2013) included some TAMs in the environments and offered robots two options for object transportation. There is an area for indirect transferences to pass objects from one region to another and an alternative path to transport the object without partitioning. This path links both regions allowing each robot to carry from a source to a nest the object by itself. In the area for transferences, there are several couples of TAMs working as a cache room, where objects that are dropped in one side become available in the other region. Thus, the authors could manipulate the required time to transfer an object (delay). In some conditions, delays were short and robots should opt for transferring the objects. In other conditions, delays were longer than traveling from a source to a nest and then robots should prefer to transport the objects without transferring it. Unlike traditional foraging with robot-landmarks (KURAZUME, HIROSE, 2000), bucket brigading (DROGOUL, FERBER, 1992), or market-based strategies (AKBARIMAJD, SIMZAN, 2014), with TAMs, all robots were foraging in (PINI et al., 2011b; PINI et al., 2013).

We noted that some of the current works (PINI et al., 2014; CASTELLO et al., 2016; BUCHANAN, POMFRET, TIMMIS, 2016) allow robots to adapt their parameters for making decisions while foraging in the environment (i.e., online). In (PINI et al., 2014), robots can get an estimated time to complete a subtask in order to adapt their decision-making strategies. Based on these estimations, robots use probability functions to decide whether to partition or not. Their robots can handle physical objects and employ direct transference. However, if a partner did not appear to complete the transfer, they can stop waiting for its arrival and go to the nest without transferring the object. Robots can also adapt their exploration region based on previous experiences. Likewise, in (BUCHANAN, POMFRET, TIMMIS, 2016), robots adjust the traveled distance depending on their success in finding objects and partners. With this model, robots reached the solution of an optimal partitioning distance for direct transferences, which was found in (PINI et al., 2014). In (CASTELLO et al., 2016), robots use a θ-threshold to decide whether to search for objects or remain in the nest. They can adapt this threshold value as a response to abrupt changes in the availability of food. Therefore, novel strategies of coordination for foraging must allow for the change and adjustment of some parameters online and locally, i.e., while each robot is foraging.

2.1.3 Related works in other fields

Some authors already employed concepts of complex networks to model their systems and evaluate the dependence of the individuals on the network structure (VISSCHER, SEELEY, PASSINO, 2006; HSU-SHAN et al., 2013; COUCEIRO et al., 2014). Visscher, Seeley and Passino (2006) abstracted a swarm of bees as a decentralized complex network. Each node exploit the perceptions of its neighbors to make its own decisions. This model allowed to identify the structural requirements for the team behavior to emerge. Hsu-Shan et al. (2013) employed a centralized network to coordinate a team of robots in a rescue mission. Their robots identify fire hotspots and share this information with a central unit, which helps the team to find safe paths of evacuation. Couceiro et al. (2014) worked with a decentralized network of robots for exploring unknown environments. In particular, two kinds of robots create a MANET (Multi-robot Ad-hoc Network). Big robots act as a router for the subgroup of small ones exploring the environment may publish information in the network. However, in these models, the authors did not consider structural parameters to improve their strategies.

Unlike robot networks, in sensor networks, researchers have introduced structural parameters to improve their strategies (ALBERT, JEONG, BARABASI, 1999). Several authors have followed this trend to work with complex networks of sensors (WU, TSE, LAU, 2014; RAMOS et al., 2014; PERILLO, HEINZELMAN, 2005; YIN et al., 2013). Commonly, sensors are static nodes in the network and they have to gather environmental information. Nodes create a path such that the information reaches a central unit that stores and processes it. Wu, Tse and Lau (2014) and Ramos et al. (2014) employed a parameter named *betweenness centrality*, which is a global measure of the number of paths that pass through a node over the total number of paths in the network. In particular, Wu, Tse and Lau (2014) decreased the number of congestions in *hub* nodes (nodes with a high number of connections), while Ramos et al. (2014) extended batteries lifetime.

The *degree* of a node is another structural parameter that is commonly considered to achieve some improvements in sensor networks. It quantifies the number of links a node has. This parameter helps to classify networks according to their degree distribution, e.g., scale-free, regular, random, among other structures. Albert, Jeong and Barabasi (1999) introduced this parameter in a strategy for routing called MDS (Maximum Degree Strategy), which achieved a lower number of messages to gather the environmental information. Afterward, Yin et al. (2013) included another structural parameter called the *clustering coefficient*, which is a local parameter that measures the fraction of neighbors of a node that are also connected among themselves. By combining both, degree and clustering coefficient parameters, the authors sped up the gathering of environmental information.

These works show that the structure of communication can affect the performance of the entire system. Furthermore, sensor networks showed that there are structural parameters that can bring benefits to the dynamics of the team. It seems promising to find a structure where robots may exploit their neighboring information. However, communication between robots requires more than an optimal structure. There is more to explore: What information can they share? How many neighbors should a robot listen to? How long should the range of message transmission be? The following works enlight the reader upon these questions and deal with foraging tasks.

Hoff, Wood and Nagpal (2013) fostered adaptability by providing the swarm with a set of sub-models to forage whose activation depends on incoming information. They followed some of the criteria of swarm robotics proposed in (ŞAHIN, 2005): robots are homogeneous and do not have a leader nor global information. The authors established a sequence for switching between three sub-models. First, the swarm begins to forage with a gradient-based sub-model, which is similar to the model presented in (SUGAWARA, KAZAMA, WATANABE, 2004). In that model, robots employ direct communication to transmit information about the nest/source locations and, thus, they could create a path between both. The second sub-model is called area-sweeping foraging and employs the idea found in (BALCH, HYBINETTE, 2000). This model allows robots to navigate by employing social potential fields. Finally, their last resort is a random exploration. In particular, the connectivity was fundamental for the group to reach an almost synchronous switching between sub-models. Robots were sharing and replicating information about the success in finding objects with the active sub-model. Whenever a robot finds food, it delivers a positive message informing about the source location while it walks back to the nest. But, if no robot finds food, they would not listen to any good news and, after a while, they would note that things are going bad with the current sub-model. After a period of unsuccessful search, the robots begin to transmit a signal indicating a change from the current sub-model. Thus, through communication, the group can detect if the active sub-model is failing (working), and robots can switch (stay) -as a group- to another (in the same) sub-model. The authors concluded that if food sources appear at fixed distances, a single static sub-model might work well. However, if food sources appear at different distances, their proposal has better performance than any of the three base sub-models.

Wei, Hindriks and Jonker (2014) studied the effects of communication in the coordination of a swarm of robots. The robots dealt with concurrent and consecutive tasks. In the latter, communication allowed robots to obtain greater improvements. Inspired by human communication, the authors proposed messages with more complexity in the information-sharing strategy. They did not focus on the range of communication but on the content of the message going beyond the results found in (BALCH, ARKIN, 1994; SUGAWARA, KAZAMA, WATANABE, 2004). Unlike those two works, the robots in this model shared messages with their goals and perceptions with one another. This leaded the team performance improved further, because robots sped up their coordination when they shared information about the room where they are/were working (i.e., goal) and the color of the object they found there (i.e., perception). Since a robot could share a message with all members of the team, it would influence their future locations. In particular, a robot may change its goal by considering information it listened: if another robot was there and found objects of the color it needs, that room becomes more likely its goal. Otherwise, if it needs another color, this information would push it toward other rooms. Finally, the authors concluded that to enhance performance, it is better for robots to share only perceptions. Thus, a robot can have more information of the object locations in the environment and forage for objects they need.

Sarker and Dahl (2011) followed a similar idea of stimulus from some of the first models in foraging (BONABEAU et al., 1997; THERAULAZ, BONABEAU; DENUEBOURG, 1998). They modeled the system as a bipartite graph holding only robot-task connections. In their model, the weight of the connection represents the stimulus of a robot for a task. Thus, a robot depends on the weights of its connections to engage in some tasks. Robots can learn and specialize in a particular task by reinforcing or decreasing the weight of its connection with each task. If the weight of each connection is low, robots can forsake a task. This forsaking skill allowed the swarm to avoid stagnation by over-commitment, i.e., to be able to continue exploring other alternatives besides the best option. To test the effects of communication, the authors worked with two swarms, each including only one kind of robots: independent ones, which have global sensing and no communication between them, and cooperative ones, which have local sensing and robot-robot communication of short range. The independent robots disregard experiences of others, replicating the effect of fast rates of pheromone evaporation. For the cooperative robots, the authors adapted some ideas found in (YOSHIDA, ARAI, 2000) to maximize information spread. They reached the effect of moderate rates of pheromone evaporation with explicit communication. By comparing both swarms, cooperative robots could reach better/similar results than those obtained with global sensing and no communication.

Pitonakova, Crowder and Bullock (2016a) found an interesting way to foster the team adaptability. They regulated direct information sharing to achieve a balance between exploitation and exploration of new information. A robot shares information related to the source it found while returning to the nest, which it can reach by using phototaxis. Its teammates navigate toward the reported source by integrating their path through a transformation of the shared information clues into a navigation route, as in (MILETITCH et al., 2013). If they do not find the source after a while, they start a random search in that near area. The authors modified several communications ranges in different scenarios where robots behave like *beggers*, that is, robots acting as recruiters could request information from other recruiters. Then, they could make a decision on whether to switch to another source or remain in their current one. When sources were too scattered, the best strategy was individual foraging with short-range communication. However, if the objects were in a heap, it was better to have a swarm of long-range *beggers*, because they created a strong commitment to a single heap. Then, they challenged the swarm in environments where the qualities of the sources changed, but not their location. They noted that long-range *beggers* disabled the swarm to self-adapt to these environments because of over-commitment, which is similar to the long-lasting pheromone effect. On the other hand, the short-range *beggers* lead the swarm to adapt and establish themselves to new profitable sources, i.e., they could adapt as bees do to new environmental conditions.

2.2 Simulation software and robots

We worked with e-puck robots. We opted for this robot because they are an academic platform offered as an open hardware concept, where all documents are distributed and submitted to a license allowing everyone to use and develop for it. This robotic platform appeared in 2004, since then, several universities have worked with e-pucks (EPFL, 2010). Moreover, our laboratory has some e-pucks. Here, we list the components of an e-puck robot. The hardware available in a standard e-puck includes a dsPIC microprocessor of 40 MHz, 2 motors, 2 encoders, a VGA camera, 8 distance sensors, Bluetooth, 10 leds, 1 accelerometer, 3 microphones, and 1 speaker. E-pucks include Bluetooth for communication, for loading the program, and for remote control. Leds help us to have an indication of whether the robot has a load. There are available several hardware extensions to improve e-puck capacities, but it means that the cost of the research would increase with each robot.

Notwithstanding, in our laboratory, the number of robots was not enough when we began this research. Thus, we opted for exploring several simulators that include the e-puck in their libraries in order to pick the best for our robots. Among those simulators, we found that some authors had implemented their models in Gazebo (KOENIG, HOWARD, 2004), V-REP (ROHMER, SINGH, FREESE, 2013), ARGoS (PINCIROLI et al., 2012), and Webots (MICHEL, 1998). We note that e-pucks in some of those simulators can perform unreal actions such as detecting occluded targets, move fast even while processing images, and robots do not create shadows. In addition, physics of e-puck acts differently. We found that Webots offers the smallest reality gap for e-pucks and delivers the following advantages: *i*) it simulates visual information of the e-puck camera, *ii*) it allows to perform remote control, and *iii*) it also converts the code directly into a .hex file to transfer it to the real robot. It only does not emulate the audio components of e-pucks. However, in our simulations, neither audio nor accelerometer nor encoders component were necessary.

In (EPFL, 2010), the official webpage of the e-puck, they recommend Webots and V-REP for 3D simulations with e-pucks. However, we compared the e-puck robot features in Webots against other simulators. The version of Gazebo we installed does not include the shadows, camera information, and the remote control option. When we were almost finishing this research, the new version of V-REP introduced the camera-like sensor and hardware extensions. Unlike Webots, V-REP does not offer the possibility for e-puck remote control option. Some authors that worked with e-pucks in ARGoS did not mention the version they employed, but they provided videos of the simulations in their supplementary material (PINI et al., 2011a). Figure 1 shows the visual differences of e-pucks in ARGoS and Webots simulators. Note that Webots have more colors and visual details.

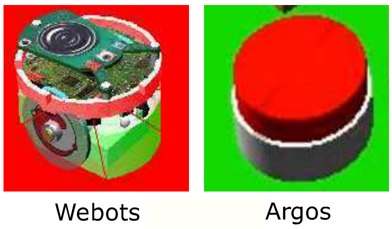


Figure 1 – E-puck robots in the respective simulators. The ARGoS image was taken from videos of the supplementary material found in (PINI et al., 2011a). The e-puck photo is from Webots 7.4.

Moreover, only Webots offers the possibility of creating the .hex file to transfer it into the real e-puck. There is an option to create this file in MPLAB X16 Microchip programmer. Therefore, our final decision was to employ Webots simulator.

2.2.1 Communication of e-pucks in Webots

Since we are exploring communication network structures to improve team adaptability, we had to check whether the simulator emulates real communication details. Fortunately, another advantage of Webots is how it emulates the communication steps. Real inter-robot communication is asynchronous, which means messages can remain in the buffer after they are emitted. This would affect robot learning, i.e., it is necessary to consider this in simulations. Figure 2 shows the process of communication in Webots. Note that Webots approaches its simulations to reality. The reader can find more details e-puck communication in (COUCEIRO, VARGAS, ROCHA, 2014).

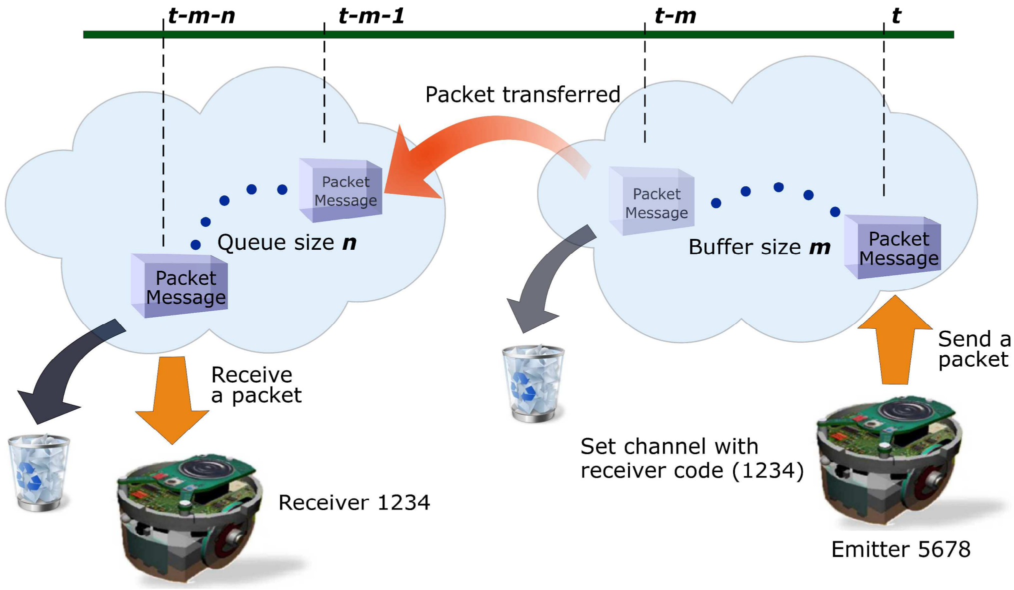
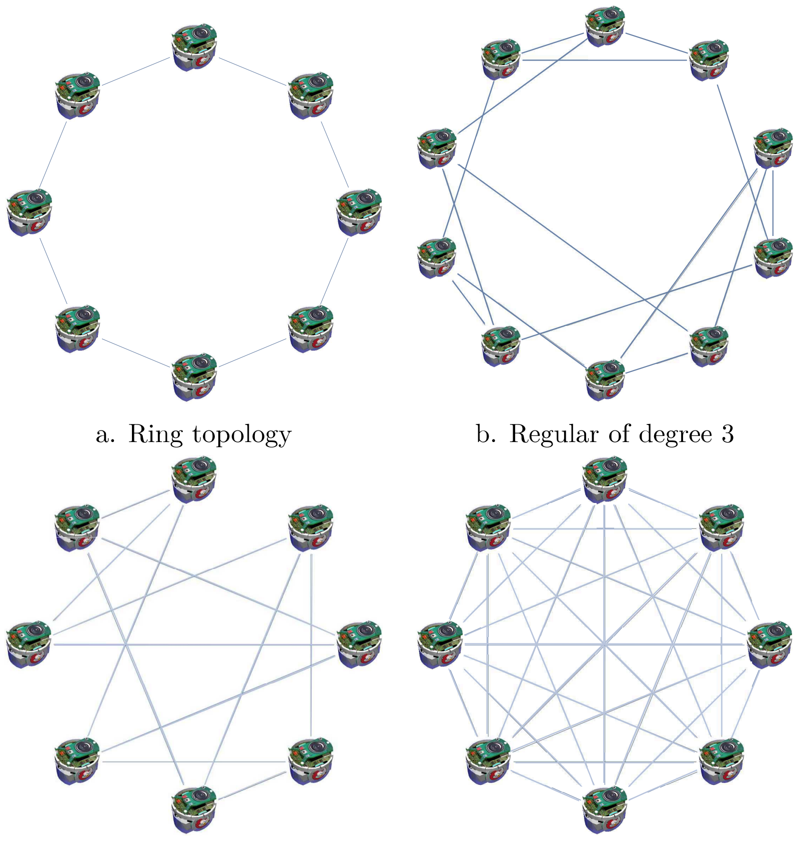


Figure 2 – Steps for inter-robot communication in Webots. Data flows from right (emitter) to left (receiver) using a packet queue.

Static communication structures

Recall that one of the key points of this thesis is to explore communication structures to find one that brings improvements no matter which environmental challenge robots have to deal with. However, previous works indicated that changes in environmental conditions could demand a more complex and adaptive structure than a static one. Therefore, we need to evaluate the strategies without communication (no connections, N), and over the following static communication structures: *i*) ring topology (R), *ii*) a regular of degree 3 (Rd3), *iii*) a small-world (SW), and *iv*) a fully connected one (F), which are commonly studied in complex networks. Figure 3 shows an example of these structures with eight e-pucks.

Each of these structures provides different advantages and weaknesses. For instance, the ring topology is a regular network of degree 2, where each robot shares information with two partners. This kind of information-sharing structure guarantees a boundary in time for all robots to receive a message from any other robot, but it lacks resilience, i.e., the capacity to resist failures or random attacks. A small-world structure emerges when some links between nodes are randomly changed from a regular topology (e.g., a ring topology). It offers a better resilience than a regular one, but does not guarantee that all robots would learn at a similar speed through the incoming messages. Finally, a fully connected topology offers both a high resilience and a high speed in learning, but its costs could increase exponentially as the team size increases. Therefore, first, we need to perform an evaluation of these static networks and, thus, we should achieve some preliminary conclusions about the communication structure effects. Therefore, this research should begin by identifying which structure(s) offers the team a good support



a. Ring topology b. Regular of degree 3

b. Small-world c. Fully connected

Figure 3 – Samples of communication structures including eight robots and their links for information sharing. Robot degree increases toward the right and down.

for their performance in different and dynamic environmental conditions. If there is no such structure, we should propose one that improves team adaptability.

2.2.2 Tools for simulations

Our license in Webots is Educational, which means it does not allow to exploit computational resources entirely. Despite we worked in a Dell computer with an Intel i-7 processor of the third generation, 8 GB of memory RAM, and a Nvidia board of 2 GB for dedicated video, the simulations ran on speeds lower than 0.4X. In particular, simulations with teams of more than six e-pucks in Webots take too long. For instance, a simulation of two hours in Webots could last up to six real hours. Such restriction made initial exploratory experiments expensive. Therefore, we decide to use a simpler and lighter platform to execute exploratory experiments. We opted for Netlogo 5.3.1 (WILENSKY, 1999), a multi-agent platform. Thus, we adjusted agent movements such that they ran as robots in Webots and their time to forage took a proportional time to forage in Webots. In particular, 2 hours in Webots $\approx$ 7000 steps in Netlogo, which only require 15 minutes. After the exploratory (and more massive) simulations in Netlogo, we used the promising conditions to perform more refined and realistic simulations back in Webots.

2.2.3 Task Abstraction Module - TAM

Since e-pucks lack handling capacities, we found a solution to emulate the picking up/dropping actions with TAMs (Task Abstraction Modules). These devices can abstract tasks as described in (BRUTSCHY et al., 2010), and later a formal modeling through Petri Nets was provided in (BRUTSCHY et al., 2015). Each TAM has a light that changes its color to show its current state, like a stoplight.

A TAM emulating a source provides a red light, when a robot seeking for objects detects this TAM, it would enter to pick a virtual object. Once the robot entered it, the TAM would change its light lest other robots try to enter. The robot would wait inside until the TAM changes its color to indicate a complete emulation of its picking action. Then, the robot could leave and the TAM restarts its process. This also works for a TAM emulating a nest, but this time, it would have different colors and its waiting time emulates the dropping action. Figure 4 shows the Petri model of the TAMs working as sources and nests, and the schematic of the hardware available in each TAM.

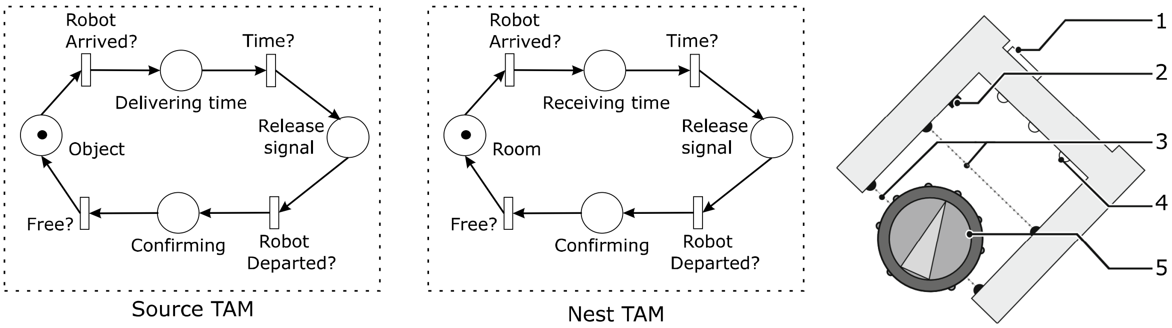


Figure 4 – Petri model of TAM devices working as a source (left) and as a nest (middle). TAM schematic proposed in (BRUTSCHY et al., 2015) (right): 1. IEEE 802.15.4 mesh networking module, 2. IR transceiver, 3. IR light barriers, 4. RGB LEDs, and 5. E-puck entering into the TAM.

TAMs working as caches enable the task partitioning option. In particular, a cache requires a couple of TAMs working as follows: one acts as a receiver -where the robot drops the object and transfers it to another region- and the other TAM acts as a deliverer -where robots may pick up the object later. Figure 5 shows a cache Petri Net model.

Note that a cache begins working as a receiver and can handle only one object at a time. The TAM working as a receiver has a color-specific light indicating that it is free. Once a robot enters this TAM, the light changes its color to indicate that it is busy. After a while, when enough time has passed for the robot to drop an object, the TAM changes its color to indicate a successful reception. Then, that robot can leave. At this moment, the other TAM working as a deliverer changes its color to indicate its availability and disables the receiver side. Once a robot enters to pick the object, the light changes to indicate that it is busy. After a while, when enough time has passed for the robot to pick

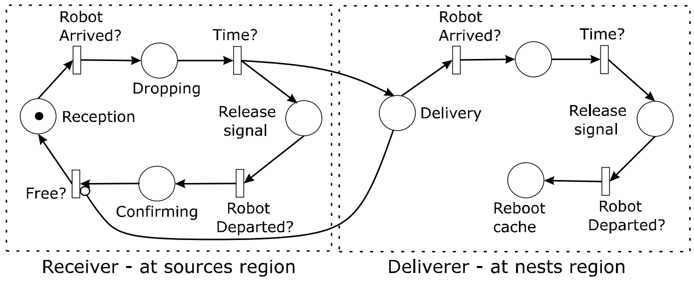


Figure 5 – Petri Net showing the functioning of a couple of TAMs working as a cache.

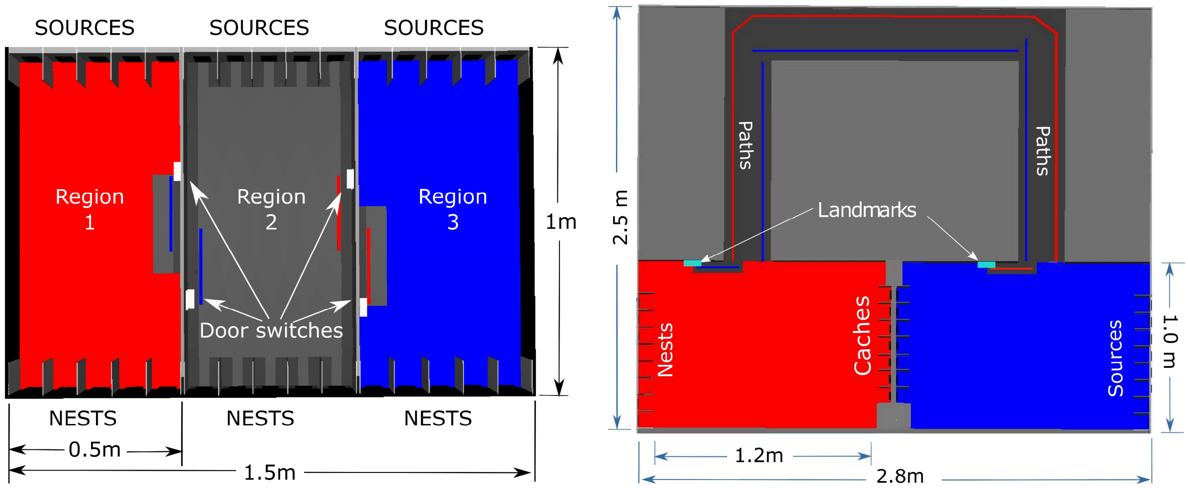
up the object, the light changes indicating to the robot that it can leave. This process repeats during the entire simulation.

The dimensions of a TAM are $9 \times 9\ cm^2$ in order that an e-puck robot can enter them, but they can be built to fit any robot platform. As proposed in (BRUTSCHY et al., 2015), these smart devices include a Zigbee component that allows them to generate a network for information sharing. Consequently, these devices can emulate more complex actions and tight-coupled tasks by communicating between them or with robots. Thus, we implemented TAMs as static robots able to share information in Webots. Since e-puck camera is also limited, according to (BRUTSCHY et al., 2010), e-pucks can distinguish only up to 6 colors with their embedded camera. The simulated TAMs could not exhibit more than this amount of colors. However, to overcome this limitation, we created codes by combining geometric shapes (rectangles and circles) and four colors to encode the available TAM states. The remaining two colors were employed for doors and path landmarks. For instance, a red rectangle code is exhibited by a TAM to indicate an object is available, while a white rectangle indicates a door. Such shape-color combination indicates the available resources and helps robots in their navigation process.

2.2.4 Navigation

Here, robot navigation does not depend on odometry; it depends on the visual clues detected through e-puck camera. Our simulated environments include geometric shapes and color codes to identify the available tasks and transportation alternatives (see Fig. 6 for details). This section provides navigation details, which were based on white-shark hunting strategies that were published in (NOGALES, ESCARPINATI; OLIVEIRA, 2017). We refer the reader to (NOGALES, OLIVEIRA, 2017a) for simulation videos in Webots.

The main aid for robot orientation while foraging is visual clues. Both, door switches, path landmarks, and TAMs are helping robots to orientate themselves while navigating. In the task allocation scenario, robots use doors to travel across regions. In the task partitioning scenario, they have an alternate path whose beginning is identified by two cyan landmarks. TAMs are working as sources, nests, or caches allowing them to emulate the repetitive picking/dropping actions of the objects.



Task allocation environment Task partitioning environment

Figure 6 – Dimensions and regions of the environments for each of the proposed strategies to tackle the foraging task.

Note that each region is short enough to avoid that robots waste too much time finding the visual clues during their navigation. Robot visual range is almost 40 cm, while regions have around 100cm in length. Their forward movements oscillate between 10 and 30 cm depending on the image features. Next, since sources and nests have a fixed location, any recording mechanism would make easier for robots the finding of the static sources. This would shadow the adaptability skills of the implemented strategies. Therefore, despite e-pucks have encoders, we avoided odometry and limited their orientation to visual clues to allow a small space for randomness.

Although robot navigation is random, they have a scarce idea of the distribution of the environment. When a robot finds a different visual clue from the one it is looking for, it knows it has to turn toward the opposite side to continue its search. For instance, a robot looking for a nest that finds a source should turn about 180 degrees and, then, it moves forward in its exploration. Thus, that robot avoids increasing traffic to those robots that are trying to reach a source. After dropping (picking) an object in a nest (source), a robot turns and begins its random search for a source (nest) because it does not remember the source or nest location.

Next, when a robot decides to travel toward another region, it has to find a switch indicated by a door switch or landmark, which are another kind of visual clues. In the task allocation environment, after a robot reaches a door switch, we force that robot to wait a time while the door opens. Once it opened, the door remains open and waits for enough time for the robot to pass through (supplementary material: entering door in (NOGALES, OLIVEIRA, 2017a)). In the task partitioning environment, robots have to reach a landmark, and then they can begin a line-following sequence to reach the other region.

As mentioned before, robot navigation depends mostly on visual clues, that is, on image processing. Some samples of the image database are illustrated in Figure 7. In particular, we used the color thresholding technique to segment the images and find the components in them. This technique depends on accurate color and illumination. However, we considered two light conditions to include little changes in color. Since our goal is to test the strategies, this problem can be controlled in laboratory experiments. Thus, we needed to find the histograms of the color distribution. Then, from the histograms, we find out the thresholds of the colors that worked for both light conditions.

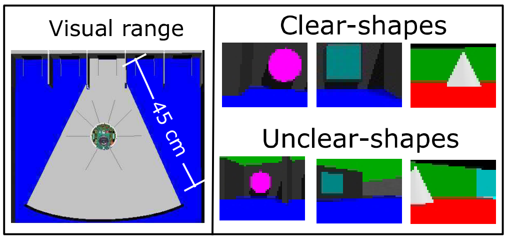


Figure 7 – Robot visual range (left). Samples of the geometric shapes from the image database (right).

Weka (HALL et al., 2009) helped us to identify the descriptors (area of shape and width of the bounding-box) to differentiate targets from the non-target shapes. Afterward, we included a second level of classification of targets: *unclear-shapes* and *clear-shapes*. The *unclear-shapes* cannot be distinguished due to distance or angle of perception, but once detected, the robot may approach and double-check them. *Clear-shapes* are trustable detections and, unlike *unclear-shapes*, they allow robots a faster approach. For the descriptors of this second stage of classification, Weka ranked the descriptors according to their *information gain ratio* and brought us the most contributive ones: eccentricity, extent, width of the bounding-box, height of shape, width of the shape. With these descriptors, we trained some supervised classifiers and compared the number of correctly classified instances (performance). In particular, Weka allowed us to test three classifiers: Decision table, Logistic regression, and J-48 Decision tree. The results of the training stage are in Table 1. Since there is a difference of only 5% between their performance, we opted for the J-48 Decision tree option because it has the lowest computational cost (rules).

Classifier	Rules	Performance
Decision table	56	89.3
Logistic Regression	36	85
J-48 Decision tree	25	84.8

Table 1 – Results of the classifiers tested in Weka. The column *Rules* means the amount of mathematical operations needed to do an object classification. The column *Performance* means the amount of correctly classified instances

To summarize, our database of images considers non-target and targets. Targets can be *clear-shapes* or *unclear-shapes*. Such classifier answers allow robots to imitate white shark's possible perceptions: non-prey, potential prey, or guaranteed prey. The following section explains how the robot switches its approaching behaviors based on these perceptions.

Shark-inspired search and approach

As white sharks can switch between behaviors, we allowed robots to switch between their approaching behaviors. Moreover, as sharks can change their sensory organs while hunting based on their perceptions, we also allowed robots to switch between their sensors too. When a robot is walking randomly in the arena, it stops periodically to obtain treatable images. In these images, the robot searches its target (a particular color and shape combination). If the robot detects a target and the classifier indicates that it is an *unclear-shape*, then the robot approaches cautiously to double-check. However, if the robot already double-checked this target, it returns to the random exploration. Whenever the robot detects its target and the classifier indicates it is a *clear-shape*, then it turns off the camera, enables its distance sensors, and increases the speed on each wheel by using the image information. Then, the robot steps forward between 10 and 30 cm according to the height of the shape. It goes faster for small images, because they are far. It goes slowly if the image is large, because it is near.

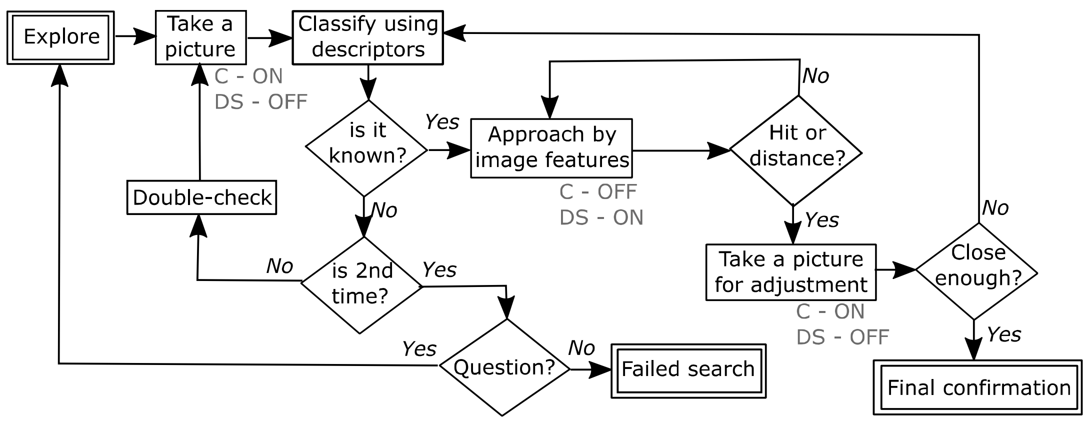


Figure 8 – Process for robot switching across its sensory systems inspired on white sharks (SKOMAL, 2015). The switch of sensors is in gray font, C for camera and DS for distance sensors. The Question? evaluation consider abandoning the search of that kind of target by declaring a failed search. The final target confirmation means a successful search and approach.

After moving forward some distance or if the infrared sensors detected an obstacle, the robot stops again, moves back few steps to avoid shadowing the target, enables again its camera and takes another picture to adjust its trajectory or to identify the obstacle, respectively. If it is an obstacle, the robot turns right, moves forward, and then it turns again to continue its walk.

As sharks need another sensory organ to make its final decision, our strategy includes a final validation. Robots do not need to bite its target. They just hit the target and move back a few steps. Since the robot visual field is saturated when there is lacking almost 6 cm to reach its target, once the robot hits the wall of the TAM, if the image is saturated with the target color, it means it has arrived. Thus, the search and approach are fulfilled successfully. The chart showing the *Shark-approach* strategy is shown in Figure 8. More details of this navigation proposal are in Appendix B.

Summary

We noted that the environments play an important role in the design. Here, we opted for TAMs as environmental aids for both task allocation and task partitioning scenarios. Both kinds of strategies offer interesting solutions to foraging tasks. The task allocation environment has only sources and nests. It consists of three regions whose rates of object production are different. The proposed solution borrowed an idea from a multi-agent strategy found in (NOGALES, FINKE, 2013), which robots should reach it through the proposed distributed control rules. In particular, TAMs keep track of the marginal utility of each region, while robots keep an estimation of it based on their own marginal contribution. The region marginal utility measures the amount of objects delivered within a period. Robots share their estimations through robot-robot connections, and TAMs share their more accurate information with robots of that region periodically. Robots based their decisions on these utilities. We explored the effects of different communication structures to find the best suited for robots to achieve a (near-)optimal solution. These results were already published in (NOGALES, OLIVEIRA, 2018b).

On the other hand, the environment with a task partitioning option includes sources, nests, and caches. Recall that each cache is a couple of TAMs programmed to deal with indirect transferences. We also handle the delay to complete the transference to challenge the implemented decision-making strategies as in (PINI et al., 2013). In this environment, TAMs do not share information with robots. Only robot-robot connections are available and each robot has to adapt its connections to deal with environmental changes. Since the solution emerges in this scenario, it is harder (if not impossible) to determine a solution beforehand. However, we wanted to improve team adaptability through the communication structure. A technique from data streaming for online learning helped us to solve the challenges of the second environment. These results were recently published on (NOGALES, OLIVEIRA, 2018a) and the dynamic network is submitted on (NOGALES, VARGAS, OLIVEIRA, 2018). The following chapters offer more details about these environments, the problem they propose, and the respective controllers, which help them to navigate and forage for virtual objects following either task allocation or task partitioning strategies.

Chapter 3

Problem

Foraging belongs to the set of canonical tasks that roboticists study to find real life strategies. It is a testbed to explore robot-robot cooperation. Drogoul and Ferber (1993) defined foraging as the activity where individuals have to collect items, which are dispersed in the environment, into a deposit or nest. Mainly, the efficiency of the designed behaviors in foraging robots is measured through the amount of collected items. Despite there are several ways to tackle this task, in this proposal, we focus in task allocation and task partitioning strategies. The rest of the chapter is organized as follows: Section 3.1 describes the task allocation problem and Section 3.2 the task partitioning one.

3.1 Task allocation problem

The idea for this first problem has its roots in (YAN, JOUANDEAU, ALI-CHéRIF, 2012). The authors created an environment where the sources have different rates of object production, which are unknown for robots. The robots have to find a balance between foraging and resting activities according to those rates. The number of active robots depends on the current rate. Here, we proposed an environment consisting of several regions whose rates of object production are unknown for robots. Moreover, robots do not have resting activities, they have to travel between regions and distribute themselves such that the entire group can be foraging. Thus, if a region has a greater rate of object production, it should have more robots and more collected objects. Consequently, robots would need local decision rules to distribute themselves across these regions because they have to find an optimal allocation based on their estimations about these rates.

In particular, we modeled the environment and its regions as a topological map. Figure 9 provides an example of a possible scenario of a topological map of the rooms of a warehouse. Let nodes represent the rooms, i.e., distributed locations where robots perform the object transportations. In this building, robots have to repetitively move objects within a room. However, rooms receive different amounts of objects (as different rates of objects production), which require a different number of robots. This condition would force robots to distribute among the rooms to serve them appropriately. In particular, the more objects appear in a room, the more robots should serve that room.

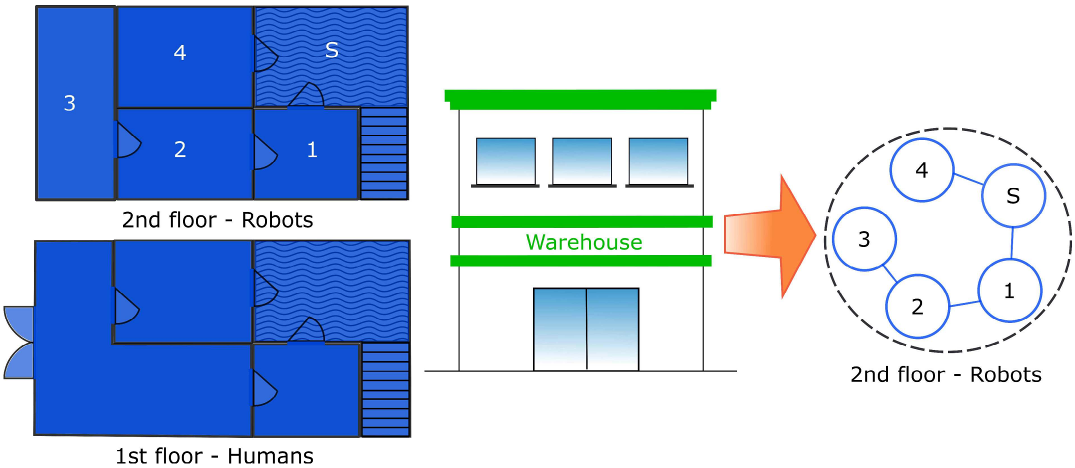


Figure 9 – Transformation of a warehouse scenario to our topological map. Small circles represent the nodes, while the dashed circle represents the floor.

Note that the topological model of the environment can be as complex as the designer needs. Moreover, following with the mathematical details of the topological model, nodes belong to a set N, indexed from 1 to n. Each node has a rate of object production defined as Op_i for node i. For a node i, the number of robots foraging in it is represented by r_i. Next, let $\Delta_q \subset \mathcal{R}^n$ denote the $(n-1)$ dimensional simplex defined by the equality constraint $\sum_{i=1}^{n} r_i = q$, where q denotes the quantity of robots available. The simplex guarantees that: i) the number of robots in any node is non-negative and ii) after their distribution, by summing up the amount of robots in all nodes, the total remains equal q. This restriction is necessary because, mathematically, we could have negative amounts of robots in any node or some of them could be outside regions. Next, the number of collected objects by having a number of robots r_i within node i is given by the utility function $f_i : \mathcal{R} \to [0, \infty)$. The total utility function is defined by $f : \mathcal{R}^n \to [0, \infty)$, $f(r) = \sum_{i=1}^{n} f_i(r_i)$, where $r = [r_1, ..., r_n]^\top$ represents the state of the system. Under the assumption of local communication and decentralized decision-making, the objective is to identify conditions that allow us to solve the following optimization problem

$$\text{maximize } f(r), \text{ subject to } r \in \Delta_q. \tag{1}$$

In other words, robots have to find the optimal distribution among regions such that they maximize the utility associated to each region. Since the utility measures the number of collected objects, it depends directly on the rate of object production. Therefore, we needed to design: i) an environment with several regions where e-pucks can forage, ii) local rules such that they may distribute themselves, and iii) a suitable communication structure with which the team performance improves. The proposed solution is in the following chapter.

3.2 Task partitioning problem

The task partitioning problem has its roots in *Atta sexdens* ants that exhibit an interesting behavior of task partitioning while foraging. Some ants climb up trees and drop leaves on the ground, where others would later pick them up and transport them to the nest (FOWLER, ROBINSON, 1979). Thus, the transportation is partitioned in two steps. These ants inspired Pini et al. (2011b), who proposed an environment including an option for robot task partitioning. Commonly, foraging robots transport objects from sources to nests through a path that links both regions. However, in this environment there is an area for transferences, where caches enable the partitioning option to forage. Their environment includes TAMs acting as sources, nests, and caches to enable virtual object foraging. (The reader may find more details of TAM functioning in Section 2.2.3.) When robots decide to partition the transportation, they leave the objects in a cache of the area for transference and other robot would pick them up to complete the task.

We took the model for foraging proposed in (PINI et al., 2011b) as our base model. It was later extended in (PINI et al., 2013) by allowing robots to share their experiences. This worked as social learning because robots learned from other members in the group. Their robot-robot connections built a communication structure upon which robots could share information. Since we want to explore information-sharing structures to deliver a communication strategy that improves team adaptability, these works including a task partitioning option were a good starting point. Besides, in both works, the authors worked with the same robotic platform, the e-puck, which is not task-specific designed. Some opportunities for new strategies and parameter adaptations emerged. Such improvements brought benefits to the team performance avoiding hardware-specific solutions.

About the environment in (PINI et al., 2011b) model

In this environment, robots have to transport objects from sources to nests repetitively, but each robot can transport an object at a time. Sources and nests have a predefined time for delivery and reception of objects, respectively. They also are in different regions and their location is static. However, the environment enables two options for carrying the objects from sources to nests: partitioning or non-partitioning.

The partitioning option allows robots to divide each transportation task into two sequential subtasks by employing an area for transferences. In this area, there are some caches that link both regions. Each cache is a couple of TAMs. One of its sides works as a receiver, while the other side delivers the object the receiver got. Thus, the deliverer side imitates a source once an object was dropped on the receiver side. (Petri models of TAMs workings as sources, nests, and caches are explained in Section 2.2.3). The non-partitioning option has an alternative path that links both regions. Through this path, robots can travel back and forth between regions carrying objects by themselves.

Robots should decide whether to partition or not according to the times they experienced in these options. For instance, if the area for transference is faster, robots should take advantage of this option and partition the transportation of their objects. Otherwise, robots should not to partition and carry the object through the path to drop it into a nest because the area for transference is taking too long. Robots should decide by which option they consider the fastest one. Figure 10 shows the schematic view of the environment and its options to forage.

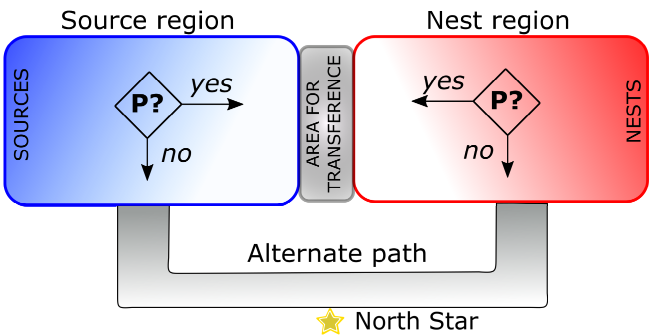


Figure 10 – Schematic view of the environment where robots forage for virtual objects. The North Star introduces a mechanism for global orientation. **P?** means the possibility to partition or not. Adapted from (PINI et al., 2013).

We believe that this environment abstracts a complete scenario of transportation. For instance, a shipment company where autonomous robots have to decide whether to carry a product from the sender to the receiver by themselves or if it is better to partition the task into regions. If environmental conditions change, maybe raining or snowing precipitations appear, robots should adjust their decisions and reach, instead of a cache, a warehouse. In this place, they could deposit the products in order that other members can pick them up and finish the transport. Thus, our decision-making model should help this company by allowing its robots to adapt according to environmental conditions.

About the decision-making in (PINI et al., 2011b) model

The authors proposed a decentralized decision-making, that is, each robot can adapt its decisions on whether to partition or not by learning from its experiences. In (PINI et al., 2013), robots might share their experiences incorporating social learning. Pini, in both works, challenged the decision-making models through environmental changes. Those challenges included a change in the delays for transferring objects in the caches. Results showed that their strategy brought a good performance despite changes. However, communication brought unclear effects: in some cases, it improved team performance, while in others, it reduced it. It depended on the luck of some of the robots to detect the environmental change to notify the team.

Robot decisions were done through comparisons of their estimated times to complete partitioning and non-partitioning activities. For that, the authors provided sigmoidal probability functions. (In Section 4.2.3, we explain how these functions behave to soften the changes in the probability.) However, despite task partitioning offers a lot of benefits, when robots are foraging with limited resources and space, conflicts and queues could emerge and decrease the overall group performance. To avoid such deterioration and stagnation, their robots needed the *give up* options. In (PINI et al., 2011b), these decisions depend on a static *give up* function, while in (PINI et al., 2013) on a fixed threshold. In both models, such kind of decision compares the current measured time struggling to complete an activity vs. the estimated time for the same activity. Moreover, in (PINI et al., 2011b; PINI et al., 2013), robots can only decide whether to give up or continue struggling in partitioning activities, i.e., those related to the area for transferences.

Furthermore, all robot decisions depended on learned estimations. Each robot could update its estimations once it ended an activity by employing the time it experienced to complete it. Those estimations helped the robot to know which option was better. In (PINI et al., 2013), communication increased the speed of learning due to shared information and reinforce what robots already know. In other words, each experience (lived or listened) helped robots to synchronize their information. However, robot-robot messages could also stick the team by inhibiting their diversity of knowledge and reducing the exploration of other alternatives in the environment. Consequently, if no robot detects the change, the whole team would continue foraging in the worst option because all robots learned the same estimations. Here is where we found an opportunity to test whether a particular communication structure could improve the team performance or if it would be necessary to include a mechanism for local adjustments of the communication structure. In particular, we tried to regulate the social learning to keep team diversity and exploration capacities.

Summarizing...

We worked in two problems whose strategies to solve the foraging task are different. The first problem has to be solved through a task allocation strategy, which we adapted from a previous work in (NOGALES, FINKE, 2013). The second one has to be solved by a task partitioning strategy, which we proposed based on (PINI et al., 2011b; PINI et al., 2013) models. For both problems, we had to provide an appropriated environment that demanded either a task allocation or a task partitioning solution.

Chapter 4

Proposals

Some difficulties emerged when we tried to replicate a navigation strategy from (PINI et al., 2011b) implemented in Argos (PINCIROLI et al., 2012). In Webots, robot vision considers object occlusion and shadows. We were forced to implement a strategy for navigation based on visual clues, which was published in (NOGALES, ESCARPINATI; OLIVEIRA, 2017) and briefly explained in Section 2.2.4. This navigation avoids global guidance, pheromone, or dead-reckoning strategies. It uses only local and visual environmental information. In particular, the implemented environments include TAMs, landmarks, and door switches to help robots in their navigation. Recall that TAMs are devices acting as sources and nests where robots may forage for and deposit virtual objects.

In the foraging workspace, the implemented environment for the task allocation strategy allowed us to propose: *i*) Two adaptations of the deterministic solution for multi-agent systems found in (NOGALES, FINKE, 2013), *ii*) An adaptation of the probabilistic decision-making found in (BONABEAU et al., 1997) such that robots could consider the three regions, and *iii*) An exploration of the effects yielded by different structures of communication on these decision-making strategies. Here, robots share messages with information about their estimations of the utility of the region in which they are foraging.

On the other hand, the implemented environment for the task partitioning strategy allowed us to propose: *i*) Two improved versions of the base models, one with static and other with online adaptation and *ii*) One mechanism that allows robots to adapt their communication structure according to the environmental conditions. With this strategy, robots share messages about the time they required to complete the transportation through both alternatives. Moreover, they can send messages to enable and disable their connections, which makes its communication structure an adaptive one.

4.1 Task allocation proposal

The solution for the task allocation problem was inspired in a previous work found in (NOGALES, FINKE, 2013) for multi-agent systems. According to Nogales and Finke

(2013), homogeneous agents should generate similar contributions to the utility. Thus, since e-pucks are similar and can attend only one task at a time, utility functions may be associated to the region where a subgroup of robots is foraging. A preliminary analysis of the region utility showed that, by adding more robots into a region, the utility increases but each time the increase gets lower until the number of robots surpasses a bound where the utility decreases. This phenomenon is known as the law of diminishing returns. For such scenarios, the benefit for a subgroup depends only on the number of robots that belongs to it (MARDEN, ARSLAN, SHAMMA, 2009). Unlike auction-based solutions, this law does not need robots to communicate their bids for an auction. Moreover, decision-making strategies based on this law allow individuals to show both divisional autonomy and distributed control (MARDEN, ARSLAN, SHAMMA, 2009).

In particular, we adapted the multi-agent solution to work with a multi-robot system. In that generic solution, agents should distribute themselves across a set of subtasks to achieve an optimal distribution. Agents base their decisions on the utility each subtask delivers. Here, each region of the environment represents a subtask where robots may forage for virtual objects. Moreover, each region has a utility that measures the number of objects that the subgroup of robots delivered within a period in that region. Note that region utility would depend directly on the rate of object production of that region and its value is affected by the number of agents working on it.

We employed TAMs to enable virtual objects handling, store the region utility, and share this information with robots that are working in their respective region. However, each robot can compute an estimation of the utility of its neighboring regions based on the messages it shares with other robots. In each decision, a robot compares the utilities and decides on which region to serve, while it seeks to optimize the group utility. Since the team has to maximize the group utility, the proposed decision-making strategies should stimulate robots to cooperate among subgroups based on the utility measurements.

In this section, we explain the implemented environment, the assumptions that enable our multi-robot system to work with the solution found in (NOGALES, FINKE, 2013). We also detail the proposed robot decision-making strategies, which vary in robot autonomy. Finally, we explain how social learning works through the message exchange between TAMs and robots, and robots with one another. In particular, since robots get information by entering into TAMs, they communicate over an intermittent network that includes robot-environment connections. The effects of robot-robot connections are explored through different network topologies that underlie their communication structure.

4.1.1 Simulated environment

In Webots, we needed to implement an environment consisting of several regions to work with decision-making strategies following a task allocation solution. Commonly, environments for foraging consists of at least two regions: one for deposit objects and other where robots explore (MATARIC, 1994). Notwithstanding, more regions are allowed: robots can divide the environment into regions (PINI et al., 2014; BUCHANAN, POMFRET, TIMMIS, 2016) or the researchers provide a predefined division (HOFF, WOOD, NAGPAL, 2013; PITONAKOVA, CROWDER; BULLOCK, 2016b). Here, we propose an environment with a predefined division, which consists of three regions that are connected through hallways with automatic doors.

Figure 11 shows the dimensions of the environment, its doors, landmarks, and their distribution. Note that every region has five TAMs working as nests, five working as sources. But, sources have a different rate for object production (as the number of objects dropped in a room of the warehouse). Therefore, robots have to distribute themselves according to these region rates. The main idea is that the sources of a region that have greater rates of production should attract more robots to forage in that region, while regions with a low rate of production should have fewer robots.

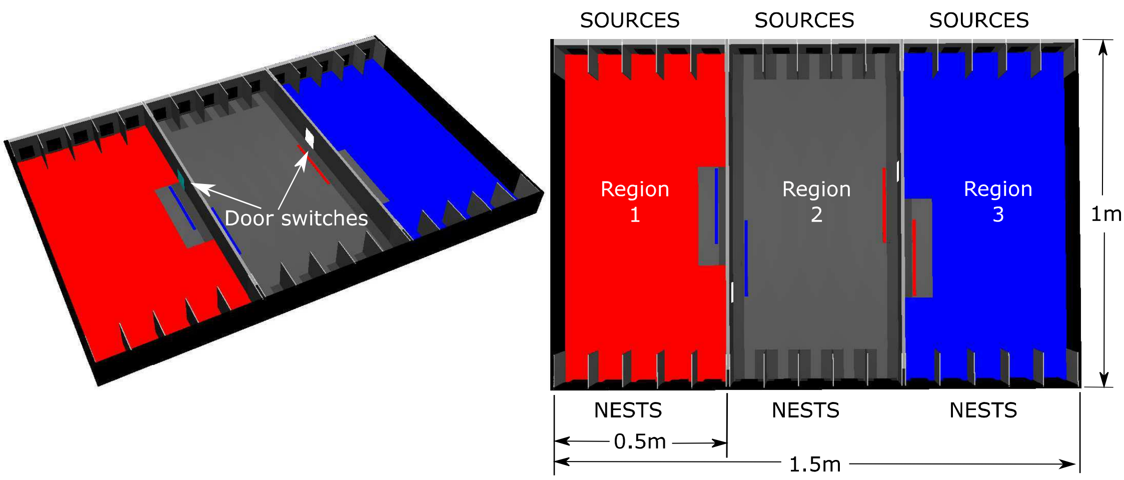


Figure 11 – Environment for the foraging task. The left image is a 3D view of the environment showing the door switches. Right image considers all dimensions.

The implemented TAMs are able to store and share information. As proposed in (BRUTSCHY et al., 2015), these smart devices include a Zigbee component that allows them to generate a network for information-sharing. Other environmental aids we included were ground colors to differentiate each region (PARKER, 1998) and fixed-lighted landmarks (KURAZUME, HIROSE, 2000) to indicate doors through which robots can reach a neighboring region (BOBADILLA et al., 2012). These aids are the cheapest solutions we found to distinguish and divide regions (Appendix C provides a broader review of environmental aids).

4.1.2 Proposed model

The strategies should try to solve the problem described in 3.1: allocate the robots among regions (nodes) such that they maximize the total utility (number of delivered objects). However, we need some assumptions to make it possible for robots to employ the multi-agents solution. First, the following three assumptions (common in economic theory (FÄRE, 1980)) are required for each utility function f_i.

A1 Each function f_i is continuously differentiable on $\mathcal{R}$.

A2 An increase in utility satisfies

$$\frac{f_i(r_i + u_i) - f_i(r_i)}{u_i} > \frac{f_i(r_i + w_i) - f_i(r_i)}{w_i} \tag{2}$$

where $r_i \in \mathcal{R}$, $w_i > u_i > 0$ represent a finite number of robots entering node i.

A3 An increase in the number of robots within a node increases the utility of that node, bounded by

$$0 < \frac{f_i(r_i + u_i) - f_i(r_i)}{u_i} < \infty \tag{3}$$

Assumption A2 represents the law of diminishing returns and implies that increasing the number of robots in a node will always yield decreasing average returns. Assumption A3 indicates that any additional robot should increase the utility moderately. Thus, according to Eq. (2) and (3), from Assumption A2 and A3, respectively, the partial derivative of f_i with respect to r_i, denoted by s_i, satisfies

$$-a \leq \frac{s_i(x_i) - s_i(y_i)}{x_i - y_i} \leq -b \tag{4}$$

for any $x_i, y_i \in \mathcal{R}$, $x_i \neq y_i$, and constants $0 < b \leq a$. It can be shown that if Assumptions A1-A3 are satisfied, the marginal utility functions $s_i(\cdot)$ are continuous on $\mathcal{R}$, strictly decreasing, and non-negative, while $f_i(\cdot)$ is strictly concave (see (NOGALES, FINKE, 2013) for details). Loosely speaking, the marginal utility functions indicate a lower average contribution per robot in the utility of a node after adding more robots. Section 5.1.1 will present the shape for the marginal utility functions of the nodes (or regions) of the proposed environment.

Until here, we have defined the assumptions about the utility functions, which allow robots to make decisions that benefit the group. Notwithstanding, we need more nomenclature and assumptions to design the distributed control rules (or decision-making strategies). Since we are using a topological map of the environment, a connection between two regions means that robots can move back and forth between them, and also get information from those regions. By moving across regions, robots may join or leave them at time indexes $t = 0, 1, 2, ...$ according to their decisions, which are asynchronous. Let $e_{u_i}^{i \to k}(t)$ denote the decision of a number u_i of robots to leave region $i \in N$ to join a neighboring region $k \in N_i$ at time t. Let $e_{u_i}^{i \to N_i}(t)$ denote the set of all possible simultaneous decisions from region i to its neighboring regions N_i. The set of events $\mathcal{E} = \mathcal{P}(\{e_{u_i}^{i \to N_i}(t)\}) - \{\emptyset\}$ represents all possible simultaneous decisions from all regions. Thus, a single event $e(t) \in \mathcal{E}$

is defined as a set where each element represents a decision of a number of robots to abandon a region.

If an event $e(t) \in \mathcal{E}$ occurs at time t, the update of the state of the system is given by $r(t+1) = g(r(t))$. For the robots belonging to region $i \in N$, $g(r(t))$ is defined as

$$r_i(t+1) = r_i(t) - \sum_{\{k: e_{u_i}^{i \to k}(t) \in e(t)\}} u_i(t) + \sum_{\{j: e_{u_j}^{j \to i}(t) \in e(t)\}} u_j(t) \tag{5}$$

In other words, after a movement of robots, the number of robots of a region would decrease by those leaving and increase by those arriving. Figure 12 shows an example of robot movements in the graph model of the environment between connected regions or nodes. We use both terms interchangeably.

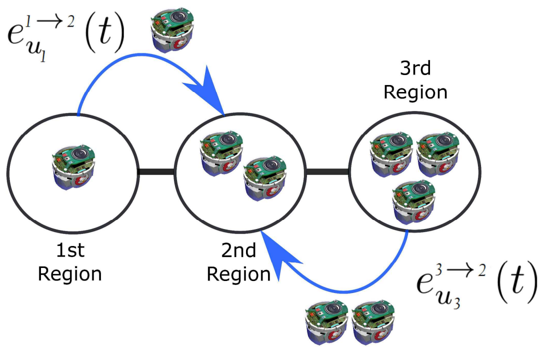


Figure 12 – Visualization of an event $e(t)$ when two simultaneous movements occur.

Note that three robots are joining the second region in this event $e(t)$. One from the first regions, whose mathematical expression is $e_{u_1}^{1\to 2}(t)$ with $u_1(t) = 1$, and two from the third region, whose mathematical expression is $e_{u_3}^{3\to 2}(t)$ with $u_3(t) = 2$.

Next, we got the mathematical notation. However, robots have to share information to reach the optimal distribution, which requires that the model to satisfy the following assumptions on the information sharing structure of regions and its robots.

- A4 The regions form a connected network $G_n = (n, A_n)$.
- A5 There is a large enough number of robots, q, such that there can be at least a robot within each region providing a positive utility when $r \in \Delta_q^\star$.

Assumption A4 implies that there is a path formed by consecutive regions that are connected by a door, i.e., a robot can reach any region after passing (probably several) doors to reach it. This places a small condition on the information-sharing and possible movements across regions. Assumption A5 demands a minimum number of robots, which in general depends on the nature of the utility functions. Thus, finally, under Assumptions A4 and A5, the optimal solution has the form

$$\Delta_q^\star = \{r \in \Delta_q | \ \forall i \in N, \ \forall k \in N_i, \ s_i(r_i) = s_k(r_k)\} \tag{6}$$

In other words, under all the above assumptions, for any finite number of robots, the optimal allocation $r \in \Delta_q^\star$ is unique (BERTSEKAS, 1999). In particular, Bertsekas (1999) found theoretically this unique and optimal point for non-linear systems and let open the challenge to propose strategies that can reach it. Here, robots need to reach the distribution $r \in \Delta_q^\star$ that captures the optimal allocation once all subgroups have the same marginal utility. Note that if all regions offer the same benefits, no robot has an incentive to abandon its region. If a robot does it, it would affect the performance of both regions, the one it arrives and also the one it leaves.

Here, we computed the optimal quantity of robots for different rates of production in each region (Op_i for region i) of the environment. But, the initial allocation is different from the optimal one. Robots have to travel between nodes seeking for one where they get the best performance. Their decision on whether to remain or leave a node depends on a learned estimation of the marginal utility. The next section explains this process.

4.1.3 TAMs and robots learning

Each robot updates its estimations from its own experiences, messages from their partners, or through TAMs. Since TAMs and robots are able to store information and share it among them, we need to explain the content of the messages and how they update their information. In a region, TAMs share their information with all robots foraging in that same region once they enter in them. On the other hand, robots can share information over three particular communication structures: fully connected, regular of degree 3, and ring topology. Thus, each communication structures provides each robot a set of links, which enable it to share information with its neighbors (i.e., those robots with which it is connected). Then, more links mean more cost in communication and processing of incoming information. In Section 2.2.1, Figure 3 provides a sample for each of these structures with 8 robots.

Robots and TAMs keep a record of estimation of the number of delivered objects. Robots keep their own performance (i.e., robot ℓ at node i keeps $s_i^\ell(t)$), while TAMs keep an estimation of the region (i.e., TAMs at region i keeps $s_i(r_i(t))$). Recall that if robots did not work in a region, information from other robots working in that region could spread up to them through their connections. Also, we allowed robots to share information they received from another region with their neighboring robots, that is, those robots with which they have a connection and share information with them. Robots and TAMs employ the following emitter-receiver messages:

- **TAM-Robot updating:** region ID and marginal utility
- **TAM-Robot order to leave:** region ID, receiver ID, and Bye message
- **Robot-Robot updating:** receiver ID, estimation, and region

- ❑ **Robot-Robot/Robot-TAM leaving a region:** emitter ID and Bye message
- ❑ **Robot-Robot/Robot-TAM arriving at a region:** emitter ID and Hi message
- ❑ **Robot-TAM refusing an order:** emitter ID and Negative message
- ❑ **Robot-TAM accepting an order:** emitter ID and Positive message

In our models, robots can receive messages from their own region and from another region. The arrival of such a message should update its estimations. If messages came from another region, they would update the estimated values a robot has of that region. The fulfillment of an activity updates the estimations, too. In particular, a robot ℓ uses the following exponential moving average function for learning with both its own measurements and incoming messages

$$s_i^\ell(t) = \alpha \cdot M_i^\ell + (1-\alpha) \cdot s_i^\ell(t-1) \tag{7}$$

where $\alpha \in (0, 1]$ is the rate of learning, M_i^ℓ is the score the robot measured (listened), and $s_i^\ell(t)$ was defined as the performance estimation that robot ℓ gets by working at node i. In robot-robot messages, robots share this estimation with all their neighboring robots. Thus, if robots have no path of connections with other regions, they would consider that their current region is the best, because they never got information about other regions. Thus, the communication structure affects their learning process, their decision-making, and their task allocation process. Moreover, we need to know how much information and how fast robots should share information, because both, message content and speed, also influence the learning process, and, as a consequence, the team performance.

Messages from TAMs are fundamental to this proposal. TAMs also indicate the end of a period for robots to update their local variables. This idea resembles the way enterprises pay their employees: they are asynchronous and payments are done at the end of a period of work (e.g., 15 days, monthly, etc.). Once TAMs send a *TAM-Robot updating* message, they indicate the end of a period to its robots. Then, they will consider how many objects transported -as a subgroup- and update their estimations of performance in that region.

Besides synchronizing robot performances, TAMs store more precise information of the subgroup of robots foraging in the region where they are allocated (i.e., the marginal utility $s_i(r_i(t))$). In particular, the set of TAMs working as nests in a node (or region) i update their marginal utility with a function of learning defined as follows:

$$s_i(r_i(t)) = \Lambda \cdot M_i + (1-\Lambda) \cdot s_i(r_i(t-1)) \tag{8}$$

where $\Lambda \in (0, 1]$ is the rate of learning of a region, M_i is the score measured during a period, and $s_i(r_i(t))$ was defined as the marginal utility of the region or node i. TAMs share this marginal utility every period in a TAM-Robot updating message, which helps robots to regulate their behavior. Thus, besides helping robots with their navigation at the environment, TAMs help robots in their information-sharing process.

4.1.4 Implemented decision-making strategies

Here, we detailed the strategies for decision-making. In particular, we implemented four decision-making strategies:

- Deterministic model (D), which we adapted from the solution for multi-agent systems that guarantees an optimal distribution; this model works as a reference.
- Two semi-stochastic proposals, which allow more autonomy for robots named (SS) without environmental aids while (SS-TAM) considers that these aids are enabled.
- A probabilistic strategy (P), which is an adaptation from (BONABEAU et al., 1997) with which robots could consider three regions at once. Robots used two thresholds for making these decisions.

These strategies of decision-making should help robots to regulate their transitions such that they could reach a (near-)optimal allocation. Movements may be stochastic, but any event $e_{u_i}^{i \to N_i}(t) \in e(t)$ must satisfy one of the following decision-making rules. In particular, how robots decide to distribute themselves into these nodes is the focus of the next sections.

4.1.4.1 Deterministic decision-making

In this strategy, according to the decision-making for multi-agent systems found in (NOGALES, FINKE, 2013), each region should decide if a robot should leave and forage in another. In other words, robots have no autonomy to decide which region to serve.

D-R1 If $s_i(r_i(t)) \geq s_j(r_j(t))$ for all $j \in N_i$, then $u_i(t) = 0$, i.e., robots remain in a node i where they get the best performance of the neighborhood.

D-R2 If there exists a node $k \in N_i$ such that $s_i(r_i(t)) < s_k(r_k(t))$, then some robots could decide to abandon i to serve in the neighboring node k. This node k is chosen such that $\forall j, k \in N_i \; s_{k\ell}(r_k(t)) \geq s_j(r_j(t))$, i.e., robots tend to move to a node that has the highest marginal utility among the neighboring nodes. In particular, the number of robots leaving node i is bounded by

$$0 < u_i(t) \leq \frac{1}{2}\phi\left[s_k(r_k(t)) - s_i(r_i(t))\right] \tag{9}$$

where $\phi \in (0, 1/a]$ represents the level of cooperation between regions, a is the fastest change in the marginal utility functions (see Eq. (4)).

Rules D-R1 and D-R2 restrain the allowable events. In particular, D-R2 regulates the tendency of robots to join a node that has a higher marginal utility value. Within a particular node, robots show divisional autonomy in the sense that they have no constraint in their decisions to serve that node. However, nodes have authority to bereave robots of node-to-node movements. In the warehouse scenario, it would be the same if robots can choose which objects they will move in a room, but not which room to serve. When robots forage with this decision-making, rooms would indicate them when to leave and towards where they must go. The messages from TAM to Robot (TAM-Robot) *order to leave*, Robot-TAM *refusing an order*, and Robot-TAM *accepting an order* work only on this deterministic model, where regions have authority upon robots to restrain their transitions. A robot answers with a *refuse message* if it is carrying an object. Otherwise, it would accept the command of the TAM and leave toward the suggested node.

4.1.4.2 Semi-stochastic decision-making

We proposed this strategy to provide robots with more autonomy, because it removes node authority upon robots, that is, robots can choose which room they want to serve. However, it introduces some randomness in their decisions. First, we need to add the following two assumptions on communication:

- A6 Each node offers information of its local utility to robots working within it.
- A7 There is a connection between any pair of nodes, that is, if $j \in N_i$, then there exists at least one link communicating a robot from node i with some robot in node j.

Since robots have to communicate and share information about neighboring regions to make a decision, Assumption A6 and A7 are guaranteeing information flow between TAMs in different regions. This is a critical requirement for the proposed robotic task allocation. Note that Assumption A7 is a local extension of Assumption A4. It allows robots to get information about performances and marginal utilities in neighboring nodes. Thus, robots can share information and compare the options to make a decision.

Note that robots following the semi-stochastic decision-making have autonomy to move across regions. No region is commanding robots to move toward another region. However, this means that when robots detect another node with a better utility, they could depart massively and that node would be empty. According to the optimal point $\Delta_q^\star$ defined in Eq. (6), if no robot is working in a node, they would have a lower team performance. The group has not reached the optimal distribution. To avoid such situations, we have to design a decision-making strategy that regulates such massive movements. In other words, we have to create distribute control rules such that some robots could remain in a node, even when there is another with a better utility.

First, robots would need to know or at least estimate how many of them are working in that node to be able to decide whether to stay or leave a node. Although such a kind of global knowledge is commonly unavailable in swarm robotics (ŞAHIN, 2005), they could estimate or guess how many are working on the same node. Recall that we are working with homogeneous robots, hence, it is expected that all robots serving a particular node would have a similar performance. Thus, by using the marginal utility of its region, $s_i(r_i(t))$, and its own, $s_i^\ell(t)$, robot ℓ might estimate how many of them are there. Recall that we let a robot ℓ keep track of its performance at node i in $s_i^\ell(t)$, which is an estimation of the number of objects it delivered in region i. Next, let $\hat{r}_i^\ell(t) = s_i(r_i(t))/s_i^\ell(t)$ represent the estimation of robot ℓ about the number of robots working in its node, which is not necessarily the real number.

Note that if robots are considering to leave node i, it is because there is a neighboring node $k \in N_i$ such that $s_k(r_k(t)) > s_i(r_i(t))$. Thus, let $p_{i\to k}^\ell(t)$ be the probability of robot ℓ departing from node i toward node k, which offers a better utility. When robots follow this strategy, any movement $e_{u_i}^{i\to N_i}(t) \in e(t)$ must also satisfy the following rules.

S-R1 If $s_i(r_i(t)) = s_i^\ell(t)$ (i.e., $\hat{r}_i^\ell(t) = 1$), then $p_{i\to k}^\ell(t) = 0$, i.e., that robot is the only one serving there and must remain in it even when there is a node with a better utility.

S-R2 If $s_i(r_i(t)) > s_i^\ell(t)$ (i.e., $\hat{r}_i^\ell(t) > 1$), then $p_{i\to k}^\ell(t) > 0$, i.e., that robot should compute its probability of departing toward node k that offers a better utility. The probability of robot ℓ departing is given by

$$p_{i\to k}^\ell(t) = \frac{1}{2}\phi_\ell \frac{[s_k(r_k(t)) - s_i(r_i(t))]}{\hat{r}_i^\ell(t)} \tag{10}$$

where ϕ_ℓ is level of cooperation of robot ℓ.

These rules stimulate robots to go after an optimal solution. Note that, unlike the deterministic solution, the semi-stochastic decision-making allows robots to use probability functions. Thus, by allowing robots to have autonomy to move between regions, we remove the warranty of achieving the optimal distribution $\Delta_c^\star$, i.e., robots could achieve sometimes the optimal solution, others, they would achieve near-optimal ones. The rule S-R2 with Eq. (10) should reduce the probability of having nodes without robots, but those situations can occur at any moment.

4.1.4.3 Probabilistic decision-making

Here, we adapted sigmoidal (s-shape) functions proposed by Bonabeau et al. (1997) for decision-making, which offer a soft change between two regions. We adapted those functions to allow robots to decide between three regions at once. In preliminary simulations, we also allowed robots to decide with only two options, that is, between its own region and the region with the maximum marginal utility. Those results showed that our adaptation delivered an improvement of at least 5% up to 18% over the decision-making comparing only two options. Moreover, we observed more frequently that robots left empty nodes when robots employed the two-options strategy. Therefore, we worked with the proposed three-option strategy. This strategy also includes Assumption A6 and A7 to guarantee communication between robots.

Next, let us assume that robot ℓ is foraging in node i. This robot has two thresholds and the best option would gain more range of probability as it becomes better. Otherwise, it would shrink its range. In particular, its probabilistic decision-making with three options employs the following thresholds

$$\theta_1(t) = \begin{cases} R_2\left[1+e^{R_1\left(1-s_i^\ell(t)/\hat{s}_k^\ell(t)\right)}\right]^{-1}, & \text{if } s_i^\ell(t) > \hat{s}_k^\ell(t) \\ R_2\left[1+e^{R_1\left(s_i^\ell(t)/\hat{s}_k^\ell(t)-1\right)}\right]^{-1}, & \text{otherwise} \end{cases} \tag{11}$$

$$\theta_2(t) = \begin{cases} R_1\left[1+e^{R_2\left(s_i^\ell(t)/\hat{s}_j^\ell(t)-1\right)}\right]^{-1}, & \text{if } s_i^\ell(t) > \hat{s}_j^\ell(t) \\ R_1\left[1+e^{R_2\left(1-s_i^\ell(t)/\hat{s}_j^\ell(t)\right)}\right]^{-1}, & \text{otherwise} \end{cases} \tag{12}$$

where $R_1 = 2/3$, $R_2 = 4/3$, and $s_i^\ell(t)$ is the performance of robot ℓ at node i, while $s_k^\ell(t)$ and $s_j^\ell(t)$ are the (probably listened) performances at nodes j and k, which are its best two neighboring nodes. Recall that Assumption A6 allows robots to share information about their performance to another robots serving neighboring nodes. Thus, if robot ℓ never worked in these nodes, it could use the listened performance of a partner working there. Figure 13 provides the shapes of both thresholds as the ratio increases.

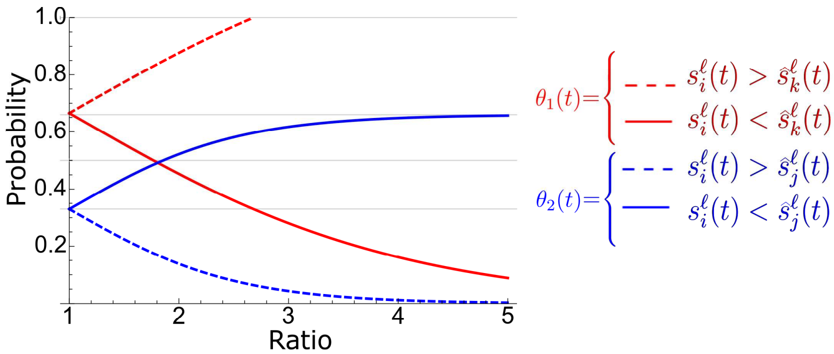


Figure 13 – Shapes of the probability thresholds to leave a node. The values of $\hat{s}_j^\ell(t)$ and $\hat{s}_k^\ell(t)$ remained constant in this plot. The ratio means the relation between $s_i^\ell(t)$ over any of the alternatives.

Note that if marginal utilities are similar in the three regions, a robot has equal probabilities to decide for any of the regions. Otherwise, that robot should adjust its thresholds to make its decisions. Dashed lines in Figure 13 represent the cases where the performance of robot ℓ in the current region i seems to be better than those listened about or previously experienced in its neighboring regions. If the marginal utility of node i increases, more chances this robot would have to remain in that node. The solid lines indicate when robot ℓ at region i has a lower performance than its neighboring regions. Note that if its marginal utility gets lower, the other options increase their probabilities until it reaches a negative value, which means robot ℓ must leave region i.

Following this decision-making strategy, any movement $e_{u_i}^{i\to N_i}(t) \in e(t)$ must also satisfy the following rules.

P-R1 If $\theta_1 < p^{\ell}_{i\to\{j,k\}}(t) < \theta_2$, then robot ℓ should not leave its current node i.
P-R2 If $\theta_1 > p^{\ell}_{i\to j}(t)$, then robot ℓ should depart toward node j.
P-R3 If $\theta_2 < p^{\ell}_{i\to k}(t)$, then robot ℓ should depart toward node k.

In other words, if robot probability falls between the thresholds, it would opt to remain. Otherwise, it should depart toward a neighboring node. In this strategy, nodes have no authority upon robots. They can travel between nodes according to the indication of their estimations. Note that this option is completely probabilistic and different from the deterministic rules D-R1 and D-R2 and the semi-stochastic rules S-R1 and S-R2. However, Assumptions A1-A7 remain. Consequently, robots are measuring utilities and have a greater dependence on information coming from robots at different regions. Unlike the semi-stochastic decision-making, robots following this probabilistic decision-making do not have one rule to regulate how many of them can leave a node with low utility. Therefore, it is possible to have massive departures of robots between nodes, which could leave some nodes without foragers.

Summarizing, all the implemented strategies of decision-making depend on the information-sharing structure. Robots following the deterministic decision-making need TAMs to command them where to go. Whereas the semi-stochastic and probabilistic strategies, TAMs help robots in the information sharing and in the synchronization of their decisions within each region. Note that the more autonomy for robots, the more they depend on information from neighbors and regions. The next section explains how robot behaviors are controlled such that they could forage for objects.

4.1.5 Controller

Robot behaviors are controlled by a hierarchical state machine, i.e., a state machine where each state may contain a sequence of sub-states as shown in Figure 14. We chose this technique to avoid an explosion of states due to some actions (as searching) and transitions that repeat in different activities (states). When robots begin to forage, they will begin with *Pick an object*, but they first identify the ground color to know which region they belong to. Then, they send a message *Robot-TAM arriving at a region* such that the TAMs in that region may know who are their foragers.

In Figure 14, we expanded the states to show details of their inner sub-states. Note that robots have two kinds of decisions, **T?** and **G?** The first represents the transitioning decision, which is achieved by the previous decision-making strategies. In other words, **T?** means that a robot is considering to leave a region to move to another where it may get a better utility. The latter decisions are only available in the *Go to a better region* state because robots can fail the searching of the door switch. After a robot ends this state, it can be in either the desired region or the same (if its searching process failed).

In that latter case, the robot could decide to stay there because it is taking too long to find the door.

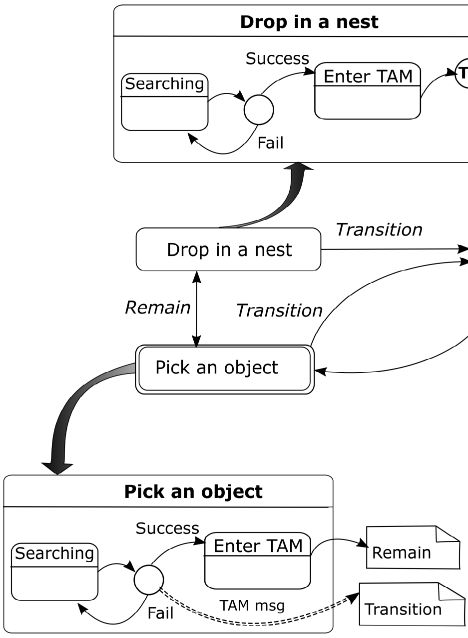


Figure 14 – Hierarchical state machine describing robot behaviors. The dashed lines represent the transitions where robots are able to abandon a previous decision. The circles in the expanded states contain the conditions **G?** (for *give up* decisions) and **T?** (for decisions on transitioning toward another region).

In particular, robots use the following *give up* probability function for that kind of decisions:

$$P_g(\hat{t}, t_i) = \left(1 + e^{K\left[M\frac{t_i - \hat{t}}{\hat{t}} + O\right]}\right)^{-1} \tag{13}$$

where $K = -0.6$ and $M = 5$ are steepness factors, $O = -5$ is the offset of the function, t_i is the time already invested in finding the door, and $\hat{t}$ is the estimated time picking and dropping an object in this region. These values were the ones that worked better in the preliminary simulations. We adapted Eq. 13, which was published in (NOGALES, ESCARPINATI; OLIVEIRA, 2017). Note that robots can only abandon *Go to a better region* state to return to pick objects instead of wasting their time looking for the door. Robots following the deterministic model should send a *Robot-TAM refusing an order* message when they decide to abandon the transition. Thus, TAMs could send another robot to depart from that region. The following section describes the preliminary simulations to configure the environment parameters and the performance of the decision-making strategies in our foraging task.

4.2 Task partitioning proposal

First, since this solution was extended from the base models, we need to summarize the differences between our model with the base models found in (PINI et al., 2011b; PINI et al., 2013) in Table 2.

Proposal	**(PINI et al., 2011b)**	**(PINI et al., 2013)**
Robot decision-making		
• Adaptive *give up* function	• Static *give up* function	• Threshold based *give up* option
• *Give up* decisions compare times on both options	• *Give up* decisions compare the waiting time vs. estimated time on the same activity	• *Give up* decisions compare the waiting time vs. estimated time on the same activity
• *Abandon* decisions in any activity	• *Abandon* decisions in activities related to the area for transfers	• *Abandon* decisions in activities related to the area for transfers
• *Abandon* decisions may occur after failing a search	• *Abandon* decisions may occur after finish an activity	• *Abandon* decisions may occur after finish an activity
Navigation		
• Based on shark-inspired strategy (only visual)	• Based on odometry information	• Based on odometry information
• Without memory or previous knowledge	• Not mentioned	• A priori knowledge of sources and nests sites
• Traveling through the alternate path employs a line-following algorithm	• Traveling through the alternate path mixes a North Star guidance with ground color information	• Traveling through the alternate path mixes a North Star guidance with ground color information
Controller		
• Hierarchical state machine	• Finite state machine	• Finite state machine
Communication		
• Adaptive	• OFF	• OFF/ON

Table 2 – Main differences between our proposal and the reference models.

Here, we also employed TAMs to enable virtual objects handling but, in this environment, they did not share information with robots. Robots should compare their experiences while transporting the objects from sources to nests through probability functions. In particular, the time they required to complete the transportation.

Moreover, in this section, we will explain: *i*) the simulated environment, which follows the schematic distribution shown in Fig. 10. *ii*) the proposed decision-making model with an adaptive function that helps robots to make faster and smarter decisions for abandon or continue struggling in any activity, *iii*) how the controller deals with these functions to regulate robots behaviors, and *iv*) our proposal for adapting the communication structure, which was embedded in the best three implemented models for decision-making to regulate robot learning speed.

4.2.1 Simulated environments

Here, we describe the parts of the environment and their distribution in the environment. Since standard e-puck robots lack handling capacities, we seek possible solutions to validate our proposal. We found several ways to help simple and limited robots: some require hardware extensions, while others are through environmental aids. Generally, hardware extensions are more expensive because they must be acquired for each member of the team. Thus, as the team grows, the cost of the experiments increases too. This does not happen with environmental aids because all robots may use one of them. Among the environmental aids, we found interesting solutions, each with a different level of complexity and cost of implementation (see Appendix C for details). We opted for TAMs, ground colors, and landmarks.

The simulated environments have a blue region, which has some TAMs working as sources, while the red one holds TAMs working as nests. Recall that TAMs indicate the available activities through a color-shape code. The area for transferences includes some caches, which join both regions. The paths have lines of specific colors to guide robots toward the other region. Finally, the scalability of the models was tested by providing two sizes for the environment: small and large. Figure 15 illustrates the short and large environments with their regions.

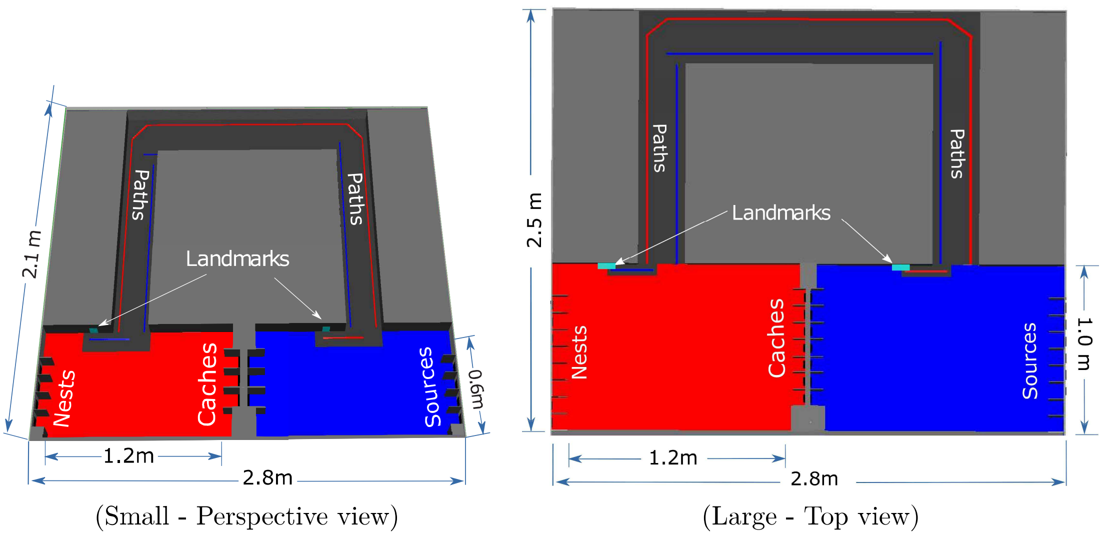


Figure 15 – Dimensions and regions of the environments for the foraging task. Both environments include two cyan landmarks indicating the start of the paths to switch across regions.

Note that the small environment includes 4 sources, 3 caches joining both regions, and 4 nests. While the large one has 8 sources, 7 caches, and 8 nests. The alternate path has the same size in both environments, likewise the position of the two landmarks. Notwithstanding, the regions in the large environment are broader to hold more TAMs. Their number is proportional to the number of robots that the environment can support. However, there are fewer caches than robots to increase the complexity of the environment.

In the base models, there was a North Star to guide robots while traveling through the alternate path (see Fig. 10). Here, it was replaced by two local landmarks. These landmarks are two fixed squares always lit with a cyan color indicating the start of the path. Once a robot reaches a landmark, it enables a line-following process. The robot begins to follow a line with the same color of the region of destination. When the robot finishes that line, it follows another line with another color lest it wanders when it arrives at its destination. Since there is only one landmark per region, a robot may occlude the landmark for a while. Other robots looking for it will wander around until it is visible again or until they decide to abandon this option. This would create queues that the implemented decision-making models should overcome.

4.2.2 Robot learning mechanism

After a robot finishes an activity whether picking or dropping an object, they update their estimated time to complete such activity. Such estimations give robots an indication about the current environmental conditions. Besides, when robots share their experiences, that is, with a social learning as in the base model (PINI et al., 2013), they would consider the experiences of others robots as if they would have finished that activity. Only neighbors can share information between them, that is, those robots having robot-robot connections between them. Those connections create a communication structure.

Since robots employ a communication structure to share information, we explore the following static communication structures: ring topology (R), small-world (SW), and fully connected (F). When they do not share information, we considered it as a no connection structure (N). The reader may find a graphic representation of these networks in Section 2.2.1 for eight e-pucks. In particular, robots can share the following messages:

- **Robot-Robot updating:** emitter ID, estimation, task code
- **Robot-Robot hi:** emitter ID, receiver ID, 1
- **Robot-Robot bye:** emitter ID, receiver ID, 0

The task code encrypts if the robot was partitioning or not and if it was picking or dropping an object. For instance, “PC” means *Picking from a cache.* We understand from this code that the robot was partitioning because it was using caches and, evidently, it was picking an object. Likewise, for the non-partitioning option, we used “PS” for source and “DN” for nest activities. These codes were researcher-friendly to make easier the processing of robot reports.

Recall that we search for a communication structure that helps robots to forage while dealing with changes in the environmental conditions, without knowing the exact conditions. In this proposal, environmental changes are modifications in the delays of the transference area. However, we confirmed what previous works had indicated: having a dynamic environment could demand a more complex and adaptive structure. This held in our changeable environment. Then, we had to provide robots with a mechanism for adjusting their connections, which is explained in further sections. Note that robots have two more messages, which they can use to change their communication structure in this way: *Robot-Robot hi* messages connects emitter and receiver robots while *Robot-Robot bye* disables their connection. In other words, these messages enable robots to create an adaptive structure that helps them to deal with a changeable environment.

Learning process

Note that environmental congestions and the delays in caches affect robot estimations. Even if cache delays remain fixed, environmental congestions yield different measures of time after each experience. We employed this delays to introduce significant changes in robot experiences. Robots should keep an estimation of the time they required to fulfill their activities. Thus, after finishing an activity, the time the robot measured updates the estimated time for that activity as follows:

$$\hat{t} \leftarrow (1-\alpha)\hat{t} + \alpha t_M \tag{14}$$

where t_M represents the measured time since the robot began to do that activity and $\alpha \in (0, 1]$ is the rate of learning. The larger the alpha, the more important the last experience is. Whereas $\hat{t}$ represents the estimated time of that last activity. For instance, in the region for sources, a robot that picked an object and opted for the non-partitioning option would update $\hat{t}_S$ (estimated time to store in a nest). That robot decided to travel through the alternative path to store its object in a nest. The time steps it measures until it stores the object in a nest (t_M) would update its estimation for that activity ($\hat{t}_S$). If the robot had opted for the partitioning option, it would have to update $\hat{t}_{Dc}$ (estimated time to drop in a cache). Likewise, for a robot in the region of nests. If the robot chooses the non-partitioning option, it would have to travel and harvest an object from a source. In this case, that robot would update $\hat{t}_H$ (estimated time to harvest from sources). Otherwise, it would pick up an object from caches and update $\hat{t}_{Pc}$ (estimated time to pick from a cache).

Furthermore, when a robot completes an activity, it can broadcast to all robots a *Robot-Robot updating* message with its ID, the measured time, and the code of the activity it was performing. Other robots will listen to this message but only its neighbors will consider this information as if it were its own experience. In other words, a neighbor would consider this information as if it itself had finished the reported activity. Consequently, that robot will update its estimations by using Eq. (14) employing the time its neighbor experienced as t_M in the activity it listened about as $\hat{t}$. Since events are related to the learning process, every time a robot uses Eq. 14, it triggers an event. In other words, an event is when a robot updates its estimations either because it fulfilled an activity (experience) or by the arrival of a message (social learning).

In swarm robotics, it is better to avoid global communication. In case of a real implementation, TAM modules, as described in (BRUTSCHY et al., 2015), could store this information, share it across the network of TAMs, and replicate it to all robots that will eventually enter them. Thus, in a real implementation, TAMs could enable communication between all robots and activate the social learning process. Robots learn from *Robot-Robot updating* messages, where they share their updated estimations. Since the *Robot-Robot updating* messages include the robot ID, other robots could decide whom they listen to, i.e., they could choose who are their neighbors. Thus, robots can vary from excessive to scarce information by increasing or decreasing their connections. Unlike (PINI et al., 2013), our robots do not share their experiences continuously; they share them only after finishing an activity. More details about how robots regulate their information-sharing resources are in the following section.

Adaptive communication structure

Since our proposal depends on communication skills to improve team performance, we observed how social learning affects team performance. Some researchers of sensor networks improved their strategies by introducing structural parameters in their strategies (ALBERT, JEONG, BARABASI, 1999; WU, TSE, LAU, 2014; RAMOS et al., 2014; PERILLO, HEINZELMAN, 2005; YIN et al., 2013). The most common one was node degree, which measures the number of connections a node has in the network of communication. This parameter has helped some of these authors to reach improvements as extending the battery life of the sensors, increasing the speed for information-sharing, decreasing the number of messages to spread information.

In particular, the proposed mechanism allows each robot to adjust its degree, i.e., the number of connections it has with other robots according to the environmental conditions. Recall that a connection between two robots means that they can share information between them and, as a consequence, they can learn twice as fast as only one could. For instance, let robots A and B be neighbors. Robot A delivers an object in the nest and shares this information with its neighbor who is also coming to deliver another object. When robot B ends the delivery, it would share a message with its experience. Then, each would have updated its estimation as if it had already finished two deliveries (one from its neighbor and one from its own experience). Following this analysis, the more connections or degree a robot has, the faster it will learn. Thus, robots would have to balance between little and excessive information, i.e., between exploring or exploiting their neighbors with which they share information to deal with environmental changes. Figure 16 shows where this mechanism should work to improve group adaptability.

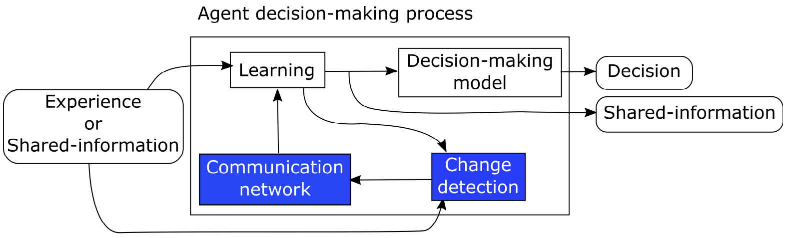


Figure 16 – Decision-making process schema for each robot. The blue squares are the core of our proposal to improve adaptability through the communication structure.

This kind of adjustment in robot degree has a similar effect to the technique of window size adaptation for online learning in data-streaming (found in (GAMA et al., 2014)). In that technique, if the controller detects a change in the data stream, it decreases the window size to speed up the learning process. Otherwise, it enlarges its learning window to the normal size. In our case, when a robot increases its degree, it would have the same effect as reducing the window size because the robot could get more experiences in a short time instead of waiting to live them. For instance, if a robot detects a change in cache delays, it could increase its connections. By increasing its degree, that robot would get more incoming messages. Then, it could update its estimations faster. However, in normal conditions, the robot just need to keep a few connections and learn at a slow rate. Fig. 17 shows a graphical comparison of the learning adjustments in both ideas.

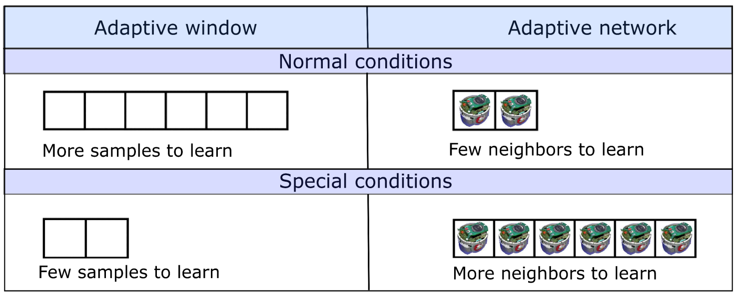


Figure 17 – Comparison of our proposal with the adaptive window size technique from data streaming.

Note that this proposal requires that each robot supervise environmental conditions to detect changes. We let robots to check the ratio between the costs of both options as follows

$$PnP(t) = \frac{t_{\phi 1}(t) + t_{\phi 2}(t)}{t_{\Phi}(t)} \tag{15}$$

where $t_{\phi 1}(t)$ and $t_{\phi 2}(t)$ are the estimated times to transport an object in each of the two regions, i.e., to get a successful transport through the area of transference; $t_{\Phi}(t)$ is the estimated time to finish the transport without partitioning, i.e., to transport an object from a source to a nest using the path. This equation evolve by the occurrence of learning events.

Recall that an event refers to a moment when a robot gets new information about the activity estimations: by its own experience or by social learning when a neighbor shares its experience. After each event, robots check for changes in the ratio $PnP(t)$. Here, they employ the following high-pass filter to detect changes

$$\gamma(t) = \beta * (\gamma(t-1) + PnP(t) - PnP(t-1)) \tag{16}$$

where the parameter β is a threshold that helps to filter noise from real changes (GAMA et al., 2014). Note that a small β means that the curve will decay quickly. Thus, the robot needs large changes between $PnP(t)$ and $PnP(t-1)$ to adjust its connections. If there is no change, i.e., $PnP(t) - PnP(t-1) = 0$, then the function $\gamma(t)$ will decay to zero. Next, working with a large β means that small changes are noticeable. As a consequence, we let a fixed small value of $\beta = 15\%$, which means that a change greater than 15% is enough for a robot to adjust its connections. We defined this value based on the standard deviation of preliminary simulations.

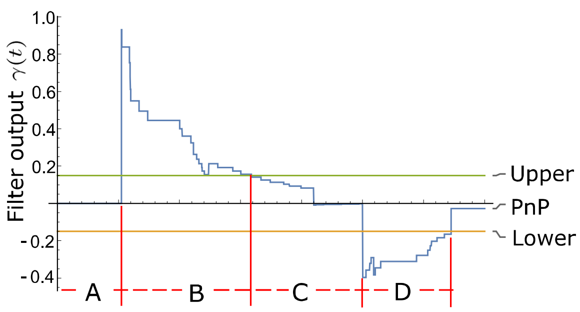


Figure 18 – Filter behavior against changes in the ratio $PnP(t)$. Red circles mark the events where the degree of a robot should change.

Figure 18 shows how the filter works. First, within the interval A, note that there is no change because no robot ended any task to update their estimations. Within the interval B, the filter output surpasses the upper bound, then, the robot will adjust its connections ($PnP(t)$ increased enough). Within the interval C, the output remains between the upper and lower bounds, then, the robot will not change its connections. Once the filter output trespasses the lower bound, the robot will adjust its connections again ($PnP(t)$ decreased enough). The change in the degree of a robot depends on how this parameter behaves in constant scenarios. Details about how robots should adjust their connections in the proposed environment are in Section 5.3.3.

To summarize, we provide robots with a simple learning mechanism. Robot-robot connections help robots to increase their rate of learning. However, either excessive or scarce incoming messages can deteriorate the team performance. To avoid this, we also propose a mechanism that enables a way for each robot to regulate between exploiting and exploring shared information. In particular, once a robot modifies the number of neighbors, it changes the rate of learning, going from fast to slow learning rates. Then, those changes in robot degree should help the team to improve their performance. Recall that we focus on improving the adaptability through the communication structure. Therefore, since robots are foraging in a dynamic environment, they require more complex communication structures. Our robots could adapt their structure according to each environmental condition, which was possible due to the change detection mechanism. It helped robots to deal better with environmental changes.

4.2.3 Implemented decision-making strategies

Here, robots have two kinds of decisions: partitioning? and giving up? Both decisions depend on the estimated times each robot learned while fulfilling these activities, whether through the caches or the alternative path. These decision-making models are local and employ sigmoidal probability functions to compare both options. The probability functions help robots to favor either the partitioning or non-partitioning activities.

We also introduced the possibility for robots to abandon any activity, not only those related to non-partitioning. They can abandon any activity not only those related to the area for transference. As previously cited studies showed, small changes at the individual level can generate improvements at the group level. Some of the benefits of having the *abandon* options and a static version of the *give up* function were submitted in (NOGALES, OLIVEIRA, 2018a). We explain in the following paragraphs both kinds of decisions.

Decision-making 1 - Partitioning?

Robot decisions about partitioning consider the ratio between partitioning and non-partitioning estimations on the following sigmoidal probability function

$$P_p = \begin{cases} 1/\left[1 + e^{R\left(\hat{t}_\Phi/(\hat{t}_{\phi_1}+\hat{t}_{\phi_2})-1\right)}\right], & \text{if } \hat{t}_\Phi > (\hat{t}_{\phi_1} + \hat{t}_{\phi_2}) \\ 1/\left[1 + e^{R\left(1-(\hat{t}_{\phi_1}+\hat{t}_{\phi_2})/\hat{t}_\Phi\right)}\right], & \text{otherwise} \end{cases} \tag{17}$$

where P_p is the probability of partitioning, R is a steepness factor, $\hat{t}_\Phi = \hat{t}_H + \hat{t}_S$ is the estimated time to transport an object by the non-partitioning option: harvesting from sources and storing it in a nest through the alternate path. While $\hat{t}_{\phi_1}$ and $\hat{t}_{\phi_2}$ represent the estimated time to transport an object by partitioning, that is, by using the caches for indirect transferences between both regions. In the region of sources, the estimated time to transport from a source to the receiver side of a cache is $\hat{t}_{\phi_1}$. While in the region of nests, $\hat{t}_{\phi_2}$ represents the time to transport an object from the deliverer side of a cache to a nest. Fig. 19 shows the behavior of these functions.

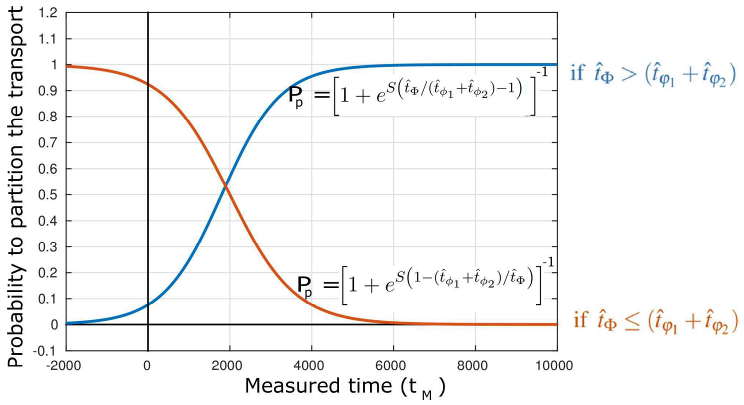


Figure 19 – Behavior of the probability to decide between partitioning and non-partitioning, i.e., between using the area for transference or the alternative path.

Since these functions follow sigmoid curves, as the estimation for an option goes lower (higher), the probability of choosing that option increases (decreases). Eq. (17) does not change from the one in (PINI et al., 2011b).

Decision-making 2 - Giving up?

Although we worked with the same robotic platform of (PINI et al., 2011b; PINI et al., 2013), we observed that the *give up* functions of the base models delivered a slow speed of adaptation. Here, we proposed a new *give up* function, which robots could adjust as they forage (online). Our robots decide to give up without using centralized task-related information and independent of neighbors success. If a robot does not find a cache available, it computes its probability of giving up through the following sigmoidal function:

$$P_g(\hat{t}, t_M) = \left(1 + e^{\Theta(\hat{t}, t_M)}\right)^{-1} \tag{18}$$

where $\hat{t}$ can be the estimated time $\hat{t}_{D_c}$ (for the dropping of an object in a cache) or $\hat{t}_{Pc}$ (for the picking up of an object in a cache), and $\Theta(\hat{t}, t_M)$ is a function that computes the effect of waiting for a cache,

$$\Theta(\hat{t}, t_M) = K\left[M\frac{t_M - \hat{t}}{\hat{t}_{\phi_1} + \hat{t}_{\phi_2}} + \mathcal{O}\right] \tag{19}$$

where K and $\mathcal{O}$ are the steepness and the offset of the function, respectively.

It is important to mention that the parameter M in Eq. (19) was introduced in the current work. That is, M is absent in the *give up* function proposed in (PINI et al., 2011b). Therefore, to reproduce that *give up* function, we let $M = 1$ to cancel its effects. Next, in the *give up* decisions based on a threshold from (PINI et al., 2013), we let robots struggle for an available cache until $\tau = 3\hat{t}_{\phi_i}$, where $i \in \{1, 2\}$. Afterward, they change to the equivalent non-partitioning activity, for instance, *drop in cache* changes to *store in a nest* and *pick from a cache* changes to *pick from a source*.

The *give up* function was fundamental to increase the speed of adaptation of the base models. Therefore, we implemented two models: M-SGU for offline adjustment of M in Eq. (19) and M-AGU for the proposed online adaptation through Eq. (20). Note that this parameter increases the effects of the current measured time (t_M) to achieve faster adaptations in robot decisions. Figure 20 shows both sigmoidal curves.

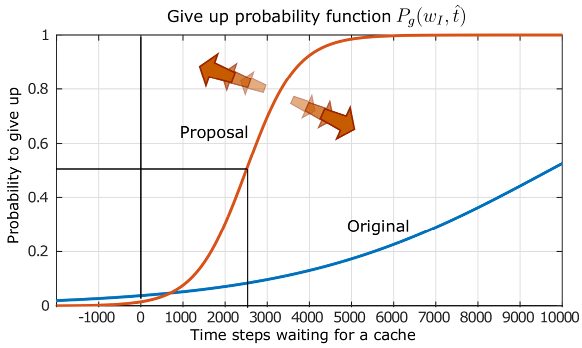


Figure 20 – Shapes of the probability to *give up* according to the time waiting to pick up or drop an object in a cache. The blue line is the function of the model described in (PINI et al., 2011b) (M-2011), while the red one is the offline adaptation proposed in (NOGALES, ESCARPINATI; OLIVEIRA, 2017) (M-SGU). Arrows show the directions of the possible adjustments for the online adaptive *give up* function (M-AGU).

As time passes, robots increase their probability to *give up* the partitioning. Note that the original curve of the base model is slower. There was no need to abandon any activity in the base models because, in Argos, robots did not fail in their search processes; they could see through other robots. Webots does not allow this to happen, robot camera captures environmental information including other robots. This yielded a need to adjust the curve for *give up* and further simulations showed that it was necessary online adjustments.

Here, we explain how to do an offline tuning of M. First, we took the average of all estimated times from preliminary simulations with a mandatory usage of the caches (always partitioning) and with a mandatory usage of the path (never partitioning). The parameter M has a value where both options are equally likely. Thus, robots have a 50% probability of giving up the partitioning activity (related to the cache usage) when they waited the average time required to travel through the alternative path (almost 2500 steps as shown in Figure 20). The general idea of the M-SGU model, with the offline tunning of M, was firstly published in (NOGALES, ESCARPINATI; OLIVEIRA, 2017). In that paper, we let $M = 5$ and compared it with the base model found in (PINI et al., 2011b).

Next, for M-AGU, we allowed robots to compute the M parameter of Eq. (19) during each simulation by using their own experiences as follows:

$$M = -2\mathcal{O}\frac{\hat{t}_{\phi_1} + \hat{t}_{\phi_2}}{\hat{t}_{\Phi} - 2\hat{t} + 1} \tag{20}$$

where $\hat{t}$ can be either $\hat{t}_{D_c}$ (for dropping an object in a cache) or $\hat{t}_{Pc}$ (for picking up an object in a cache). Note that when the estimated time for a partitioning activity surpasses the estimation for the non-partitioning option ($\hat{t}_{\Phi}$), the sign of M changes along with the slope of Eq. (18). Consequently, the robot will favor the non-partitioning option and abandon the partitioning activity. Eq. (20) allows robots to compare the estimation of both options: the estimated time to do a subtask under partitioning against the estimated time for the alternative activity in the non-partitioning option.

Moreover, recall that resources are scarce and robots have to deal with emerging conflicts and queues. Thus, after a while trying to find an available TAM or landmark, they could fail to find it. The probability functions in Eq. (17) and (18) also help robots to consider whether to abandon or continue struggling to complete any activity. This was one of the improvements we made to the base models. It includes the possibility for robots to reconsider their previous decisions through these functions in any activity and if the robot fails in its searching, it can employ the *abandon* transitions we added in all activities. The following section introduces the *abandon* transitions through which robots can adapt faster to current environmental conditions.

4.2.4 Controller

A hierarchical state machine, i.e., a state machine where each state may contain a sequence of sub-states (as shown in Figure 21). controls robot behaviors. We chose this technique to avoid an explosion of states due to some actions and transitions that repeat in different activities (states). When robots begin to forage, they first identify the ground color to know in which state to initialize. For instance, if a robot starts in the blue region, it will begin with *Pick from a source*. In the red region, robots begin with *Pick from a cache*, because they begin favoring the partitioning option.

In Figure 21, we expanded two states to show details of their inner sub-states. In particular, the *Pick from a cache* state that considers a partitioning activity and *Go harvest a source* that considers a non-partitioning one. Note that both activities have three possible outcomes: Non-partitioning, Partitioning, and Abandon. The orange circles inside these zoomed activities represent the options where a robot may consider to abandon that activity. Note that they are different from the *give up* function. If robots decide to abandon an activity, they will try its equivalent version in the other option. For instance, if a robot abandons *Pick from a cache*, it will change to *Pick from a source*. The dashed lines illustrate abandon transitions, which are coming from every state, not only states involving the cache usage as in the base models.

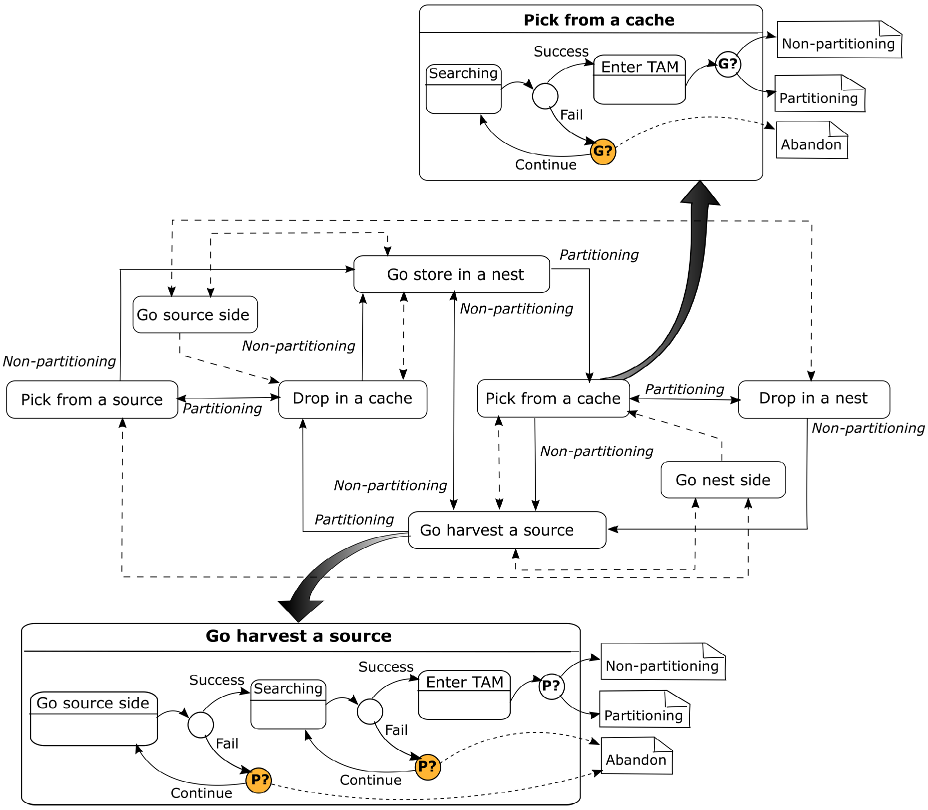


Figure 21 – Hierarchical state machine describing robot behaviors. The dashed lines represent the transitions where robots are able to abandon a previous decision. The circles in the expanded states contain the conditions **G?** for *give up* decisions and **P?** for decisions on partitioning

These *abandon* transitions allow reconsidering previous decisions more frequently. With these transitions, a robot evaluates the same condition of the host activity after it fails in the search process. Such evaluation is done as if the robot had already finished the activity, but instead of using the estimated time, it uses the measured time. Thus, they compare new information with old estimations. For instance, if a robot performs *Go harvest a source* and the landmark is occluded for a long time, it can consider abandoning this activity. This time, it would evaluate the function for partitioning decisions in Eq. (17) with the measured time (t_M) instead of the estimation (for this case $\hat{t}_H$). Likewise, if a robot performs *Drop in a cache* or *Pick from a cache* and no cache is available, it can use the *abandon* transition by evaluating the *give up* function in Eq. (18). But, it uses the measured time instead of the estimation it has for it.

Summarizing...

We have provided two environments where the task allocation and task partitioning strategies can be exploited. The mathematical background for both solutions is deeply explained and how the robots and TAMs update their knowledge of the environment (individually and through messages). Moreover, since both environments required different behaviors, we proposed two controllers, one for each environment. In the task allocation problem, robots are helped by TAMs to share information. But, in the task partitioning problem, we had to introduce a mechanism for robots be able to adapt their communication structure. The following chapter shows how these proposals dealt with the challenges in their respective environments.

Chapter 5

Simulations and results

We explored different communication structures into different models of decision-making, which vary in how information is handled. Here, simulations ran in two simulators Webots (MICHEL, 1998) and Netlogo (WILENSKY, 1999). In Webots, simulations run at a slow pace because the academic license does not allow to increase the speed and exploit the hardware resources entirely (in some occasions, it was lower than 0.4X). In other words, in Webots, the exploratory experiments would take too long. Such exploratory experiments should allow us to identify the pattern between a structural parameter and the environmental conditions. As a consequence, we had to implement some of these models in a faster simulator. We opted for Netlogo, which has fewer details and features than Webots but it takes only a small fraction of time to complete the experiments. We adjusted robots movements in Netlogo such that the models delivered similar performances to those found in Webots. Netlogo allowed us to find a pattern between robot degree and the environmental conditions. Then, with the most promising results and configurations, we turn back to Webots to perform the more refined and realistic simulations. The rest of the chapter is organized as follows: Section 3.1 shows the task allocation results and Section 3.2 the task partitioning ones.

5.1 Task allocation simulations

Aiming to check whether the information-sharing skills help robots to forage in a three-region environment while keeping autonomy in their distribution, we implemented four decentralized decision-making strategies: a deterministic (D), the semi-stochastic with (SS-TAM) and without TAMs help (SS), and the probabilistic (P). Although without TAMs information, robots would violate Assumption A6, we wanted to see if only receiving information from neighbors in other regions was enough to reach a good distribution (i.e., the SS model). The decision-making strategies should keep the system robustness, flexibility, and scalability in the foraging task. Moreover, robots should keep their autonomy and the distributed decision-making rules must restrain their decisions. We are measuring the performance of the decision-making models in the amount of objects delivered. Note that robots have to focus more on transporting objects than traveling between regions because if they keep traveling, they are not transporting objects. We provided a video of e-pucks foraging with the deterministic strategy in the supplementary material available in (NOGALES, OLIVEIRA, 2017a).

Since TAMs can share information with TAMs in other regions, they get more accurate information. For the deterministic model, TAMs decide how many robots should leave and toward which region. Thus, in each region, TAMs control and allocate the robots to attend their needs of foragers. Therefore, this model works as a reference. In the other strategies, robots make their own decisions about leaving toward a promising region. Robots need a structure of communication to share information between them (i.e., working with Assumption A6 and A7). We tested three kinds of structures: one fully connected and two regular networks with degree 2 and 3. We also tested a variation with switching-links strategy in the information-sharing structure to observe the effects of changing neighbors. For this variation, we added a suffix (-S). In particular, the switching consists of changing neighbors after a robot leaves a region. It would connect to some of the robots in the new region. This means they can use a communication hardware of a lower range and cost.

Here, robots and TAMs have the same rate of learning, $\alpha = \Lambda = 0.6$ and both have full cooperation between regions, $\phi = \phi_\ell = 100\%$ in Eqs. (9) and (10), respectively. The value of α guarantees that robots would consider both new and previous information, while we expect to have a full cooperation in their activities. Each region (or node) measures its utility as the number of foraged objects and its marginal utility as the change in the utility generated by robots working in that region within a period. We established the same period of 1,000 steps for evaluation of the marginal utilities in each region. We ran these experiments in Netlogo, and then we worked in Webots with the best conditions we found in Netlogo. The following section shows the non-linear effects of the rate of production of the regions, with which we could find the marginal utility functions.

5.1.1 Preliminary analysis

Recall that robots required the marginal utility estimations to find the optimal distribution point ($\Delta_q^\star$ of Eq. (6)), i.e., a distribution when all marginal utilities reach the same value and no robot has an incentive to abandon its region. Then, initially, we need to find the marginal utility functions for each region and its rates of object production. In particular, the capacity of sources to produce objects is the rate of object production, which can be either 20%, 40%, 60%, 80%, or 100%. After robots remove objects, these rates indicate the probability of reappearance.

For reasons of time, we let all regions with the same dimensions, same amount of TAMs available. Thus, choosing a region i and variating its rate Op_i would be enough to find the marginal utilities of all regions. Next, regions can have different rates of object production, i.e., $Op_i \neq Op_j \ \forall i, j \in N$, which would demand a different number of robots serving each region.

The period of evaluation of performance in each region is 1,000 steps. Since there are five sources, each could produce objects every 200 steps. Thus, for instance, let us assume that robots removed all objects from sources at once from TAMs in two regions whose rates are 100% and 20%, respectively. Then, the region whose rate is 100% could have all objects in 200 steps and the region with 20% could likely restore one object within the same 200 steps. At the end of the 1,000 steps (a period), the region with 100% would have produced 25 objects, while the other region only 5 objects. Therefore, the higher the rate of a region, the more robots it would need. However, it is important to mention that the need for foragers is non-linear across rates. The average of 30 simulations at each possible rate delivered the functions shown in Figure 22.

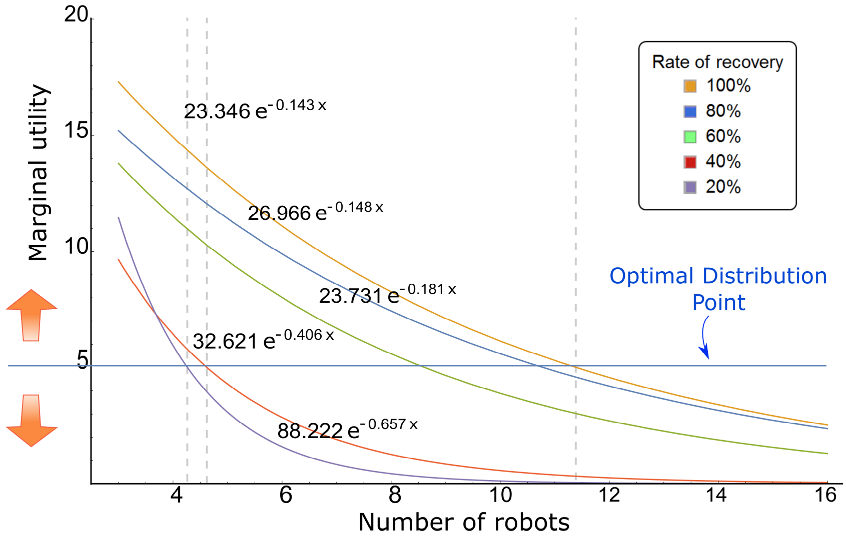


Figure 22 – Marginal utility functions for the different rates of object production of the sources. The arrows show the possible movement of the optimal distribution point to work with different team sizes.

Using these marginal utility functions, we could get an estimation of the number of robots the environment requires. First, let us assume that the regions have the following rates: the first region (red ground) with 20%, the second (gray ground) with 40%, and the third (blue ground) with 100%. Moreover, let us assume that we have only 20 robots available. Then, we need to define an optimal distribution point and look for the number of robots at the interception with the utility functions. By letting the optimal point to be the one of Figure 22, you may follow the dotted lines at its interception with the utility functions and look for the number of robots in the X-axis. Note that the first region, whose rate is 20%, would need almost 4 robots, the second region (rate of 40%) around 4, and the third region (rate of 100%) needs almost 12 robots. Therefore, by letting this optimal point and these rates of production, we could employ all 20 robots. Note that if we move up the optimal distribution point, the number of robots reduces. Otherwise, it increases at each rate. If we have more or less robots, the line should move down or up to define the number of required robots for the current rates of production. Such changes in the number of required robots are due to the non-linearity of the robot performance at each rate. However, team performances follow the law of diminishing returns.

Finally, we can evaluate all decision-making strategies (D, P, SS, and SS-TAM) communicating over the three predefined structures with and without switching-links option (suffix -S). Since robots do not decide in the deterministic decision-making (D), the variation D-S does not make sense. Thus, we got 21 different configurations to test with two group sizes. The following sections show the results of the experiments.

5.1.2 Small group

For the small group, after settling the rates in 20%, 40%, and 80%, the optimal distribution point (with marginal utility around 6 in Figure 22) indicates that we needed 16 robots for this configuration. The optimal distribution for this point is 2-5-9 (i.e., 2 in the red region, 5 for the gray, and 9 for the blue one). We tried a different initial distribution 5-5-6 in order to avoid starting in the optimal robot distribution. Figure 23 shows the results from simulations of 10,000 steps.

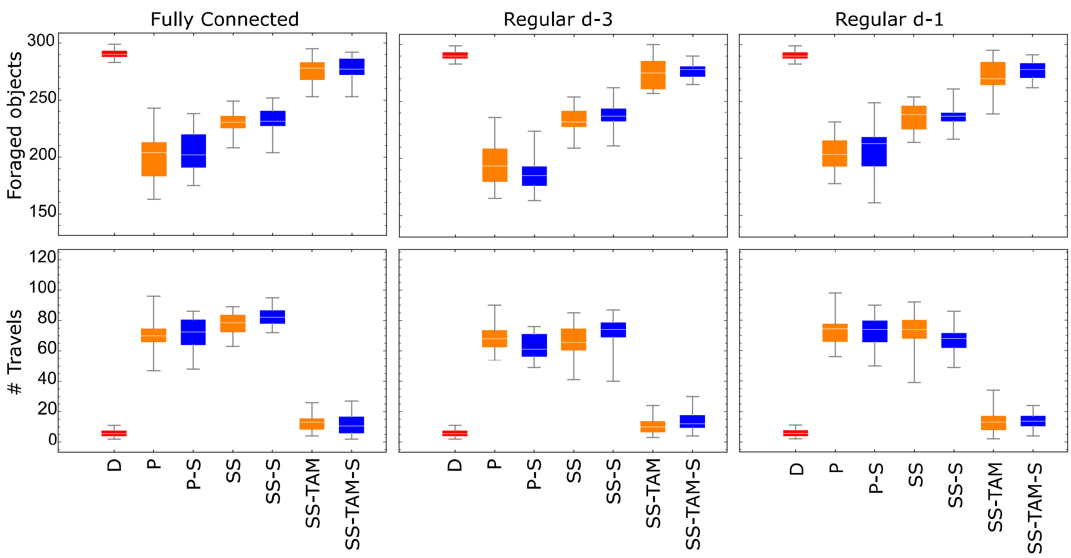


Figure 23 – Performance and travels of the small group of robots while foraging in the environment with different communication structures. Decision-making models should go from 5-5-6 toward the desired distribution 2-5-9. Orange box-plots are the results without the switching-links strategy, while blue box-plots for results with that strategy.

Note that despite the decision-making strategies should move only 3 robots from region red to blue, given the initial distribution, the distributed decision-making strategies did more travels. Recall that the deterministic model is the ideal reference. Therefore, as expected, it delivered the best performance, the t-test showed that it has no competitor.

However, this model sacrifices robot autonomy because TAMs decide when a robot could leave and towards where it should move. Among the decision-making models that enable robot autonomy, the SS-TAM-S delivered the best performance. In particular, SS-TAM-S with a regular network of degree 3 got the second place (losing by an average of 18 objects against the deterministic one, with $p = 0.001$). Moreover, by comparing the different network structures on the SS-TAM-S, the t-test indicates that there is no significant difference between them. The fully connected network got $p = 0.933$ for the regular network of degree 2 and $p = 0.822$ for the regular one with degree 3. From these results, we could conclude that by sharing information over a regular network of low degree, robots could reach similar results to a fully connected one. This can be translated to a lower communication cost to reach similar performances in the foraging task, both in hardware and computing.

Furthermore, the box-plots show that by allowing robots to switch their links decreases the variation of the results. Moreover, the t-test indicates there is no significant advantage in keeping the structure of communication static. This holds for any model combined with any network structure. Switching robot-robot connections would not affect the group performance significantly. This can be translated to a cheaper, low-range, and low-consumption hardware of communication.

By comparing SS-TAM and SS, we find that the information shared with TAMs was fundamental to reduce the number of travels. Robots in these two strategies have more autonomy while foraging than those with the deterministic model. They made their own decisions about transitioning between regions. In the SS model, information of TAMs was unavailable, robots distributed themselves based only on the information shared among neighbors. This model delivered a bad performance because robots spent more time traveling than foraging. Therefore, without Assumption A6, there was a lack of information that yielded inaccurate decisions and increased the number of travels between regions of robots.

Next, by comparing the results of the semi-stochastic model (SS) vs. the probabilistic (P), the t-test confirmed that SS won in all communication structures. In particular, the lowest difference was almost 28 objects for the fully connected network, with $p >> 0.05$. The greatest difference was almost 52 objects, with $p >> 0.05$, for the regular network with degree 3. Finally, note that these two decision-making models got the greater amount of travels and their performances are the worst, that is, they did not help robots to focus on their foraging tasks instead of wasting their time in traveling.

We computed the Euclidean distance of the last distribution to see which strategy was near to the optimal distribution point $\Delta_q^\star$. The results of the distances are in Table 3. These values were taken from simulations that ran 3 periods of 1,000 steps after the deterministic model reached a value near the equilibrium. Note that the probabilistic model has the greatest distance to the optimal point and the worst performance. The SS and SS-TAM models (with and without TAM information) delivered a value near to the optimal distribution.

	Structure		
Model	Fully connected	Regular d-3	Regular d-1
D	1.31	1.31	1.31
SS	1.75	1.72	3.42
SS-S	2.23	2.39	2.60
SS-TAM	1.71	1.04	1.74
SS-TAM-S	1.53	0.80	1.48
P	10.32	11.25	11.12
P-S	11.09	12.60	11.16

Table 3 – Values of the Euclidean distance of the last distribution of robots found in each implemented model and the computed optimal distribution 2-5-9.

We also observed the evolution of the marginal utilities. Figure 24 shows the settling time for the evolution of marginal utilities of each model. It is clear that the faster model to reach the balance is the deterministic one. Note that the SS-TAM model was faster than SS to reach a point where all regions delivered similar marginal utilities. Again, the information of TAMs was crucial for this faster allocation.

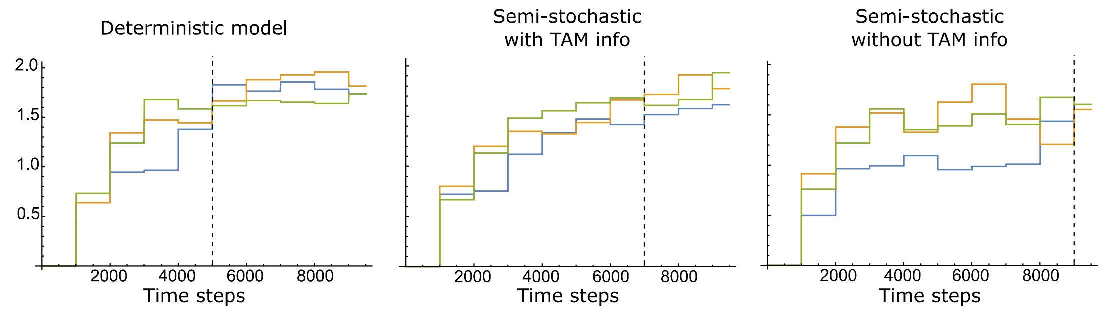


Figure 24 – Evolution of the performance with a period of update in TAMs of 1,000 steps. The dashed line represents the moment when the marginal utilities reached an almost similar value to the expected one.

Note that though robots reached a near-optimal solution, the semi-stochastic model without information of the TAMs tarried too much to reach a good distribution of the robots. The SS-TAM model shows the importance of the environmental aids offered by TAMs. We infer that the SS model delayed to reach a near-optimal value due to diversity in the estimations of each robot, i.e., it lacks of more accurate information.

5.1.3 Large group

For the large group, after settling the rate of the regions in 20%, 40%, and 60%, the optimal distribution point was moved towards 3 in Figure 22. Such point indicated that this configuration would need 32 robots (double size). The optimal distribution of robots for these rates is 5-11-16 (i.e., 5 in the red region, 11 for the gray, and 16 for the blue region). We tried a different initial distribution 1-15-16 in order to avoid starting in the optimal robot distribution. Figure 25 shows the results of 30 simulations, each of 10,000 steps.

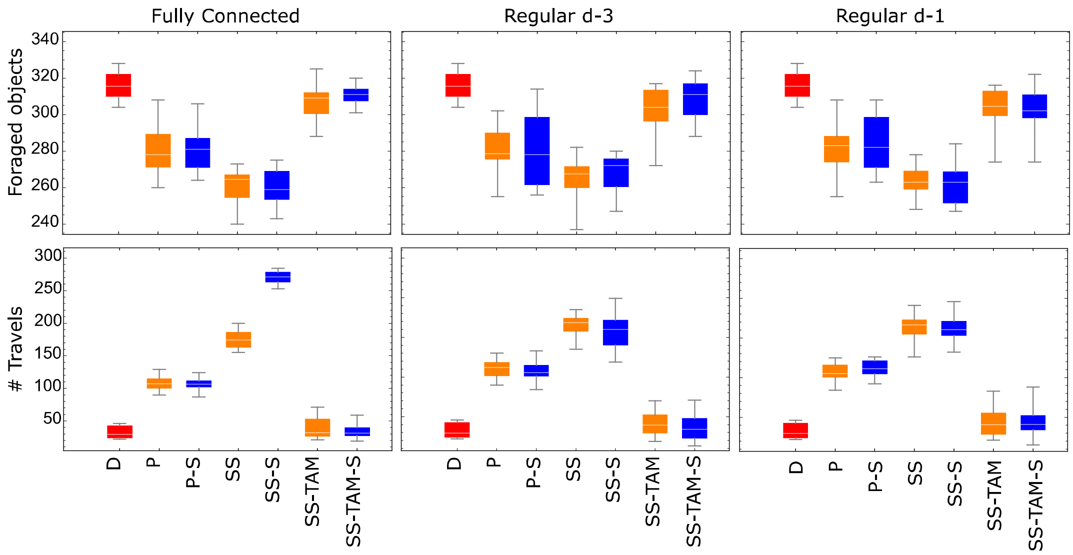


Figure 25 – Performance and travels of the large group of robots while foraging in the environment with different communication structures. Decision-making strategies should go from 1-15-16 toward the desired distribution 5-11-16. Orange box-plots are the results without the switching-links strategy, while blue box-plots for results with that strategy.

In these experiments, we opted for an initial distribution requiring only 4 robots to move from the blue region toward the red one. However, again the distributed decision-making strategies did more than 4 travels. As expected, the deterministic model (D) delivered the best performance; the t-test showed it has surpassed all other models. Recall that this model delivers the authority for travels between regions to the TAMs, i.e., regions bereave robots of their autonomy.

Among the models that enable robot autonomy, the SS-TAM-S model delivered the best performance. In particular, the SS-TAM-S model with a fully connected network got the second place with a difference of 13 objects in the average performance and $p = 0.002$. But, the t-test indicates that there is no significant difference between the fully connected structure and the regular one with degree 3 ($p = 0.511$). This time, decreasing to much the number of connection in the network yielded a significant difference between the fully connected network when compared to the regular one of degree 2 ($p = 0.017$). Therefore, robots should share information over a regular network of low degree (but not too low) and they would reach similar results of foraging with a expensive fully connected one. This means a moderate cost in communication hardware.

Next, by comparing the models with and without the switching-link option (-S), there is no significant difference in any network structure. Note that there is a lower variability among the simulations (i.e., the box-plots seem shorter) with the switching variation (-S) in all models. Moreover, the t-test results confirmed that robots could forage with a cheaper, low-range, and low-consumption hardware of communication because they delivered good performance with the (-S) option.

In this large group scenario, the comparison between the semi-stochastic model (SS) vs. the probabilistic (P) indicates that the latter won in all communication structures. In particular, the lowest difference was almost 15 objects (and $p >> 0.05$) in the regular network with degree 3, while the greatest difference was around 28 objects (and $p >> 0.05$) for the regular network with degree 2. This happened because more robots brought more diversity in their estimations, as a consequence, robots made more inaccurate decisions and traveled more to explore other regions. Moreover, recall that SS is violating the Assumption A6, that is, the condition of information-sharing between TAMs and robots. In the travel plots, we could confirm that these two models delivered the lowest amount of objects, because robots focused more on traveling than foraging.

Next, we computed the Euclidean distance of the last distribution to see which strategy was near to the optimal distribution point $\Delta_q^\star$. We allowed simulations to run 3 periods of 1,000 steps after the deterministic model reached a value near the equilibrium. Table 4 shows the results of these distances.

	Structure		
Model	Fully connected	Regular d-3	Regular d-1
D	1.34	1.34	1.34
P	21.06	19.12	20.69
P-S	21.38	20.50	17.90
SS	6.19	3.40	4.48
SS-S	2.86	4.86	4.58
SS-TAM	2.27	1.96	2.45
SS-TAM-S	1.74	2.24	2.33

Table 4 – Values of the Euclidean distance of the last distribution of robots found in each implemented model and the computed optimal distribution 5-11-16.

We arrived at similar results of the small group. Nevertheless, the semi-stochastic strategy without help from TAMs worsened its performance. It took too long for the SS strategy to reach a near-optimal allocation, i.e., it had a slow allocation process. Since the group had a greater number of robots, the diversity in the information increased and, as a consequence, robot decisions were far from the optimal one. In the travel plots, we could confirm that SS had the greatest amount of travels and the lowest number of foraged objects.

5.2 Task partitioning simulations

The decision-making models should keep the system robustness, flexibility, and scalability in the foraging task. Moreover, robots should keep their autonomy, but distributed control rules must restrain their decisions. We measured the decision-making performances in the amount of objects delivered. We provided a video of e-pucks foraging with the M-SGU strategy in the supplementary material available in (NOGALES, OLIVEIRA, 2017b). Aiming to compare our proposal adaptation skills, we consider the following:

- **Always:** robots will always choose to partition, $P_p = 1$.
- **Never:** robots will never partition, $P_p = 0$.
- **M-2011:** $M = 1$ parameter is static as in (PINI et al., 2011b).
- **M-2013:** *give up* decisions use a threshold defined in (PINI et al., 2013), $\tau = 3\hat{t}$.
- **M-SGU:** static value defined offline, $M = 5$.
- **M-AGU:** M adapts online by following Eq. (20).
- **Greedy:** consider $\varepsilon = 10\%$ of probability to choose the worst option.

Always and Never strategies lack adaptability, but we could compute an upper bound of the group performance by combining them. In preliminary simulations, they also helped us to define short, long cache delays, and the static M parameter of the M-SGU model. We could define a short delay of 75 time steps, which is good for task partitioning, i.e., robots should partition the transportation. On the other hand, we let 1000 time steps for long delays to avoid task partitioning, i.e., if cache delays are long is better for robots to employ the non-partitioning option. Thus, short delays indicate when robots should transfer the objects through caches and long delays refer when robots should travel through the path and complete the task alone.

The offline value of M for M-SGU came from the evaluation of the average of the estimations from preliminary simulations with Always and Never strategies in Eq. (20). When estimations in both options are equal, robots get 50% of chances to abandon the caches or keep struggling in them to complete a transference. Furthermore, the **G?** and **P?** functions also change for each strategy. For instance, the strategy M-SGU considers Eq. 18 with $M = 5$, while M-AGU is adjusted online, i.e., robots foraging with M-AGU can adjust M parameter while they perform the tasks.

Moreover, we enabled or disabled in the controller the *abandon* transitions according to the strategy of decision-making. For instance, robots following M-2011 and M-2013 can only abandon activities involving partitioning decisions, as indicated in (PINI et al., 2011b; PINI et al., 2013). Then, we disabled the *abandon* transitions from all states that are not related to the caches (see Fig. 21). Robots following the proposed decision-making, M-SGU and M-AGU, can abandon any activity by employing the proposed *abandon* transitions. Then, we enabled all of them. The other parameters for M-2011, M-2013, M-AGU, and M-SGU were taken from the best results of the model in (PINI et al., 2011b): $S = -2.5$, $K = -0.6$, and $\mathcal{O} = -5$. The parameter for learning estimations is $\alpha = 0.6$.

According to the results in (PINI et al., 2013), the Greedy strategy worked as a reference strategy with a small $\varepsilon = 0.1$. This strategy forces robots to follow the best option once they discover the actual delay in caches. But, it provides a small probability ε of exploration of other alternatives. Besides, the adaptability of this strategy depends only on the performance of old experiences and does not consider current conditions.

Based on the estimated times for each activity working with short cache delays, we defined the interval [200, 800] for activities in the partitioning option. While the average estimation for robots foraging with the Never strategy helped to define the interval [1000, 2500] for activities in the non-partitioning option. Thus, robot estimations were randomly initialized within these intervals. Note that such initial values would favor the partitioning option, then robots would begin favoring the partitioning activities. Therefore, we have a set of challenges to test the decision-making models. The following section details the challenges the implemented strategies should overcome.

5.2.1 Challenges for the decision-making models

Environmental congestions and the delays to emulate a successful dropping/picking up of a virtual object in caches affect robot estimations. Congestions are created by robot motions and cannot be predicted due to the randomness of the robot navigation. However, we can control the delays on the transferences in caches to test the adaptability of robots to environmental changes. If cache delays are short, then foraging through the caches will be faster than traveling through the path, i.e., robots should favor task partitioning transport. Otherwise, the robots should travel to the nest and avoid the caches, i.e., they should opt for the non-partitioning option because delays are too long.

Afterward, all models deal with a set of challenges to test their adaptability. Each challenge consists of a particular behavior in the delays of the area for transference (e.g., a constant delay, step-down or step-up change). Moreover, we explored different static networks to find a pattern between a structural parameter and the environmental condition. Once, we identified a relationship between the degree and each environmental condition, we fostered each robot with a mechanism for adjusting its connections according to the environmental conditions.

5.2.1.1 Challenges nature

The proposed challenges defy the team adaptability. Recall that adaptability refers to the adjustments a system does to cope with negative effects of external influence

(MARTIN et al., 2009). In particular, we challenge the robots by considering constant, abrupt, and reoccurring changes in the cache delays. Figure 26 shows the nature of delay changes. Neither oscillations nor spiky changes apply because robots need a steady time to learn. Even when they can notify their neighbors, they have to fulfill an activity and it requires time. Besides, new information requires more time to spread through the communication structure.

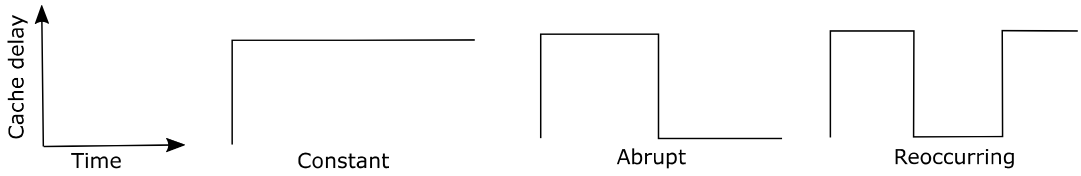


Figure 26 – Visual representation of challenges with which the robots have to deal. Long delays are up while the short value is down.

Each simulation takes 120 minutes of simulated time, the same time the battery of our real e-puck takes to discharge. In the environments, nests and sources have a fixed value of delays during the 120 minutes of each simulation. But, delays in receiving and delivering virtual objects in caches can change.

We executed some preliminary simulations to find out those values. First, we set a value of delay and ran some simulations with robots always opting for partitioning (Always strategy). Then, we increased the delay and tested again. More simulations ran with robots foraging through the paths, i.e., opting for the non-partitioning option (Never strategy). Once the average performance of robots partitioning became lower than the one by non-partitioning, we discovered the values for both kinds of delays. Those preliminary simulations helped us to identify which delays were short and which ones were long delays. Then, we change them during simulations Those changes allowed us to validate the effect of the proposed decision-making models as well as the mechanism for adjusting the connections of each robot.

5.2.1.2 Challenges for Webots

In Webots, robots have to deal with two abrupt challenges that defy their adaptability: a step-up change in delay and a step-down. Since robots begin favoring the partitioning option, in the step-up challenge, they do not need to adapt their decisions because this challenge begins with short delays. But, after 45 minutes delays change to the large value and robots must adapt to it. Next, in the step-down change, the environment begins with a different condition to the one that robots favor initially. Thus, they need the first adaptation. Then, after 45 minutes, when most robots have learned that the non-partitioning option is faster, the cache delays decrease. After this environmental change, the team requires adapting a second time.

Note that the step-up challenge only requires robots to learn the large values and adapt their decisions accordingly. Since robots initialize favoring task partitioning, this challenge would be easy to overcome. However, the step-down challenge requires both to adjust and to reconsider alternative and old options for a better adaptability.

5.2.1.3 Challenges for Netlogo

In Webots, robots follow a Levy exploration and do not have dead-reckoning or global guidance. But, their navigation depends on their small range of vision from which they can detect objects. This range of vision was replicated in Netlogo by emulating through the distance between a robot and its target. Also, while a robot travels through the path, its speed reduces in order to reach a similar ratio between the estimated time to complete the transportation through partitioning and non-partitioning options in both simulators.

Since simulations in Netlogo run faster, we extended the time to 15,000 steps, that is, about 5 hours of Webots. In particular, 3,000 steps in Netlogo $\approx$ 1 hour in Webots. Simulations in Netlogo ran in less than 15 minutes. The interval of simulation in Netlogo was divided into steady sub-intervals as the challenge considered. For instance, a reoccurring challenge going from short to long and returning to short delays again, it would have 5,000 steps for each value of delay. Step-down and step-up challenges consist of two periods of 7,500 steps.

For the robot-robot communication, Netlogo allowed us to test several static networks to find the pattern between degree and the delays of caches (the environmental change). This local parameter helps to regulate the speed of learning by modifying the communication structure where information spreads. Once we found a pattern between robot degree and the environmental conditions in Netlogo, we move back to Webots. We embedded the mechanism for adjusting connections in the decision-making models to help each robot. This mechanism allows each robot to increase or decrease its degree.

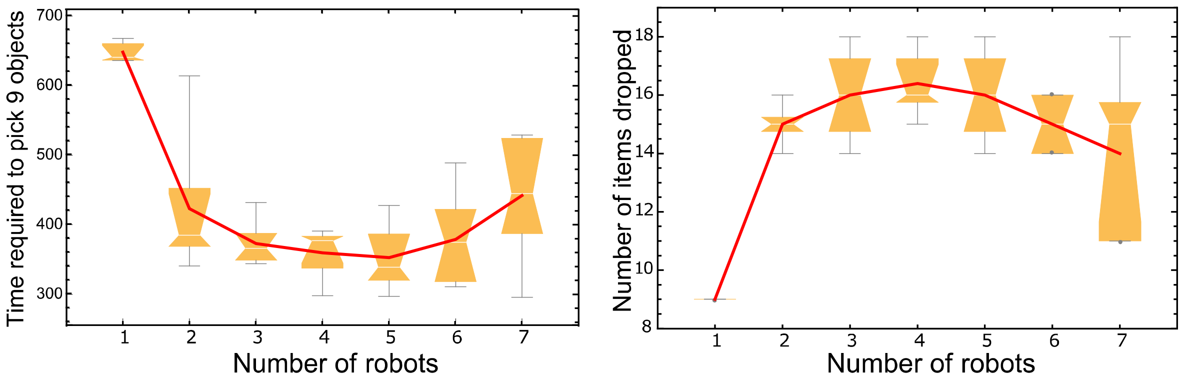
5.3 Simulations and results

This section details how decision-making models dealt with environmental challenges. Besides, it explains how the law of diminishing returns helped us to find the optimal number of robots for each environment. We implemented seven decision-making models, but two of them lack adaptability. However, these two helped us to compute an upper bound on the performance. We only compared the performance of the models with adaptability to check their speed for adaptation. Then, we took the best three models to embed the mechanism for changing connections in them. We provided a video of e-pucks foraging in the supplementary material available in (NOGALES, OLIVEIRA, 2017b).

Recall that all simulations initialize robot estimations favoring the partitioning options. But, each decision-making model has to deal with several challenges. In Netlogo, robots deal with four challenges: constant with long and short delays, abrupt with step-down change, and reoccurring with up-down-up change. In Webots, they deal with two: one where robots need to adapt their decisions after detecting the change and other where they have to adapt twice. The later challenge forces robots to reconsider their decisions, i.e., their skill to keep exploring alternatives. Moreover, we tested the scalability of the decision-making models with different team sizes. The scalability was tested with more robots and resources in the environment.

5.3.1 Preliminary analysis

Since TAMs and other environmental resources are limited, we must first find out the optimal number of robots for each environment. Adding robots could not only increase the number of collected objects, but traffic and conflicts. They could reach a point where team performance would decrease. To avoid this, we employ a diminishing returns analysis, which requires all variables to remain constant while increasing the variable of interest to observe its effect on the output. In our case, we added another robot in the environment, ran some simulations, and analyzed the results in both the time of execution and objects collected. This was done until the utility behavior changes. At that point, we found an optimal number of robots for each environment. Figure 27 shows the results of this analysis in the small environment.



(a) (b)

Figure 27 – Diminishing return analysis obtained by increasing the number of robots in the source region. The red curve in (a) shows the decreasing effects upon the time when adding more robots to transport 9 objects. The red curve in (b) shows the effect over the amount of transported objects within a fixed time.

In these experiments, we were looking for the inflection point in which the performance begins to deteriorate in a region. Since both regions are symmetric, we picked the region of sources and measured two things: how much time robots required the team to transport 9 objects and the second part consists of measuring how much objects they may forage within a fixed interval. We let simulations run 11.5 minutes, the same amount of time required for a robot to transport 9 objects. Finally, we got from the curves a similar point of inflection around 4. After this size, adding more robots does not bring any benefit. Therefore, the small environment can support up to 4 robots in each region, i.e., 8 robots in total. Next, since the large environment has broader regions and more TAMs, more robots could forage in it. In particular, a similar analysis delivered us that the large environment can support up to 16 robots.

The following section compares two sets of experiments over the decision-making models: one without structural changes and another with structural changes. decision-making models were compared by their performance. We also check if robots following the expected decisions would forage for more objects. Some simulations ran as if robots had perfect information about the delays to make the expected decisions. We ran these simulations to compare whether perfect knowledge of the environmental decisions could deliver a better performance.

5.3.2 Experiments using a fixed communication structure

Aiming to validate the adaptability of the different strategies, we executed the following three stages: (*i*) Find the optimal number of robots for the environment through the diminishing returns analysis, (*ii*) Run preliminary simulations to find the performance in constant challenges with short and long cache delays favoring either task partitioning or non-partitioning options, and (*iii*) Validate adaptability through simulations with the decision-making models: M-2011, M-2013, M-SGU, M-AGU, and Greedy. We executed all these three stages for both environments of Figure 15. We also performed the t-test to check which strategy brought significant improvements to performance.

5.3.2.1 Simulations in the small environment

For these simulations, robots had to collect objects in the small environment shown in Figure 15-a. The diminishing returns analysis indicated that 8 robots are an optimal number to forage in this environment. Then, we executed 30 simulations for the Always and Never strategies to get an expectation of the performance of the strategies with adaptability. Table 5 shows the performance of Always and Never with short and long delays in caches.

Strategy	Delay (in time steps)	Number of objects transported	Standard Dev.
Always	75	78.00	4.65
	1000	28.09	2.06
Never	-	39.89	3.92

Table 5 – Average number of objects successfully delivered to nests.

Results in the Always and Never strategies provide an idea of how many objects the robots should transport if they chose the best option as soon as the cache delays changed (ideally). We weighted the performances of the Always and Never strategies according to the simulated time in which caches were working on short and long delays to estimate the number of objects. For instance, the step-up challenge combines 45/120*(objects transported in the Always strategy with short delays) + 75/120* (Objects transported with the Never strategy). Thus, robots should transport around 54 objects in this challenge.

Step-up challenge: the decision-making models have to deal with cache delays beginning with 75 steps. After 45 minutes of simulated time, delays change to 1000 steps on each side of the cache. Since a complete simulation takes 120 min, robots have to adapt their decisions to the non-partitioning option after this change occurs. Thus, more robots should end traveling through the alternative path.

Figure 28 shows the performance of the strategies M-2011, M-2013, M-AGU, M-SGU, and Greedy dealing with this challenge. Note that by combining the Always and Never strategies results in Table 5, the expected number of foraged objects is 54. Figure 28 shows that global communication brought an improvement in all strategies. T-test results show that the proposed strategies surpass Greedy when robots can share their experiences. However, without communication, only M-SGU and M-AGU reach a similar performance to the Greedy. In particular, although the Greedy strategy fits better in this challenge, M-SGU gets a small improvement vs. the Greedy, $p = 0.09$ for the t-test.

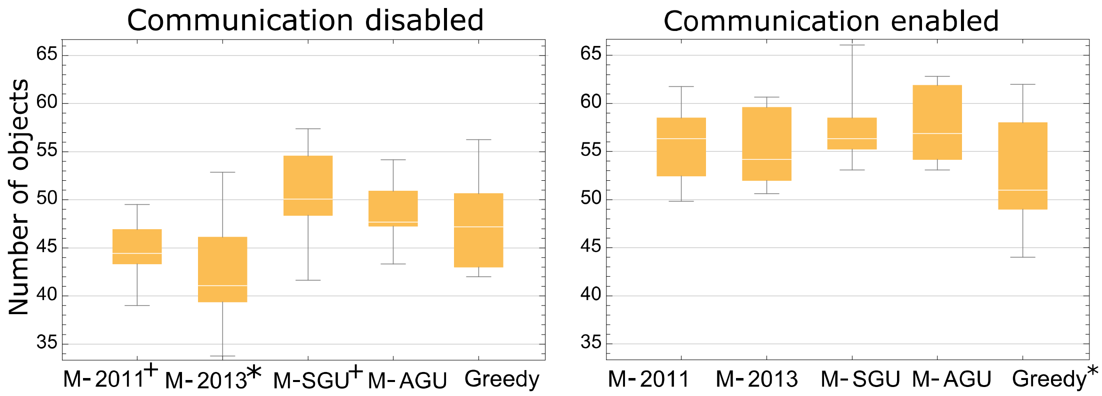


Figure 28 – Amount of objects successfully delivered to nests when cache delays change from 75 steps to 1000 steps at 45 minutes of simulation. $^*p < 0.05$ t-test vs. both M-SGU and M-AGU.

Step-down challenge: this time, caches begin with long delays (of 1000 steps) and, after 45 minutes, the delays decrease to 75 time steps. Since robots initialize their estimations favoring the partitioning option, robots need to adapt twice. They begin expecting short delays in the caches but cache delays begin with 1000 steps. Then, it is easier to use the alternative path, the non-partitioning option. Consequently, they need to do the first adaptation as soon as possible to cope with this environmental condition.

Later, when cache delays decrease to 75 steps, robots need to keep exploring alternative options to discover this change and adapt to it.

Figure 29 shows the performance of the strategies M-2011, M-2013, M-AGU, M-SGU, and Greedy in this challenge with step-down change. By combining Always and Never preliminary results from Table 5, the expected number of foraged objects is 64 objects. Since no strategy was near this value, we can affirm that a step-down challenge is more difficult and requires more adaptability from the decision-making models. In this case, global communication deteriorated the group performance of almost all strategies. Concluding, communication was not good for all environmental conditions.

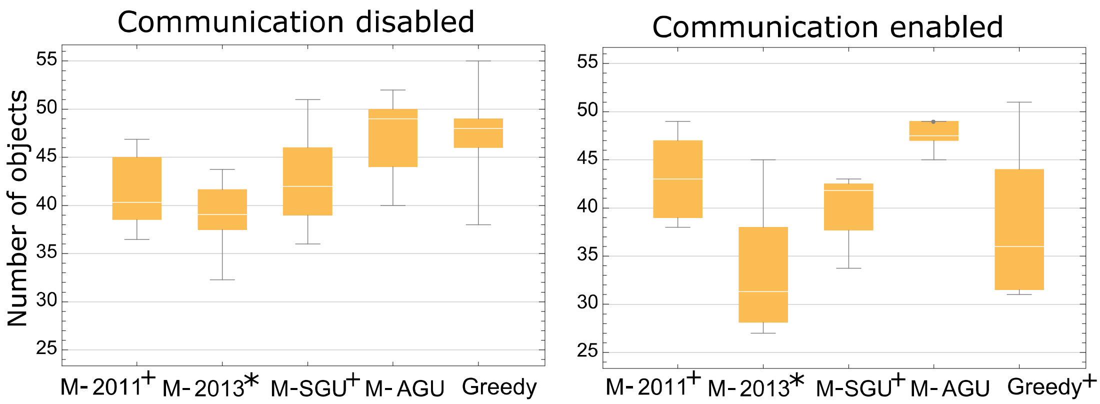


Figure 29 – Amount of objects successfully delivered to nests when cache delays change from 1000 steps to 75 steps at 45 minutes of simulation. $^{*}p < 0.05$ t-test vs. both M-SGU and M-AGU, while $^{+}$ refers $p < 0.05$ only vs. M-AGU.

T-test results showed a significant improvement of M-AGU against the strategies from the base models, M-2011 and M-2013. But, M-AGU did not surpass the Greedy strategy without communication. Note that communication favors M-AGU because its average is greater and its standard deviation is lower, i.e., teams exhibit similar performance in all simulations. Next, although the M-SGU strategy had an offline adjustment for this environment, its performance was poor. Each robot knew the best M parameter for ideal and not for real conditions. This confirms that a static value does not work well in dynamical environments.

Robot decisions: in the step-down challenge, robots were following different solutions to the expected decisions and reaching good performances. We also noted that robots foraging with the Greedy strategy had problems dealing with this challenge. Thus, we verified whether teams that followed the best option were among the ones that got good performances.

Figure 30 shows the evolution of the number of robots partitioning. Note that although the M-2013 strategy presented a close behavior to the ideal decisions (Expected curve), and it offered the worst performance. Seeking an explanation, note that in Figure 30-5, the number of robots partitioning is low in the first part, because almost all robots decide to travel at the same time. Next, since local landmarks are small, only the nearest robots can visualize them and begin their travel. However, they keep it occluded for other robots. Consequently, queues of robots looking for the landmarks emerge.

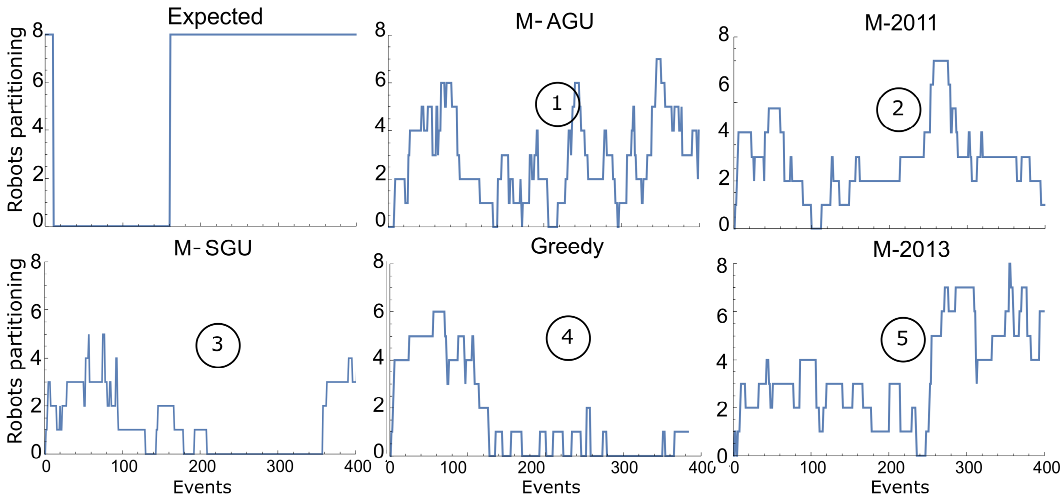


Figure 30 – Number of robots partitioning when cache delays change from 1000 to 75 steps. The Expected plot shows the expected number of robots partitioning based on results from Table 5. The numbers within circles show the ranking of the average performance. Data from results with communication enabled.

Queues and traffic conflicts affect the group performance, because those robots are not foraging, they are waiting to enter the path. Moreover, note that when the cache changed to a short delay, robots continue struggling to travel through the path. Only when robots end the trip, they would change their decisions and begin to use the caches (note that the number of robots partitioning increased). Note also that almost all robots decided to use the caches at the same time, and then, they began to compete for the available caches. In M-2013, the option of abandoning previous decisions fails. In particular, the M-2013 strategy did not allow robots to abandon earlier those activities. They have to wait threefold the estimated time in activities related to the caches. Note that the Greedy also failed in providing this option, too.

Robots foraging with M-2011 and M-2013 could not compare both alternatives. The strategies M-SGU and M-AGU allows robots to do this kind of comparisons and to abandon any activity. However, M-AGU delivered a better position. It offers robots the possibility to adjust their own functions according to their experiences. Therefore, robots that found the landmark and traveled through the path can inform others about their good experience. Note that there is a lot of oscillations in their decisions. The faster adaptation of decisions in M-AGU could cope with these conflicts and deliver a good performance.

If robots cannot abandon any previous decision faster, they will affect the group performance. Figures 28 and 29 show that the adaptive *give up* function and the new *abandon* transitions brought a faster adaptation. Recall that those transitions allow robots to abandon any previous decision after struggling for a while to complete it. Since the M-SGU strategy also includes those transitions, we confirm that the online adaptation of the parameter M in the new M-AGU strategy is an essential feature. Note that M-AGU could overcome both challenges. It delivered a good performance level with and without communication.

The current environmental conditions (interference and queues) push robots to choose an alternative option. No matter if that option has a longer estimated time; it is the best option for the group at that moment. These results show that the combination of online adaptability with a continuous comparison of both options to reconsider whether to abandon or continue struggling may lead to better performances. The following section presents simulations that check scalability of these strategies.

5.3.2.2 Simulations in the large environment

In these simulations, robots were foraging in the environment shown in Figure 15-b. The diminishing returns analysis indicated that 16 robots are the optimal number for this environment (8 robots for each region). Since the regions of foraging and the number of TAMs almost doubled those of the small environment, the number of robots also doubled. We executed 30 simulations for the Always and Never strategies to get an expectation of the ideal performance. As in the previous experiments, the results from Table 6 provided the expected number of foraged objects for this environment. But, only the more difficult challenge was tested in these experiments.

Since we wanted to test if the static M-SGU model could satisfy the scalability requirement, we did not compute a new M parameter. Although using the average of the estimated times from the Always and Never simulations in Eq. (20), we could find the offline value of M for this environment. It is important to mention that robots initialized their estimations within the same intervals of the previous experiments.

Strategy	Delay (in time steps)	Number of objects transported	Standard Dev.
Always	75	123.37	5.59
	1000	62.09	2.78
Never	-	84.51	2.59

Table 6 – Average number of objects successfully delivered to nests working in the environment shown in Figure 15-b

Step-down change: 16 robots must adapt twice to overcome this challenge. Delays begin with a value of 1000 time steps (long) and, after 45 minutes of simulated time, caches change to short delays of 75 steps on each side. By combining Always and Never results in Table 6, the expected number of foraged objects results in 98.

Figure 31 shows the performance of each strategy in the large environment. Note that the new strategy M-AGU offered better results for this challenge. It surpassed M-SGU, M-2011, and M-2013; but it did not surpass the Greedy strategy with communication. This time, global communication helped some strategies more than others. In particular, Greedy and M-AGU improved by sharing experiences among robots. Note also that communication decreases the performance in teams working with the M-2013 and M-SGU strategies. For M-2011, the standard deviation decreased, i.e., performances are more similar in all simulations.

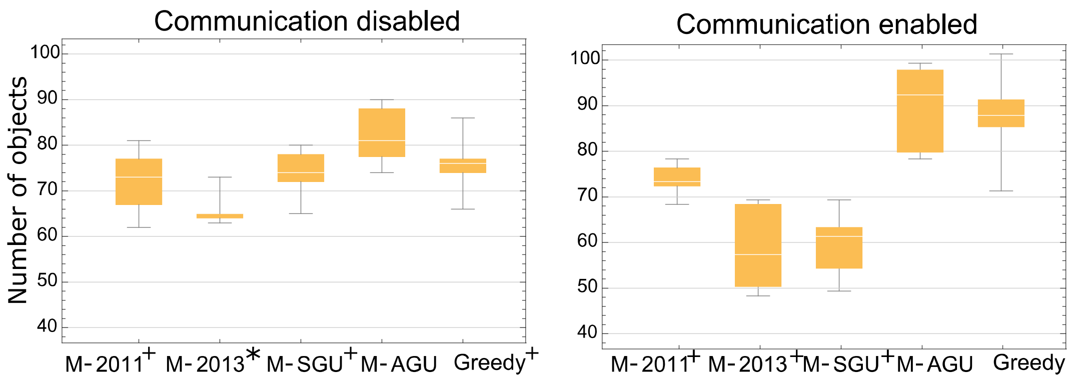


Figure 31 – Amount of objects successfully delivered at nests. Cache delays begin with 1000 steps and decrease to 75 steps at 45 minutes. $^{*}p < 0.05$ t-test vs. both M-SGU and M-AGU, while $^{+}$ refers $p < 0.05$ only vs. M-AGU.

Robot decisions: To verify whether robots were following the ideal decisions, we tracked the number of them working in partitioning and non-partitioning activities. In particular, Figure 32 shows the evolution of the number of robots partitioning over time. The Greedy strategy got a good performance. Robots began by partitioning. When they learned that cache takes longer, the number of robots partitioning decreases faster. Afterward, due to the small probability of exploration, ε, a few robots reconsidered partitioning. M-AGU and M-2011 seem to follow a contrary evolution of the expected one. The slope of the partitioning function probably yields that. Both need a greater number of experiences than the Greedy strategy does to adjust robot decisions. In other words, robots have a slow adjustment of their decisions due to Eq. (17). However, due to the randomness in decisions, some robots took the alternative path and traveled through it.

Again, massive group decisions created queues and conflicts because many robots were trying to find an available TAM or landmark at the same time. Note that M-2013 fails in both sets of experiments. Note also that M-AGU shows sharper changes in robot decisions. Such changes could be linked to *abandon* decisions, because, this time, there were more robots that could occlude both landmarks and TAMs for longer periods. However, M-AGU offered again the best performance dealing with dynamical environments. Therefore, decision-making models through which robots could change

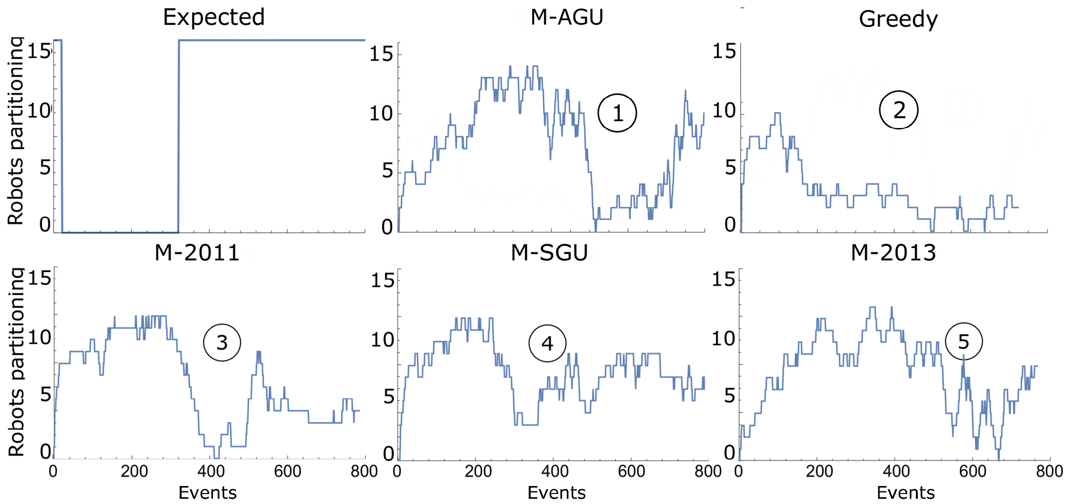


Figure 32 – Number of robots partitioning when cache changes from 1000 to 75 steps. The Expected plot shows the expected number of robots partitioning. The numbers within circles show the ranking of the average performance. Data with communication enabled.

their decisions and still keep exploring other options faster allows robots to detect more available resources (an available TAM or a landmark to travel to).

In general, robots following M-AGU could abandon any activity and cope with the conflicts and queues. In this strategy, robots follow the online adaptive *give up* function, which considers estimated times of both options. Moreover, M-AGU performed well in both environments satisfying scalability requirement and dealing well with both abrupt challenges. Robots increased the group adaptability by reconsidering their previous decisions using the current measured time vs. learned estimations. For instance, when robots abandon the travel option and try to forage through caches, even though the estimations of cache activities were longer, they were better for the group at that particular moment.

Note that global communication can generate conflicts and can help on certain occasions. Its negative effects occurred when all robots did the similar decisions based on the similar information they got. On the other hand, communication helped some strategies; it depends on how they took advantage of the information to improve their adaptability. Since the difference between M-SGU and M-AGU is the online adjustment of the M parameter in the *give up* function, we can affirm that this parameter is essential for the adaptability. How robots define and adjust the M parameter could change the way in which the group may adapt to the non-deterministic nature of the environment.

5.3.3 Experiments using an adaptive communication structure

In this section, we show the results of the experiments we performed to investigate how the communication structure can improve the adaptability of the decision-making strategies. We also introduced the Borda count concept to rank the communication structures that helped to improve locally robot adaptability. Note that all the implemented strategies have the same learning function given by Eq. (17). But, the social learning process influences the strategies differently. Each strategy has a different speed to adjust its decisions. By speed, we refer to how fast incoming information can influence robots.

The Greedy strategy has the fastest decision-making; robots always compare the costs of both alternatives and choose the best. The M-2011 and M-AGU strategies allow robots to compare their estimations in both options through a sigmoidal probability function before making a decision. A noticeable change in the estimations requires several interactions of a robot with the environment. Each time a robot picks up (drops) an object from source/deliverer-cache (nest/receiver-cache), it gets one experience. Then, both are slower than Greedy. In M-2011, the decision-making is even slower, because robots keep struggling too long to complete any activity. While in M-AGU, the decision-making has a middle-speed, because robots have the abandon transitions where they reconsider to continue working or abandon in any activity by using current information.

These strategies struggle with four challenges; Constant up, Constant down, Abrupt, and Reoccurring. Recall that a challenge is a particular behavior of the delays in the caches of the area for transferences, as those illustrated in Figure 26. Foraging under Constant challenges, we found a pattern between robot connections and the environmental conditions. This opened an opportunity to indicate to the robots how they should adapt their connections to overcome changes in the environmental conditions.

Initially, we explore four static structures and one dynamic structure in these three strategies of decision-making. Later, based on the major conclusions obtained in these exploratory simulations in Netlogo, we could perform and test our proposal in Webots.

5.3.3.1 Simulations in Netlogo

Here, simulations considered four communication structures that influence the information-sharing: Fully connected (F), Ring topology (R), small-world (SW) obtained by randomly changing some connections of a ring topology, and without connections (N). Aiming to identify the best structures, we adapted the Borda count technique to rank the four structures through their performance in each decision-making model. This technique shows the preference of a candidate by weighting the votes he/she got in each position of a list (EMERSON, 2013). Thus, the structures are the candidates, and the votes are the number of victories a communication structure got in the different team sizes and in the different challenges. For each strategy and for each size, the winner gets 3 points, the points decrease at each subsequent position such that the last one gets zero points. All values are the rounded average performance of 20 simulations, and short delays have 10 steps while long delays 1000 steps.

Table 7 shows the results of the Greedy strategy dealing with constant cache delays. To provide an example of the weighted sum of the Borda count, we detail the process for

Greedy (Fast decision making)

No. robots	Short cache delay						Long cache delay					
	1st		2nd		3rd		1st		2nd		3rd	
10	N	230	SW	214	F	211	F	76	SW	76	R	74
14	N	319	F	308	SW	295	F	107	SW	104	R	101
20	N	454	F	454	R	453	F	153	SW	145	R	141
24	N	558	R	546	SW	546	F	182	SW	176	R	171
30	N	691	R	693	SW	686	F	229	SW	221	R	207

Table 7 – The 3 best performances chosen by structure of foraging robots with Greedy.

Table 7 under short cache delays. The N structure got first place in 5 different team sizes (number of robots), which gives it 15 points. The R structure was second twice, which gives it 4 points, and one time it got third place, adding 1 point to its score; summing up, it gets 5 points in total. The SW structure got 2 points for its second place and 3 points for getting third place three times; it gets a score of 5 points. The F structure was second twice, which gives 4 points, and one in third place, which adds one point; it gets a score of 5 points. Since F structures are expensive for real implementation, if the Borda count of F ties with another structure, it will lose by cost. Table 8 summarizes the Borda count for the results presented in Table 7.

Ranking of best structures for Greedy

	Short cache delay				Long cache delay			
Structure	1st	2nd	3rd	Borda	1st	2nd	3rd	Borda
N	5	0	0	15	0	0	0	0
R	0	2	1	5	0	0	5	5
SW	0	1	3	5	0	5	0	10
F	0	2	1	5	5	0	0	15

Table 8 – Borda count ranking of robots foraging with the Greedy strategy.

For convenience, only the first two positions of the Borda count process are present in the M-2011 and M-AGU strategies. Two rows at the end of their respective table show their scores. The reader may compute the Borda count for M-AGU and M-2011 strategies following a similar process to the one described to create Table 8.

In the results from Tables 8, 9, and 10, note that when cache delays are long, the F structure ranked first in all strategies. In other words, it seems better to have a great number of connections to improve team performance when cache delays are long. Next, for short cache delays, structures with few or no connections ranked better. Only for the M-2011 (slowest) decision-making models, it seems better to have a fully connected structure no matter what challenges it deals with. Results delivered no absolute winner for the other strategies, i.e., no structure works as an optimal universal one.

M-AGU decision making (Middle)

No. robots	Short cache delay						Long cache delay					
	1st		2nd		3rd		1st		2nd		3rd	
10	N	149	R	148	SW	139	F	59	R	58	SW	57
14	SW	222	R	218	F	210	F	80	SW	79	R	78
20	R	323	SW	323	F	304	R	116	F	115	SW	115
24	F	421	SW	389	R	372	R	141	F	140	SW	139
30	F	507	SW	483	R	476	F	178	R	177	SW	173
Ranking of best structures												
	SW 10 points						F 13 points					
	R 9 points						R 11 points					

Table 9 – The 3 best performances chosen by structure of robots foraging with M-AGU.

M-2011 decision making (Slow)

No. robots	Short cache delay						Long cache delay					
	1st		2nd		3rd		1st		2nd		3rd	
10	SW	140	F	134	R	130	F	64	R	63	SW	62
14	F	202	SW	195	R	185	R	87	F	87	N	86
20	F	290	SW	290	R	277	SW	127	F	127	N	126
24	F	358	R	339	SW	332	SW	155	F	154	N	154
30	R	458	F	458	SW	430	R	194	F	192	SW	192
Ranking of best structures												
	F 13 points						F 11 points					
	SW 9 points						R and SW 8 points					

Table 10 – The 3 best performances chosen by structure of robots foraging with M-2011.

Following the observed pattern, we established a change in the number of connections with which robots could forage while dealing with the environmental changes. A robot should increase its degree such that it will connect with all robots once it detects long delays in caches (due to congestions or by the challenge). But, it should decrease its degree to two connections when it detects short delays (i.e., when caches are faster to score a successful transport). Note that robots need a mechanism that allows them to detect environmental changes to adjust their connections. Our strategy, called Gama (G), allows each robot to detect changes and adjust its degree to deal with the detected environmental conditions.

Here, our mechanism to detect environmental conditions described in Section 4.2.2 enters to help robots, the filter described in Eq. (18). This filter helps robots to supervise the evolution of the $PnP(t)$ function described in Eq. (15) after each event (by itself or listening to experiences of its neighbors). Figure 33 shows the evolution of robot degree and some detection points. Each detection point means that the filter output trespasses the β threshold. This indicates to the robot that it should switch from few connections to more connections or back. The degree values were normalized according to the team size. Long cache delays are considered as 100% while short delays are over 0%. We do not plot the changes in decisions between partitioning or not because they depend on the decision-making model.

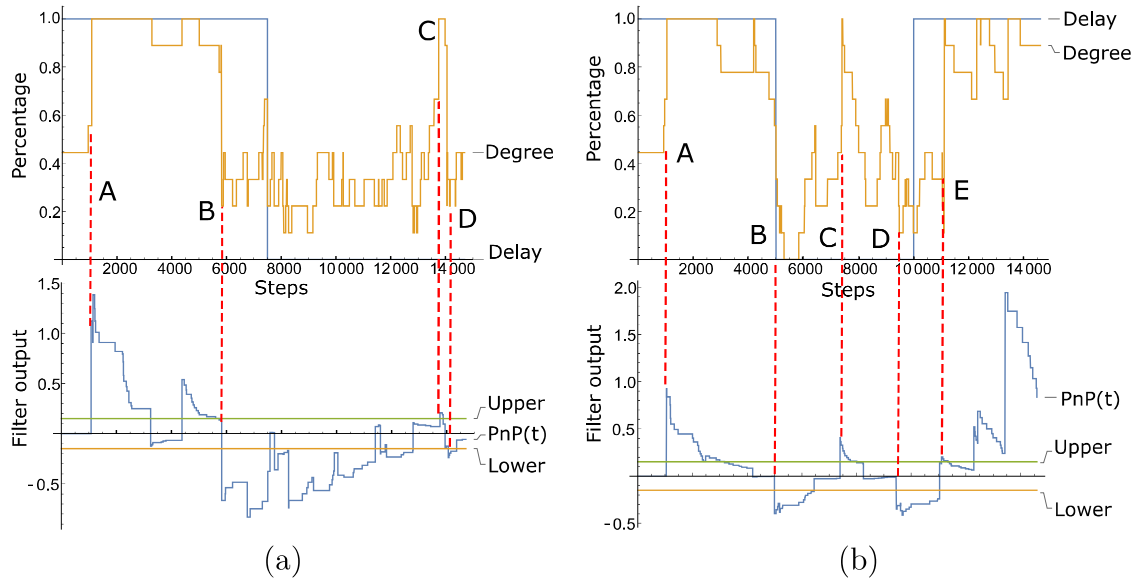


Figure 33 – Evolution of $PnP(t)$ and the respective changes in degree in a certain robot, as well as its filter output. The left plot (a) shows the evolution for an abrupt challenge with step-down change in delays. The right plot (b) shows the evolution for a reoccurring challenge going from long to short, and back to long delays.

We first analyzed Figure 33-a. In moment A, the robot detected an increase in $PnP(t)$ due to a long cache delay. Then, the robot increased its degree to share this information with all robots and also to be able to listen to all. In moment B, since the robot reduced its degree, we can infer that most of the robots should learn that taking the alternate path is the best option. Then, when all robots try to travel between regions, they would increase the congestions to reach the entrance of the path. From moment B to C, there is no change detected, but the degree oscillates because other robots are adapting their connections. In moment C, the robot increased its degree, because all robots could note that using the caches is the best option. Since there are fewer caches than robots, the time to find an available cache could increase, resembling that caches have a long delay. However, note in moment D that the filter also helps the robot to detect this false report because it reduces the number of connections again.

Next, for Figure 33-b, in moment A, the robot detected a change in $PnP(t)$, indicating long cache delays. Then, the robot increased the number of connections. In moment B, the robot detected another change, this time. The robot decreased its connections because the event indicates that caches are faster than the path. In moment C, the robot fails to detect noise, but some of its neighbors do not. Note that its degree decreases despite no change is detected by the robot. In moment D, the robot decreased its degree to readjust its connections after a false detection. Sometimes its own experiences help the robot to discover a false detection. Sometimes the neighboring robots are the ones that help a neighbor to discover false reports. Finally, in moment E, the robot detects a change in the environment again and increases its degree.

When robots increase their degree, they are exposing themselves to excessive information, because they could share information to influence all other robots but they also can listen to others robots. However, if the environmental conditions demand the avoidance of excessive information, robots could decrease its degree. This helps them to adapt their speed of learning and dealing with new environmental conditions.

Since we added the mechanism for adjusting connections as another structure candidate (an adaptive one), the Borda count changes. We performed a new series of simulations considering the Gama communication structure. Thus, the Borda count calculates the ranking considering five structures: F, R, SW, N, and G. In this case, the first rank position will get 4 points, the second 3 points, and the third 2 points. To increase the pressure on the competition, the two last positions get zero points. Results in Table 11 show that the dynamic structure Gama performs well against all challenges, i.e., it offers a good level of adaptability to deal with the proposed environmental changes. Besides, it seems good to use the Gama structure because a fully connected structure means every robot is connected, which can be expensive and inviable as the team increases. In some occasions, Gama lost by few objects. We tested the filter and the Gama structure in M-2011, although it showed that a fully connected structure is the best for it no matter which challenge it deals with (see Table 10).

Table 11 – Values are the rounded average performance of 20 simulations with different amounts of robots in each region for each of the three strategies. Borda count was done for all challenges. A perfect score would be 64 points.

	Greedy						M-AGU						M-2011					
No.	Challenge 1 - Long cache delay																	
robots	1st	Perf	2nd	Perf	3rd	Perf	1st	Perf	2nd	Perf	3rd	Perf	1st	Perf	2nd	Perf	3rd	Perf
10	F	76	R	76	G	75	F	59	R	58	G	57	F	64	R	63	SW	62
14	F	107	G	107	SW	104	G	81	F	80	SW	79	R	87	F	87	G	86
20	F	153	G	151	SW	145	R	116	F	115	SW	115	SW	127	F	127	G	126
24	G	183	F	182	SW	176	R	141	F	140	SW	139	G	155	SW	155	F	154
30	F	229	G	228	SW	221	F	178	R	177	SW	173	R	194	G	193	F	192
	Challenge 2 - Short cache delay																	
10	N	230	SW	214	G	214	G	151	N	149	R	148	SW	140	G	136	F	134
14	N	319	F	308	G	297	SW	222	G	218	R	218	G	207	F	202	SW	195
20	N	454	F	454	R	453	R	323	SW	323	G	311	F	290	SW	290	R	277
24	N	558	R	546	G	546	F	421	SW	389	G	387	G	361	F	358	R	339
30	G	692	N	691	R	693	G	510	F	507	SW	483	R	458	F	458	G	446
	Challenge 3 - Abrupt change from longer delays to short delays																	
10	N	84	F	76	G	75	G	106	R	100	F	99	G	103	R	100	F	98
14	N	120	F	107	G	106	F	138	G	135	SW	134	F	143	SW	139	G	137
20	N	163	G	152	F	152	G	197	F	195	R	190	G	203	F	199	R	193
24	N	210	F	183	G	182	G	244	R	230	SW	229	SW	247	G	242	F	240
30	N	257	G	230	F	228	G	299	SW	293	R	289	G	299	R	294	F	294
	Challenge 4 - Reoccurring change from longer to short and long again																	
10	G	76	F	76	SW	75	SW	85	R	83	F	83	F	92	SW	87	R	87
14	R	108	G	106	F	105	N	124	G	120	R	118	N	120	R	120	SW	118
20	R	159	F	152	G	150	R	176	G	175	F	173	F	179	SW	172	G	170
24	F	184	SW	181	G	181	SW	212	R	207	G	199	SW	206	G	204	F	204
30	F	230	G	226	N	224	R	266	F	261	SW	259	G	261	F	260	R	255
	Ranking of best structures																	
	F	54 points					G and R	48 points					F	55 points				
	G	51 points					F	40 points					G	50 points				

5.3.3.2 Simulations in Webots

Since our Webots license does not run at a good speed (not at least 1X), we only run simulations with abrupt challenges considering step-down and step-up changes. Recall that step-up challenge begins favoring the caches and requires only one adaptation. Whereas the step-down challenge requires twice as many adaptations because it begins with long delays in caches, and robots initialize as if the caches had short delays. In these simulations, robots could go at faster speeds and the range of vision was increased to improve their navigation. We also simulated a perfect knowledge strategy to confirm whether robots having knowledge of the environmental conditions could deliver a better performance as a group. Tables 12, 13, and 14 show the average performance of 15 simulations. The candidates for the Borda count are: a fully connected structure (F), without connections (N), Perfect knowledge (P), and the adaptive structure (G).

Greedy decision making (Fast)

No. robots	Abrupt Challenge - step down								Abrupt Challenge - step up							
	1st		2nd		3rd		4th		1st		2nd		3rd		4th	
8	G	126	N	120	P	116	F	113	N	111	F	110	G	109	P	100
16	G	199	F	180	P	169	N	152	G	178	F	173	N	171	P	167
Ranking of best structures																
	G	10 points							N	6 points						

Table 12 – The average performances by structure of robots foraging with Greedy.

M-AGU decision making (Medium)

No. robots	Abrupt Challenge - step down								Abrupt Challenge - step up							
	1st		2nd		3rd		4th		1st		2nd		3rd		4th	
8	G	119	F	118	N	117	P	116	F	116	G	115	N	112	P	100
16	G	182	F	178	N	172	P	169	N	183	G	172	P	167	F	166
Ranking of best structures																
	G	10 points							F	7 points						

Table 13 – The average performances by structure of robots foraging with M-AGU.

M-2011 decision making (Slow)

No. robots	Abrupt Challenge - step down								Abrupt Challenge - step up							
	1st		2nd		3rd		4th		1st		2nd		3rd		4th	
8	N	118	P	116	G	114	F	113	F	108	G	106	N	105	P	100
16	N	180	G	179	P	169	F	163	N	175	F	169	P	169	G	168
Ranking of best structures																
	N	10 points							G and F	5 points						

Table 14 – The average performances by structure of robots foraging with M-2011.

These simulation results confirm those of Netlogo. Our proposal is reaching a good position by its performance. We can confirm that the adaptability does increase when robots have the possibility to change their connections, which was embedded in the decision-making models. Results in Webots showed that the adaptive strategies could find a better solution than robots having perfect knowledge of the environmental conditions. As we inferred from the evolution of robot decisions in Figures 30 and 32, robots made massive similar decisions increasing queues and conflicts that deteriorated the team performance. In particular, the other strategies allowed robots to regulate the speed of information-sharing according to environmental conditions that each of them detects. As a consequence, they could reach better performances in dynamical environments.

Chapter 6

Conclusions and future work

The major objective of this work was to investigate how robots can adapt their individual behaviors when dealing with dynamical environments through their communication structures. In other words, the focus of our approach was to improve team adaptability by providing suitable communication structures. In some scenarios, it was necessary to provide them an adaptive communication structure. We validated this approach in two foraging tasks performed by teams of robots. Different team sizes defied the scalability of the solutions. Its effectiveness was evaluated in the improvements of the team performance. Since we worked on foraging tasks, their performance was measured as the number of collected objects within a fixed interval of time.

We proposed several specific objectives in the introduction of this text, which helped us to achieve the major goal. They are revised here and the obtained results are also highlighted:

- To identify and implement some models of foraging robots where communication was relevant to the fulfillment of robot tasks

After a broad and deep search in the literature, we identified and implemented two foraging models. The first model worked on the task allocation paradigm, which had a greater dependence on communication. Previous works inspired the proposed model where authors consider different regions to distribute their robots (BONABEAU et al., 1997; KRAUSE, GUESTRIN,). The proposed solution was reached through the adaptation of a multi-agent strategy found in (NOGALES, FINKE, 2013), which required some mathematical extensions. This model worked with e-pucks working on Webots simulator. The second model was designed over the task partitioning paradigm. The work that inspired this model began in (PINI et al., 2011b) and was extended in (PINI et al., 2013). We could reproduce their task in Webots. However, due to the restrictions of our license, we had to replicate it in Netlogo, which delivered faster and simple exploratory experiments. Once we pinpointed some promising patterns, we returned to the small reality gap offered by Webots for e-puck robots.

❏ To select different communication structures based on the topologies investigated in complex networks

We did an exploratory search about sensor and robot networks that considered the communication structure based on concepts of complex networks. From those works, we spotted some common topologies. In particular, in the task allocation model, we opted for three structures: a fully connected, regular of degree 3, and a regular of degree 2 (also known as a ring topology). This model also considered communication with the environmental aids. We let TAMs working as nests to share information with robots that entered to drop objects. Next, for the task partitioning model, we selected four structures: fully connected, ring topology, small-world, and without connections. Unlike the task allocation model, the task partitioning one had a lower dependence on communication. Indeed, the first version of the base model found in (PINI et al., 2011b) does not consider communication. However, in (PINI et al., 2013), they introduced communication by allowing a continuous broadcasting of the last experiences a robot had. They arrived to inconclusive results: sometimes communication helps, others not.

❏ To evaluate the performance of the team of robots in the implemented foraging tasks while they share information over the selected communication structures (static topologies)

We proposed a set of messages to be shared between robots, and between robots and the environmental aids. Then, we could begin to explore the effects of the communication over the team performance. We ran some simulations for the task allocation model and the four strategies: deterministic, semi-stochastic with and without environmental aids, and probabilistic. They were evaluated over three static network structures: fully connected, regular of degree 3, and a ring topology. In the task partitioning model, we worked with seven strategies: always partitioning, never partitioning, one based on (PINI et al., 2011b) (M-2011), other based on (PINI et al., 2013) (M-2013), a greedy one, our first improvement published in (NOGALES, ESCARPINATI; OLIVEIRA, 2017) (M-SGU), and its adaptive version (M-AGU). They were evaluated using four static network structures: fully connected, ring topology, small-world, and no-connected.

❏ To identify a pattern between structural parameters of the communication network and the performance of the team of robots in the dynamic environments

Results from the exploratory simulations revealed which structural parameters were fundamental for the team of robots to complete their tasks. We compared the performance while we kept the same environmental conditions for all strategies in both models. In the task allocation, results of all strategies showed that both connections were important for the team performance, that is, robot-robot and robot-environment connections. However, robot-environment ones were more relevant to achieve a good team performance. The robot-robot connections could be reduced to a small number, but leaving enough connections for robots to have a path to share information with other members. In the task partitioning simulations, the interpretation of the results was harder. Recall that only robot-robot connections were available in this model. This time, the kind of network that worked better depended on the speed of the strategy to process new information. In other words, some strategies are so fast that they need a low degree (ring or small-world structures) to regulate learning speeds. Other strategies that are slower could need a high number of connections (a fully connected structure). This conclusion appeared after we compared the performance of each strategy over the different network structures. We confirmed that there is no optimal (static) structure, as previous researchers suggested (SARKER, DAHL, 2011; PITONAKOVA, CROWDER; BULLOCK, 2016a).

- ❏ To discover which environmental events require robots to adjust their connections while they execute the implemented foraging tasks in dynamic environments

This was the most complex part of the research because it required the exploration of several environmental conditions under the different strategies and communication structures. In other words, this was the most time-consuming stage of our research. The task allocation model was easier because we already had a deterministic model that guided us toward the optimal solution. We explored the effects of traveling across the three regions on the performance. We also evaluated how changing connections could help them to improve their performance. In particular, robots should work with intermittent connections. Once they decided to travel, they disabled their connections with both robots and TAMs in their initial region. At their arrival, they should connect with some of the robots and TAMs of that region to get information from this region. On the other hand, in the task partitioning model, it was harder to identify the key events. Unlike the task allocation model, it was expected the swarm behavior would emerge in this model. Thus, reaching the optimal solution would be difficult (if not an impossible) goal. This makes the task partitioning environment broader to explore. It is important to mention that we could not control the individual movements. We only provided them some behaviors they needed to adapt according to the environmental conditions. Finally, we found the key events and proposed a set of challenges by modifying an environmental variable that we could control - the delay in the transference of objects.

- ❏ To elaborate local mechanisms of perception for the robots be able to identify the events that demand adjustments in their communication structure

As previously mentioned, the task allocation model allowed an easy detection of the events. It also provided a way for robots to supervise the environmental conditions. This was, in part, due to the environmental aids that delivered a regional synchronization and more accurate information. For this model, when a robot travels, it affects two regions, the one it left and that where it arrives. However, *Robot-Robot* and *Robot-TAM leaving/arriving at a region* messages helped to overcome the event that changes the environmental conditions. The environmental aids and robots considered this information and adjusted their utilities and connections. In the task partitioning model, it was necessary to borrow a technique for change detection from data streaming, which is a high-pass filter. Thus, robots could perceive the changes in the ratio of the costs of both alternatives for object transportation. According to the previous results, we concluded that it was better for a robot to increase its connections when it detected an costly time for the transference. Otherwise, when the transference was faster than traveling through the alternative path, it was better to reduce its connections. Notwithstanding, this conclusion did not appear in all of the explored strategies. For these simulations, we continued with the more promising strategies: M-AGU, Greedy, and M-2011.

- ❏ To improve the adaptability of the team of robots while foraging in the implemented environments by embedding the mechanism of (local) structural adaptations

This final step of our research allowed us to show that robots that locally adjusting their communication structure could adapt their behaviors to improve the team performance. In the task allocation model, robots recovered their autonomy with a smaller dependence on the environmental aids. Recall that environmental aids could offer information of the actual environmental conditions with more accuracy. However, unlike the deterministic strategy, they could decide whether to travel or remain in a region by themselves. In the task partitioning model, the adaptive structure emerging from the Gama strategy achieved good results. In this scenario, we tested a variation where robots had perfect knowledge of the environmental conditions. Such variation did not deliver the best performance. All robots were massively looking for the optimal alternative (path or interfaces) at the same time. Therefore, the adaptive Gama structure allowed robots to have inaccurate and diverse information that brought benefits for the team. Robots could adapt to and choose those options that seemed bad but yielded good team performance. This structure also showed to be useful for the decision-making strategy that did not exhibit the pattern of changes in the communication structure. Finally, when compared to the static structures, the Gama adaptive structure offered more transported objects, i.e., it helped robots to deal better with environmental challenges.

To summarize, we explored different structures for information-sharing such that robots could achieve a good performance while keeping a low cost in communications. In some cases, we observed that a fixed and predefined structure could offer the benefits and help robots to overcome environmental challenges. In other cases, an adaptive structure is the best solution to help robots to learn and overcome the environmental changes. This would depend on the way the designer tries to tackle the problem.

If we analyze these findings in terms of costs, in the task allocation scenario, robots could work with few connections between them but depended on the environmental aids. Thus, the optimal number of connections generates a low cost in the communication structure. The implemented environmental aids are a one-time investment and do not increase the costs with the team size. For the task partitioning scenario, since the best solution requires continuous adjustments of the structure according to what each robot considers appropriated to the detected environmental condition, the costs in hardware and energy would depend on the environmental conditions.

Major contributions

This work fulfilled its objective of investigation. It shows how to improve team adaptability by locally modifying the communication structure in foraging scenarios. Besides, it also yielded some contributions that brought improvements over the evaluated models:

- A navigation strategy inspired by white shark hunting strategies. It is based on processing and classifying images of the environment to find visual clues the robot is looking for. This strategy was applied to both scenarios: task allocation and task partitioning.

- Two controllers for the robot behaviors based on hierarchical state machines, one for each scenario. Such controllers enabled us to achieve more complex activities than those found in the base models. In particular, we dealt with robots and not theoretical agents as in (NOGALES, FINKE, 2013). On the other hand, we overcome the limitations of the model in (PINI et al., 2011b) by adding more transitions in the proposed controller.

- Two decision-making strategies for robots working on the task allocation model, semi-stochastic and probabilistic. Unlike them, the deterministic one is an implementation of the model published in (NOGALES, FINKE, 2013) for a multi-agent system. We had to extend those mathematical conditions such that robots could have autonomy to move between regions.

- Two decision-making strategies for robots working on the task partitioning model, named M-SGU and M-AGU. M-SGU is an improvement upon the base model found in (PINI et al., 2011b). It considers a new function for the *give up* options and *abandon* transitions, which makes robots more flexible while foraging. M-AGU is an adaptive version of M-SGU, where robots can make online adaptations of the *give up* function. We allowed each robot to compare its estimations in both options, and to check how is it going its current experience against the expectations it has.

- The Gama mechanism for structural adaptation of the robot communication. This mechanism is able to detect environmental conditions such that each robot could adapt to them by adjusting its connections. The Gama mechanism was inspired by a technique of window size adaptation for online learning in the field of data-streaming (GAMA et al., 2014). It required some adaptations to work on a robotic system. It enables the adjustments on the communication structure of each robot by using a high-pass filter to detect significant environmental changes.

On the other hand, note that the environment is fundamental in helping robots with the navigation. We found that the tendency of home automation in environments (i.e., environments with smart aids) is growing as the costs to obtain such technologies decrease (details in Appendix C). Thus, we opted for implementing two environments that offer the possibility of exploration of strategies in Webots:

- A smart environment for the task allocation model that offers the possibility of foraging. We provided a video in the supplementary material available in (NOGALES, OLIVEIRA, 2017a).

- A smart environment for the task partitioning model that offers the possibility of fulfilling the transport with two alternatives: transference through caches and the path linking both regions. We provided a video in the supplementary material available in (NOGALES, OLIVEIRA, 2017b).

Both environments were also adjusted in Netlogo for the exploratory stages. Notwithstanding, Webots delivered more complexity and realism.

Publications and on-going works

We have publised three papers on international conferences related to multi-agent and robotics systems.

- NOGALES, J. M., ESCARPINATI, M. C., and OLIVEIRA, G. M. B. de. Shark-inpired target approach strategy for foraging with visual clues. In: Proc. of the 18th Conference Towards Autonomous Robotic Systems, pag. 182-198, 2017.

  This paper focuses on the navigation strategy inspired by the hunting strategy of the (white) sharks. Here, we proposed the M-SGU strategy and compared it against the M-2011 strategy (adapted to Webots from (PINI et al., 2011b)), M-2013 (adapted to Webots from (PINI et al., 2013)), and Greedy. However, these results do not explore different communication structures between robots, that is, robot learning process is not regulated.

- NOGALES, J. M. and OLIVEIRA, G. M. B. de. Team distribution between foraging tasks with environmental aids to increase autonomy. In: International Conference on Agents and Artificial Intelligence, pag. 25-36 ScitePress, 2018.

  This paper includes the results we obtained in the task allocation model using the deterministic and semi-stochastic strategies. It does not include the results obtained with the probabilistic strategy. The reviewers recommended us to include a strategy of such kind. We developed the probabilistic strategy and its results are included in this thesis.

- NOGALES, J. M and OLIVEIRA, G. M. B. de. Adaptive give-up decisions for a team of robots foraging with task partitioning. In: IEEE International Conference on Tools with Artificial Intelligence. IEEE, 2018.

  This paper includes the preliminary results of the exploratory simulations, where we achieved the first improvements on the base models. The proposed strategy was named M-AGU and delivered better results by including more options to give up and abandon failed decisions. This new model was compared against 6 more strategies. It brought more adaptability to the team, which was visible on the performance results.

Furthermore, we have been working on one manuscript reporting the major contributions of this thesis:

- NOGALES, J. M., VARGAS, P. A., and OLIVEIRA, G. M. B. de. Introducing plasticity into the communication structure to deal with dynamic environments. Submitted in: Journal of Intelligent and Robotic Systems. [S.l.], 2018.

  This manuscript includes the results in the task partitioning scenario using three of the adaptive strategies we studied here: M-2011, M-AGU, and Greedy. Besides, it is the first work where we discuss the effects of employing different communication structures: fully connected, ring topology, small-world, and no-connection. This manuscript also includes the embedding of the Gama mechanism in the strategies and the benefits of using a dynamic communication structure. We believe this is the major contribution of this thesis.

Future work

Clustering and centrality of betweenness are included in the strategies that helped researchers in sensor networks. Clustering measures the quantity of neighbors of a node are also connected between them. Thus, the higher the clustering, the faster information spreads in that neighborhood of nodes. Centrality of betweenness measures the ratio of the number of shortest paths to the total paths that pass through a node. These are very useful parameters. We propose to explore other structural parameters like these (e.g., clustering) and other well-known networks topologies (e.g., power-law). We believe there is a way to improve the speed required to reach the optimal point without depending on environmental aids. One where robots could keep their entire autonomy, constrained only by the group goal. Maybe, as decentralized auctions, establishing a token, i.e., a way to identify or name a robot as a temporal controller of the region. This robot could keep track of the region information. This could be done using its clustering (or another structural parameter) that delivers information about the influence that robot has in that region. On the other hand, we have seen solutions to handle information in gossip and consensus algorithms, where certain nodes gather an amount of information and then they shared a condensed version of it (RAMOS et al., 2014). Structures like the power-law have a few nodes named hubs, which could work as temporal controllers. Thus, combining the structures with other structural parameters, robots could get information about the environmental conditions with more accuracy.

Future investigations could also involve real applications of the foraging tasks we investigated here. For instance, the foraging with task allocation was guided by a deterministic strategy based on the law of diminishing returns, a common concept used in economics. Such phenomenon occurs in many other models, because their workers are commonly sharing limited resources. In particular, in those models where an additional worker improves the performance, but each time in smaller amounts. An interesting scenario to employ the proposed solution is agriculture. Suppose a land with different crops, whose time for sowing, maintenance, and harvesting require a different number of robots. Besides that, they would need to adjust their distribution according to the season. Therefore, applications of this kind could be modeled as a task allocation scenario taking advantage of the results and strategies we developed here.

Considering the task partitioning model, we propose to explore the advantages of changing connections in other tasks where robots could share information. This idea can be explored in other tasks in which robot learning depends on shared information. Therefore, we believe that some improvements could be achieved once the network adaptability is mingled with the decision-making strategies of those tasks. For instance, a similar task could be defined in aerial navigation, when a team of robots is synchronizing their positions while they reach a location (MERINO et al., 2012). If the environmental conditions change (e.g., a strong wind), they could adjust their communications to overcome such challenge. Another potential application is the surveillance task performed by a team of robots (KHAN et al., 2016). If there is suspicion of an invader, the communication structure could change in order to inform the most of individuals.

Future research could involve a new mechanism to control the way robots adapt their communication structure. This mechanism could aim to make soft changes in the degree. An immediate proposal is to investigate the Gama strategy by employing an adaptive

β parameter in the filter (defined in Eq. 16). It could bring soft changes and better improvements to the strategy. Besides, the function to supervise the ratio of the costs on both ways to transport the objects (Eq. 15) can be extended to more alternatives and more parameters. Furthermore, other local structural parameters than degree (e.g., clustering) can help to improve the adaptability of the team of robots. Future works could also bring improvements by adding pheromone information (TINOCO, LIMA, OLIVEIRA, 2017), that is, mixing explicit and implicit communication.

Finally, we are currently working on the implementation of the models with real robots. Even providing a smaller reality gap, going from Webots to real e-pucks is costly. Therefore, we had to let both models be entirely implemented in future works. We have already done some preliminary exploration in the real e-pucks, but we found some obstacles: the light conditions, the camera resolution, the time for remote control took too long for the implemented code, among others. However, we implemented a limited version of the TAMs and could enter robots into them through the remote control option. TAM circuit details are in Appendix C and a video of the e-puck entering a TAM is available in (NOGALES, OLIVEIRA, 2017a), file name is real-TAM.mp4.

Bibliography

ABDALLAH, S., LESSER, V. Learning the task allocation game. In: *Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. New York, NY, USA: ACM, 2006. (AAMAS '06), p. 850–857. ISBN 1-59593-303-4. DOI: <http://doi.acm.org/10.1145/1160633.1160786>.

AHUJA, R. K. et al. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, Elsevier, v. 123, n. 1, p. 75–102, 2002. DOI: <https://dx.doi.org/10.1016/S0166-218X(01)00338-9>.

AKBARIMAJD, A., SIMZAN, G. Application of artificial capital market in task allocation in multi-robot foraging. *International Journal of Computational Intelligence Systems*, v. 7, n. 3, p. 401–417, 2014. DOI: <http://dx.doi.org/10.1080/18756891.2014.922814>.

ALBERT, R., JEONG, H., BARABASI, A.-L. The diameter of the world wide web. *Nature*, v. 401, p. 130–131, 1999. DOI: <https://doi.org/10.1038/43601>.

ANDERSON, C., RATNIEKS, F. Task partitioning in insect societies. i. effect of colony size on queueing delay and colony ergonomic efficiency. *The American Naturalist*, JSTOR, v. 154, n. 5, p. 521–535, 1999. DOI: <http://dx.doi.org/10.1086/303255>.

ANTOUN, A. et al. Kilogrid: A modular virtualization environment for the kilobot robot. In: IEEE. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Daejeon, South Korea, 2016. p. 3809–3814. DOI: <http://dx.doi.org/10.1109/IROS.2016.7759560>.

BALCH, T., ARKIN, R. C. Communication in reactive multiagent robotic systems. *Autonomous robots*, Springer, v. 1, n. 1, p. 27–52, 1994. DOI: <http://dx.doi.org/10.1007/BF00735341>.

BALCH, T., HYBINETTE, M. Social potentials for scalable multi-robot formations. In: IEEE. *Proc. of the IEEE International Conference on Robotics and Automation*. San

Francisco, CA, USA, 2000. v. 1, p. 73–80. DOI: <http://dx.doi.org/10.1109/ROBOT.2000.844042>.

BAYINDIR, L. A review of swarm robotics tasks. *Neurocomputing*, v. 172, p. 292 – 321, 2015. ISSN 0925-2312. DOI: <http://dx.doi.org/10.1016/j.neucom.2015.05.116>.

BERTSEKAS, D. *Nonlinear programming.* [S.l.]: Athena scientific Belmont, 1999. ISBN 1886529000.

BERTSEKAS, D. P. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, INFORMS, v. 20, n. 4, p. 133–149, 1990. ISSN 00922102, 1526551X. DOI: <http://dx.doi.org/10.1287/inte.20.4.133>.

BOBADILLA, L. et al. Controlling wild mobile robots using virtual gates and discrete transitions. In: IEEE. *American Control Conference*. 2012. p. 743–749. DOI: <http://dx.doi.org/10.1109/ACC.2012.6315569>.

BONABEAU, E. et al. Adaptive task allocation inspired by a model of division of labor in social insects. In: *Proc. Biocomputing and emergent computation.* [s.n.], 1997. p. 36–45. DOI: <http://dx.doi.org/10.1.1.25.8016>.

BRAMBILLA, M. et al. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, v. 7, n. 1, p. 1–41, 2013. DOI: <http://dx.doi.org/10.1007/s11721-012-0075-2>.

BRUTSCHY, A. et al. The TAM: abstracting complex tasks in swarm robotics research. *Swarm Intelligence*, Springer, v. 9, n. 1, p. 1–22, 2015. DOI: <http://dx.doi.org/10.1007/s11721-014-0102-6>.

BRUTSCHY, A. et al. The IRIDIA TAM: a device for task abstraction for the e-puck robot. *TR/IRIDIA/2010-015, IRIDIA*, Université Libre de Bruxelles, Belgium, v. 1, n. 15, 2010.

BRUTSCHY, A. et al. Self-organized task allocation to sequentially interdependent tasks in swarm robotics. *Autonomous Agents and Multi-Agent Systems*, v. 28, n. 1, p. 101–125, Jan 2014. ISSN 1573-7454. DOI: <https://dx.doi.org/10.1007/s10458-012-9212-y>.

BUCHANAN, E., POMFRET, A., TIMMIS, J. Dynamic task partitioning for foraging robot swarms. *Proc. of the 10th International Conference on Swarm Intelligence*, p. 113–124, 2016. DOI: <http://dx.doi.org/10.1007/978-3-319-44427-7_10>.

CAMPO, A. et al. Artificial pheromone for path selection by a foraging swarm of robots. *Biological cybernetics*, Springer, v. 103, n. 5, p. 339–352, 2010. DOI: <http://dx.doi.org/10.1007/s00422-010-0402-x>.

CAO, Y. U., FUKUNAGA, A. S., KAHNG, A. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, v. 4, n. 1, p. 7–27, Mar 1997. ISSN 1573-7527. DOI: <https://dx.doi.org/10.1023/A:1008855018923>.

CASTELLO, E. et al. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence*, Springer, v. 10, n. 1, p. 1–31, 2016. DOI: <http://dx.doi.org/10.1007/s11721-015-0117-7>.

CHAIMOWICZ, L., CAMPOS, M. F. M., KUMAR, V. Dynamic role assignment for cooperative robots. In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 2002. v. 1, p. 293–298 vol.1. DOI: <http://dx.doi.org/10.1109/ROBOT.2002.1013376>.

COSTA, E. et al. Método de alocação de tarefas para transporte cooperativo. In: *Anais do X Simpósio Brasileiro de Automação Inteligente*. [S.l.: s.n.], 2011.

COUCEIRO, M. S. et al. Darwinian swarm exploration under communication constraints: initial deployment and fault-tolerance assessment. *Robotics and Autonomous Systems*, v. 62, n. 4, p. 528–544, 2014. DOI: <http://dx.doi.org/10.1016/j.robot.2013.12.009>.

COUCEIRO, M. S., VARGAS, P. A., ROCHA, R. P. Bridging the reality gap between the webots simulator and e-puck robots. *Robotics and Autonomous Systems*, Elsevier, v. 62, n. 10, p. 1549–1567, 2014. DOI: <http://dx.doi.org/10.1016/j.robot.2014.05.007>.

DASGUPTA, P. *Multi-Robot Task Allocation for Performing Cooperative Foraging Tasks in an Initially Unknown Environment*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. 5–20 p. ISBN 978-3-642-17764-4. DOI: <https://doi.org/10.1007/978-3-642-17764-4_2>.

DENEUBOURG, J.-L. et al. The dynamics of collective sorting robot-like ants and ant-like robots. In: *Proc. of the 1st International Conference on Simulations of Adaptive Behaviors: From animals to animats*. [s.n.], 1991. p. 356–363. DOI: <http://dx.doi.org/116517/116557>.

DROGOUL, A., FERBER, J. From tom thumb to the dockers: some experiments with foraging robots. In: *Proc. of the 2nd International Conference on From animals to animats*. [s.n.], 1992. v. 2, p. 451–459. DOI: <http://dx.doi.org/10.1.1.56.6647>.

DROGOUL, A., FERBER, J. Some experiments with foraging robots. In: MIT PRESS. *Proc. of the 2nd International Conference on Simulations of Adaptive Behaviors: From animals to animats*. 1993. v. 2, p. 451. DOI: <http://dx.doi.org/10.1.1.56.6647>.

EMERSON, P. The original borda count and partial voting. *Social Choice and Welfare*, Springer, p. 1–6, 2013.

EPFL. *E-puck camera*. 2010. DOI: <http://www.e-puck.org/index.php?option=com_content&view=article&id=5&Itemid=14>.

FÄRE, R. *Laws of diminishing returns*. Springer Verlag, 1980. DOI: <http://dx.doi.org/10.1007/978-3-642-48314-1>.

FERRANTE, E. et al. Evolution of task partitioning in swarm robotics. In: *Proc. of the Workshop on Collective Behavior and Social Dynamics of the European Conference on Artificial Life*. [S.l.: s.n.], 2013.

FONG, T., NOURBAKHSH, I., DAUTENHAHN, K. A survey of socially interactive robots. *Robotics and autonomous systems*, Elsevier, v. 42, n. 3, p. 143–166, 2003. DOI: <http://dx.doi.org/10.1016/S0921-8890(02)00372-X>.

FOWLER, H. G., ROBINSON, S. W. Foraging by atta sexdens (formicidae: Attini): seasonal patterns, caste and efficiency. *Ecological Entomology*, Wiley Online Library, v. 4, n. 3, p. 239–247, 1979. DOI: <http://dx.doi.org/10.1111/j.1365-2311.1979.tb00581.x>.

GAMA, J. et al. A survey on concept drift adaptation. *ACM Computing Surveys*, ACM, v. 46, n. 4, p. 1–12, 2014. DOI: <http://dx.doi.org/10.1145/2523813>.

GARNIER, S. et al. Alice in pheromone land: An experimental setup for the study of ant-like robots. In: IEEE. *Proc. of the IEEE Symposium on Swarm Intelligence*. 2007. p. 37–44. DOI: <http://dx.doi.org/10.1109/SIS.2007.368024>.

GERKEY, B. P., MATARIć, M. J. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, v. 23, n. 9, p. 939–954, 2004. DOI: <http://dx.doi.org/10.1177/0278364904045564>.

HALL, M. et al. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009.

HOFF, N., WOOD, R., NAGPAL, R. Distributed colony-level algorithm switching for robot swarm foraging. In: *Distributed Autonomous Robotic Systems*. Springer, 2013. p. 417–430. DOI: <http://dx.doi.org/10.1007/978-3-642-32723-0_30>.

HOLLAND, O., MELHUISH, C. Stigmergy, self-organization and sorting in collective robotics. *Artificial life*, v. 5, n. 2, p. 173–202, 2000. DOI: <https://dx.doi.org/10.1162/106454699568737>.

HSU-SHAN, S. et al. Mobile robots based intelligent fire detection and escaping system. *Journal of Computers*, v. 8, n. 5, p. 1298–1302, 2013. DOI: <http://dx.doi.org/10.4304/jcp.8.5.1298-1302>.

HUNTSBERGER, T. et al. Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, v. 33, n. 5, p. 550–559, Sept 2003. ISSN 1083-4427. DOI: <http://dx.doi.org/10.1109/TSMCA.2003.817398>.

KHAN, M. U. et al. Cps oriented control design for networked surveillance robots with multiple physical constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 35, n. 5, p. 778–791, May 2016. ISSN 0278-0070. DOI: <http://dx.doi.org/10.1109/TCAD.2016.2524653>.

KOENIG, N., HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. 2004. v. 3, p. 2149–2154. DOI: <http://dx.doi.org/10.1109/IROS.2004.1389727>.

KRAUSE, A., GUESTRIN, C. Near-optimal observation selection using submodular functions. In: *Proc. of the 22nd Conference on Artificial Intelligence*. [s.n.]. v. 7, p. 1650–1654. DOI: <http://dx.doi.org/10.1.1.64.2463>.

KUBE, C., BONABEAU, E. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, v. 30, n. 1, p. 85 – 101, 2000. ISSN 0921-8890. DOI: <http://dx.doi.org/10.1016/S0921-8890(99)00066-4>.

KURAZUME, R., HIROSE, S. An experimental study of a cooperative positioning system. *Autonomous Robots*, Springer, v. 8, n. 1, p. 43–52, 2000. DOI: <http://dx.doi.org/10.1023/A:1008988801987>.

LERMAN, K., GALSTYAN, A. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, v. 13, n. 2, p. 127–141, Sep 2002. ISSN 1573-7527. DOI: <https://dx.doi.org/10.1023/A:1019633424543>.

LIU, Y., NEJAT, G. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, v. 72, n. 2, p. 147–165, Nov 2013. ISSN 1573-0409. DOI: <http://dx.doi.org/10.1007/s10846-013-9822-x>.

LUO, L., CHAKRABORTY, N., SYCARA, K. Distributed algorithm design for multi-robot task assignment with deadlines for tasks. In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 2013. p. 3007–3013. ISSN 1050-4729. DOI: <http://dx.doi.org/10.1109/ICRA.2013.6630994>.

MARCO, R. D., FARINA, W. Changes in food source profitability affect the trophallactic and dance behavior of forager honeybees (apis mellifera). *Behavioral Ecology and Sociobiology*, Springer, v. 50, n. 5, p. 441–449, 2001. DOI: <http://dx.doi.org/10.1007/s002650100382>.

MARDEN, J. R., ARSLAN, G., SHAMMA, J. S. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 39, n. 6, p. 1393–1407, Dec 2009. ISSN 1083-4419. DOI: <http://dx.doi.org/10.1109/TSMCB.2009.2017273>.

MARTIN, J. A. et al. Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature. *Natural Computing*, Springer, v. 8, n. 4, p. 757, 2009. DOI: <http://dx.doi.org/10.1007/s11047-008-9096-6>.

MATARIC, M. J. Reward functions for accelerated learning. In: *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994. p. 181–189. DOI: <http://dx.doi.org/10.1016/B978-1-55860-335-6.50030-1>.

MERINO, L. et al. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent & Robotic Systems*, v. 65, n. 1, p. 533–548, Jan 2012. ISSN 1573-0409. DOI: <https://dx.doi.org/10.1007/s10846-011-9560-x>.

MICHEL, O. Webots: a powerful realistic mobile robots simulator. In: *Proc. of the 2nd International Workshop on RoboCup*. [S.l.: s.n.], 1998.

MILETITCH, R. et al. Information aggregation mechanisms in social odometry. In: *Proc. of the 12th European Conference on Synthesis and Simulation of Living Systems*. [s.n.], 2013. p. 102–109. DOI: <http://dx.doi.org/10.7551/978-0-262-31708-2-ch016>.

MOHAN, Y., PONNAMBALAM, S. An extensive review of research in swarm robotics. In: IEEE. *Proc. of the World Congress on Nature & Biologically Inspired*. [S.l.], 2009. p. 140–145.

NEWMAN, M. E. J. The structure and function of complex networks. *SIAM Review*, v. 45, n. 2, p. 167–256, 2003. DOI: <http://dx.doi.org/10.1137/S003614450342480>.

NOGALES, J. M., ESCARPINATI, M. C., OLIVEIRA, G. M. Shark-inpired target approach strategy for foraging with visual clues. In: . Proc. of the 18th Conference Towards Autonomous Robotic Systems (TAROS), 2017. p. 182–198. ISBN 978-3-319-64107-2. DOI: <http://dx.doi.org/10.1007/978-3-319-64107-2_15>.

NOGALES, J. M., FINKE, J. Optimal distribution of heterogeneous agents under delays. In: *2013 American Control Conference*. [s.n.], 2013. p. 3212–3217. ISSN 0743-1619. DOI: <http://dx.doi.org/10.1109/ACC.2013.6580326>.

NOGALES, J. M., OLIVEIRA, G. M. *Supplementary material for: Improving adaptability of a multi-robot foraging model with task allocation*. 2017. DOI: <https://sites.google.com/site/juanmnogales/projects/task-allocation>.

NOGALES, J. M., OLIVEIRA, G. M. *Supplementary material for: Improving adaptability of a multi-robot foraging model with task partitioning*. 2017. DOI: <https://sites.google.com/site/juanmnogales/projects/task-partitioning>.

NOGALES, J. M., OLIVEIRA, G. M. Adaptive give-up decisions for a team of robots foraging with task partitioning. *To be published in IEEE International Conference on Tools with Artificial Intelligence*, Manager, Nov. 2018.

NOGALES, J. M., OLIVEIRA, G. M. B. de. Team distribution between foraging tasks with environmental aids to increase autonomy. In: *Proc of the 10th International Conference on Agents and Artificial Intelligence (ICAART)*. SciTePress, 2018. p. 25–36. ISBN 978-989-758-275-2. DOI: <http://dx.doi.org/10.5220/0006588700250036>.

NOGALES, J. M., VARGAS, P. A., OLIVEIRA, G. M. Introducing plasticity into the communication structure to deal with dynamic environments. September 2018. DOI: <http://dx.doi.org/>.

PARKER, L. E. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE transactions on robotics and automation*, IEEE, v. 14, n. 2, p. 220–240, 1998. DOI: <http://dx.doi.org/10.1109/70.681242>.

PERILLO, M. A., HEINZELMAN, W. B. *Wireless sensor networks protocols*. [S.l.]: University of Rochester, 2005.

PESSIN, G. et al. Intelligent control and evolutionary strategies applied to multirobotic systems. In: *Proc. of the IEEE International Conference on Industrial Technology*. [s.n.], 2010. p. 1427–1432. DOI: <http://dx.doi.org/10.1109/ICIT.2010.5472498>.

PHAN, C., LIU, H. H. T. A cooperative uav/ugv platform for wildfire detection and fighting. In: *Proc. of the 7th International Asia Simulation Conference on System Simulation and Scientific Computing*. [s.n.], 2008. p. 494–498. DOI: <http://dx.doi.org/10.1109/ASC-ICSC.2008.4675411>.

PINCIROLI, C. et al. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, Springer, v. 6, n. 4, p. 271–295, 2012. DOI: <http://dx.doi.org/10.1007/s11721-012-0072-5>.

PINI, G. et al. *Supplementary material for: Task partitioning in swarms of robots: An adaptive method for strategy selection*. 2011. DOI: <http://iridia.ulb.ac.be/supp/IridiaSupp2011-003/index.html>.

PINI, G. et al. Task partitioning in swarms of robots: An adaptive method for strategy selection. *Swarm Intelligence*, Springer, v. 5, n. 3-4, p. 283–304, 2011. DOI: <http://dx.doi.org/10.1007/s11721-011-0060-1>.

PINI, G. et al. Task partitioning in a robot swarm: Object retrieval as a sequence of subtasks with direct object transfer. *Artificial life*, MIT Press, v. 20, n. 3, p. 291–317, 2014. DOI: <http://dx.doi.org/10.1162/ARTL_a_00132>.

PINI, G. et al. Task partitioning in a robot swarm: a study on the effect of communication. *Swarm Intelligence*, Springer, v. 7, n. 2-3, p. 173–199, 2013. DOI: <http://dx.doi.org/10.1007/s11721-013-0078-7>.

PITONAKOVA, L., CROWDER, R., BULLOCK, S. Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, Springer, v. 10, n. 1, p. 33–63, 2016. DOI: <http://dx.doi.org/10.1007/s11721-016-0118-1>.

PITONAKOVA, L., CROWDER, R., BULLOCK, S. Task allocation in foraging robot swarms: The role of information sharing. *Proceedings of the Artificial Life Conference*, p. 306, July 2016. DOI: <http://dx.doi.org/10.7551/978-0-262-33936-0-ch053>.

RAMAN, V., KRESS-GAZIT, H. Explaining impossible high-level robot behaviors. *IEEE Transactions on Robotics*, IEEE, v. 29, n. 1, p. 94–104, 2013. DOI: <http://dx.doi.org/10.1109/TRO.2012.2214558>.

RAMOS, H. S. et al. Topology-related metrics and applications for the design and operation of wireless sensor networks. *ACM Transactions on Sensor Networks*, v. 10, n. 3, p. art. 53, 2014. DOI: <http://dx.doi.org/10.1145/2512328>.

RANJBAR-SAHRAEI, B., WEISS, G., NAKISAEE, A. A multi-robot coverage approach based on stigmergic communication. In: SPRINGER. *German Conference on Multiagent System Technologies*. Germany, 2012. p. 126–138. Disponível em: <http://dx.doi.org/10.1007/978-3-642-33690-4_13>.

RATNIEKS, F., ANDERSON, C. Task partitioning in insect societies. *Insectes sociaux*, Springer, v. 46, n. 2, p. 95–108, 1999.

RATY, T. D. Survey on contemporary remote surveillance systems for public safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 40, n. 5, p. 493–515, Sept 2010. ISSN 1094-6977. DOI: <http://dx.doi.org/10.1109/TSMCC.2010.2042446>.

ROHMER, E., SINGH, S. P., FREESE, M. V-rep: A versatile and scalable robot simulation framework. In: IEEE. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. 2013. p. 1321–1326. DOI: <http://dx.doi.org/10.1109/IROS.2013.6696520>.

ŞAHIN, E. Swarm robotics: From sources of inspiration to domains of application. Springer Berlin Heidelberg, Santa Monica, CA, p. 10–20, 2005. DOI: <https://doi.org/10.1007/978-3-540-30552-1_2>.

SANDHLOM, T. W., LESSER, V. R. Coalitions among computationally bounded agents. *Artificial Intelligence*, v. 94, n. 1, p. 99 – 137, 1997. ISSN 0004-3702. Economic Principles of Multi-Agent Systems. DOI: <http://dx.doi.org/10.1016/S0004-3702(97)00030-1>.

SARKER, M. O. F., DAHL, T. *Bio-Inspired Communication for Self-Regulated Multi-Robot Sytems*. INTECH Open Access Publisher, 2011. DOI: <http://dx.doi.org/10.1016/j.robot.2016.12.006>.

SCHMICKL, T., THENIUS, R., CRAILSHEIM, K. Swarm-intelligent foraging in honeybees: benefits and costs of task-partitioning and environmental fluctuations. *Neural Computing and Applications*, Springer, v. 21, n. 2, p. 251–268, 2012. DOI: <http://dx.doi.org/10.1007/s00521-010-0357-9>.

SCHNEIDER, E. et al. Auction-based task allocation for multi-robot teams in dynamic environments. In: SPRINGER. *Conference Towards Autonomous Robotic Systems*. 2015. p. 246–257. DOI: <http://dx.doi.org/10.1007/978-3-319-22416-9_29>.

SICILIANO, B., KHATIB, O. *Handbook of robotics*. [S.l.]: Springer Science & Business Media, 2008. ISBN 978-3-540-30301-5.

SIMONIN, O., CHARPILLET, F., THIERRY, E. *Collective construction of numerical potential fields for the foraging problem*. Tese (Doutorado) — INRIA, 2007. DOI: <http://dx.doi.org/inria-00143302-v2>.

SKOMAL, G. *Sharkopedia: interview to Dr. Gregory Skomal*. 2015. DOI: <http://sharkopedia.discovery.com/shark-topics/feeding-hunting-diet/#great-white-hunting-tactics-an-interview-with-greg-skomal>.

STONE, P., VELOSO, M. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, v. 110, n. 2, p. 241 – 273, 1999. ISSN 0004-3702. DOI: <http://dx.doi.org/10.1016/S0004-3702(99)00025-9>.

SUGAWARA, K., KAZAMA, T., WATANABE, T. Foraging behavior of interacting robots with virtual pheromone. In: IEEE. *Proc. of the IEEE Conference on Intelligent Robots and Systems*. 2004. v. 3, p. 3074–3079. Disponível em: <http://dx.doi.org/10.1109/IROS.2004.1389878>.

SUNG, C., AYANIAN, N., RUS, D. Improving the performance of multi-robot systems by task switching. In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 2013. p. 2999–3006. ISSN 1050-4729. DOI: <http://dx.doi.org/10.1109/ICRA.2013.6630993>.

TAMBE, M. Towards flexible teamwork. *Journal of artificial intelligence research*, v. 7, p. 83–124, 1997. DOI: <http://dx.doi.org/10.1613/jair.433>.

THERAULAZ, G., BONABEAU, E., DENUEBOURG, J. Response threshold reinforcements and division of labour in insect societies. In: *Proc. of the Royal society of London: Biological Sciences*. [s.n.], 1998. v. 265, n. 1393, p. 327–332. DOI: <http://dx.doi.org/10.1098/rspb.1998.0299>.

TINOCO, C. R., LIMA, D. A., OLIVEIRA, G. M. An improved model for swarm robotics in surveillance based on cellular automata and repulsive pheromone with discrete diffusion. *International Journal of Parallel, Emergent and Distributed Systems*, Taylor & Francis, p. 1–25, 2017. DOI: <http://dx.doi.org/10.1007/978-3-319-10401-0_3>.

VIGURIA, A., MAZA, I., OLLERO, A. Set: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets. In: *Proc. of IEEE International Conference on Robotics and Automation*. [s.n.], 2007. p. 3339–3344. ISSN 1050-4729. DOI: <http://dx.doi.org/10.1109/ROBOT.2007.363988>.

VISSCHER, K., SEELEY, T., PASSINO, K. Group decision making in honey bee swarms. *American Scientist*, v. 94, n. 3, p. 220–229, 2006. DOI: <http://dx.doi.org/10.1511/2006.59.993>.

WAWERLA, J., VAUGHAN, R. T. A fast and frugal method for team-task allocation in a multi-robot transportation system. In: IEEE. *Proc. of the IEEE Conference on Robotics and Automation*. 2010. p. 1432–1437. Disponível em: <http://dx.doi.org/10.1109/ROBOT.2010.5509865>.

WEI, C., HINDRIKS, K. V., JONKER, C. M. Performance of communicating cognitive agents in cooperative robot teams. In: *Proc. of the 6th Int. Conf. on Agents and Artificial Intelligence, Angers, France*. [s.n.], 2014. p. 13–31. ISBN 978-3-319-25210-0. DOI: <http://dx.doi.org/10.1007/978-3-319-25210-0_2>.

WILENSKY, U. *Netlogo:*. [S.l.]: Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. Http://ccl.northwestern.edu/netlogo/.

WU, J., TSE, C. K., LAU, F. C. M. Effective routing algorithms based on node usage probability from complex network perspective. In: *IEEE Int. Symposium on Circuits and Systems*. [s.n.], 2014. p. 2209–2212. DOI: <http://dx.doi.org/10.1109/ISCAS.2014.6865608>.

YAN, Z., JOUANDEAU, N., ALI-CHéRIF, A. Multi-robot heuristic goods transportation. In: *6th IEEE International Conference Intelligent Systems*. [s.n.], 2012. p. 409–414. ISSN 1541-1672. DOI: <http://dx.doi.org/10.1109/IS.2012.6335251>.

YIN, Y. et al. An improved search strategy for even degree distribution networks. *Journal of Networks*, v. 8, n. 7, p. 1558–1565, 2013. DOI: <http://dx.doi.org/10.4304/jnw.8.7.1558-1565>.

YOSHIDA, E., ARAI, T. Performance analysis of local communication by cooperating mobile robots. *IEICE Transactions on Communications*, The Institute of Electronics, Information and Communication Engineers, v. 83, n. 5, p. 1048–1059, 2000.

ZAHADAT, P. et al. From honeybees to robots and back: division of labour based on partitioning social inhibition. *Bioinspiration & biomimmics*, IOP Publishing, v. 10, n. 6, p. 66005, 2015. DOI: <http://dx.doi.org/10.1088/1748-3190/10/6/066005>.

Appendix A

Appendix A

A Environments

Over the last decade, technological advances have reduced both costs and sizes of devices allowing robotic research to grow more quickly. In particular, swarm robotics has taken advantage of such advances because its functioning relies upon a large number of simple robots. The designer has to regulate robot-robot and robot-environment interactions in order that the group of robots may accomplish a complex task. Several works focus on robot-robot interactions. There is a lack of works highlighting robot-environment interactions. We provide a brief review of smart environmental solutions that helped swarm of robots to solve complex tasks. This review classifies environmental solutions to foraging, navigation, and searching tasks. We also detail current tendencies in robotic environments.

A.1 Inspiration

As human beings, we possess several sensor inputs and internal memory information that help us to identify the environment and its conditions. In particular, the environment is constantly saturating us with aids that may come from sound, smell, or visual sources. We exploit these aids, whether dynamic or static, through complex processes to navigate, coordinate, and locate ourselves or other things in the environment. For instance, while walking down a street, any human has enough environmental information to locate herself and coordinate her navigation with others. The sign of a grocery store, despite being static, provides us with enough information to know we arrived at our destination. A stoplight, being dynamic information, helps us to avoid accidents, when we behave according to the meaning of each color.

As the reader may note, it is the environment that helps us and we are part of the environment as well. Thus, it happens with robots. Sahin et al. in (ŞAHIN, 2005) proposed one of the most accepted definitions of swarm robotics: *the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment.* Next, it is evident that many studies have pushed forward this field by focusing on control the interactions among agents. However, there is an aspect of this definition, which has not been treated with the importance it deserves: the interactions between the agents and the environment.

The concept of swarm robotics began to take strength at the beginning of the 2000s. It was a novel concept that took its inspiration from insect strategies to solve complex tasks. In particular, such solutions offer a far better alternative through the employment of simpler units. Designing simple robots seems easier than creating a big, expensive, and heavy robot. In (ŞAHIN, 2005), the authors considered pertinent to describe the desirable properties of swarm robotics before the concept became blurred through time. These properties continue till our days:

- **Robustness:** redundancy and decentralization should foster the swarm to continue operating despite failures or disturbances in the environment, although at a lower performance.

- **Flexibility:** requires the swarm to be able to generate modularized solutions to different tasks.

- **Scalability:** considers that the coordination mechanism would be able to deal with a large number of relatively simple robots.

We found several reviews of swarm robotics research; it is interesting that many of them focus on robot-robot interactions (ŞAHIN, 2005), (MOHAN, PONNAMBALAM, 2009), (BRAMBILLA et al., 2013), and (BAYINDIR, 2015); while few are highlighting the environmental conditions in which robots worked (HARPER, 2003) and (ZUEHLKE, 2008). Note that, over the last decade, many technological advances have emerged as solutions to provide smart devices for monitoring (MAO, FIDAN, ANDERSON, 2007), vision (JALOBEANU et al., 2015), and communication (ARNDT, BERNS, 2012). In swarm robotics, since robots should be simple, small, and relatively cheap, the environment plays an important role to make the tasks easier and the programming of robots.

Next, in (MOHAN, PONNAMBALAM, 2009), the authors provided a classification of the previous works considering four aspects: the inspiration (insects, mammals, etc), kind of communication (explicit or implicit), control (centralized or distributed), and the task (mapping, navigation, etc). In (BRAMBILLA et al., 2013), the authors summarized previous reviews and their taxonomies. They also provided their own classification by design (behavior-based, learning, micro, and macroscopic modeling) and by the kind of task. Since the field has grown enough, the most complete and recent review provided a taxonomy that focuses on the problems to solve (BAYINDIR, 2015).

From those reviews of robot-environment interactions, we found smart environments that include smart objects. A smart object is a standard device equipped with a wireless sensor node that provides whether sensing, communication, or computational capabilities. Mostly, these smart devices are working in environments as houses (HARPER, 2003) or industries (ZUEHLKE, 2008). Such devices help to regulate interactions among robots, and between robots and humans. Communication between robots, the environment, and human beings underlies the possibility of these entities to work simultaneously (SANFELIU, HAGITA, SAFFIOTTI, 2008). For instance, a mobile cleaning robot can localize itself easily with the help of an ambient camera-based monitoring system (SAFFIOTTI, BROXVALL, 2005). Robot vision is commonly deficient due to onboard sensors, but with the help of environmental aids, the robot can extend its "vision" (ARNDT, BERNS, 2012). Different localization techniques, especially in WSNs (Wireless Sensor Networks), can be found in (MAO, FIDAN, ANDERSON, 2007).

Although swarm robotics has tackled a lot of tasks; we focus on environments for foraging tasks and its subtasks: navigation, search, and transport of objects. It is important to mention that many of the works in foraging with robots began as mathematical analyses and, then, they came alive with physical robots (e.g., (DENEUBOURG et al., 1991) and (HAYES, MARTINOLI, GOODMAN, 2003)). However, as we mentioned before, solutions of social insects inspired swarm robotics solutions, bio-mimic. Entomologists found that insects can work collectively by sharing information, and, in swarm robotics, communication is indispensable for effective cooperative working (SUGAWARA, KAZAMA, WATANABE, 2004). Foraging is not the exception; robots could share information directly (OCA et al., 2011) or indirectly (MOHAN, PONNAMBALAM, 2009).

Many of these works exploit indirect communication for navigation purposes as (CAMPO et al., 2010), (GARNIER et al., 2007), and (RANJBAR-SAHRAEI, WEISS; NAKISAEE, 2012), just to cite some in which robots leave pheromone clues. Robots working with pheromone drop some (virtual) chemical in the environment, whose rate of evaporation is essential in order that the group may deliver a good performance. When pheromone clues remain for longer periods, the team could stagnate in a single option because robots would have a high chance to reinforce initial clues. Fast rates of evaporation could not bring any benefit to the group. It would be the same as working individually because each robot is despising others' experiences. Some works replicated the same effects of pheromone in guiding other members of the swarm through direct communication (THERAULAZ, BONABEAU; DENUEBOURG, 1998), (BALCH, ARKIN, 1994) and (PITONAKOVA, CROWDER, BULLOCK, 2016). Therefore, as a communication strategy, the pheromone is beneficial for the group.

A.2 Environmental solutions

In the following sections, we organize the solutions by their level of complexity in the implementation and reproduction. Complexity increases as the designer has to invest more money or time programming the environmental aids. Going from simple solutions, because they have low cost and require few or no programming stage, to complex solutions where smart objects with communication capabilities help robots to execute their task. We do not consider within this classification the global aid of a north-star or sun guidance, which is common and present in several works for location (MAYET et al., 2010), (PINI et al., 2011), (PINI et al., 2013), (CHEN, GAUCI, GROß, 2013), and (PITONAKOVA, CROWDER, BULLOCK, 2016).

A.2.1 Simple solutions

Within simple solutions, we found cheap solutions as templates (MELHUISH, WELSBY, EDWARDS, 1999), ground tapes (BOBADILLA et al., 2012), barcodes (ALERS et al., 2013), big objects (CHEN, GAUCI, GROß, 2013), and photo-sensible grounds (MAYET et al., 2010) and (RANJBAR-SAHRAEI et al., 2013). The authors of these and similar works provide crafty solutions that help robots to perform complex tasks. Such aids guide robots without bereaving them of their autonomy.

Inspired by how insects build without having a blueprint schematic, the authors of (MELHUISH, WELSBY, EDWARDS, 1999) tried to regulate a wall construction with robots. The environment included a template that helped robots to find a good place to build a wall. Such template is a combination of light-field and ground lines. The authors employed a white tape in the ground and a bank of halogen lamps to guide robots while they built the wall. Robots include a photo-sensor, an obstacle detector, and a ground color sensor. They did not build a 3D wall, but pushed blocks to form linear patterns of them, as ants do in a two glass anthill. Later, in (STEWART, RUSSELL, 2006), another robot provided the light-gradient information to the builders. The purpose of this robot was to guide other robots by moving the light-gradient information like the environment did in (MELHUISH, WELSBY, EDWARDS, 1999) statically.

In (BOBADILLA et al., 2012), the authors divided the environment into a discrete set of regions linked through gates. Ground colored tapes allowed robots to identify the gates and know the direction they were going. In particular, robots remembered the last gate color to avoid returning to previously visited regions. Moreover, robots could complete complex tasks as patrolling, disentanglement, and navigation with this environmental information and simple sensor feedback. The only sensors required were a color sensor that can detect simple landmarks in the environment and some contact-sensors that detect obstacles and walls. Besides the ground colored tapes, when the room light changed, the behaviors of the robots might change. In the patrolling task, for instance, the robots return to previously visited places when the light changes. They also analyzed the effect of the number of gates through simulations. With a fewer number of gates, the outliers increased, and the expected time to complete the patrolling was slightly higher.

Another interesting solution is to use a combination of far and near distance clues. In (CHEN, GAUCI, GROß, 2013), the authors create a strategy to move objects collectively. The objects were taller than the robots to occlude their visual field. Robots could push the obstacle once they were behind the obstacle being unable to see the target. To be able to transport the object farther, the authors of (CHEN et al., 2015) improved this idea. They introduced a robot working as a beacon guiding the robots through a path of checkpoints, which lead the team to reach the target. In (ALERS et al., 2013), the authors suggested objects with two kinds of information: a color ring on the top of the object, and a barcode on the bottom. In particular, the color ring offered the far distance information providing a coarse guidance. The barcode provides detailed information for navigation orientation. To avoid the barcode image processing, which has a great computational weight, the authors of (NOGALES, ESCARPINATI; OLIVEIRA, 2017) preferred to encode the information by combining colors and shapes.

Next, we found (MAYET et al., 2010) and (RANJBAR-SAHRAEI et al., 2013) emulating pheromone trails, through an idea that came from an artistic project (BLOW, 2005). In particular, both environments required robots equipped with UV-LEDs and dark rooms such that the phosphorescent trails might be visible. The synthetic raisin on the floor includes grains that react to UV-light. They absorb UV-light and re-emit it at a lower intensity for several minutes after the original excitation. This solution presents a completely autonomous way for the robots to lay trails unlike centralized solutions (as the projectors in (SUGAWARA, KAZAMA, WATANABE, 2004)) and does not require any data storage or computational power for the robots to store the ground information (as the gradient-field in (LIMA et al., 2017)). In (MAYET et al., 2010), the authors adapted the phosphorescent glow-paint to mimic ant trails in a foraging task. The authors of (RANJBAR-SAHRAEI et al., 2013) worked on a coverage task, which could function for monitoring and patrolling missions.

A.2.2 Middle-term solutions

Middle solution refers to environmental aids that are cheap and easy to implement. Most of these works do not intend to be real life applications, but laboratory tools that help robots to complete their tasks. Among the creative solutions with which researchers test their algorithms, we found the usage of smart objects (BERG, KARUD, 2011), (BRUTSCHY et al., 2015), and (NOGALES, ESCARPINATI; OLIVEIRA, 2017); augmented reality (SUGAWARA, KAZAMA, WATANABE, 2004), (GARNIER et al., 2007), (GARNIER et al., 2013), (ARVIN et al., 2015), (REINA et al., 2015), and (REINA et al., 2017); and sacrifice of other robots (STEWART, RUSSELL, 2006), (GRIBOVSKIY, MONDADA, 2009), and (CHEN et al., 2015). By smart objects, we refer to a standard item equipped with communication and coordination capabilities. The sacrificed robots do not act directly in the task but orientate other robots.

In (BERG, KARUD, 2011), the authors provide objects with a set of infrared emitters, which allow robots to find them. Once robots detect these emitters, they modify their behavior and begin to push the object. The direction of the transport depends on the available emitters. If a robot does not find an available emitter, it has to search for another to dock itself and participate in the pushing task. Despite these objects do not have inputs, their outputs are enough to influence robot behaviors. The experiments described behaviors similar to those that ants exhibit when moving large objects.

The authors of (BRUTSCHY et al., 2015) proposed a booth-like module that abstract complex tasks through color indications. They called these booths TAMs (Task Abstraction Module), which work like a stoplight. They include sensors to detect the presence of the robot, light-color indicators for their current state, and a wireless communication through ZigBee antenna. Unlike the objects in (BERG, KARUD, 2011), each TAM has inputs and outputs to interact with robots. The designer needs to know the logical relation of the abstracted tasks. For instance, a set of three TAMs could emulate the load of a big truck. One of these TAMs would be indicating that its room for trucks is available. Once a truck robot enters, the light of the other two TAMs would change in order that other robots may know the truck is ready. Two robots could bring their loads, enter the TAMs, and deposit their virtual loads in the truck. When the TAMs finish emulating the load process, they would change their light color or send a message to indicate robots they can depart. Since robots employed in (NOGALES, ESCARPINATI; OLIVEIRA, 2017) could distinguish few colors, the authors adapted the TAMs. They suggested TAMs that combine color and shape to encode the different tasks. Thus, robot capacities increase through the environmental help.

We also found different works employing localization systems with cameras (SCHWAGER et al., 2011), Kinect devices (JALOBEANU et al., 2015), and radio or wireless sensor signals (CARO, DUCATELLE, GAMBARDELLA, 2009), (DORIGO et al., 2013). These aids are commonly employed for navigation tasks. Among them, a practical solution is projecting a virtual trail by including a projector and a camera in the environment. By linking these devices through a computer, the pheromone trail could become an *augmented* reality. The camera tracks the robots, and the projector updates their trails. Allowing communication through virtual pheromone seems an effective way to help robots to navigate collectively. Since the trail keeps information, its rate of evaporation is fundamental for robots to exploit the experiences of other robots. If the evaporation rate is low, the trails could trap the robots into useless searches, while a fast rate of evaporation could erase useful information. Therefore, experiments with pheromone need to adjust their rate of evaporation such that it may bring benefits to the group.

One of the first works to implement virtual trails is (SUGAWARA, KAZAMA, WATANABE, 2004). For their experiment, the communication was indispensable to obtain an effective cooperative team. V-DEAR (Virtual Dynamic Environment for Autonomous Robots) projection mimicked pheromones trails with white traces on the ground. The authors tested this solution with homogeneously scattered objects and localized sources. As time passes, the disappearance of the luminous trails emulates the evaporation of pheromone and removes the useless information. But, the objects do not disappear (i.e., sources were endless). One interesting conclusion of this work is that the distribution of the objects in the environment could affect the optimal rate of evaporation. With localized sources, the robots could work with faster rates of evaporation (less information). But with scattered objects, they require lower rates of evaporation.

In (GARNIER et al., 2007), the authors explained that projection of pheromone trails and augmented reality solutions are cheap and easy to handle as a laboratory tool. However, these solutions are unsuitable to be real life applications. They validated their algorithm with a group of five robots, which could perceive their environment and adapt accordingly. The environment offers two alternative paths linking two different areas, and the luminous trails helped the group to make a collective decision. Their results suggest that if evaporation is too fast, no stable choice can take place. Results also suggest that a quick and stable collective choice requires an optimal number of robots.

In (REINA et al., 2017), the authors worked with Kilobots (RUBENSTEIN, AHLER, NAGPAL, 2012), which are small, slow, and simple robots. The inexpensive nature of the Kilobots limits their range of capabilities because they only possess a single sensor, but they can communicate through an infrared emitter/receiver module. In some studies, this limitation can be a source of motivation and inspiration, while in others it is an impediment. Notwithstanding, they allowed to validate an algorithm for collective decisions in (VALENTINI et al., 2016), and to test searching strategies in (DIMIDOV, ORIOLO, TRIANNI, 2016). The work described in (REINA et al., 2017) extends robot capacities by adding a system for Augmented Reality for Kilobots (ARK).

The ARK system includes a flexible base control software, which allows users to define varying virtual environments within a single experiment using an integrated overhead tracking and control. This system allows Kilobots access to customized information based on their location and state. The environment requires three components:

- A tracking system with an overhead camera that provides real-time data on robot location and state
- A modified overhead emitter which broadcasts infrared signals to communicate to the Kilobots
- A base control station to coordinate the system and simulate the virtual environments

Unlike (SUGAWARA, KAZAMA, WATANABE, 2004), the available sources depleted through time in (REINA et al., 2017). The software obtains images from four cameras covering the Kilobot arena. These environmental components updated 150 Kilobots once every 2.5 s. Since a robot can move a maximum of 3 cm during the same time, this is a suitable frequency for updating. Although this solution seems complex, the authors share it as an open source project and it is available in ⟨https://github.com/DiODeProject/KilobotArena⟩.

This kind of system does not present a fully autonomous way for the robots to release and follow trails because they depend on a central unit, an external computer. However, new algorithms could come to life by employing the augmented reality idea (GARNIER et al., 2013), (REINA et al., 2015), and (RAMAITHITIMA et al., 2016). We also found

(ARVIN et al., 2015); a work where the authors followed the idea of augmented reality, but employing an LCD screen instead of a projector.

Many researchers have introduced direct communication among robots, and in most cases, employing physical media such as light, sound, radio waves for it. Inspired in audiovisual communication, the authors of (GRIBOVSKIY, MONDADA, 2009) proposed a task where robots have to search dynamic audiovisual sources. In particular, each task comes from a robot imitating a chicken's call for help. The robot searching for chickens includes a special hardware device with a set of microphones to distinguish robot-chicken sounds. An overhead camera provides the robot positions by tracking a visual marker upon them. The open-source SwisTrack software(LOCHMATTER et al., 2008) helped the authors in the visual tracking process. As a consequence, the environment required a controlled light, color-specific walls, and an overhead camera. The searching algorithm was tested in three main tasks for audio-localization of the sound sources, with different separation of the sounds that chickens emit and meanings of these sounds.

A.2.3 Complex solutions

Here are the more complex and expensive aids we found. In particular, some of these aids have to deal with real chemical materials (RUSSELL, 1999), (SVENNEBRING, KOENIG, 2004), and (RUSSELL, 1997); which are a difficult endeavor when working with robots. Besides the technical difficulties, the cost of the hardware for chemical regulation would rise the cots of the research. Some solutions include patches of ground sensors (ANTOUN et al., 2016) and (LIMA et al., 2017). Other solutions tend to act or work with the concept of the Internet of things, which allows several devices to share information through a network (O'HARA, BALCH, 2007), (CARO, DUCATELLE, GAMBARDELLA, 2009), (BARDELLA et al., 2012), (KHALIQ, ROCCO, SAFFIOTTI, 2014), and (KHALIQ, SAFFIOTTI, 2015).

An interesting solution is to use a paper-based environment (SVENNEBRING, KOENIG, 2004), which allows a special ink to evaporate.Robots could draw their trails in such paper-based environment. The ink to write their paths is special and yields thin trails. The authors validated their algorithm in a coverage task, where robots had to avoid the trails as in (MAYET et al., 2010) and (RANJBAR-SAHRAEI et al., 2013), but to regulate the ink evaporation is a hard task. In (RUSSELL, 1999), the authors introduced alcohol trails, which robots can follow with the help of a specific sensor. However, the chemical sensors used in this setup and the combination of substances (e.g., alcohol, the special ink) have been shown to be unreliable and impractical.

Former ideas to allow robots to release pheromone and create trails include heating mechanisms. In (RUSSELL, 1997), robots possessed a heater to mark their path once they find food. The big problem of this solution is the energy consumption of the robot. They worked with a medium size robot and a lamp. The electrical power to generate heat depends greatly on the battery. Therefore, our current technology makes this an impossible solution for smaller robots. Moreover, to control the heat such that the path can have different evaporation (dissipation) rates is difficult. As mentioned before, the rate of evaporation is fundamental in these tests. The environment needs a well-conditioned laboratory: chemicals depend on the humidity and temperature, and the heat could fail if you are in a hot/cold city. Therefore, these solutions require an expensive investment in the laboratory where the robots would work.

In (ANTOUN et al., 2016), the authors worked with Kilobots, too. To overcome robot limitations (i.e., limited sensors and actuators), they created a virtual environment called Kilogrid. The Kilogrid includes a set of cells with sensing and communication capacities. Moreover, each cell offers a visual feedback by displaying a color that is related to its state. This allows a reconfigurable environment. But, it is necessary to add a barrier between cells to avoid infrared interference while robots communicate with cells. The authors created two cases of study in which they could: i) virtualize robot sensors by adding a virtual wall, and ii) virtualize robot actuators by allowing robots to irrigate a virtual field. The communication between robots and the environment is direct and explicit. For instance, when irrigating a field, the robot explores the field by sending messages to the cell where it stands. If the cell identifies itself as a dry patch, the robot will stay there circling while delivering a virtual unit of water. Unfortunately, the Kilogrid is costly and time-consuming to construct. Each cell module costs almost the same price of a Kilobot.

We found a similar solution in (LIMA et al., 2017), where the authors suggested a ground consisting of a grid of sensors to coordinate and synchronize their robots. Since their model uses cellular automata principles to work, the coordination mechanism requires each robot to be in the center of a cell to compute its following movement. Each cell has a sensor to detect the presence of a robot and indicates through a color whether it is an object, an obstacle, or a free patch. Robots could forage and avoid collisions thanks to a centralized mechanism that uses the ground information. Three layers contain the ground information: One considers the weights of the gradient orientation, another layer contains the visual information exhibited through the ground colors. Finally, the last layer mixes objects and robots into one map. Note that this approach is complex and unpractical because robots are working in a continuous world; they do not need to center at each movement.

Next, we found some environments with pre-deployed nodes in (CARO, DUCATELLE, GAMBARDELLA, 2009) and (O'HARA, BALCH, 2007). Nodes are like wireless sensors, but they measure nothing. In (CARO, DUCATELLE, GAMBARDELLA, 2009), the nodes create a data route, which allows information of the estimated distance and the angle of the target may reach the searching robot. In (O'HARA, BALCH, 2007), the nodes create points where the robot should pass. Since the goal is to visit all nodes, each node informs the robots of its attractor value to push them toward unvisited nodes. Both solutions work even when the environment is dynamic. The underlying idea is to help robots to share information across something like a WSRN (Wireless Sensors and Robots Network), but sensors are just sharing information. These networks consist of a great number of nodes, and their placement can affect the group performance.

Working with more complex devices in (BARDELLA et al., 2012), the authors mimicked a similar approach in a complex searching task. The robots have to find an object by its descriptors. The object provides its own descriptors to a robot whenever it is close and asking for this information. If the object descriptors match the robot target, then the robot takes a picture and sends it to the operator indicating a successful search. Note that this idea is similar to the trend mentioned before. Allowing robots and objects to share information in a network can lower the computational burden in the robots and the programming complexity requested.

Although the use of RFID technology may boost the cooperation and interaction between objectd and robots, it has a limited scope. In particular, the RFID technology has intrinsic constraints such as a rather short operational range (especially for the passive RFIDs) and the extremely limited computational capabilities of the tags. However, tags can work similarly to the smart objects of the networks in (BARDELLA et al., 2012). In (KHALIQ, SAFFIOTTI, 2015), the tags in the environment helped robots to plan its navigation. While in (MAMEI, ZAMBONELLI, 2005), the tags helped to control the spread of pheromones. Environments with RFID seems to be nearer to a possible real life implementation, i.e., to go beyond the laboratories.

A.3 Electronic Implementation

We opted for TAMs mechanisms based on (BRUTSCHY et al., 2010), where the authors also employed communication devices. In our version of TAMs, the devices are simply indicating the state of the task with a RGB LED and detecting robot presence through infrared sensors. They offer a middle complexity because they need a programming stage, but are cheap to implement. The performance is recorded in the Arduino Board and displayed in a LCD 16x2 screen for the experimenter. We also added a receptor of infrared signals to change the delays to receive/deliver objects in TAMs. Figure 1 shows the circuit implementation in our laboratory. Figure 2 shows the ideal components for a bigger world.

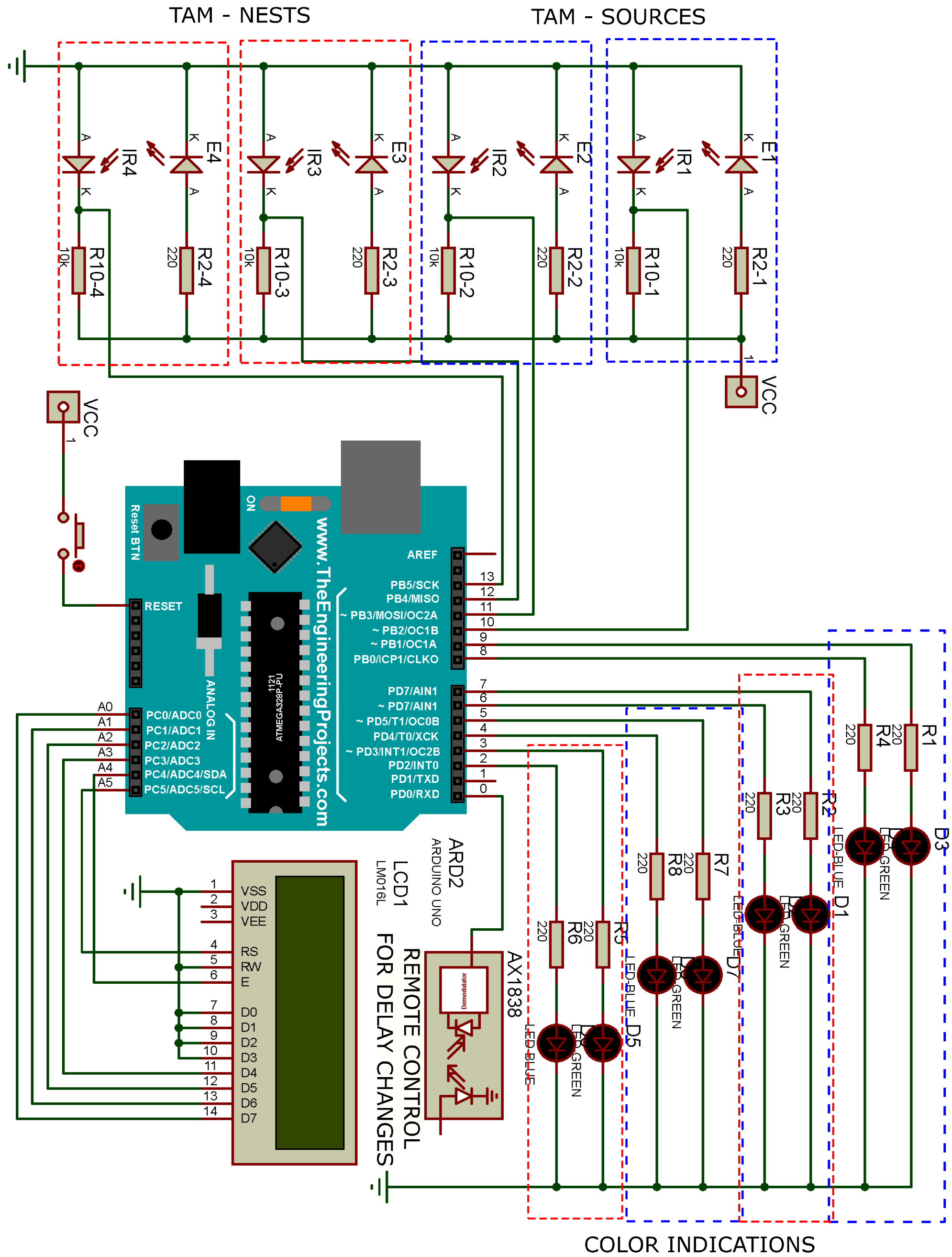


Figura 1 – Circuit for implementing 4 adapted TAMs working with an Arduino UNO

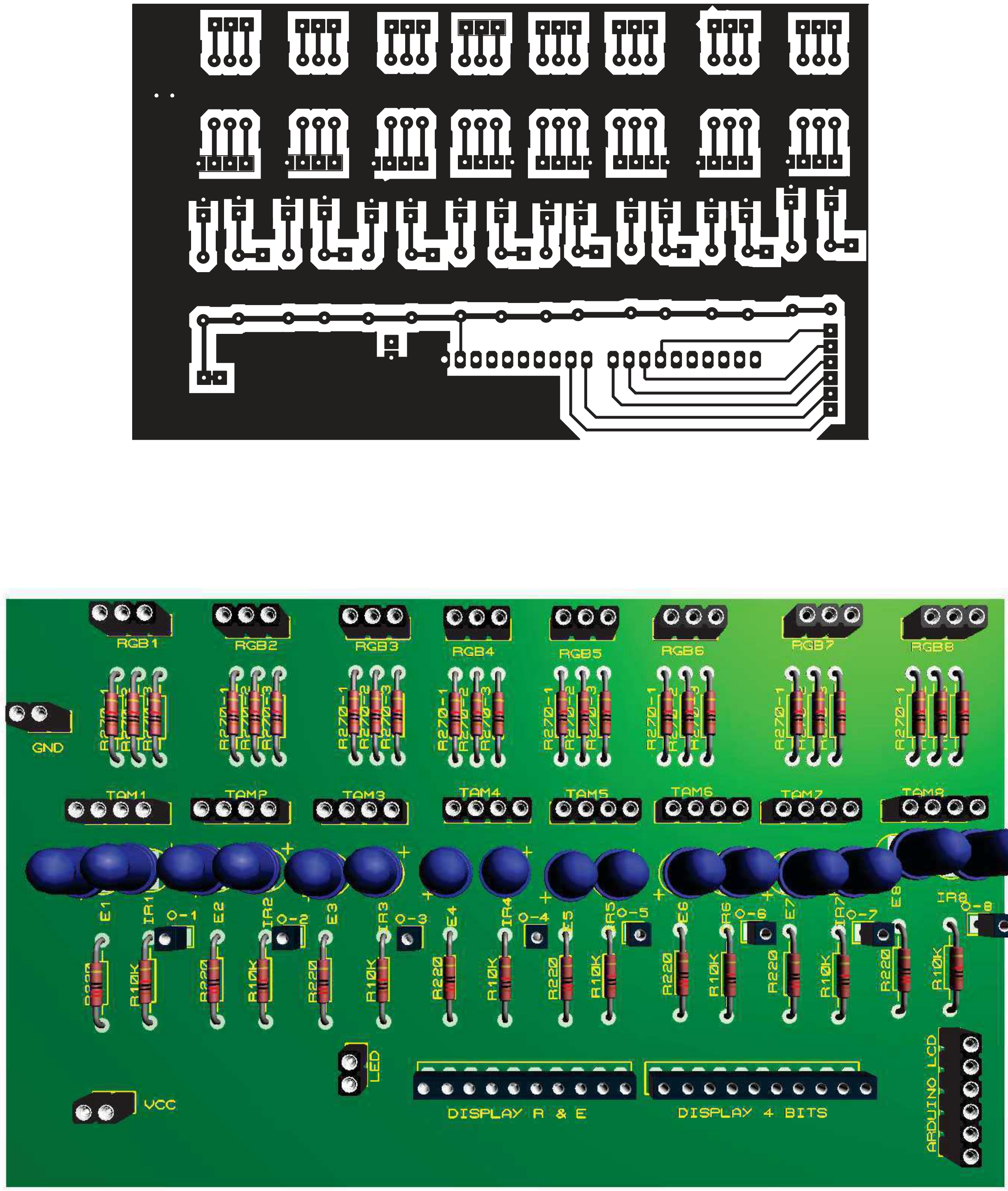


Figura 2 – Circuit for printing a PCB of 8 adapted TAMs to work with an Arduino MEGA

During this research, we created some manuals where we explain how to restore the firmware of e-pucks (with Claudiney Tinoco), recover e-puck PIN number, and a short guide to reading the fundamental aspects of the tutorial of Webots. However, these tools were not the focus of our work and are not written here.

A.4 Conclusions

These kinds of solutions encourage us to find what is the least amount of information needed to solve a certain task. Designers need to have insights of the inherent complexity of the task. There is a threshold between making smarter robots and making smarter environments. Smarter robots require complex behaviors and sensors. Smarter environments provide aids through different kinds of objects and patterns in them. For both, it is the designer's responsibility to balance between robot-robot and robot-environment interactions.

Instead of spending resources on inessential aspects of a real life application, the purpose of these solutions is to provide a cheap and very easy way to handle laboratory tools to test algorithms. These environments allow robots to perceive and adapt their behavior in an almost fully autonomous way, as humans do. It is important to make explicit that robots need to know the meanings of the signs of the environment. Otherwise, they would be useless.

We observe that many of the simulated works are coming to real life scenarios (working with physical robots). However, experiments involving swarms of robots tend to be expensive, and they present technical problems because robots generally have limited sensors and actuators. The time needed to calibrate is extensive, and it is difficult to track and communicate with each individual robot. Researchers are dealing with this transition because reality and simulation can be very different. In particular, noise and the reality gap could make the simulated behaviors diverge from its goal.

Referências

ALERS, S. et al. An experimental framework for exploiting vision in swarm robotics. In: IARIA XPS. *Proc. of the 5th International Conference on Adaptive and Self-Adaptive Systems and Applications*. Valencia, Spain, 2013.

ANTOUN, A. et al. Kilogrid: A modular virtualization environment for the kilobot robot. In: IEEE. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Daejeon, South Korea, 2016. p. 3809–3814. DOI: ⟨http://dx.doi.org/10.1109/IROS.2016.7759560⟩.

ARNDT, M., BERNS, K. Optimized mobile indoor robot navigation through probabilistic tracking of people in a wireless sensor network. In: *Proc. of the 7th German Conference on Robotics*. Germany: [s.n.], 2012. p. 1–6. ISBN 978-3-8007-3418-4.

ARVIN, F. et al. Colias-φ: an autonomous micro robot for artificial pheromone communication. *International Journal of Mechanical Engineering and Robotics Research*, v. 4, n. 4, p. 349–353, 2015. DOI: ⟨http://dx.doi.org/0921/20150921021314970⟩.

BALCH, T., ARKIN, R. C. Communication in reactive multiagent robotic systems. *Autonomous robots*, Springer, v. 1, n. 1, p. 27–52, 1994. DOI: ⟨http://dx.doi.org/10.1007/BF00735341⟩.

BARDELLA, A. et al. Autonomous robot exploration in smart environments exploiting wireless sensors and visual features. *Annals of telecommunications*, v. 67, n. 7, p. 297–311, Aug 2012. ISSN 1958-9395. DOI: ⟨https://doi.org/10.1007/s12243-012-0305-z⟩.

BAYINDIR, L. A review of swarm robotics tasks. *Neurocomputing*, v. 172, p. 292 – 321, 2015. ISSN 0925-2312. DOI: ⟨http://dx.doi.org/10.1016/j.neucom.2015.05.116⟩.

BERG, J., KARUD, C. H. *Swarm intelligence in bio-inspired robotics*. [S.l.]: Institutt for datateknikk og informasjonsvitenskap, 2011.

BLOW, M. Stigmergy: biologically-inspired robotic art. In: AISB. *Proc. of the Symposium on Robotics, Mechatronics and Animatronics in the Creative and Entertainment Industries and Arts*. Hatfield, UK, 2005. p. 1–8. DOI: ⟨http://dx.doi.org/10.1.1.669.1582⟩.

BOBADILLA, L. et al. Controlling wild mobile robots using virtual gates and discrete transitions. In: IEEE. *American Control Conference*. 2012. p. 743–749. DOI: ⟨http://dx.doi.org/10.1109/ACC.2012.6315569⟩.

BRAMBILLA, M. et al. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, v. 7, n. 1, p. 1–41, 2013. DOI: ⟨http://dx.doi.org/10.1007/s11721-012-0075-2⟩.

BRUTSCHY, A. et al. The TAM: abstracting complex tasks in swarm robotics research. *Swarm Intelligence*, Springer, v. 9, n. 1, p. 1–22, 2015. DOI: ⟨http://dx.doi.org/10.1007/s11721-014-0102-6⟩.

BRUTSCHY, A. et al. The IRIDIA TAM: a device for task abstraction for the e-puck robot. *TR/IRIDIA/2010-015, IRIDIA*, Université Libre de Bruxelles, Belgium, v. 1, n. 15, 2010.

CAMPO, A. et al. Artificial pheromone for path selection by a foraging swarm of robots. *Biological cybernetics*, Springer, v. 103, n. 5, p. 339–352, 2010. DOI: ⟨http://dx.doi.org/10.1007/s00422-010-0402-x⟩.

CARO, G. A. D., DUCATELLE, F., GAMBARDELLA, L. M. *Wireless Communications for Distributed Navigation in Robot Swarms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. 21–30 p. ISBN 978-3-642-01129-0. DOI: ⟨https://doi.org/10.1007/978-3-642-01129-0_3⟩.

CHEN, J., GAUCI, M., GROß, R. A strategy for transporting tall objects with a swarm of miniature mobile robots. In: IEEE. *Proc. of the IEEE International Conference on Robotics and Automation*. 2013. p. 863–869. ISSN 1050-4729. DOI: ⟨http://dx.doi.org/10.1109/ICRA.2013.6630674⟩.

CHEN, J. et al. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, v. 31, n. 2, p. 307–321, April 2015. ISSN 1552-3098. DOI: ⟨http://dx.doi.org/10.1109/TRO.2015.2400731⟩.

DENEUBOURG, J.-L. et al. The dynamics of collective sorting robot-like ants and ant-like robots. In: *Proc. of the 1st International Conference on Simulations of Adaptive Behaviors: From animals to animats*. [s.n.], 1991. p. 356–363. DOI: ⟨http://dx.doi.org/116517/116557⟩.

DIMIDOV, C., ORIOLO, G., TRIANNI, V. *Random Walks in Swarm Robotics: An Experiment with Kilobots*. Cham: Springer International Publishing, 2016. 185–196 p. ISBN 978-3-319-44427-7. DOI: ⟨https://dx.doi.org/10.1007/978-3-319-44427-7_16⟩.

DORIGO, M. et al. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics Automation Magazine*, v. 20, n. 4, p. 60–71, Dec 2013. ISSN 1070-9932. DOI: ⟨http://dx.doi.org/10.1109/MRA.2013.2252996⟩.

GARNIER, S. et al. Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? a swarm robotics test bed. *PLoS computational biology*, Public Library of Science, v. 9, n. 3, p. e1002903, 2013. DOI: ⟨http://dx.doi.org/10.1371/journal.pcbi.1002903⟩.

GARNIER, S. et al. Alice in pheromone land: An experimental setup for the study of ant-like robots. In: IEEE. *Proc. of the IEEE Symposium on Swarm Intelligence*. 2007. p. 37–44. DOI: ⟨http://dx.doi.org/10.1109/SIS.2007.368024⟩.

GRIBOVSKIY, A., MONDADA, F. Real-time audio-visual calls detection system for a chicken robot. In: *Proc. of the International Conference on Advanced Robotics*. [S.l.: s.n.], 2009. p. 1–6. ISBN 978-1-4244-4855-5.

HARPER, R. Inside the smart home: Ideas, possibilities and methods. *Inside the smart home*, Springer, p. 1–13, 2003. DOI: ⟨http://dx.doi.org/10.1007/1-85233-854-7_1⟩.

HAYES, A. T., MARTINOLI, A., GOODMAN, R. M. Swarm robotic odor localization: off-line optimization and validation with real robots. *Robotica*, Cambridge University Press, v. 21, n. 4, p. 427–441, 2003. DOI: ⟨http://dx.doi.org/10.1017/S0263574703004946⟩.

JALOBEANU, M. et al. Reliable kinect-based navigation in large indoor environments. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2015. DOI: ⟨http://dx.doi.org/10.1109/ICRA.2015.7139225⟩.

KHALIQ, A. A., ROCCO, M. D., SAFFIOTTI, A. Stigmergic algorithms for multiple minimalistic robots on an rfid floor. *Swarm Intelligence*, v. 8, n. 3, p. 199–225, Sep 2014. ISSN 1935-3820. DOI: ⟨https://doi.org/10.1007/s11721-014-0096-0⟩.

KHALIQ, A. A., SAFFIOTTI, A. Stigmergy at work: Planning and navigation for a service robot on an rfid floor. In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 2015. p. 1085–1092. ISSN 1050-4729. DOI: ⟨http://dx.doi.org/10.1109/ICRA.2015.7139311⟩.

LIMA, D. A. et al. Coordination, synchronization and localization investigations in a parallel intelligent robot cellular automata model that performs foraging task. In: INSTICC. *Proc. of the 9th International Conference on Agents and Artificial Intelligence*. SciTePress, 2017. p. 355–363. ISBN 978-989-758-220-2. DOI: ⟨http://dx.doi.org/10.5220/0006081403550363⟩.

LOCHMATTER, T. et al. Swistrack - a flexible open source tracking software for multi-agent systems. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. [s.n.], 2008. p. 4004–4010. ISSN 2153-0858. DOI: ⟨http://dx.doi.org/10.1109/IROS.2008.4650937⟩.

MAMEI, M., ZAMBONELLI, F. Physical deployment of digital pheromones through rfid technology. In: *Proc. of the IEEE Swarm Intelligence Symposium*. [s.n.], 2005. p. 281–288. DOI: ⟨http://dx.doi.org/10.1109/SIS.2005.1501633⟩.

MAO, G., FIDAN, B., ANDERSON, B. D. Wireless sensor network localization techniques. *Computer Networks*, v. 51, n. 10, p. 2529 – 2553, 2007. ISSN 1389-1286. DOI: ⟨http://dx.doi.org/10.1016/j.comnet.2006.11.018⟩.

MAYET, R. et al. *Antbots: A Feasible Visual Emulation of Pheromone Trails for Swarm Robots*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 84–94 p. ISBN 978-3-642-15461-4. DOI: ⟨https://doi.org/10.1007/978-3-642-15461-4_8⟩.

MELHUISH, C., WELSBY, J., EDWARDS, C. Using templates for defensive wall building with autonomous mobile antlike robots. In: *Proc. of Conference Towards Intelligent Mobile Robots*. [S.l.: s.n.], 1999.

MOHAN, Y., PONNAMBALAM, S. G. An extensive review of research in swarm robotics. In: *Proc. of the World Congress on Nature Biologically Inspired Computing*. [s.n.], 2009. p. 140–145. DOI: ⟨http://dx.doi.org/10.1109/NABIC.2009.5393617⟩.

NOGALES, J. M., ESCARPINATI, M. C., OLIVEIRA, G. M. Shark-inpired target approach strategy for foraging with visual clues. In: . Proc. of the 18th Conference Towards Autonomous Robotic Systems (TAROS), 2017. p. 182–198. ISBN 978-3-319-64107-2. DOI: ⟨http://dx.doi.org/10.1007/978-3-319-64107-2_15⟩.

OCA, M. A. Montes de et al. Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, v. 5, n. 3, p. 305–327, Dec 2011. ISSN 1935-3820. DOI: ⟨https://doi.org/10.1007/s11721-011-0062-z⟩.

O'HARA, K. J., BALCH, T. R. *Pervasive Sensor-less Networks for Cooperative Multi-robot Tasks*. Tokyo: Springer Japan, 2007. 305–314 p. ISBN 978-4-431-35873-2. DOI: ⟨https://doi.org/10.1007/978-4-431-35873-2_30⟩.

PINI, G. et al. Task partitioning in swarms of robots: An adaptive method for strategy selection. *Swarm Intelligence*, Springer, v. 5, n. 3-4, p. 283–304, 2011. DOI: ⟨http://dx.doi.org/10.1007/s11721-011-0060-1⟩.

PINI, G. et al. Task partitioning in a robot swarm: a study on the effect of communication. *Swarm Intelligence*, Springer, v. 7, n. 2-3, p. 173–199, 2013. DOI: ⟨http://dx.doi.org/10.1007/s11721-013-0078-7⟩.

PITONAKOVA, L., CROWDER, R., BULLOCK, S. Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, Springer, v. 10, n. 1, p. 33–63, 2016. DOI: ⟨http://dx.doi.org/10.1007/s11721-016-0118-1⟩.

RAMAITHITIMA, R. et al. Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots. *IEEE Robotics and Automation Letters*, v. 1, n. 2, p. 746–753, July 2016. ISSN 2377-3766. DOI: ⟨http://dx.doi.org/10.1109/LRA.2016.2523600⟩.

RANJBAR-SAHRAEI, B. et al. Stico in action. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. *Proc. of the International Conference on Autonomous agents and multi-agent systems*. 2013. p. 1403–1404. DOI: ⟨http://dx.doi.org/10.1145/1858171.1858249⟩.

RANJBAR-SAHRAEI, B., WEISS, G., NAKISAEE, A. A multi-robot coverage approach based on stigmergic communication. In: SPRINGER. *German Conference on Multiagent System Technologies*. Germany, 2012. p. 126–138. Disponível em: ⟨http://dx.doi.org/10.1007/978-3-642-33690-4_13⟩.

REINA, A. et al. Ark: Augmented reality for kilobots. *IEEE Robotics and Automation Letters*, v. 2, n. 3, p. 1755–1761, July 2017. ISSN 2377-3766. DOI: ⟨http://dx.doi.org/10.1109/LRA.2017.2700059⟩.

REINA, A. et al. Augmented reality for robots: virtual sensing technology applied to a swarm of e-pucks. In: IEEE. *Proc. of the Conference on Adaptive Hardware and Systems*. 2015. p. 1–6. DOI: ⟨http://dx.doi.org/10.1109/AHS.2015.7231154⟩.

RUBENSTEIN, M., AHLER, C., NAGPAL, R. Kilobot: A low cost scalable robot system for collective behaviors. In: IEEE. *Proc. of the IEEE International Conference on Robotics and Automation*. 2012. p. 3293–3298. DOI: ⟨http://dx.doi.org/10.1109/ICRA.2012.6224638⟩.

RUSSELL, R. A. Heat trails as short-lived navigational markers for mobile robots. In: *Proc. of International Conference on Robotics and Automation*. [s.n.], 1997. v. 4, p. 3534–3539 vol.4. DOI: ⟨http://dx.doi.org/10.1109/ROBOT.1997.606882⟩.

RUSSELL, R. A. Ant trails - an example for robots to follow? In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 1999. v. 4, p. 2698–2703 vol.4. ISSN 1050-4729. DOI: ⟨http://dx.doi.org/10.1109/ROBOT.1999.774005⟩.

SAFFIOTTI, A., BROXVALL, M. Peis ecologies: Ambient intelligence meets autonomous robotics. In: *Proc of the International Conference on Smart Objects and Ambient Intelligence*. [s.n.], 2005. p. 275–280. Disponível em: ⟨http://dx.doi.org/10.1145/1107548.1107615⟩.

ŞAHIN, E. Swarm robotics: From sources of inspiration to domains of application. Springer Berlin Heidelberg, Santa Monica, CA, p. 10–20, 2005. DOI: ⟨https://doi.org/10.1007/978-3-540-30552-1_2⟩.

SANFELIU, A., HAGITA, N., SAFFIOTTI, A. Network robot systems. *Robot. Auton. Syst.*, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, v. 56, n. 10, p. 793–797, out. 2008. ISSN 0921-8890. DOI: ⟨http://dx.doi.org/10.1016/j.robot.2008.06.007⟩.

SCHWAGER, M. et al. Time scales and stability in networked multi-robot systems. In: *Proc. of the IEEE International Conference on Robotics and Automation*. [s.n.], 2011. p. 3855–3862. ISSN 1050-4729. DOI: ⟨http://dx.doi.org/10.1109/ICRA.2011.5979847⟩.

STEWART, R. L., RUSSELL, R. A. A distributed feedback mechanism to regulate wall construction by a robotic swarm. *Adaptive Behavior*, Sage Publications Sage CA: Thousand Oaks, CA, v. 14, n. 1, p. 21–51, 2006. DOI: ⟨http://dx.doi.org/10.1177/105971230601400104⟩.

SUGAWARA, K., KAZAMA, T., WATANABE, T. Foraging behavior of interacting robots with virtual pheromone. In: IEEE. *Proc. of the IEEE Conference on Intelligent Robots and Systems*. 2004. v. 3, p. 3074–3079. Disponível em: ⟨http://dx.doi.org/10.1109/IROS.2004.1389878⟩.

SVENNEBRING, J., KOENIG, S. Building terrain-covering ant robots: A feasibility study. *Autonomous Robots*, v. 16, n. 3, p. 313–332, May 2004. ISSN 1573-7527. DOI: ⟨https://doi.org/10.1023/B:AURO.0000025793.46961.f6⟩.

THERAULAZ, G., BONABEAU, E., DENUEBOURG, J. Response threshold reinforcements and division of labour in insect societies. In: *Proc. of the Royal society of London: Biological Sciences*. [s.n.], 1998. v. 265, n. 1393, p. 327–332. DOI: ⟨http://dx.doi.org/10.1098/rspb.1998.0299⟩.

VALENTINI, G. et al. Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, Hingham, MA, USA, v. 30, n. 3, p. 553–580, maio 2016. ISSN 1387-2532. DOI: ⟨http://dx.doi.org/10.1007/s10458-015-9323-3⟩.

ZUEHLKE, D. Smartfactory – from vision to reality in factory technologies. *IFAC Proc. Volumes*, v. 41, n. 2, p. 14101 – 14108, 2008. ISSN 1474-6670. 17th IFAC World Congress. DOI: ⟨http://dx.doi.org/10.3182/20080706-5-KR-1001.02391⟩.