
# THESIS

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :** *l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*
Cotutelle internationale *PEDECIBA, Universidad de la República, Uruguay*

---

**Présentée et soutenue le** *Date de défense (14/02/2019)* **par :**

FACUNDO BENAVIDES OLIVERA

---

**Multi-robot exploration under non-ideal communication conditions**

---

**JURY**

| | | |
|---|---|---|
| FRANÇOIS CHARPILLET | Professeur d'Université | Rapporteur |
| ALFREDO WEITZENFELD | Professeur d'Université | Rapporteur |
| EDUARDO GRAMPÍN | Professeur d'Université | Directeur de Thèse |
| JANETTE CARDOSO | Professeur | Directrice de Thèse |
| ALVARO MARTÍN | Professeur Associé | Examinateur |
| ISABEL AMIGO | Professeur Assistant | Examinateur |

---

**École doctorale et spécialité :**
*EDSYS : Robotique 4200046*
**Unité de Recherche :**
*Aerospace Vehicles Design and Control Department (DCAS)*
**Directeur(s) de Thèse :**
*Eduardo GRAMPÍN*, *Janette CARDOSO*, *Pablo MONZÓN* et *Caroline P. CARVALHO CHANEL*
**Rapporteurs :**
*François CHARPILLET* et *Alfredo WEITZENFELD*

# Acknowledgements

... .

# Contents

# List of Figures

# List of Tables

# Table of acronyms

| | |
|---|---|
| **AAMO** | Auto-Adaptive Multi-Objective approach |
| **CC** | Connected Component |
| **CR** | Coverage Ratio |
| **EbC** | Event-based Connectivity approach |
| **DLR** | Disconnection Last Ratio |
| **DR** | Dual Role approach |
| **LoS** | Line-of-sight |
| **MANET** | Mobile Ad-hoc NETwork |
| **MDLR** | Maximum Disconnection Last Ratio |
| **MDP** | Markov Decision Process |
| **MORSE** | Modular Open Robots Simulation Engine |
| **MP** | MinPos approach |
| **MRS** | Multi Robot System |
| **OSR** | Over-Sensing Ratio |
| **PL** | Path length |
| **POMDP** | Partially Observable Markov Decision Process |
| **TT** | Total exploration Time |
| **UAV** | Unmanned Aerial Vehicle |
| **Y** | Yamauchi approach |

# Introduction

The exploration problem is a fundamental subject in autonomous mobile robotics that deals with achieving the complete coverage of a previously unknown environment. There are several scenarios where completing exploration of a zone is a central part of the mission, e.g. planetary exploration, reconnaissance, search and rescue, agriculture, cleaning, or dangerous places as mined lands and radioactive zones. Additionally, due to the inner qualities -mainly efficiency and robustness- of multi-robot systems, exploration is usually done cooperatively.

Nevertheless, when multiple robots are involved in an exploration task, it is advisable to avoid several of them moving to the same place. Wireless communication plays an important role in collaborative multi-robot strategies. Unfortunately, the assumption or requirement of stable communication and end-to-end connectivity may be easily compromised in real scenarios due to interference, fading or simply robots moving beyond the communication range. When robots are unconnected they have no possibilities to coordinate their actions and damages or inner failures can lead to information losses. Therefore, depending on the application field, the exploration strategy should take this into account to prevent isolation situations.

In this thesis, two novel approaches to tackle the problem of multi-robot exploration of communication constrained environments are proposed. For the sake of robustness, a decentralised approach is followed; coordination is addressed implicitly through localisation and mapping data exchanging.
Robots make decisions asynchronously and following a human operator criterion.
A simple, yet effective model for the signal strength and attenuation effects provide the robots with connectivity awareness. Thereupon, connectivity level measurements are considered together with paths costs to build utility functions which permit to decrease the disconnection periods duration without degrading the performance regarding completion time for the multi-robot system. To this end, a less restrictive connectivity strategy where the robots are motivated but not compelled to keep connected –or even regain connectivity– is followed.
An auto-adaptive multi-objective function is designed to support the selection of tasks regarding both exploration performance and connectivity level. Two roles – explorer and communication relay – are considered to improve the benefits of the task selection strategy. The relay positioning problem is addressed using the signal strength model in a way that explicitly avoids solving the corresponding Steiner Minimum Spanning Tree problem.

## Contributions

The main contributions of these proposals can be summarised as follows.

**Ease to deploy and flexibility** The solution follows a multi-objective strategy where the tasks under consideration are assessed regarding two objectives: travelling costs and connectivity levels. The weights of these potentially conflicting objectives are derived from formal analysis instead of training. The human operator is asked to use his application-field expertise to play a part in the task assessment process by setting a distance threshold until which the tasks that preserve or enlarge connectivity are preferred over the rest.
All this leads to a more flexible system where the robots can deal with communication constraints adjusting the weights of each objective independently of any scenario in a more intuitive manner and saving a lot of training time too.

**Good performance** Robots can compute the same optimal tasks-to-robots distributions in a short time. Asynchronism is taken as a natural way of avoiding waiting times and making locally optimal decisions linearly computable most of the time. Compared with others, the proposed approaches are capable of decreasing the last of disconnection periods without noticeable degradation of the completion exploration time.

**Novel Relay placement polynomial-time solution** Based on the communication model, a novel polynomial-time relay placement approach for multi-robot exploration missions is introduced in detail. Since the approach does not make strong working assumptions it should be flexible enough to apply to systems which have static or dynamic role assignment policies, indistinctly.

## Outline

The present document is organised as follows. Chapter 1 presents the main contributions and drawbacks from a set of state-of-art proposals. Chapter 2 provides the exploration problem formalisation including models and goals. Next, an auto-adaptive multi-objective approach, as well as the task allocation algorithm and the decentralised coordination mechanism, are thoroughly described in Chapter 3. Experimental results related to a baseline and the *Auto-Adaptive Multi-Objective (AAMO)* approach itself are discussed in Chapter 4. From this, another multi-role based approach is introduced in Chapter 5. The assessment of this last approach is presented in Chapter 6. Finally, the document is concluded highlighting some future research directions in Chapter 7.

Chapter 1

# State of Art on Multi-Robot Exploration Systems

## Contents



The exploration problem is a fundamental subject in autonomous mobile robotics and deals with achieving the complete coverage of a previously unknown environment and fulfilling some optimal criteria. There are several scenarios where completing exploration of a zone is a central part of the mission, e.g. planetary exploration, reconnaissance, search and rescue, agriculture, cleaning, or dangerous places as mined lands and radioactive zones. Additionally, due to the inner qualities -mainly efficiency and robustness- of multi-robot systems, exploration is usually done cooperatively (Burgard et al. 2005). Typically, the overall exploration time is the most commonly used quality indicator to measure and to compare different proposals (Yan et al. 2015).

Schematically, the exploration of an environment can be seen as the composition of *Mapping* and *Motion Planning* tasks. A map is needed in order to plan new motions. Moreover, choosing a correct motion sequence based on this map is also needed to expand the knowledge about the environment optimally. Consequently, *Mapping* is regularly interleaved with *Motion Planning*, and vice versa during the whole process (Wurm, Stachniss, and Burgard 2008; Burgard et al. 2005; Simmons et al. 2000).

## 1.1 Task assignment

When multiple robots are involved in an exploration task, it is advisable to avoid several of them moving to the same place. The task assignment problem concerns the choice of new places[1] to visit in a coordinated way. To reach this purpose is usual to split up the task into two steps. The first one, called *Task Identification*, concerns the identification of the points of interest that should be visited next. It strongly depends on both the sensory robot capabilities and the underlying environment representation. The second one, called *Task Allocation*, concerns the search of a distribution of tasks to robots that maximises the overall system utility and minimises the amount of overlapped information obtained by all of them (Korsah, Stentz, and Dias 2013; Burgard et al. 2005; Gerkey 2004).

### 1.1.1 Task identification

The most widely used representation for this purpose is the well-known *Occupancy Grid* structure (Elfes 1989). Based on it, a method to identify points of interest was proposed by (Yamauchi 1998). The strategy assumes that the closer to the frontier between known and unknown regions the tasks are defined, the more information the team can gather. Since then, the majority of exploration proposals have adopted this scheme known as *Frontier Points* or *Frontier Regions* (Keidar and Kaminka 2014; Keidar and Kaminka 2012; Yuan et al. 2010). See Fig.1.1.

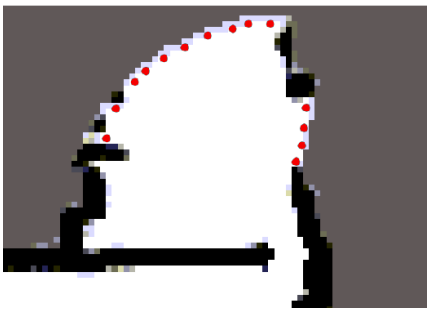


**Figure 1.1:** Occupancy Grid representation. *Obstacle* cells are black; *Free* cells are white, *Unknown* are grey, and small red circles mark *Frontier Points*.

### 1.1.2 Task Allocation

There exist a wide variety of proposed solutions to this problem where a family of methods based on market economy are probably the most popular ones. These methods are based on the notion of *Auctions* from which the robots can bid for the tasks to decide who goes to where at each moment. The market may be managed centrally either by a virtual agent at the base station as in (Simmons et al. 2000) where the bids are processed centralised by a greedy

[1] These singular places will be referred along the document as tasks or targets, indistinctly.

algorithm or by a robotic agent as in (Burgard et al. 2005). Conversely, the fleet can manage to exchange the bids among all the members in order to take decentralised decisions (Sheng et al. 2006; Zlot et al. 2002), avoiding, in turn, the single point of failure. All these methods owe their popularity to their simplicity and ease of implementation, but they suffer from a significant shortcoming: falling in local minima (Cavalcante, Noronha, and Chaimowicz 2013).

Far from economy inspired approaches, a scheduling based approach is presented in (Wurm, Stachniss, and Burgard 2008). This method combines an environment segmentation technique with the centralised task allocation method proposed by (Kuhn 1955). The exploration is performed after dividing the environment into disjoint segments (see Fig.1.2). Thus, the expected sensory overlap between agents is decreased as much as possible.

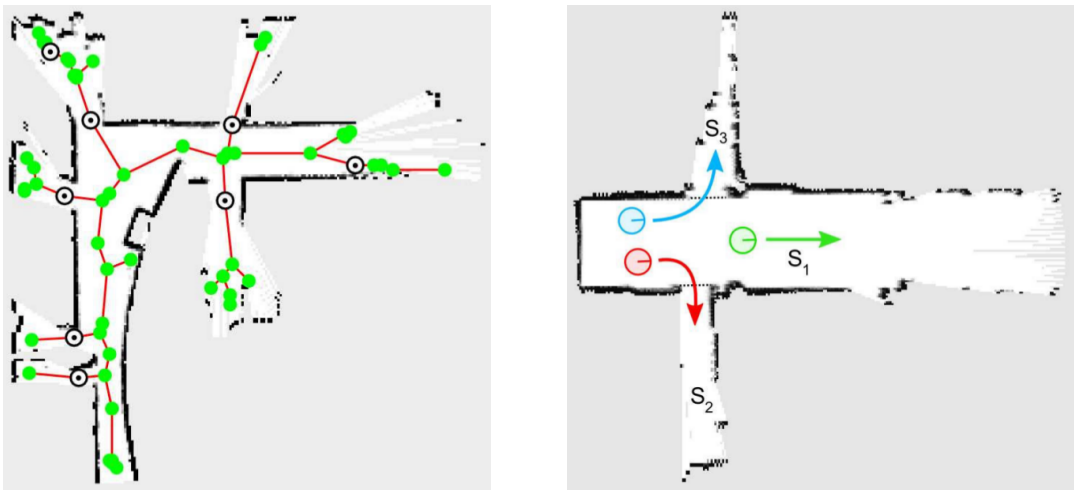
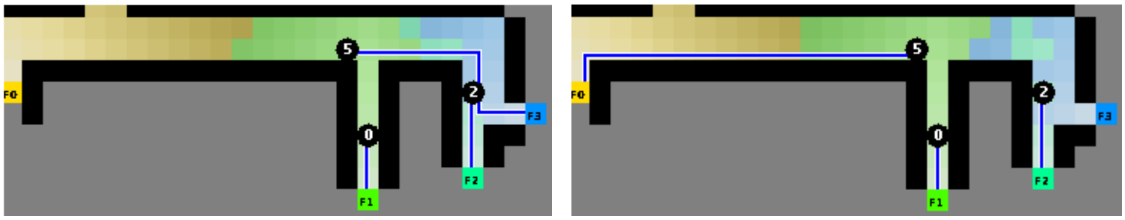



**Figure 1.2:** Task allocation of a segmented environment (Wurm, Stachniss, and Burgard 2008). (a) Segmentation of a small fraction of an environment. (b) Typical coordination of robots obtained by assigning them to different segments of the partial map.

In (Hollinger and Singh 2012) the authors address coordination implicitly through localisation data exchanging. Robots are forced to wait for others before making a decision. Task selection is made iteratively –one robot after another– employing an objective function which rewards the right choices. In (Pham and Juang 2013) a centralised approach is used. The tasks-to-robots distribution is computed balancing information gain, localisation quality and navigation costs. Another centralised approach computes a utility function enabling the robots to locally prioritise the tasks within its scope and, potentially, also enabling the whole team to search for the best global distribution as well (Korsah, Stentz, and Dias 2013).
On the contrary, a decentralised approach, called *minPos* (Bautin and Simonin 2012), attempts to distribute the robots over the unexplored locations as much as possible. By doing so, it has outperformed several reference proposals decreasing the completion-exploration time for a big set of practical scenarios. The working principle is to rank robots concerning their distance to every possible task. An example is depicted in Fig.1.3. The robots coordinate

their actions implicitly and may choose to visit the tasks for which they are best ranked at each point in time.




**(a)** **(b)**

**Figure 1.3:** Task allocation based on robot ranking (Bautin and Simonin 2012). (a) Greedy like allocation. (b) minPos allocation based on a distance ranking of robots. In the scene, $R_0$ is ranked first to task $F_1$, $R_2$ is ranked first to both tasks $F_2$ and $F_3$, $R_5$ is ranked first to task $F_0$.

Finally, the strategy described in (Valentin et al. 2014) is mainly devoted to deal with uncertainties in sensing and motion processes of a multi-robot system. To this end, the authors model the exploration and mapping problem as a Partially Observable Markov Decision Process (POMDP) that is solved centrally. In (Banfi et al. 2016) the assignment algorithm work in an asynchronous fashion assuming that not all robots must be ready for new plans at the same time.
Unfortunately, all these approaches do not take into account communication conditions. Moreover, ideal communication conditions are assumed (implicitly or explicitly) as a working condition.

## 1.2 Communication issues

Wireless communication plays an important role in collaborative multi-robot strategies. Nevertheless, the assumption or requirement of stable communication and end-to-end connectivity may be easily compromised in real scenarios due to interference, fading or robots moving beyond the communication range. A first critical issue concerns the collective knowledge of the environment. Under communication restrictions, such knowledge cannot be assumed to be always accessible and depending on the coordination mechanism could be the cause of significant performance degradation (Amigoni, Banfi, and Basilico 2017). Therefore, depending on the application, the exploration strategy should take this into account in order to prevent the robots from becoming completely unconnected, let say isolation situations. Such an isolation situation, as well as its possible effects, are illustrated in the example scene depicted in Fig.1.4.

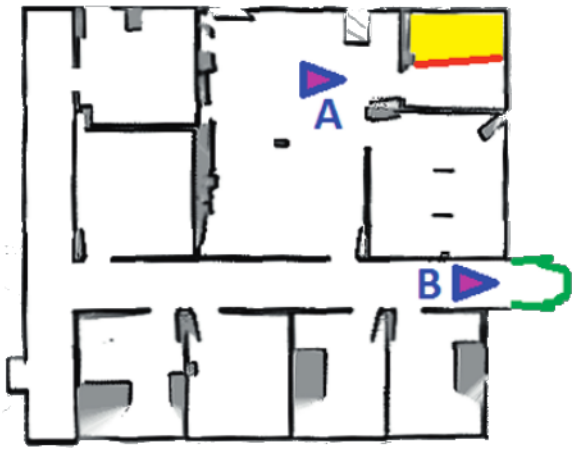



**Figure 1.4:** Re-exploration caused by restricted communication. (Amigoni, Banfi, and Basilico 2017). The yellow portion of the map is only known by robot B. Thus, robot A goes to re-explore the region beyond the red frontier.

### 1.2.1 Connection requirements

Three categories are mainly identified (Amigoni, Banfi, and Basilico 2017):

- *None.* Robots are not required to communicate.
- *Event-based connectivity.* The need for regaining connectivity is triggered by particular events such as the discovery of new information or just periodically.
- *Continuous connectivity.* Every robot must be connected at all times to any other fleet member either directly or in a multi-hop manner.

Please note that these requirements could have an impact on the fleet mobility and, in turn, on the availability of exploration strategies to be adopted. For instance, under a continuous connectivity scheme, the fleet is more restricted to move around than in other cases.

### 1.2.2 Communication models

Communication model refers to the prior knowledge about communication capabilities that support the decision making of the robots along the exploration. Nevertheless, sometimes no communication model is assumed and, consequently, robots do not depend on communicating to decide where to go next. In such cases, explicit coordination only occurs opportunistically due to random encounters (Amigoni, Banfi, and Basilico 2017).
The communication models typically adopted are (Amigoni, Banfi, and Basilico 2017):

- *None.* Robots do not make any assumption on the communication possibilities between any pair of arbitrary locations.

- *Line-of-sight (LoS).* Two robots can communicate if and only if their positions belong to a free-of-obstacle line segment. Usually, the distance is also restricted to a maximum value that is often related to the scope of the communication device.

- *Disc or Circle.* Communication with any other robot is permitted when its location is within a fixed maximum distance (communication radius) regardless of the presence of obstacles.

- *Signal.* Communication is available with a certain probability that depends on the estimated signal power between the robot positions. The higher the signal power, the higher the probability.

- *Traces.* Robots can communicate with each other by dropping messages in the environment.

Additionally, to these five categories that cover an essential aspect of the communications, say connectivity, there exist other formulations aimed at cover bandwidth or throughput as well. Clear examples of its use are the applications with a strong dependence on video streaming like search and rescue applications.

### 1.2.3 Surveyed proposals

Multirobot exploration for building communication maps with prior from communication models

A Multi-Objective Exploration Strategy for Mobile Robots Under Operational Constraints

Multirobot Online Construction of Communication Maps

Collaboration in Multi-Robot Exploration: To Meet or not to Meet?

Multi-Robot Coordination through Dynamic Voronoi Partitioning for Informative Adaptive Sampling in Communication-Constrained Environments

Cooperative Frontier-Based Exploration Strategy for Multi-Robot System

BoB: an online coverage approach for multi-robot systems

Multirobot rendezvous planning for recharging in persistent tasks

Despite its well-known inefficiencies, there exist some few approaches without any connection requirements where robots meet each other by chance. Nevertheless, this section only surveys the proposals that depend on connectivity in one way or another.

In (Vazquez and Malcolm 2004) a behaviour-based approach is presented. The architecture is designed to guide the exploration constraining the fleet to keep within the communication range, establishing a mobile network. The well-known disk model and a graph structure are used to model the network connectivity and identify possible disconnections as is depicted in Fig.1.5a.

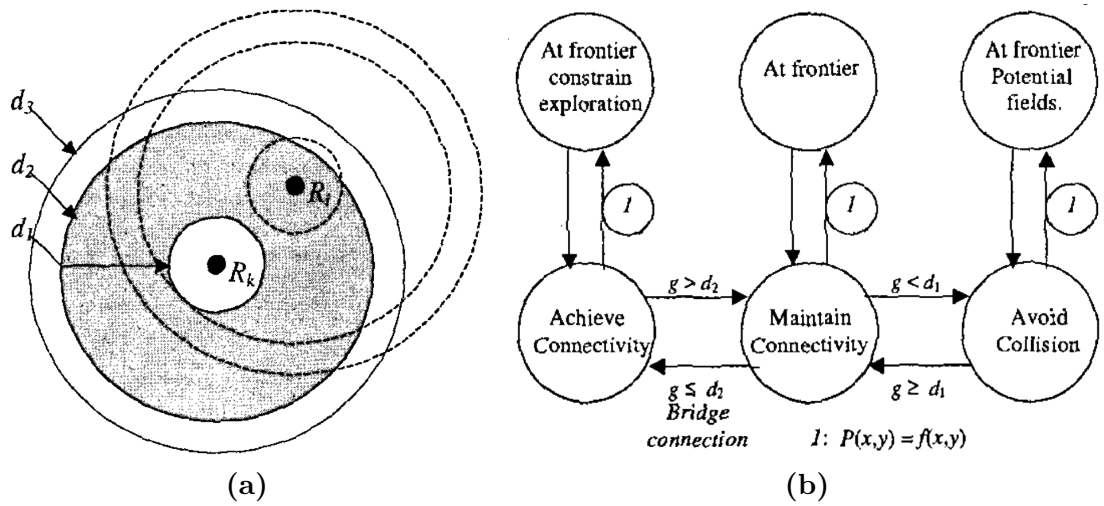
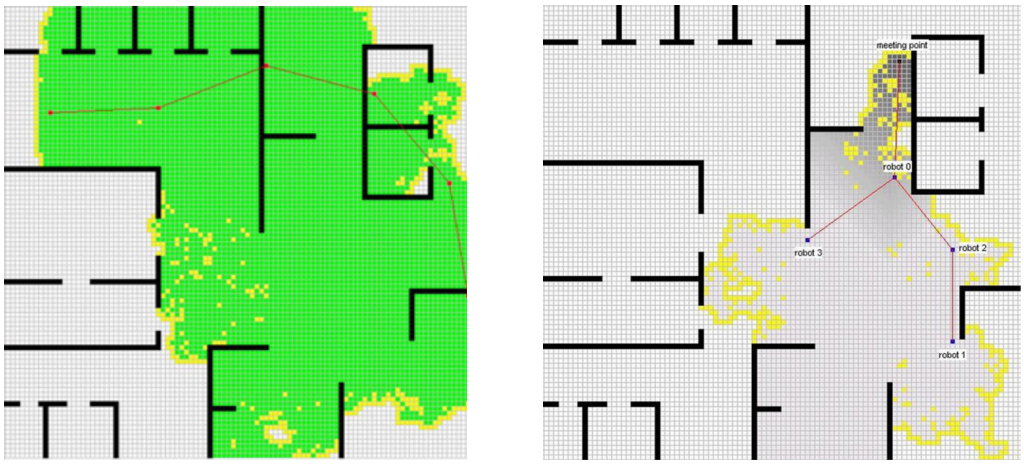



**Figure 1.5:** Behaviour-based exploration. (Vazquez and Malcolm 2004). (a) The comfort zone – represented by the shadowed region– is a collision-free and connectivity-guaranteed zone ($d_1 < g < d_2$). $d_1$ represents a collision constraint while $d_2$ represents a preventive disconnection constraint. When $d > d_3$ the connection between robots is lost. $g$ represents the Euclidean distance between robots. (b) Schematic switching of behaviours. $P$ is the robot position and $f$ is the current frontier.

Frontier cells are evaluated regarding costs (computed utilising a flooding algorithm) and information utility (based on the ideas proposed in (Simmons et al. 2000)). Behaviours are selected according to the network topology conditions. The general schema of behaviour switching is presented in Fig.1.5b.

In (Rooker and Birk 2007) a centralised communicative exploration algorithm is proposed. Communicative exploration implies that the team of robots have to maintain connections between each other at all times. The target selection is based on a utility function that weights the benefits of exploring new regions versus the goal of keeping connected. While connectivity is valued using the classic *disc* model, the costs of the shortest paths are computed from the Manhattan distance notion. Due to spatial and movement restrictions, specific behaviours are defined to deal with deadlocks. See Fig.1.6.

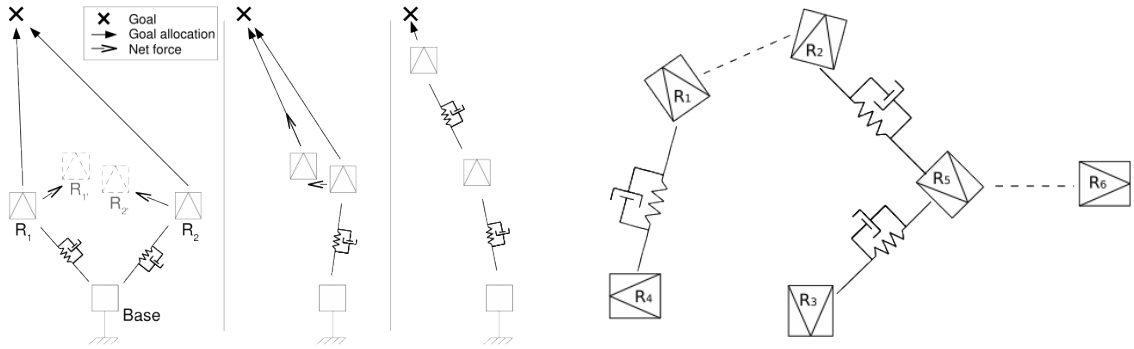
Also following a centralised approach, (Mosteo, Montano, and Lagoudakis 2008) present four fully reactive exploration strategies. They consist in translating the distance to tasks and disconnection situations into artificial forces that pull and push the robot to reach new positions smoothly, avoiding them to lose connectivity. These forces are depicted in Fig.1.7. The radio signal quality is modelled considering both the communication range and the distance attenuation effect. Deadlocks are avoided by assigning tasks to a cluster of robots. This allocation guarantees that robots belonging to the same cluster do not exert conflicting forces




(a) (b)

**Figure 1.6:** Behaviour-based exploration. (Rooker and Birk 2007). Black solid lines represent obstacles. Yellow cells represent *Frontier* cells while grey cells represent the remaining unknown region. (a) Deadlock situation caused by opposed goals: exploring or keeping connected. The green zone represents the known region while red dots and lines, respectively represent robots and communication links. (b) Distance map for meeting point behaviour. The grey scale in the cells represents the distances to the meeting point on the known region. Recovery from deadlocks is carried out through behaviour changes.

upon each other towards different directions.




(a) (b)

**Figure 1.7:** Reactive exploration strategy (Mosteo, Montano, and Lagoudakis 2008). (a) Chain formation achieved from the application of artificial (network and goal) forces. (b) Example of a MANET with virtual springs over links of low quality. Dashed lines represent good quality communication links. The remaining are virtual-springs-like links that pull the robots to avoid disconnections.

In (Le et al. 2009), the authors propose a decentralised version of the strategy proposed in (Rooker and Birk 2007) based on message exchanging and a graph structure where the group always tries to keep a biconnected network efficiently. Communication model is based on the classic *disc* model. In consequence, robot mobility is restricted by the communication range. Using the same graph theory, in (Michael et al. 2009) the experimental validation of a

distributed algorithm that preserves connectivity is also discussed. Nevertheless, a different coordination mechanism –supported by a market-based negotiation algorithm– is adopted. Unfortunately, only results on connectivity maintenance are shown, lacking exploration metrics reports.

The proposal of (Derbakova, Correll, and Rus 2011) aims to maintain and repair the underlying wireless mesh network while the coverage task is being performed, all at once. The system works in a fully asynchronous and distributed way. Differently, from previous works, the authors propose a network disconnection detection by checking the real state of connections without assumptions on communication range or propagation model. In Fig.1.8 the algorithm's reaction to a network failure is shown. On the contrary, all nodes require knowledge about the area to be covered and on global positions.

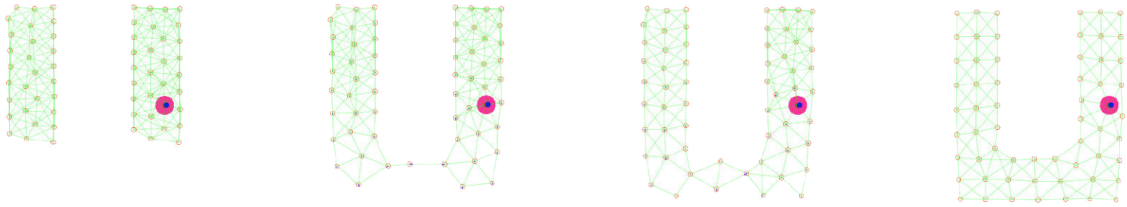


**Figure 1.8:** Network repairing strategy (Derbakova, Correll, and Rus 2011). From left to right. Network failure is simulated; the failure is detected triggering the reaction of the nodes towards a gateway construction; the nodes disperse in order to redistribute themselves across the coverage area.

In (Hollinger and Singh 2012) the robots can disconnect as long as they regain connectivity periodically following a distributed but synchronous strategy. Authors address coordination implicitly through localisation data exchanging. Robots are forced to wait for others before making a decision. The system works as an optimisation method where each variable is optimised at a time in a round-robin while the others remain unchangeable. Fig.1.9 shows how robots adjust their plans in order to regain connectivity.

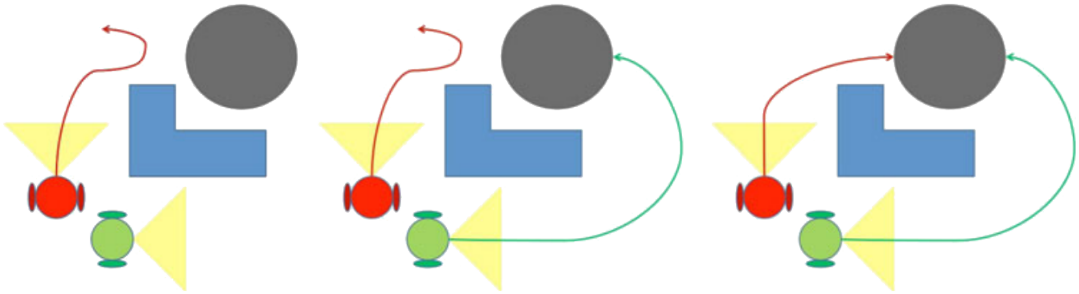


**Figure 1.9:** Periodic connectivity strategy (Hollinger and Singh 2012). The robots (green and red) must move around the obstacle (blue L-shape) to gather information in the grey zone. To do so, they must regain connectivity beyond the obstacle.

In (Laëtitia Matignon and Mouaddib 2012) the problem of exploration and mapping is addressed by using a *Decentralised POMDP*. This technique takes advantage of local interaction and coordination from the interaction-oriented resolution of decentralised decision makers. *Distributed value functions (DVS)* are used by decoupling the multi-agent problem into a set

of individual agent problems. In order to address full local observability, limited information sharing and communication breaks, an extension of the *DVS* methodology is proposed and applied in multi-robot exploration so that each robot computes locally a strategy that minimizes the interaction between fleet members and maximizes the coverage achieved by the team, even in communication constrained environments. The global software architecture of the decision framework from the viewpoint of one robot is depicted in Fig.1.10. A decision step consists in building the model, computing the policy from the *DVS* and producing a trajectory.

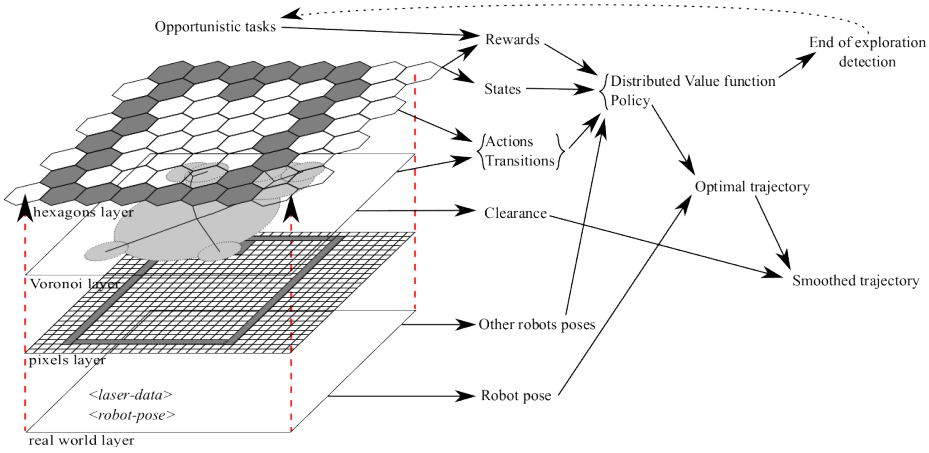



**Figure 1.10:** MDP Framework (Laëtitia Matignon and Mouaddib 2012). The MDP data structures are derived from four grid layers.

Rendezvous-based techniques have also been used to deal with limited communication ranges. In (Pham and Juang 2013) robots are enabled to move out of the communication range but forced to rejoin the group frequently. After moving out the communication range robots have to return to a pre-arranged meeting point to exchange the information gathered during the disconnection period in order to avoid exploration overlaps.

The proposal presented in (Couceiro et al. 2014) describes a *Particle Swarm Optimisation* based approach to achieving fault-tolerance in preventing communication network splits. The principal objective is to keep the fleet $k$-connected. An initial deployment of k-connected robots formation is shown in Fig.1.11. Considering that the application domain defines the fault-tolerance level required to the system, a MANET connectivity algorithm is extended with the concept of $k$-fault-tolerance.

In (Jensen, Nunes, and Gini 2014) a fully distributed approach for multi-robot sweep exploration is introduced. The proposal aims to guarantee full coverage using a minimum number of messages and to maintain connectivity at all times, even under severe restrictions on the communication type, range and quality. The algorithm proposed uses communication not only to exchange information but to direct the robot movements. Communication intensity is used in order to disperse the fleet while beacons are used to mark locations of interest.

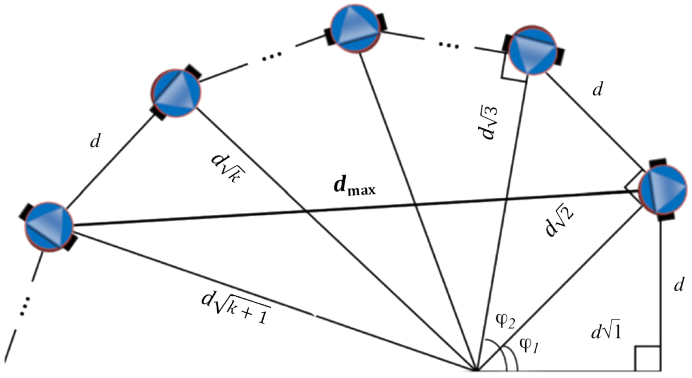



**Figure 1.11:** $k$-connected Network (Couceiro et al. 2014). $d_{max}$ represents the maximum communication distance. The inter-robot distance can be approximated as follows: $d = \frac{d_{max}}{k}$.

#### 1.2.3.1 Multi-Role based approaches

Optimal Redeployment of Multirobot Teams for Communication Maintenance

Mobile Ad-hoc NETworks constitute a particular example of scenarios where the topology of the robot network varies dynamically over time. This kind of network is recommended when the fixed infrastructure is no longer available, e.g. in disasters to support the communication among rescue team members. In such cases, connectivity is of utmost importance because the loss of communication could imply human losses. Since that, it is useful to consider relay robots to guarantee the connectivity of the fleet. This kind of peculiar behaviour, that implies selecting the best locations to forward the information among the remaining robots, may be incorporated to the robot network either by means of fleet members with specific communication maintenance goals or changing dynamically the role of existing members depending on the current conditions, e.g. when an explorer becomes a relay.

This strategy was followed by several authors in different ways. In (Pei and Mutka 2012) the authors describe a heterogeneous multi-robot system for exploration tasks. They consider several explorer robots and conceive a particular robot playing the role of relay dispenser. This agent is in charge of place relays when and where it is necessary to support the video/audio streaming generated by explorers. Fig.1.12 depicts an example scene where the explorer robots need to move out of the communication range.

A multi-robot system for crisis management is described in (Pralet and Lesire 2014). The system is composed of mobile sensors (ground robots - UGV) and mobile relays (aerial vehicles - UAV). However, some robots may change roles dynamically during the mission (e.g. UAVs equipped with both wireless routers and cameras). The problem is modelled and solved using constrained-based local search on a communication model based on graph theory.

In (Cesare et al. 2015) a multi-robot exploration algorithm based on multiple behaviours is proposed. Quad-rotors are asked to explore and map an indoor zone with unreliable communication and limited battery life. Robots are enabled to change roles both dynamically

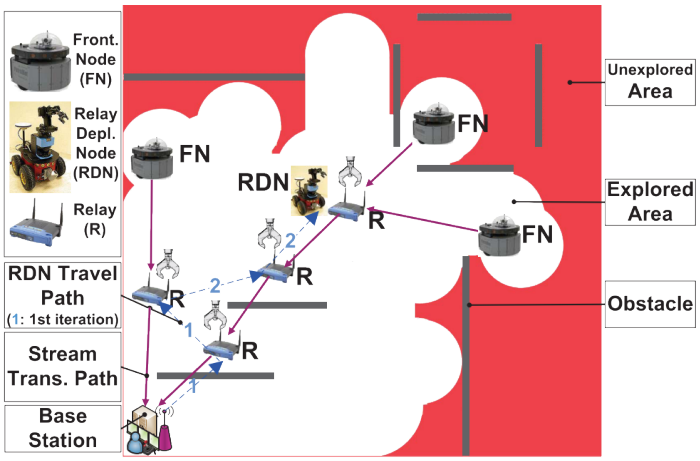



**Figure 1.12:** Heterogeneous multi-robot system (Pei and Mutka 2012). A relay dispenser is supporting the exploration activity by adding relays, accordingly. Explorer robots do not care about connectivity at all.

according to intrinsic and extrinsic factors (e.g. boundaries/distances and battery level) and hierarchically in order to explore and avoid collision among each other. Remaining battery level is considered in order to avoid losing gathered information. Quad-rotors are also able to leave the network but after a fixed period they search for regaining connectivity. Relay robots are designated to forward information from/to the more distant robots improving communication between team members. Although no optimal relay placement is computed, the existence of relays is crucial in the proposed scheme. The state diagram proposed for the adaptive exploration algorithm is depicted in Fig.1.13.

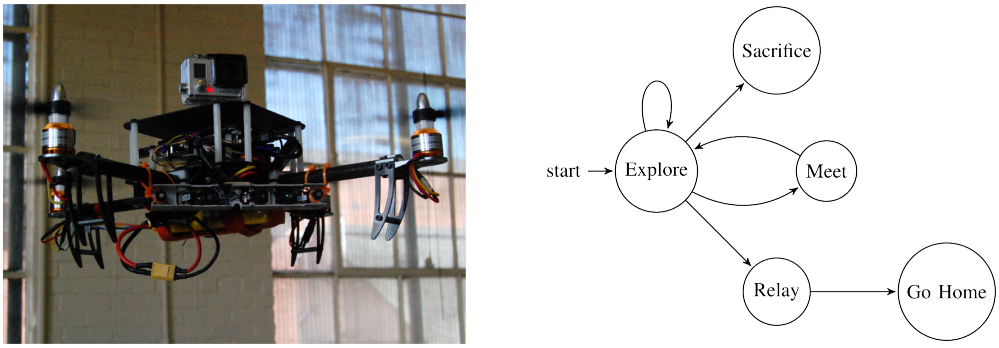



**Figure 1.13:** Multi-role-based exploration strategy. (Cesare et al. 2015) (a) Unmanned aerial vehicle (UAV). UAVs can "sacrifice" themselves by continuing to explore even when they do not have sufficient battery for returning to the base station. (b) State diagram. It concerns different behaviours that adjust for limitations on communication and battery life.

In (Nestmeyer et al. 2017) the exploration problem is addressed ensuring a time-varying connected topology in *3D* cluttered environments but following a decentralised control strategy which enables simultaneous multi-task exploration. In Fig.1.14a the four possible motion

behaviours are shown as well as their transitions while Fig.1.14b shows how the robots can explore and continuously maintain connectivity.

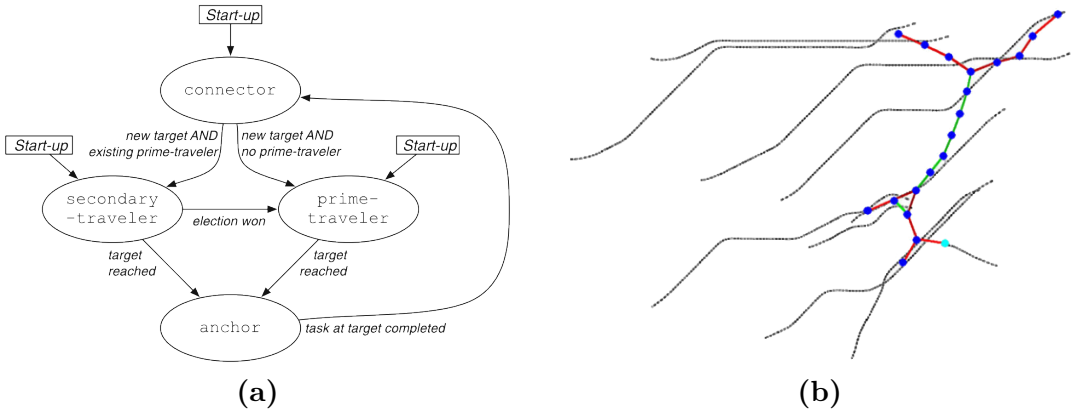
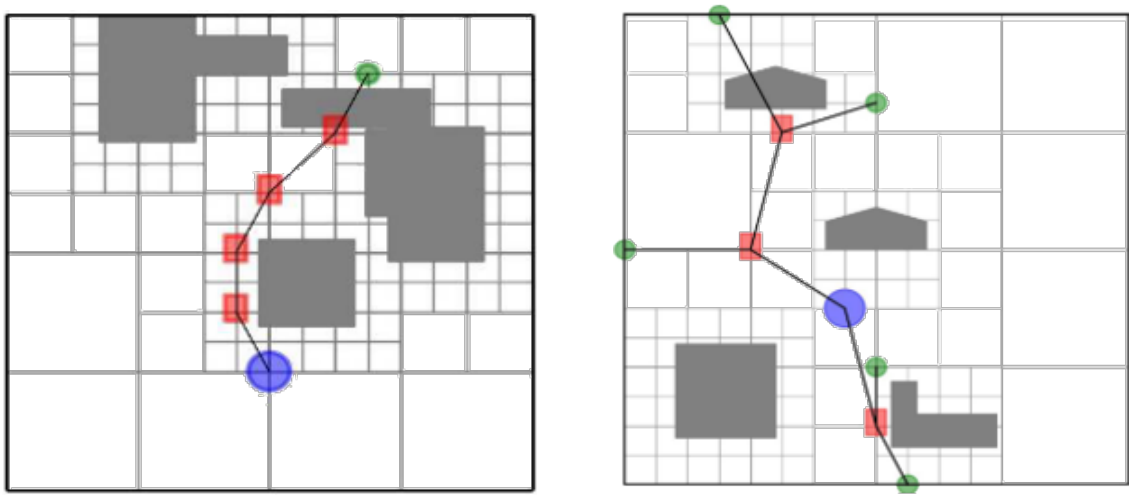



**Figure 1.14:** Multi-role-based exploration strategy (Nestmeyer et al. 2017) (a) State machine diagram of the planning algorithm of the robots. (b) Snapshot of 20 UAV simulation. Dotted black curves represent planned paths; blue dots represent the robots and line segments represent the communication links between them (green is well connected, and red is close to disconnection).

In (Magán-Carrión et al. 2017; Magán-Carrión et al. 2016) the relay node dynamic repositioning problem is tackled. The proposed solution relies on optimisation procedures and evolutionary algorithms to find the best relay locations and how the robots should move to these points. The authors follow a centralised multi-stage approach where one node is in charge of computing the best assignment regarding both connectivity and throughput.

In (Rahman et al. 2017), the problem of how to connect one or more remote units to a base station investing a limited number of intermediate relay robots in constrained communication environments is investigated. The authors study the complexity of the optimal relay placement problem and propose methodologies to create chains or trees of relays as required by different static scenarios. By contrast, in changing environments static solutions cannot be successfully applied because the location optimality does not hold over time. Different examples of static scenarios are depicted in Fig.1.15.

Another centralised but asynchronous strategy is followed in (Banfi et al. 2018),(Banfi et al. 2016) in order to address the problem of multi-robot exploration under recurrent connectivity. In these works, the authors leverage a variant of the Steiner tree problem that appears as a particular case of different known graph optimisation problems. Robot placement is treated as an optimisation problem through Integer Linear Programming. Exact and approximated algorithms are compared on particular scenarios.



**(a)** **(b)**

**Figure 1.15:** Multi-relay exploration strategy (Rahman et al. 2017) (a) Multi-relay chain. Four relay robots are connecting a unit to the base station. (b) Multi-relay multi-unit tree. Three relay robots are connecting several units to the operator.

## 1.3 Conclusions

Some conclusions arise from this brief survey of recent works. Firstly, it is remarkable that despite being the most restrictive class of exploration algorithms, the exploration strategies based on continuous connectivity are prevalent in applications where real-time image streaming are needed (e.g. search and rescue), or simply when human operators at the base station need to enforce timely information updates, or even when a high level of coordination is needed (i.e. when globally shared knowledge between robots is assumed). Additionally, robustness is also highly appreciated in hostile or inaccessible scenarios. In these missions, fault-tolerance is typically achieved adding redundancy (e.g. systems that guarantee k-connected time-varying network topologies) and employing distributed systems.
Nevertheless, when these strong requirements do not condition the mission, the event-based connectivity –that is less restrictive than the former concerning the fleet mobility– seems to be more appropriate.

Now, moving up from essential aspects as communication to the top of the software architecture stack. There exists a large set of distributed reactive and behaviour-based proposals. Compared to the centralised approaches, distributed approaches have the advantage of not presenting the single-point-of-failure weakness. However, in many cases it suffers from deadlocks at the individual or collective level.
Market-based coordination methods represent another popular option. There exists a wide variety of implementations that mainly differ from each other in the way the bids are computed by the robots (e.g. single-item or multiple-item auctions). These difference are not insignificant and typically trade simplicity and computational efficiency off for proper coordination and local optima avoidance. Besides, since each auction involves a period of synchronicity

between robots, fully asynchronous market-based systems have no place. Nevertheless, asynchronous systems may be advantageous over the ones which periodically ask the robots to wait for others before making a decision.

Finally, in communication-restricted environments, there seems to be a general agreement on the benefits of spreading out the fleet as long as the robots can regain connectivity in disconnection case. From this, and trying to well balance these potentially opposed goals, some multi-objective utility-based approaches have been proposed.
Also, defining multiple roles (including communication relays) has demonstrated to be a worthy strategy to address the multi-robot exploration problem when communication restrictions are present. All surveyed multi-role approaches –in one way or another– start from accepting that as a generalisation of the Euclidean Steiner tree problem, the minimum relay placement is NP-hard (Krupke et al. 2015). Hence, it is natural either to try exact approaches only for small and particular instances of the problem or to try approximated approaches instead (e.g. based on meta-heuristics).

In conclusion, the survey suggests that in the context of decentralised systems there is room to try new ideas related to connectivity-regaining policies and rendezvous places. On the one hand, the event-based connectivity framework imposes the execution of connectivity-regaining actions in the presence of some events. On the other hand, rendezvous-based approaches imply the definition of particular meeting points where robots have to meet in order to regain connectivity. Leaving apart the fact that the selection of these places could be a hard issue itself, once the connectivity-regaining action is triggered and the meeting place is known by robots, they should interrupt its exploration plans deviating from its current trajectories in order to accomplish the new goal. This action probably leads to global time performance degradation and individual energy consumption increasing. However, what would happen if robots are only influenced to keep or recover connectivity at all times instead of being demanded to regain connectivity? Furthermore, what would happen if they are free to meet by chance, having been motivated to stay close but without having to meet at specific places? This thesis tries to answer these research questions from the development of a decentralised multi-objective approach where the robots, when selecting their targets, are always considering the opportunity cost of keeping connected or regaining connectivity, implicitly. Besides, in reconnection cases, the location of the selected task becomes the meeting point itself avoiding deviations from planned paths.

The following chapters aim to state the exploration problem to be tackled in this document as well as to introduce two novel exploration strategies that address these questions as their main ingredients.

Chapter 2

# Problem statement

Contents



This chapter defines the instance of the multi-robot exploration problem, which constitutes the basis for the proposals formulated in this work. All particular assumptions are mentioned throughout the following sections. Firstly, the environment, robot and communication models are defined. Namely, some real communication constraints are taken into account and formalised into the model. A *task* definition is given as well as the task identification method. Finally, the global exploration objectives are stated.

## 2.1 Environment model

Gaussian processes autonomous mapping and exploration for range-sensing mobile robots

The environment $E$ is defined as a bounded planar workspace $E \subseteq \mathbb{R}^2$ previously unknown. Besides, $E$ is represented by an occupancy grid structure (Elfes 1989) where each cell $c$ can belong to three different probabilistic states $S = \{f, o, u\}$, standing for free, occupied and unknown, respectively. Typically, $\mathbb{P}(state(c) = f) = 1 - \mathbb{P}(state(c) = o)$ is assumed. When $|\mathbb{P}(state(c) = f) - 0.5| < \epsilon$ the cell $c$ is labelled as *unknown*; otherwise it is labelled as *free* or *occupied*, accordingly. These states represent all possible theoretical situations in which a point of the environment can be classified over time. The mapping algorithm frequently updates the probability value of each cell on each robot. Despite this, only the current classification of each cell at a given decision time step is considered. Consequently, the

representation of $E$ belongs to the domain of matrices $S^{m \times n}$. Furthermore, the region already explored $E_{known}$ and the remaining that is yet unexplored $E_{unknown}$ at time $t$ may be defined from this representation as follows: $E_{unknown}(t) = \{c \in E \mid |\mathbb{P}(state(c,t) = f) - 0.5| < \epsilon\}$ and $E_{known}(t) = \{c \in E \setminus E_{unknown}(t)\}$.

## 2.2 Robot model

Given a robot team $R = \{R_1, R_2, \ldots, R_M\}$ consisting of $M$ homogeneous circular rigid mobile robots with wireless communication capabilities, a traditional representation defines each robot: $R_i = (x_i, y_i, \theta_i, r_i, s_i, c_i)$ where $i \in [1..M]$ and $X_i(t) = \{x_i(t), y_i(t), \theta_i(t)\}$ represents the configuration vector of the robot $i$ at time $t$ (position of its centre and heading with respect to the inertial frame), $r_i$ represents the radius of the robot body and $s_i, c_i$ represent the sensory capabilities as maximum radius of sensing and maximum range of communication, respectively.

## 2.3 Communication model

This model aims to support the connectivity awareness ability of robots needed to deal with disconnection situations during the exploration. Given the position of their teammates and obstacles, robots can estimate the connectivity degree of a specific location considering some of the communication constraints that are widely present in real scenarios, mainly indoor (e.g. office-like and buildings).
The signal strength function[1] $\Gamma_i : \mathbb{N} \times S^{m \times n} \times \mathbb{R} \to \mathbb{R}$ is defined as follows:

$$\begin{aligned}
\Gamma_i(j, E_{known}(t), t) &= \Gamma_i^0 - d_{Att}(i,j,t) - w_{Att}(i,j,E_{known}(t),t) \\
\Gamma_i^0 &= 10 \cdot D_{af} \cdot log_{10}(c_i/r_i) \\
d_{Att}(i,j,t) &= 10 \cdot D_{af} \cdot log_{10}(d_i(j,t)/r_i) \\
d_i(j,t) &= \|X_i(t), X_j(t)\|_2 \\
w_{Att}(i,j,E_{known}(t),t) &= \begin{cases} w_i(j, E_{known}(t), t) \cdot W_{af} & \text{if } w_i(j, E_{known}(t), t) < C \\ C \cdot W_{af} & \text{otherwise} \end{cases}
\end{aligned} \tag{2.1}$$

where, $d_{Att}$ and $w_{Att}$ stand for *distance attenuation* and *wall attenuation* terms, respectively. In addition, $d_i(j,t)$ represents the Euclidean distance between two robot locations at time $t$: typically the transmitter $(X_i(t))$ and receiver $(X_j(t))$, $w_i(j, E_{known}(t), t)$ represents the number of walls[2] present in the known region between transmitter and receiver locations at time $t$, $D_{af}$ represents a distance attenuation factor and $W_{af}$ represents a wall attenuation factor. Finally, $C$ represents the maximum number of walls up to which the $W_{af}$ factor causes

[1] $\Gamma_i$ represents a slight adaptation of the signal strength function presented in (Bahl and Padmanabhan 2000)

[2] Robots cannot distinguish between different kind of rigid obstacles but the term *wall* is used for simplicity and in order to be consistent with the underlying proposal.

a significant effect in function $\Gamma_i$. When $w_i(j, E_{known}(t), t) \geq C$, the distance attenuation effect dominates. Finally, note that in (Bahl and Padmanabhan 2000) the independent term $\Gamma_i^0$ is suggested to be either derived empirically or obtained directly from the wireless network device specification. Nevertheless, in this work the model is adapted in order to become independent from specific deployments (communication devices), deriving the $\Gamma_i^0$ value so that the signal strength $\Gamma_i(j, E_{known}(t), t) = 0$ when $d_i(j, t) = c_i$.
In Fig. 2.1 the shape of the function $\Gamma_i$, as well as the attenuation effects caused by both distances and walls, are plotted.

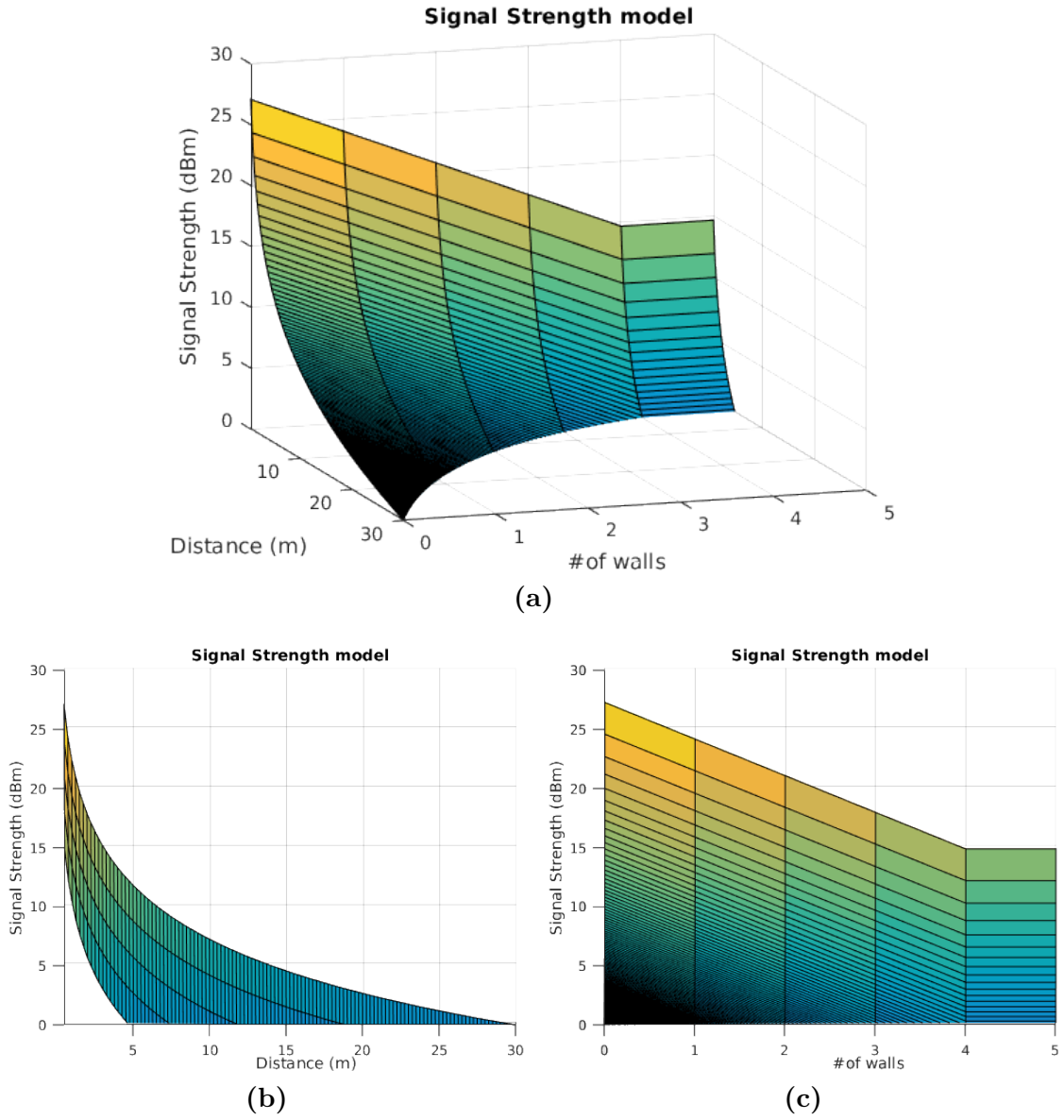



**Figure 2.1:** Behaviour of the signal strength model. (a) The signal strength function $\Gamma_i$ (dBm) is plotted for a [0..5] range of walls and [0..30] (m) range of distances. (b) Attenuation caused by distance. (c) Attenuation caused by wall interference.

Unfortunately, due to uncertain and incomplete knowledge, the $\Gamma_i$ function only can either

confirm the absence of connectivity or deliver an optimistic estimation of connectivity level instead. Although this model represents a valuable improvement in relation to others (e.g. the classic disk or line of sight models (Amigoni, Banfi, and Basilico 2017)), for the sake of simplicity other impairments also common in communication (e.g. bandwidth, information losses, fading and multi-path propagation phenomenon (Caccamo et al. 2017; Fink, Ribeiro, and Kumar 2013)) are not considered in this work.

## 2.4 Task Identification method

Given that the lack of knowledge is essentially inherent to exploration missions, the best choice for the robots is to visit the places where the gain of information can be potentially higher. Consequently, the task identification problem is addressed following a frontier point approach (Yamauchi 1998) where the *free* cells (cf. Section 2.1) that belong to a frontier are over labelled as *frontier points (FP)*. Besides, the resulting set of *FP* cells is clustered (using procedures such as *K-Means* (MacQueen 1967) or *Affinity Propagation* (Frey and Dueck 2007)) in order to identify the cells that better represent each frontier, defining a set of tasks[3] $T = \{T_1, T_2, \ldots, T_N\} \mid T_j \in \mathbb{R}^2, \forall j \in \{1 \ldots N\}$. Thus, $T$ represents at each moment, the smallest set of promising locations that the robots could be interested in visiting to explore all frontiers. In Fig.2.2 these *task* cells are coloured in yellow.

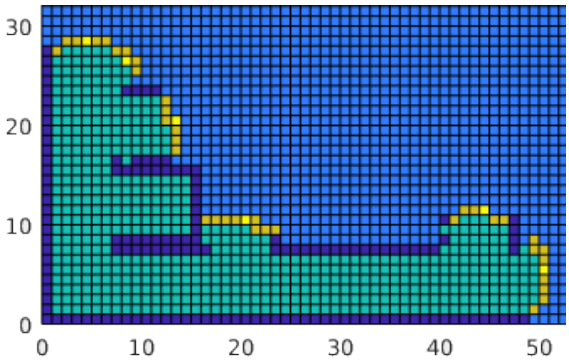



**Figure 2.2:** Frontier points. The different cell types are identified according to the following colour code: dark blue cells are **Obs**tacles, light blue cells are **Unk**nown, green cells are **Free**, orange cells are **FP** cells, and yellow cells are *tasks*.

## 2.5 Global Objectives

The exploration aims for the full coverage of a bounded indoor environment, a priori totally unknown, with a team of terrestrial robots, in minimal time and avoiding isolation situations as much as possible. In this context, isolation refers to the fact of being unconnected from any

[3]In the remainder of the document, the terms *task* and *target* are used indistinctly.

other fleet member. Therefore, in this thesis, the multi-robot system is designed to address these objectives from the following definitions.

### 2.5.1 Full coverage

Given the $E_{known}$ and $E_{unknown}$ previously defined in Sec.2.1, it is possible to claim that the completion condition is reached when $E = E_{known}$ or equivalently $E_{unknown} = \emptyset$. Although this condition is straightforward, it is useless in practice. Alternatively, the completion condition is conceived considering the sensing activity of the robots over time. Let $sen_i(t) = sen(X_i(t))$ be the information gathered by the robot $i$ at time $t$ in the configuration $X_i(t)$. From this $E_{known}$ at completion time $\mathcal{T}$ is defined as follows:

$$E_{known} = \bigcup_{i=1}^{M} \bigcup_{t=0}^{\mathcal{T}} sen_i(t) \tag{2.2}$$

Finally, the completion condition may be written as in (2.3) implying that there are no reachable configurations where any robot can gather new information.

$$\nexists X_i \mid sen_i \bigcap E_{unknown} \neq \emptyset \tag{2.3}$$

### 2.5.2 Completion time optimisation

Additionally to full coverage, the multi-robot system is asked to perform the exploration in minimal time. Therefore, from (2.3), the minimal completion time condition can be expressed as:

$$\min \mathcal{T} \mid \nexists X_i(\mathcal{T}), sen_i(\mathcal{T}) \bigcap E_{unknown} \neq \emptyset \tag{2.4}$$

### 2.5.3 Isolation avoidance

In multi-robot exploration missions the individual isolation situations (when a robot becomes unconnected from any other) are non-desirable. The key motivations to avoid them are *i*) When robots are unconnected they have no possibilities to coordinate their actions hence they could visit the same regions. Therefore, keeping the fleet connected is a way to decrease inefficiency. *ii*) Damages or inner failures during isolation periods can lead to information losses. Therefore, keeping the fleet connected is also a way to decrease the risk of re-work and to prevent time performance degradation, consequently.
Thus, in addition to (2.4), the last of possible individual disconnections should be minimised. To this end, concepts of graph theory are borrowed in order to model a time-varying network topology of mobile robots. Such network is represented by means of an undirected graph defined as $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ where the nodes $\mathcal{V} = \{1 \dots M\}$ represent the robots $R_i \mid i \in [1..M]$ and the edges $\mathcal{E}(t) = \{i, j \mid i, j \in \mathcal{V}, j \in \mathcal{N}_i(t)\}$ represent the operative communication links between any pair of robots $(R_i, R_j)$.

The function $\mathcal{N}_i(E_{known}(t), t) = \{j \mid \Gamma_i(j, E_{known}(t), t) > 0\}$ computes the neighbours of a robot $i$ at time $t$. From this is possible to define the isolation situations of any robot $i$ like the periods when the corresponding node $i$ has no incident edges ($degree(i) = 0$). Furthermore, isolation situations may repeat several times along the exploration.
In Fig.2.3 an example of an exploration timeline concerning disconnections is depicted.


**Figure 2.3:** Disconnection events representation. The disconnection events $dE_k$ can appear distributed along the exploration timeline. Its last is variable and depends on the movements realised by the fleet during the exploration. The starting and ending times of each disconnection are represented by the timestamps $t^i_{sk}$ and $t^i_{ek}$, respectively.

From this model, the expression for the disconnection last optimisation may be obtained by the equation (2.5):

$$\min \sum_{i \in \mathcal{V}} \sum_{k} \Delta dE_k \tag{2.5}$$

where : $k$ indexes the disconnection events $dE$
$\Delta dE_k = t^i_{ek} - t^i_{sk}$, represents the last of the disconnection $dE_k$
$t^i_{sk} = \min t_k \mid \mathcal{N}_i(E_{known}(t), t_k) = \emptyset$, represents the starting time of the disconnection $dE_k$
$t^i_{ek} = \max t_k \mid \mathcal{N}_i(E_{known}(t), t_k) = \emptyset$, represents the ending time of the disconnection $dE_k$

## 2.6 Closing statements

In this chapter, a particular instance of the multi-robot exploration problem has been defined. It is delimited by both the models employed and the goals to be pursued. Next, in Chapter 3, a novel multi-objective based approach that takes these previous definitions as its working assumptions is introduced.

Chapter 3

# Auto Adaptive Multi-Objective Task Selection Approach

## Contents



In this chapter, a novel multi-objective based approach for multi-robot exploration missions is introduced. In order to make the system robust and efficient, a decentralised and asynchronous coordination mechanism is defined. An auto-adaptive multi-objective task utility function is defined in accordance with both the objectives of the exploration problem defined in Sec.2.5 and the task identification method presented in Sec.2.4. Its primary purpose is to integrate travelling costs and connectivity levels finding solutions with a right balance between the benefit of visiting the closer targets and the usefulness of keeping the team connectivity level as high as possible.

Given that the lack of knowledge is inherent to exploration tasks, the best choice for the robots is to visit the places where the gain of information can be potentially higher. Besides, gaining information is the only way to conclude the exploration task. Therefore, the connection between path-cost-based target selection strategies and the completion time performance obtained resides in the fact that this way the fleet expand its territorial knowledge potentially faster.

Nevertheless, when communication is not ideal, there exists a good reason to do something else. The key motivations in considering communication constraints are *i*) when robots are unconnected they have fewer possibilities to coordinate their actions hence they could visit the same regions unnecessarily. Thus, keeping them connected is a way to decrease inefficiency. *ii*) in the presence of damages or inner failures the exploration strategy should take those events into account preventing the need of re-exploration.

Furthermore, to make the system more flexible, an analytic approach through which the relative importance of each goal is set independently of the scenarios is followed. As a result, an auto-adaptive procedure –where the human operator is asked to use his application field expertise in order to influence the robot decisions defining a criterion to balance the importance of both objectives– is developed. Several proofs of correctness on such a procedure are conducted demonstrating that the robots are always capable of auto-adapt the objectives weight to select the tasks accordingly with the human-operator criterion.

## 3.1 Task utility function

This function will guide the optimal task distribution search regarding well-balanced solutions where both the travelling cost and the team connectivity level are considered to evaluate the current targets. The objectives are implemented using utility functions such as *i*) *path* utility function takes the travelling costs to deliver a notion of how beneficial –concerning distance– are the tasks under consideration. *ii*) *connectivity* utility function gives to the robots the connectivity awareness ability.
The *task* utility function $\Phi_i : [0,1] \times T \times R^M \times S^{m \times n} \rightarrow [0,1]$, is defined as follows:

$$\Phi_i(\alpha, T_j, R, E_{known}) = \alpha \cdot \Psi_i(T_j, E_{known}) + \beta \cdot \Omega_i(T_j, R) \tag{3.1}$$

s.t.

$$i \in [1..M] \mid M = |R|$$
$$j \in [1..N] \mid N = |T|$$
$$\alpha + \beta = 1 \mid \alpha, \beta \in [0,1]$$

Given the current state of the fleet $R$ and the current environment knowledge $E_{known}$, the function $\Phi_i$ estimates the utility obtained by a robot $R_i$ in case of selecting the task $T_j$. The current fleet state refers to both the location of the assigned tasks in case of assigned robots and the robot positions otherwise. The terms $\Psi$ and $\Omega$ represent *path* utility and *connectivity* utility functions, respectively. The weights $\alpha$ and $\beta$ work as tuning parameters that permit to adjust the kind of solutions the system will search for. If $\alpha = 1$ during the whole exploration, then the system would only intent to spread out the fleet. On the contrary, if $\alpha = 0$ then the system would always search for potentially fully-connected solutions. Otherwise, when $(0 < \alpha < 1)$ the system will balance both *path* utility and *connectivity* utility. As a result, sometimes the robots could choose other tasks than the closest to favour the team connectivity level. This possibility is deeply analysed further below in Sec.3.2.

Although in this double-objective function the symbol $\beta$ could be substituted by $1-\alpha$, it is preserved for the sake of generality: if the weighted sum had more than two terms, it would not be possible to express all weights as $\alpha$ functions.

### 3.1.1 Path Utility

Path utility measures the relative effort needed for a robot to reach a task from its current location. The *path* utility function $\Psi_i : T \times S^{m \times n} \rightarrow [0,1]$ is defined as follows:

$$\Psi_i(T_j, E_{known}) = 2 \left( \frac{\overline{\Delta} - \Delta_i(T_j)}{\overline{\Delta}} \right)^{\gamma} - 1 \tag{3.2}$$

s.t.

$$i \in [1..M] \mid M = |R|$$
$$j \in [1..N] \mid N = |T|$$

where:

$$\overline{\Delta} = \overline{d} - \underline{d}$$
$$\overline{d} = \max \left\| X_i, T_j \right\|_{sp}, \forall j$$
$$\underline{d} = \min \left\| X_i, T_j \right\|_{sp}, \forall j$$
$$\Delta_i(T_j) = \left\| X_i, T_j \right\|_{sp} - \underline{d}$$
$$\left\| X_i, T_j \right\|_{sp} = \min_{\substack{wp_k \in E_{known} \forall k \in \{1 \ldots e\} \\ wp_1 = X_i, wp_e = T_j}} \sum_{k=1}^{e-1} \left\| wp_k - wp_{k+1} \right\|_2$$

Given the current environment knowledge $E_{known}$, the function $\Psi_i$ estimates the path utility obtained by a robot $R_i$ in case of selecting the task $T_j$. The parameter $\gamma$ works as a shaping factor that could be used to tune the relation between distance and utility. The ordered sequence of waypoints $wp_k$ represents the shortest path between the robot configuration $X_i$ and the target $T_j$. All segments $(wp_k, wp_{k+1})$ are safe given that they are always built regarding only the collision-free pathways present in the known region $E_{known}$. The wavefront propagation method proposed by (Bautin, Simonin, and Charpillet 2013) is employed to determine the way point sequence. The shape and behaviour of the $\Psi$ function are depicted in Fig. 3.1.

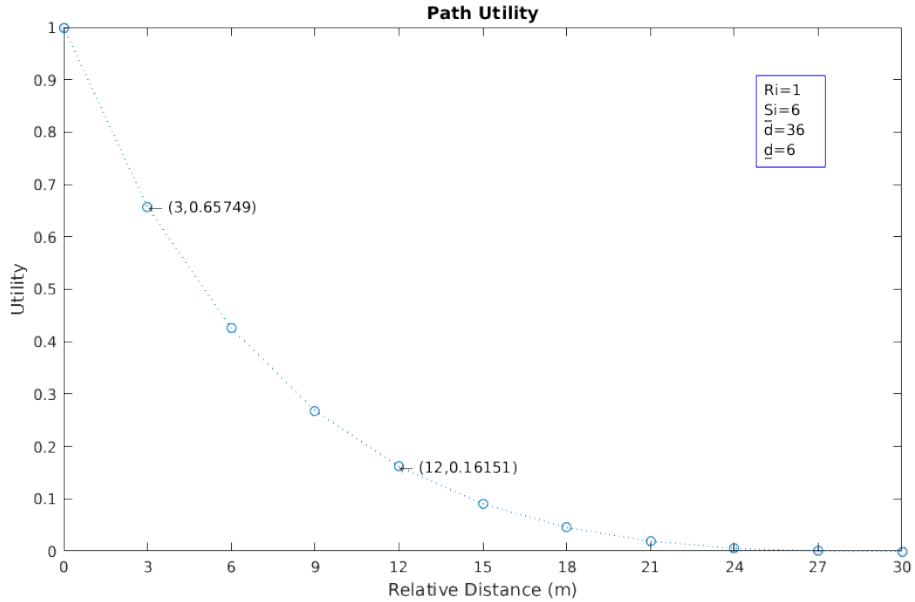



**Figure 3.1:** Path utility function behaviour. There are several tasks in the scene (blue circles). The closest is located 6m away from the robot while the furthest is 36m far away. The closest and furthest tasks always return 1.0 and 0.0, respectively.

### 3.1.2 Connectivity Utility

Connectivity utility computes, optimistically, the connectivity level present in a location at certain moment. The *connectivity* utility function $\Omega_i : T \times R^M \times S^{m \times n} \rightarrow [0, 1]$ is defined as follows:

$$\Omega_i(T_j, R, E_{known}) = \frac{log_2\left((2^{\rho} - 1) \cdot \frac{|\mathcal{N}_i(E_{known}(t), t)|}{M - 1} + 1\right)}{\rho} \tag{3.3}$$

s.t.

$$i \in [1..M] \mid M = |R|$$
$$j \in [1..N] \mid N = |T|$$

Given the current state of the fleet $R$ and the current environment knowledge $E_{known}$, the function $\Omega_i$ estimates the connectivity utility obtained by a robot $R_i$ in case of selecting the task $T_j$. Particularly, it is interesting to do so concerning the arrival time to $T_j$. The current fleet state refers to both the location of the assigned tasks in case of assigned robots and the robot positions otherwise. The parameter $\rho$ works as a shaping factor that could be used to tune the relation between connectivity level and utility. Note that the utility is decreasing in the number of robots, and may favour the adoption of *Mobile Ad-Hoc Networks* connectivity techniques. In such networks, messages travel from source to destination members in more than one hop, where intermediate nodes forward messages until the destination is reached. The shape and usefulness of the $\Omega$ function may be appreciated in Fig. 3.2.

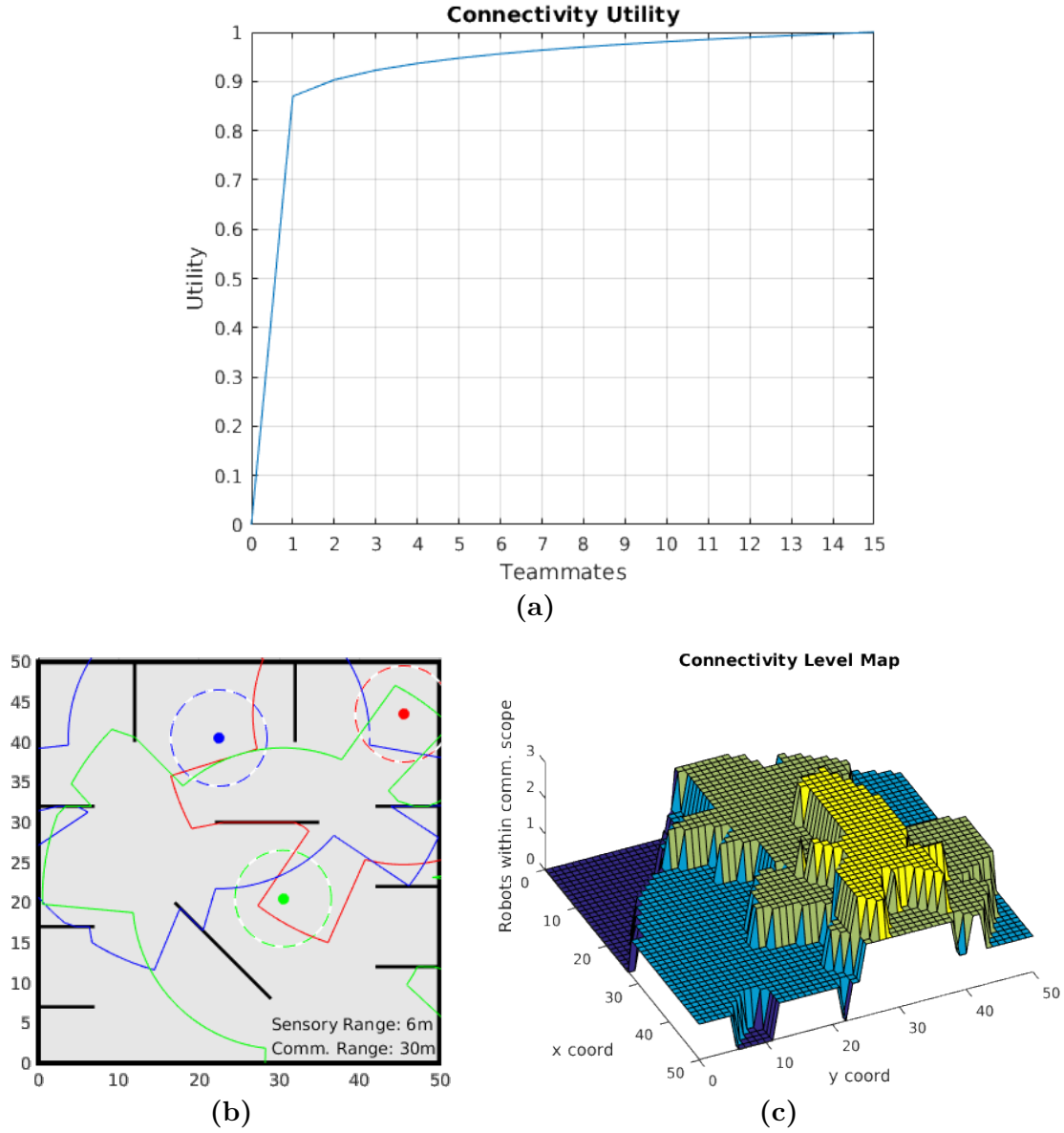



**Figure 3.2:** (a) Connectivity utility function shape. (b) A scene where the usefulness of the Ω function can be appreciated. Three robots (coloured dots) and several walls were arranged simulating an ongoing exploration process. Robots are surrounded with two lines of the same dot colour indicating sensory (dashed) and communication ranges (solid). (c) Shows the connectivity level map corresponding to (b). Therefore, as long as the will of another robot is to keep connected with the fleet, it would be able to take this perspective into account when deciding where going to.

### 3.1.3 General Considerations

The definition of multi-objective weights is usually accomplished as an empirical matter. Typically, a search process is run in order to find – after a lot of trials – values that fit some optimal criteria. This kind of methods is typically used when the parametric function is planned to be used many times. However, in the exploration context, this assumption or even

the possibility of running trials are frequently out of the question. It is not possible to assume that all scenarios where the exploration will be conducted will be similar between each other and, for this reason, is neither possible to assume that the best $\alpha$ and $\beta$ values can remain unchangeable.
Furthermore, when these procedures are followed, at the end of the training stage it is often tough to associate the resultant parameter values with real aspects of the problem (e.g. performance metrics like time, distances, energy or even connectivity levels). This lack of understanding may, in turn, wrongly influence the fine tuning of such parameters without rerunning a portion of trials. Taking those shortcomings into account, an analytic approach –through which the $\alpha$ and $\beta$ values might be set independently of the scenarios– is explored.

## 3.2 Adaptive $\alpha$-value computation

When a multi-robot exploration process is going to run under communication constrained conditions, choosing between *only exploring* or *exploring preserving connectivity level* is a crucial decision. The first choice would be suitable when either connectivity is out of the question, or it is impossible for a robot to keep connected and explore at once. In such a case, connectivity does not play any role in the decision-making process. On the contrary, the second choice is suitable when it is necessary to interleave high-performance exploration (minimising the total exploration time) and acceptable connectivity level (avoiding robot isolation as much as possible).
To this end, the human operator is let to use his application field expertise in order to influence the robot decision –defining a criterion to balance the importance of both objectives– by merely setting a parameter before the exploration starts.

Therefore, since $\alpha$ and $\beta$ parameters determine the behaviour of the robots concerning target selection, two questions come up: i) how can the value of those parameters be defined in order to ensure the applicability of the human-operator criterion along the exploration process? ii) should these values be adapted during the exploration process?

Henceforth, the task selection framework and the human-operator criterion are formalised. Besides, several proofs to demonstrate the existence and correctness of an adaptive $\alpha$-value that makes the robots behave following the criterion mentioned above are conducted.

### 3.2.1 Task selection framework

This process is always made iteratively from a list, comparing the currently best task against the rest, one by one. Therefore, without loss of generality, the most relevant aspects can be studied just analysing all the possible relations between an arbitrary pair of tasks. Regarding the distance to a specific robot location and the connectivity level (number of connections with the rest of the fleet), any task can be classified according to Table 3.1.

Therefore, the meaning of these categories is straightforward: regarding the assignment

**Table 3.1:** Task classification.

| Connectivity \ Distance | Closest (Cl) | Furthest (F) |
|---|---|---|
| Connected (Co) | Cl/Co | F/Co |
| Non-Connected (NC) | Cl/NC | F/NC |

of the fleet, *Co* means that the task location would offer at least the minimum level of connectivity to the robot (i.e. one connection to another fleet member); *NC* means the opposite; regarding the spatial distribution of tasks, *Cl* means that the task under consideration is closest to the robot than any other; *F* means that the task is furthest to the robot than any other task.

Moreover, let $R_i$ a robot and $T_j$ and $T_k$ two tasks such that $class(T)$ can belong to any class defined in Table 3.1. In any scenario, these tasks can be related to each other according to Table 3.2. Given that $T_j$ and $T_k$ are arbitrary tasks, the matrix can be considered symmetric. Thus, taking one of the triangular matrices is enough to study all possible cases.

**Table 3.2:** Possible cases when selecting from two tasks.

| $T_k$ \ $T_j$ | Cl/Co | F/Co | Cl/NC | F/NC |
|---|---|---|---|---|
| Cl/Co | | [F/Co Cl/Co] | | [F/NC Cl/Co] |
| F/Co | [Cl/Co F/Co] | | [Cl/NC F/Co] | |
| Cl/NC | | [F/Co Cl/NC] | | [F/NC Cl/NC] |
| F/NC | [Cl/Co F/NC] | | [Cl/NC F/NC] | |

From the lower triangular, it is possible to identify some cases where one task is better (regarding both path utility and connectivity utility) than the other. Such an example is the [Cl/Co;F/NC] where $T_j$ is closer to the robot than $T_k$, and it is the only one that keeps the robot connected as well. Similarly, in the [Cl/NC;F/NC] case neither task can keep the robot connected, and in consequence, the closest task $T_j$ results more convenient than $T_k$. Thus, in both previous cases, the criterion to choose a task is clear: the closest task should be selected. However, in the other cases, it is not clear at all which task should be selected. On the one case, [Cl/NC;F/Co], whichever selection implies either traversing longer distances or losing connectivity. On the other case, [Cl/Co;F/Co], selecting the closest task $T_j$ ensures traversing the shortest path but could imply losing connectivity. By contrast, selecting the furthest task $T_k$ would be acceptable only when the gain in connectivity oppose the bigger travelling effort.

**Definition 3.1**
*The human operator threshold* HO-Threshold *expresses the human operator criterion through a distance that represents the extra effort made by robots that the human operator is willing to accept in order to maintain or enlarge the size of the robot communication network.*

In other words, the human operator criterion is determined by setting the distance threshold until which the targets that preserve or enlarge connectivity are preferred over the rest, for all robots.

For instance, in the [Cl/NC;F/Co] case the selection will be conditioned as follows: $T_k$ will be selected if and only if the length of the shortest path between $T_k$ and the robot location is less than or equal to *HO-Threshold*. $T_j$ will be selected otherwise.

In order to make the influence of *HO-Threshold* clearer, an example scene is depicted in Fig. 3.3. Note that all tasks are within the *HO-Threshold*, but only $T_3$ is able to enlarge the connectivity level of the robot $R_1$. Thus, applying Definition 3.1 leads to the selection of task $T_3$ because it enables the robot $R_1$ to travel more distance to gain connectivity. On the contrary, whether the *HO-Threshold*$\leq 3$, $T_3$ would be no longer preferred over the rest, and consequently the closest task $T_2$ would be selected instead.

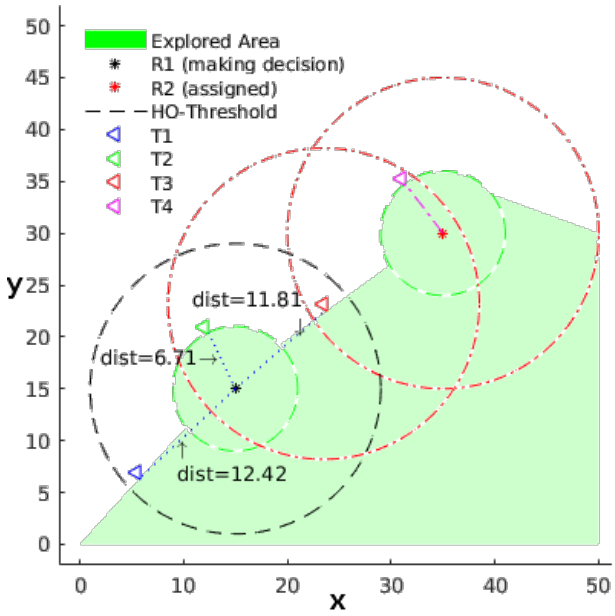



**Figure 3.3:** Two robots are carrying out an exploration mission. The communication and sensory ranges are drawn around the robots with red and green dashed lines, respectively. It is assumed that $R_2$ has already chosen the task $T_4$ whereas $R_1$ is still selecting from $T_1, T_2$ and $T_3$. Dotted lines are used to show the sight-line between $R_2$ and the tasks. The corresponding Euclidean distance is also shown. *HO-Threshold* is set to 6.

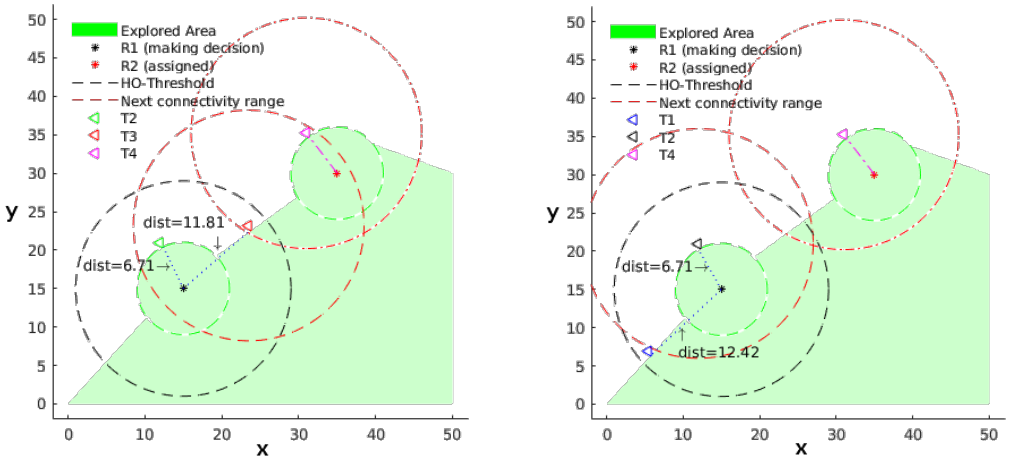
Hence, in the presence of some specific conditions, it is expected that the application of the *HO* criterion can make the fleet more cohesive than following approaches that do not take communication constraints into account and less restrictive than the ones that do not permit

disconnections or force re-connections as well.

Next, the proofs of correctness and existence of $\alpha$ (and $\beta$) values that implement the *HO* criterion are conducted regarding the cases present in the lower triangular of Table 3.2. The cases {[Cl/Co,F/NC];[Cl/NC,F/NC]} are considered at first while the remaining {[Cl/Co,F/Co];[F/Co,Cl/NC are considered afterward.

### 3.2.2 [Cl/Co,F/NC] and [Cl/NC,F/NC] cases

In Figs.[3.4a,3.4b] two instances of these cases are depicted, respectively.




**(a)** Cl/Co,F/NC case **(b)** Cl/NC,F/NC case

**Figure 3.4:** Two robots are carrying out an exploration mission. It is assumed that $R_2$ has already chosen the task $T_4$ whereas $R_1$ is still making its decision. The communication and sensory ranges are drawn around the robots with red and green dashed lines, respectively. Dotted lines are used to show the sight-line between $R_2$ and the tasks. The corresponding Euclidean distance is also shown. (a) [Cl/Co,F/NC] case: robot $R_1$ is selecting from targets $T_2$ that is the closest and keeps it connected and $T_3$ that is the furthest and cause a disconnection. (b) [Cl/NC,F/NC] case: robot $R_1$ is selecting from targets $T_1$ –the closest– and $T_2$ –the furthest–, giving that both targets cause a disconnection.

**Proposition 3.2.1.** *When $T_j$ and $T_k$ belong to $[Cl/Co, F/NC]$ or $[Cl/NC, F/NC]$, the values of $\alpha$ and $\beta$ do not make any difference in the selection process.*

*Proof.* This claim can be derived directly from the following facts:

- in the [Cl/Co,F/NC] case the furthest task $T_k$ makes the robot become unconnected and then, applying (3.1) to $T_j$ and $T_k$ leads to:

$$\Phi_i(\alpha, T_k, R) = \alpha \cdot \Psi_i(T_k) \leq \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_i(T_j, R) = \Phi_i(\alpha, T_j, R), \ \forall \alpha \tag{3.4}$$

$$\text{s.t.} \quad \Omega_i(T_j, R) > 0, \Omega_i(T_k, R) = 0$$
$$\Psi_i(T_k) \leq \Psi_i(T_j)$$

- in the [Cl/NC,F/NC] case both tasks make the robot to be unconnected and thus the $\Phi$ function value will depend only on the $\Psi$ term:

$$\Phi_i(\alpha, T_k, R) = \alpha \cdot \Psi_i(T_k) \leq \alpha \cdot \Psi_i(T_j) = \Phi_i(\alpha, T_j, R), \quad \forall \alpha \tag{3.5}$$
$$\text{s.t.} \quad \Omega_i(T_j, R) = 0, \Omega_i(T_k, R) = 0$$
$$\Psi_i(T_k) \leq \Psi_i(T_j)$$

In conclusion, in any of these cases, the task selection is not affected by $\alpha$. □

### 3.2.3 [Cl/Co,F/Co] and [F/Co,Cl/NC] cases

In [Cl/Co,F/Co] case both tasks offer the possibility to be connected. On the contrary, in [F/Co,Cl/NC] case opposite objectives are present: one task is closer but unconnected while the other is connected but further. Thus, the latter case is taken to prove the existence of an $\alpha$, that can respect any given *HO* criterion. The former case is finally used to corroborate the non-existence of any possible unwanted side effect caused by the achieved $\alpha$ expression.

#### 3.2.3.1 [F/Co,Cl/NC] case.

Based on the human-operator criterion (set by a threshold value) we want an $\alpha$-value that makes, following the scenario depicted in Fig.3.5, $T_3$ preferred over $T_2$ if and only if $T_3$ belongs to the circular area defined by the *HO-Threshold*.

Next, the existence of such an $\alpha$ parameter will be demonstrated, and its value will be derived as well.

**Proposition 3.2.2.** *When $T_j$ and $T_k$ belong to [F/Co,Cl/NC] is always possible to find an $\alpha$-value that satisfies the following inequality:*

$$\Phi_i(\alpha, T_j, R) = \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_i(T_j, R) \geq \alpha \cdot \Psi_i(T_k) = \Phi_i(\alpha, T_k, R) \tag{3.6}$$
$$s.t. \quad \Omega_i(T_j, R) > 0, \Omega_i(T_k, R) = 0$$
$$\Psi_i(T_j) \leq \Psi_i(T_k)$$

*Proof.* Let $\Omega_1$ the utility assigned to the fact of being connected with only one teammate. Then, from (3.3) is possible to state that: if $T_j$ belongs to any $[*/Co]$ class, $\Omega_1 \leq \Omega_i(T_j, R), \forall (i, j)$ over time. Moreover, if the number of robots does not change, it is also possible to state that $\Omega_1$ remains invariant over time. Applying this result into (3.6) leads to the inequality presented next in (3.7):

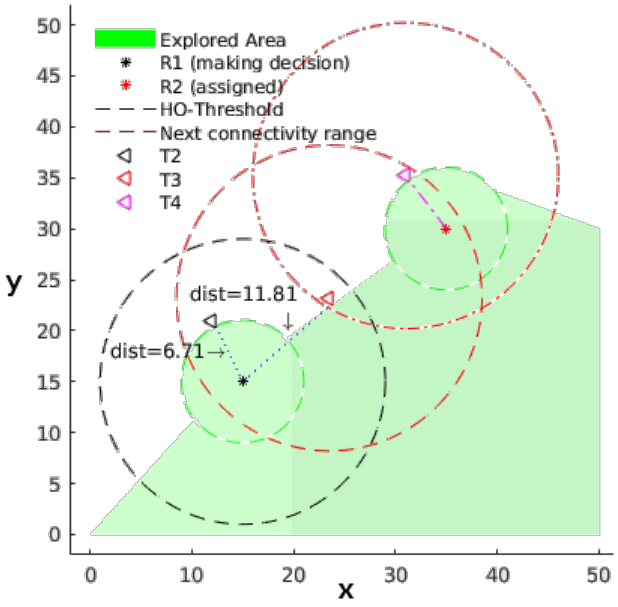



**Figure 3.5:** [F/Co,Cl/NC] case.

$$\begin{aligned}
\Phi_i(\alpha, T_j, R) \geq \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_1 \geq & \alpha \cdot \Psi_i(T_k) = \Phi_i(\alpha, T_k, R) \\
\alpha \cdot \Psi_i(T_j) + (1-\alpha) \cdot \Omega_1 \geq & \alpha \cdot \Psi_i(T_k) \\
\alpha \cdot (\Psi_i(T_j) - \Omega_1) + \Omega_1 \geq & \alpha \cdot \Psi_i(T_k) \\
\frac{\Omega_1}{\Psi_i(T_k) - \Psi_i(T_j) + \Omega_1} \geq & \alpha
\end{aligned} \tag{3.7}$$

Besides, substituting $\Omega_1 = x$ and $\Psi_i(T_k) - \Psi_i(T_j) = u$, equation (3.7) may be rewritten as follows:

$$\begin{aligned}
\alpha \leq \frac{x}{u+x} \implies \alpha \leq inf\left(\frac{x}{u+x}\right) \\
\text{s.t. } 0 < c \leq x \leq 1 \\
0 \leq u \leq 1 \\
\text{given that: } \Omega_1 = c \\
\Psi_i(T_k) \geq \Psi_i(T_j)
\end{aligned} \tag{3.8}$$

From (3.8) is possible to claim the existence of an $\alpha$-value that obey any *HO-Threshold* if and only if, the function $\frac{x}{u+x}$ presents an absolute minimum on the domain:

$$D = \{(x, u) \mid 0 < c \leq x \leq 1, 0 \leq u \leq 1\}.$$

This fact can be stated employing *Weierstrass* theorem[1]. Besides, the minimum point might be calculated analysing both: *i*) the relative extrema; *ii*) the points lying on the border of $D$. Following this procedure is possible to find the absolute extreme of the function $\frac{x}{u+x}$ in $(x, u) = (c, 1)$.
Moreover, it is remarkable that this extreme represents a place where the most demanding conditions are reached: task $T_j$ presents the lowest positive connectivity utility, and the distance between both tasks is the largest. Hence, the existence of a positive value $\alpha \leq \frac{\Omega_1}{1+\Omega_1}$ (regardless of how demanding can be the distance relation between tasks) that might alter the task selection in favour of connectivity has been demonstrated. □

Nevertheless, in (3.7) $\alpha$ is independent of the *HO-Threshold*. Consequently, its direct application would result in a strictly connectivity-guided exploration, where tasks that offer connectivity are always preferred over the rest no matter how far they are. Therefore, to relate it with an *HO-Threshold* the value of the term $\Psi_i(T_j)$ in (3.7) must be substituted by the utility of being *HO-Threshold* far from the robot, say $\Psi_i(T_{\text{HO}})$. Next, the value of the term $\Psi_i(T_k)$ is substituted by 1 since $\Psi_i(T_k) = 1$ represents the necessary condition to reach the extreme coordinate $u = 1$ that arose from (3.8). Finally, the expression for an *HO-Threshold* dependent $\alpha$, say $\alpha_{\text{HO}}$, is expressed in (3.9) as follows:

$$\alpha_{\text{HO}} = \frac{\Omega_1}{1 - \Psi_i(T_{\text{HO}}) + \Omega_1} \tag{3.9}$$

**Proposition 3.2.3.** *The applicability of the $\alpha_{HO}$ referred in* (3.9) *causes that any task within the threshold scope that also offers any positive connectivity level would be favoured over the rest of tasks that do offer none connectivity level regardless how close to the robot they are.*

*Proof.* $\Phi_i(\alpha, T_j, R) \geq \Phi_i(\alpha, T_k, R)$ is imposed to any tasks $(T_j, T_k)$ that respect the [F/Co,Cl/NC] conditions:

$$\begin{aligned}
\Phi_i(\alpha, T_j, R) = \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_i(T_j, R) &\geq \alpha \cdot \Psi_i(T_k) = \Phi_i(\alpha, T_k, R) \\
\alpha \cdot \Psi_i(T_j) + (1 - \alpha) \cdot \Omega_i(T_j, R) &\geq \alpha \cdot \Psi_i(T_k) \\
\alpha \cdot (\Psi_i(T_j) - \Omega_i(T_j, R)) + \Omega_i(T_j, R) &\geq \alpha \cdot \Psi_i(T_k) \\
\frac{\Omega_i(T_j, R)}{\Psi_i(T_k) - \Psi_i(T_j) + \Omega_i(T_j, R)} &\geq \alpha
\end{aligned}$$

Then, applying (3.9) leads to (3.10):

[1] A function $f$ has an absolute extreme if it is continuous and its domain is compact.

$$\frac{\Omega_i(T_j,R)}{\Psi_i(T_k)-\Psi_i(T_j)+\Omega_i(T_j,R)} \geq \frac{\Omega_1}{1+\Omega_1-\Psi_i(T_{\mathrm{HO}})}$$
$$\Psi_i(T_j) \geq \frac{\Omega_i(T_j,R)}{\Omega_1}\cdot(\Psi_i(T_{\mathrm{HO}})-1)+\Psi_i(T_k) \tag{3.10}$$
$$\Psi_i(T_j) = sup\left(\frac{\Omega_i(T_j,R)}{\Omega_1}\cdot(\Psi_i(T_{\mathrm{HO}})-1)+\Psi_i(T_k)\right)$$

Since *i)* $\Omega_1$ is constant. *ii)* $\frac{\Omega_i(T_j,R)}{\Omega_1} \geq 1$ *iii)* $(\Psi_i(T_{\mathrm{HO}})-1) \leq 0$, it is possible to conclude that:

- $\Psi_i(T_j)$ is monotonically decreasing with respect to $\Omega_i(T_j,R)$.
- the upper bound is reached when:
  (a) $\Psi_i(T_{\mathrm{HO}}) = 1$
  (b) $0 \leq \Psi_i(T_{\mathrm{HO}}) < 1$, $\Omega_i(T_j,R) = \Omega_1$ and $\Psi_i(T_k) = 1$.

Please note that (a) is out of the proposition conditions. Instead, (3.10) can be rewritten imposing (b), leading to:

$$\Psi_i(T_j) \geq \frac{\Omega_1}{\Omega_1}\cdot(\Psi_i(T_{\mathrm{HO}})-1)+1)$$
$$\Psi_i(T_j) \geq \Psi_i(T_{\mathrm{HO}})$$

What is true if, and only if, $\Delta_i(T_j) \leq$*HO-Threshold*, which is indeed what the human operator would like to get from his criterion application to tasks within the *HO-Threshold*. Hence, following (3.9) under the [F/Co,Cl/NC] conditions is always possible to compute an $\alpha_{\mathrm{HO}}$-value that makes the robots behave following the human-operator criterion. □

Likewise, it is important to highlight that the $\alpha_{\mathrm{HO}}$-value needs to be calculated every time a robot is ready to make a decision. This need for adaptation arises from $\Psi_i(T_{\mathrm{HO}})$, which is not constant. Its value depends on the relation between the *HO-Threshold* and the relative distance to the current furthest task. That way, the robots can autonomously adapt the weights of the *task* utility function according to the changing conditions of the environment in order to be always consistent with the human-operator criterion.

#### 3.2.3.2 [Cl/Co,F/Co] case.

This analysis is devoted to checking the applicability of the $\alpha_{\mathrm{HO}}$ when the conditions to achieve a good trade-off between path cost and connectivity level are less demanding than in the [F/Co,Cl/NC] case. In [Cl/Co,F/Co] case, although one task is closer than the other, the differences in the positive connectivity level offered by them could make the furthest task more attractive than the closest. From that, considering the connectivity level offered by the closest, two cases may be identified: *i)* When $T_j$ offers a higher level of connectivity than $T_k$. In such a case, there is no doubt that independently of the $\alpha_{\mathrm{HO}}$ value, the selection would

always favour the task $T_j$ because it is the closest as well. *ii*) On the contrary, when $T_k$ offers a higher level of connectivity than $T_j$, the selection of $T_k$ will depend on both how distant from robot it is and how much more connected would be the robot on $T_k$ respect to $T_j$.

Finally, to show that the $\alpha_{\text{HO}}$ value does not introduce any unwanted side effect on the task selection process when tasks belong to the [Cl/Co,F/Co] case, it is needed to prove that it neither contradicts the first case nor restricts the occurrence of the second case.

**Proposition 3.2.4.** *In the presence of two tasks subject to the [Cl/Co,F/Co] case conditions, if $T_j$ is the closest and the one which provides the highest level of connectivity at once, then the application of the $\alpha_{HO}$ value will never result in the selection of $T_k$.*

*Proof.* By contradiction, it is assumed that under these conditions the selection could be in favour of $T_k$, implying that the following inequality holds:

$$\begin{aligned} &\Phi_i(\alpha, T_j, R) = \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_i(T_j, R) \leq \alpha \cdot \Psi_i(T_k) + \beta \cdot \Omega_i(T_k, R) = \Phi_i(\alpha, T_k, R) \\ &\alpha \cdot (\Psi_i(T_j) - \Psi_i(T_k)) + \beta \cdot (\Omega_i(T_j, R) - \Omega_i(T_k, R)) \leq 0 \end{aligned} \tag{3.11}$$

Which implies that, independently of the $\alpha_{\text{HO}}$ value, the terms $(\Psi_i(T_j) - \Psi_i(T_k))$ and $(\Omega_i(T_j, R) - \Omega_i(T_k, R))$ should not be positive simultaneously. Thus, either $(\Psi_i(T_j) \leq \Psi_i(T_k))$ or $(\Omega_i(T_j, R) \leq \Omega_i(T_k, R))$. However, this contradicts the hypothesis where $T_j$ is stated as the closest and the one which provides the highest level of connectivity at once and accordingly; the proposition has been demonstrated. □

**Proposition 3.2.5.** *In the presence of two tasks subject to the [Cl/Co,F/Co] case conditions, if $T_j$ is the closest and $T_k$ the one which provides the highest level of connectivity, then the application of the $\alpha_{HO}$ value will never be conclusive concerning the task selection.*

*Proof.* The relation between the utility of tasks is written as follows in (3.12):

$$\begin{aligned} &\Phi_i(\alpha, T_j, R) = \alpha \cdot \Psi_i(T_j) + \beta \cdot \Omega_i(T_j, R) \lesseqgtr \alpha \cdot \Psi_i(T_k) + \beta \cdot \Omega_i(T_k, R) = \Phi_i(\alpha, T_k, R) \\ &\alpha \cdot (\Psi_i(T_j) - \Psi_i(T_k)) \lesseqgtr \beta \cdot (\Omega_i(T_k, R) - \Omega_i(T_j, R)) \\ &\alpha \cdot (\Psi_i(T_j) - \Psi_i(T_k)) \lesseqgtr (1 - \alpha) \cdot (\Omega_i(T_k, R) - \Omega_i(T_j, R)) \\ &\alpha \lesseqgtr \frac{\Omega_i(T_k, R) - \Omega_i(T_j, R)}{(\Omega_i(T_k, R) - \Omega_i(T_j, R)) + (\Psi_i(T_j) - \Psi_i(T_k))} \end{aligned} \tag{3.12}$$

Substituting $(\Omega_i(T_k, R) - \Omega_i(T_j, R)) = x$ and $(\Psi_i(T_j) - \Psi_i(T_k)) = u$ it is possible to state that in order to favour the selection of $T_j$ the inequality (3.13) must be held, otherwise the (3.14):

$$\alpha \geq \frac{x}{u+x} \implies \alpha = sup\left(\frac{x}{u+x}\right) \tag{3.13}$$

$$\alpha \leq \frac{x}{u+x} \implies \alpha = inf\left(\frac{x}{u+x}\right) \tag{3.14}$$

$$\begin{aligned} \text{s.t. } & 0 \leq x \leq 1 \\ & 0 \leq u \leq 1 \\ \text{given that: } & \Omega_i(T_k, R) \geq \Omega_i(T_j, R) \\ & \Psi_i(T_j) \geq \Psi_i(T_k) \end{aligned}$$

On this domain, the function $\frac{x}{u+x}$ present an absolute maximum equal to 1 in the point $(x, u) = (1, 0)$ and absolute minima equal to 0 along the line segment defined by $(x, u) = (0, u)$. Assessing the $\alpha_{\text{HO}}$ expression derived in (3.9) with $(0, u)$ leads to (3.15) and (3.16), respectively:

$$0 = \frac{\Omega_1}{1 + \Omega_1 - \Psi_i(T_{\text{HO}})} \quad \therefore \quad 0 = \Omega_1 \tag{3.15}$$

$$\begin{aligned} 1 &= \frac{\Omega_1}{1 + \Omega_1 - \Psi_i(T_{\text{HO}})} \\ 1 - \Psi_i(T_{\text{HO}}) &= 0 \quad \therefore \quad \Psi_i(T_{\text{HO}}) = 1 \end{aligned} \tag{3.16}$$

From which, while the condition expressed in (3.15) is reached when $|R| \to \infty$, the one expressed in (3.16) is reached when *HO-Threshold* tends to 0. The condition (3.15) is unreachable in practice implying that no $\alpha_{\text{HO}}$ can make the task $T_k$ always preferred over $T_j$. Conversely, the condition (3.16) is reachable if, and only if, the human operator deliberately does not want to care about connectivity. Otherwise, there is no positive $\alpha_{\text{HO}}$-value that can make the task $T_j$ always preferred over $T_k$.

Consequently, when $\alpha_{\text{HO}} \in (0..1]$ under the [Cl/Co,F/Co] conditions, it is not possible to hold a single preference over time. □

### 3.2.4 Considerations and usefulness

In order to establish the task selection criterion, the human operator only needs to choose the extra distance *HO-Threshold* –according to his expertise and knowledge– he is willing to ask the robots to travel in order to keep or enlarge the connectivity level of the fleet. Once the *HO-Threshold* is set, robots are capable of selecting tasks consistently with the *HO* criterion following Equation (3.9). Furthermore, it is important to note that the *HO-Threshold* value does not change along the exploration but, as was pointed out, the $\alpha_{HO}$ does, due to the dependency on the $\Psi$ function. This explains the need for auto-adaptive capabilities concerning the multi-objective $\Phi$ function.

Additionally, it also worth noticing that setting *HO-Threshold*$= \infty$ it is a practical way to implement an event-based connectivity approach where the tasks that provide connectivity will always be preferred over the rest no matter how close they are.

## 3.3 Task allocation scheme

The allocation scheme is founded on two pillars: the coordination method and the task selection algorithm.

### 3.3.1 Coordination method

In order to take advantage of the individual computing power of the robots, to avoid the single point of failure and to deal better with the presence of real communication constraints during the exploration, a decentralised approach is followed. Typically, estimation of travelling costs and target benefits, as well as mapping and localisation, are the tasks chosen to be made locally by the robots. However, to achieve a cooperative behaviour, both the local map and localisation information must be shared among team members.

Additionally, the relation between $|T|$ and $|R|$ can result in two quite different behaviours. *i*) if $|T| < |R|$, not all robots would be needed to reach all targets. Some robots may choose to keep quiet. *ii*) when $|T| \geq |R|$ all robots would be needed in order to reach the maximum amount of targets at a time. When robots decide to explore, the task selection is made coordinately. Robots coordinate their actions implicitly, sharing specific information (such as locations, eventually already-done-selections and local maps) and running the same selection algorithm. Thus, it is possible for the multi-robot system to compute a coordinated-tasks-to-robots distribution in a decentralised way (Bautin, Simonin, and Charpillet 2011; Bautin and Simonin 2012; Hollinger and Singh 2012).

To do so properly, the exchanging information time is carefully set up. The system is fully asynchronous, meaning that: *i*) Robots do not wait for others. *ii*) After selecting a task, the robots do exchange their selection in order to prevent future overlappings. *iii*) Local maps and –by this mean– the sets of new available tasks are periodically exchanged, each time two conditions are met: *1*) A waypoint of the planned path is reached. *2*) New information has been gathered. *iv*) Localisation data is exchanged at a higher rate than maps because its influence on the task selection algorithm is higher too.
While localisation data is exchanged periodically, the rest of data exchanging is triggered by events instead. These policies make the system more efficient and flexible because *i*) No data is transmitted when there is no new information to exchange. *ii*) There is no need to set up any rate parameter when exploring different environments. The robot life-cycle algorithm is sketched in Alg.1.

**Algorithm 1** Robotic Agent Life-cycle algorithm.

1: **function** EXPLORE(i,R,HO-Threshold)
▷ i stands for the robot position in vector R.
▷ R stands for the robots location vector.
2: $atT \leftarrow true$ ▷ *atTarget* flag.
3: $pose \leftarrow R[i]$
4: $gMap \leftarrow getMap(pose)$ ▷ Occupancy Grid map.
5: **while** true **do**
6: $R^* \leftarrow \emptyset$ ▷ Vector of connected robot locations.
7: **for** $j \in R \wedge j \neq i$ **do**
8: **if** $\Gamma_i(j, gMap) > 0$ **then** ▷ Connected robot.
9: $R^*[j] = rcvPose(j)$ ▷ Asking for localisation data.
10: $sndPose(pose, j)$ ▷ Sending own localisation data.
11: $gMap = mapMerge(gMap, rcvMap(j))$ ▷ Asking for local maps.
12: **end if**
13: **end for**
14: **if** $atT$ **then**
15: $T = getFrontierTasks(gMap)$ ▷ Tasks location vector.
16: $task \leftarrow getAssignment(i, R^*, T, |R|, \text{HO-Threshold})$
17: $goto(task)$
18: **end if**
19: $pose = getPose()$ ▷ Global localisation.
20: $atT \leftarrow pose = task$
21: $[gMap, ni] = mapMerge(gMap, getMap(pose))$ ▷ Mapping.
22: **if** $atT \vee ni$ **then** ▷ $R_i$ arrives at *task* or new information was gathered.
23: **for** $j \in R^* \wedge j \neq i$ **do**
24: $sndMap(gMap, j)$ ▷ Sending local map.
25: **end for**
26: **end if**
27: **end while**
28: **end function**

### 3.3.2 Task selection algorithm.

The task selection process employs the multi-objective utility function $\Phi$ defined in (3.1) with $\alpha_{\text{HO}}$ values dynamically adapted by (3.9). The corresponding algorithm is sketched in Alg. 2.

Firstly, the input parameter $R^*$ specifically corresponds to the locations of the teammates currently connected with the robot $R_i$. Next, in line 2 and 3, both the *task* and *robot* location sets are split up into two subsets each one (assigned and unassigned items, respectively). Line 4 is in charge of taking only the unassigned tasks that are within the *HO-Threshold* scope from every robot. Afterwards, from lines 5 until 9, the path utility matrix is computed regarding all possible task-robot combinations. Next, lines 10 and 11 aim to compute the $\alpha_{\text{HO}}$ and $\beta$ values according to (3.9). The set of tasks-to-robots distributions is calculated

**Algorithm 2** Task selection algorithm.

1: **function** GETASSIGNMENT($i, R^*, T, M$, HO-Threshold)
▷ i stands for the robot position in vector $R^*$.
▷ $R^*$ stands for the robots location vector.
▷ T stands for the current tasks location vector.
▷ M stands for the fleet size.

2: $[T^u, T^a] \leftarrow T$
3: $R^u \leftarrow R^*$
4: $T^{\text{HO}} \leftarrow \{T_j \mid T_j \in T^u, \Delta_k(T_j) \leq \text{HO-Threshold}, \forall R_k \in R^u\}$ ▷ Relative distance $\Delta_k(T_j)$ is defined in Sec.3.1.1.
5: **for** each $k$ in $R^u$ **do**
6: **for** each $j$ in $T^{\text{HO}}$ **do**
7: $PU[k, j] = \Psi_k(T_j)$ ▷ $\Psi$ matrix.
8: **end for**
9: **end for**
10: $\alpha_{\text{HO}} \leftarrow \dfrac{\Omega_1}{1 - \Psi_i(\text{HO-Threshold}) + \Omega_1}$ ▷ (3.9)
11: $\beta = 1 - \alpha_{\text{HO}}$
12: $N^{\text{HO}} = |T^{\text{HO}}|$
13: $M^u = |R^u|$
14: $T2RDist \leftarrow Ar^{N^{\text{HO}}}_{M^u}$ ▷ Tasks-to-robots distributions $T2RDist \in \mathbb{N}^{|Ar^{N^{\text{HO}}}_{M^u}| \times M^u}$
15: **for** each row $r$ in $T2RDist$ **do**
16: $\Phi[r] = \sum_{c=1}^{M^u} \alpha_{\text{HO}} \cdot PU[i, c] + \beta \cdot \Omega_i \left(T^{\text{HO}}_c, [R^u, T^a]\right), i = T2RDist(r, c)$
17: **end for**
18: $T^* \leftarrow T2RDist[r, i] \mid \arg\max_r \Phi[r]$
19: **return** $T^*$
20: **end function**

from line 12 to 14. Finally, from line 15 to 17 all possible assignments are evaluated using the $\Phi$ function while the task corresponding to robot $i$ of the best assignment is selected in line 18.

Some considerations on Alg. 2 are hereafter discussed. Concerning the computation of the set of tasks-to-robots distributions (lines 12 to 14), it provides a way to potentially avoid falling in local minima or even taking wrong decisions. Note that the *connectivity utility* function is subject to locality conditions and thus, it is not possible to compute optimal distributions from the application of iterative polynomial-time assignation algorithms such as the *Hungarian method* (Wurm, Stachniss, and Burgard 2008).

On the contrary, Alg. 2 can choose the optimum tasks-to-robots distribution by evaluating all possible $T^{\text{HO}}$-to-$R^u$ distributions. Nevertheless, this process may be potentially very hard since $|Ar^{N^{\text{HO}}}_{M^u}| = \frac{N^{\text{HO}}!}{(N^{\text{HO}} - M^u)!} = \Pi_{m=1}^{n}(N^{\text{HO}} - m + 1) = \Pi_{m=0}^{n-1}(N^{\text{HO}} - m) \rightarrow O(N^{*M^u})$. Therefore, the smaller $|T^{\text{HO}}|$ and $|R^u|$ the faster the algorithm will run. In the first case $|T^{\text{HO}}|$ is bounded by pruning $|T^u|$ with the help of *HO-Threshold*.

On the contrary, even being naturally bounded ($|R| \geq |R^*| \geq |R^u|$), the set $R$ could imply a large $R^u$. Besides, all efforts are to keep the fleet connected as much as possible, leading to $|R^*| \to |R|$. Fortunately, in a fully asynchronous multi-robot system the probability of two or more robots being simultaneously making a decision is negligible.

Finally, note that Alg. 2 assumes $|T^{\mathrm{HO}}| > |R^u|$; otherwise the tasks-to-robots distribution cannot be computed. In such a case, the input parameters are managed in order to conduct a robots-to-tasks distribution instead. In turn, $|Ar_{N^{\mathrm{HO}}}^{M^u}|$ does not represent a significant effort since $M^u \geq N^{\mathrm{HO}}$ holds for small values.

## 3.4 Closing statements

In this chapter, a novel multi-objective based approach for multi-robot exploration missions was described in detail. All existence and correctness proofs conducted on the task selection procedure support the fact that the robots are always capable of auto-adapting the objectives weights of the *task utility* function in order to select the tasks accordingly with the human-operator criterion.
Besides, since these weights are set up independently of any scenario, the system is expected to be more flexible than others concerning its deployment.

Next, in Chapter 4 this approach is assessed and compared with both an ideal-communication-conditions baseline of results and with the performance obtained by other approaches under non-ideal communication conditions.

Chapter 4

# Baseline statement and AAMO approach Results

## Contents



The aims of this chapter are *i*) To establish a baseline on the main figure of merits that will be defined to asses the benefits of different approaches. *ii*) To assess and analyse the performance of different instances of the *Auto-Adaptive Multi-Objective (AAMO)* approach[1] under non-ideal communication conditions. *iii*) To compare *AAMO* instances against other approaches under non-ideal communication conditions.

Regarding the first purpose, the baseline is established regarding two state-of-art approaches so that the simulation runnings concern the comparison between a *Yamauchi*-based algorithm (Yamauchi 1998) and the *minPos* algorithm (Bautin and Simonin 2012) under ideal communication conditions. These algorithms were chosen since they are decentralised, as are the author's proposals; while *Yamauchi* is a reference on exploration and typically serves itself as a comparison baseline, the *minPos* proposal has demonstrated a very good performance,

[1]The *AAMO* approach is introduced in Chap.3 and different instances –from now on– refers to different *HO-Threshold* setup values.

outperforming other important reference algorithms.
On the contrary, regarding the *AAMO* assessment and the comparison with other approaches, the simulation runnings concern exploration missions subject to non-ideal communication conditions. In this case, the primary purpose is to understand how compromised could be the exploration time performance when the connectivity level is prioritised and to reveal possible improvements concerning previous techniques. In consequence, there are experiments which compare only the performance achieved by different instances of *AAMO*, while in other experiments, where relevant, comparison with state-of-art performance is taken into account too.

## 4.1 Simulation setup

All simulations were conducted over *MORSE* physics simulator[2] using *ATRV*-like robots equipped with laser range sensors. The more relevant simulation parameters are shown in Table 4.1.

Furthermore, it is important to precise that except for the *Communication range* that depends on the device, the rest of communication factors were taken from (Bahl and Padmanabhan 2000) in attention to their strong dependency on the materials present in the environment. The values of *HO-Threshold* correspond to 66%, 50% and 33% of the communication range $c_i$, respectively. In all simulations localisation and low-level motion control are taken for granted.

### 4.1.1 Scenarios

Simulations are conducted over synthetic scenarios (See Fig. 4.1) where long distances and obstacle presence may offer similar challenging conditions that would be expected in the real world. The *Loop* and *Cross* scenarios (see Fig.4.1b and Fig.4.1a) were mainly used to confirm the correctness of the implemented solutions and to show the advantages of using a multi-robot approach over a single one. Unfortunately, and caused by the shape and size of the free zones, on those scenarios, there are nearly no possibilities to demonstrate any advantage of the proposed approaches over the others. Finally, the *Maze* scenario (Fig.4.1c), that represents the most challenging environment, was used to establish comparative results among the approaches. These results are further analysed below in Sec.4.3 and Sec.4.4, respectively. Due to the significant amount of collected data, only the values related to *Maze* runnings are summarised and discussed here. Even so, all charts and screen-shots generated from data concerning all three environments are available online: `www.fing.edu.uy/~fbenavid/projects/MuRE/mure.html`.

[2] `www.openrobots.org/morse/doc/stable/morse.html`

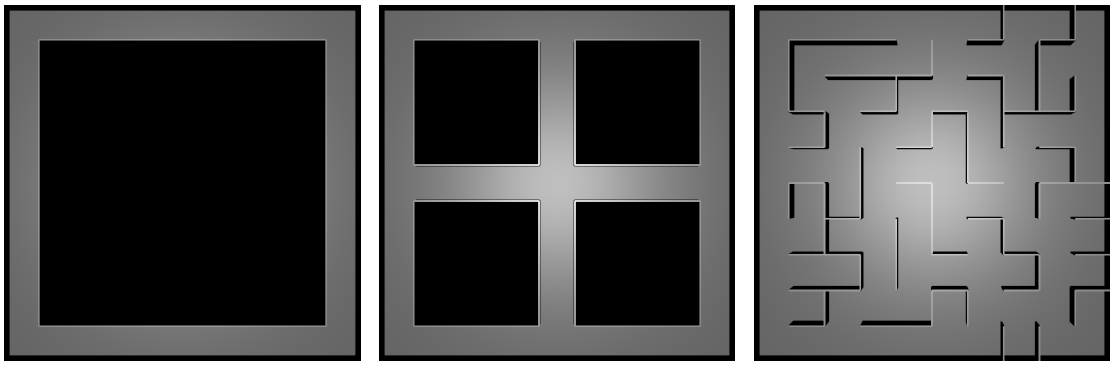
**Table 4.1:** Simulation setup.

| Robot features & capabilities. | |
|---|---|
| Model | *ATRV* |
| Maximum speed ($s_i$) | 1 (m/s) |
| Laser scan window | 360 (°) |
| Laser range | 6 (m) |
| Laser resolution | 3 (°) |
| **Communication parameters.** | |
| Range ($c_i$) | 30 (m) |
| Wall attenuation factor ($W_{af}$) | 3.1 (dBm) |
| Distance attenuation factor ($D_{af}$) | 1.523 |
| C factor | 4 (walls) |
| **Fleet features.** | |
| Heterogeneity | Homogeneous |
| Initial positions | Left Bottom corner |
| **Environment features.** | |
| Terrain | 80x80 ($m^2$) |
| Wall height | 2 (m) |
| Wall thickness | 0.2 (m) |
| Corridor width | 8 (m) |
| **Grid Map features & parameters.** | |
| Mesh | Cartesian grid |
| Cell side | $2r_i$ |
| **AAMO parameters.** | |
| $\gamma$ | 3 |
| $\rho$ | $2 \cdot (\lvert R \rvert - 1)$ |
| HO-Threshold | 20,15,10 (m). |

### 4.1.2 Explorer Robot Architecture

In this section, the software architecture of a robotic explorer agent is presented. In Fig.4.2 the main components are roughly depicted. From a software architecture point of view, each robot is organised in three layers. In turn, each layer is responsible for different aspects grouped by abstraction levels so that the higher layer, the more abstract are the issues which the software components are devoted to.

Going bottom-up in the layer stack, in the first layer the components are in charge of the interaction between the robotic agent and the environment. The *Motion Control* component



(a) (b) (c)

**Figure 4.1:** Benchmark scenarios. All terrains cover an 80x80 $m^2$ flat surface with static obstacles (walls including the outer perimeter). Proposed as benchmarks in (Yan et al. 2015). (a) Loop like scenario. (b) Cross like scenario. (c) Maze like scenario.

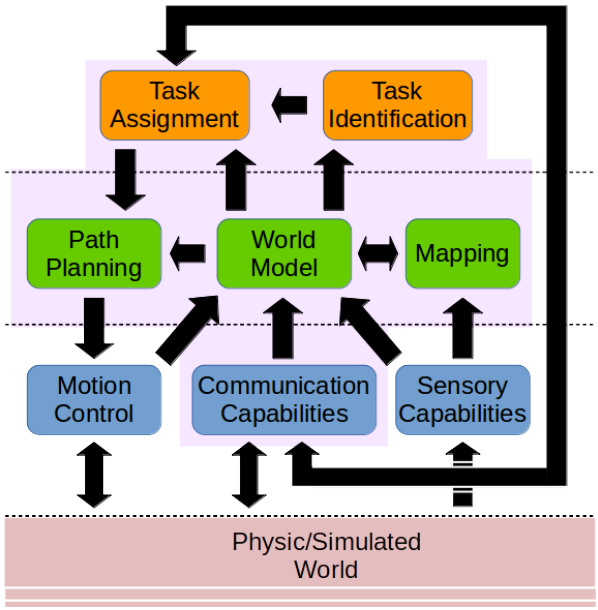



**Figure 4.2:** Explorer Agent Architecture. The first layer includes the software components that represent systems or devices through which the agent can interact with the environment. The second layer includes models and algorithms to keep the models up to date. The third layer includes task identification and task selection algorithms. Components on the shadowed zone were developed entirely during this work.

is taken from the *MORSE*[3] repository and is responsible for controlling the motors. Besides, in this work, the component follows a way-point-based motion strategy. In the *Sensory Capabilities* component are grouped all sensory systems in charge of gathering environmental information. The most relevant information comes from the *Pose* and the *Laser scanner* sensors, also taken from the *MORSE* repository. From the *Pose* sensor it is possible to know

[3] *MORSE* physics simulator `www.openrobots.org/morse/doc/stable/morse.html`

the robot configuration $X_i(t) = \{x_i(t), y_i(t), \theta_i(t)\}$ at any time –implementing the localisation capability, while the laser gives an array of distance measurements $z(t)$ from which is possible to build the map of the close surroundings. Finally, the *Communication Capabilities* component is asked to manage every aspect related to communications receiving/sending information from/to (see incoming/outgoing arrows) other team members. In this work, and since only the distance and wall attenuation effects (discarding other sources of perturbation) are considered, the communication is simulated in a straightforwardly directly applying the communication model introduced in Sec.2.3.

The second layer represents the core of the system where the models and algorithms that support the highest level functionalities –namely related to the exploration purpose of the system– are allocated. On the one side, the *World Model* component is in charge of modelling all physic interaction between the robotic agents and its surroundings. By keeping several structures up-to-date (e.g. occupancy grid map, the position of the fleet members, assignment of the fleet members, etc.), it is also able to support foretelling services that would be required for the highest level algorithms. On the other side, *Mapping* and *Path Planning* components are also supported by the *World Model* component since it gives an access point to the mapping structures and the kinematic models as well. The *Mapping* component implements a standard occupancy grid approach (Thrun, Burgard, and Fox 2005) where the posterior of the map is calculated from a collection of separate problems of estimating $p(m_k|z(t), X_i(t))$ for all grid cell $m_i$ and where each $m_i$ has attached to it one of the occupancy values $S = \{f, o, u\}$ (previously defined in Sec.2.1 of Chapter 2). The *Path Planning* component implements the wave-front propagation approach introduced in (Bautin, Simonin, and Charpillet 2013).

Finally, high-level decisions as coordination are taken in the third layer when the task allocation scheme is executed by the *Task Assignment* component. In particular, the arrow between *Task Assignment* and *Communication Capabilities* components represents the exchange of current positions and task assignments from the agent to the fleet and vice-versa.

## 4.2 Figure of merits

The performance of approaches is assessed in terms of the following figures of merit. The first three are very popular and represent the most reliable quality indicators (Amigoni, Banfi, and Basilico 2017). The fourth has been taken from (Pal, Tiwari, and Shukla 2013) and sometimes can be useful to explain the results concerning the first two. The fifth was inspired by (Satici et al. 2013) in order to measure the connectivity quality. Besides, a sixth indicator is proposed here in order to have a better qualitative analysis of the connectivity aspects. Moreover, the connected components of the topology along the exploration are also plotted. The indicators are defined as follows:

- ***T**otal exploration **T**ime (TT)*: time elapsed from the beginning until the end of exploration measured in seconds.
- ***P**ath **L**ength (PL)*: the sum of the distance travelled by each robot measured in meters.

- ***C**overage **R**atio (CR)*: percentage of the accessible terrain covered by the team. Calculated as: $\frac{\text{explored cells} \cdot 100}{\text{accessible cells}}$.

- ***O**ver-**S**ensing cell **R**atio (OSR)*: percentage of cells sensed as new by more than one robot. Calculated as: $\frac{\text{over-sensed cells} \cdot 100}{\text{explored cells}}$.

- ***D**isconnection **L**ast **R**atio (DLR)*: percentage of TT where at least one robot is totally unconnected. Calculated from the Fiedler number corresponding to the network connectivity graph (see Section 2.5.3).

- ***M**aximum **D**isconnection **L**ast **R**atio (MDLR)*: calculated as: $\frac{\text{longest disconnection period} \cdot 100}{TT}$.

## 4.3 Baseline statement

In this section, a baseline of performance on the main indicators is established from the runnings of both *Yamauchi* and *minPos* approaches under ideal communication conditions. Since the exploration problem is expected to be more difficult under non-ideal communication conditions than otherwise (Amigoni, Banfi, and Basilico 2017), the obtained results may be considered as a baseline of the first four indicators –defined before in Section 4.2– with respect to the corresponding performance achieved in runnings conducted under non-ideal communication conditions.

### 4.3.1 Collected data

In order to conduct the assessment and comparison stated above, at least ten realistic software-in-the-loop simulations were executed on the *Maze* scenario presented in Fig. 4.1. All collected data is presented in Table 4.2 and are organised obeying the following scheme. The columns refer to (from left to right): **f**igure of **m**erits (FM); approaches, where $Y$ and $MP$ stand for ***Y**amauchi* and ***MinPos***, respectively; and the fleet size $|R|$. In each fleet size, the average $AVE$ and standard deviation $StD$ values are registered.

**Table 4.2:** Yamauchi and MinPos results under ideal communication conditions on Maze environment.

| FM | | $\|R\|$ = 1 | | 2 | | 3 | | 4 | | 5 | | 8 | | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD |
| TT | Y | 1958 | 121.8 | 1288 | 255.9 | 902 | 128.6 | 791 | 125.6 | 647 | 78.5 | 516 | 41.3 | **459.5** | 41.5 |
| | MP | **1898** | 148.0 | **1044** | 110.1 | **779** | 72.8 | **615** | 52.0 | **505** | 48.8 | **496** | 19.7 | 482 | 37.4 |
| PL | Y | 1308 | 94.0 | 1581 | 157.4 | 1665 | 158.9 | 1831 | 225.7 | 1896 | 186.2 | 2093 | 95.9 | **2294** | 173.9 |
| | MP | **1268** | 59.0 | **1413** | 124.1 | **1420** | 85.4 | **1467** | 94.8 | **1592** | 107.8 | **2053** | 50.3 | 2438 | 140.7 |
| CR | Y | 99.1 | 0.09 | 99.0 | 0.06 | 99.0 | 0.04 | 99.1 | 0.05 | 99.0 | 0.04 | 99.0 | 0.05 | 99.1 | 0.04 |
| | MP | 99.0 | 0.04 | 99.1 | 0.06 | 99.1 | 0.06 | 99.1 | 0.05 | 99.1 | 0.05 | 99.1 | 0.07 | 99.1 | 0.07 |
| OSR | Y | 0.0 | 0.00 | 1.23 | 0.57 | 2.22 | 0.48 | 3.68 | 0.70 | 5.25 | 0.81 | 5.45 | 0.21 | 7.00 | 0.12 |
| | MP | 0.0 | 0.00 | **0.93** | 0.19 | **2.31** | 0.38 | **2.94** | 0.25 | **4.37** | 0.50 | 5.45 | 0.11 | 7.00 | 0.08 |

### 4.3.2 Baseline assessment

We start the analysis highlighting that both approaches can adequately explore all the environments presented above in Section 4.1.1. Coherently, both approaches achieve high levels of *CR*. This can be seen clearer in Fig.4.3.

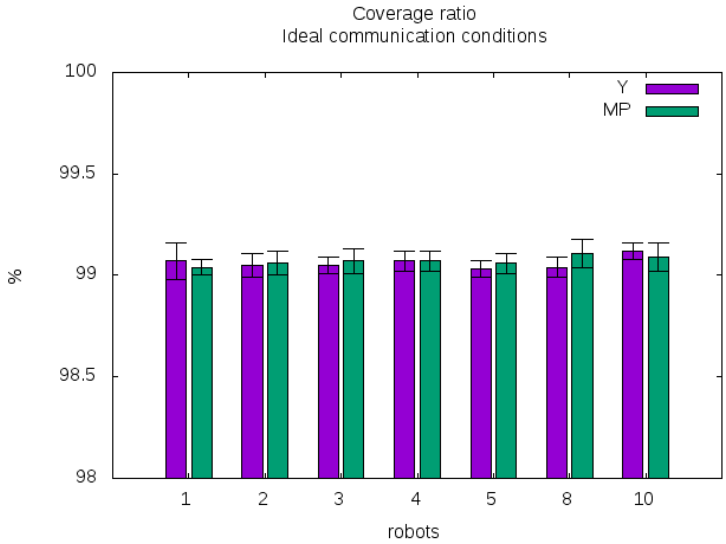



**Figure 4.3:** Coverage ratio (CR) under ideal communication conditions. Both approaches cover more than 99% of the terrain regardless of the fleet size.

Furthermore, the *minPos* approach outperforms *Yamauchi* concerning TT as was expected. However, the most notorious differences of performance are observed on fleets which size is less than or equal to five robots, as can be seen in Fig.4.4.

In crowded environments going from one location to another is often more difficult than in the presence of few robots. Therefore, due to collision avoidance manoeuvres, both approaches show an increasing *PL* when the fleet size increases. This behaviour may be observed in the corresponding chart in Fig.4.5. On the one hand, *Yamauchi* presents a trend with an almost invariant slope along the different fleet size values. On the other hand, under *MinPos*, the trend of *PL* presents a positive but minor slope from one to five-robot-sized fleet after what it becomes very steep.

Hence, the analysis is divided into two cases. Firstly, when fleet size is less than or equal to five robots, *MinPos* is more efficient than *Yamauchi* since both approaches achieve very similar coverage ratios (see Fig.4.3) but in the latter robots need to traverse longer distances than in the former, on average. That is entirely expected since the *Yamauchi* approach does not take care about the dispersion of the fleet as the *MinPos* does and consequently, in the former robots are forced to deal with crowding more frequently than in the latter. This is a remarkable difference given that energy needed to support an exploration mission will be closely related to the distance traversed by robots.
Contrarily, as the fleet size increase beyond five robots, the shape of the scenario and the

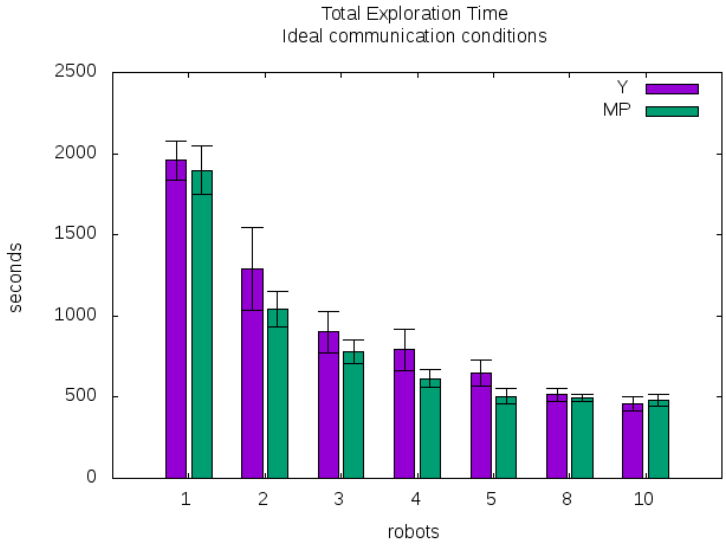



**Figure 4.4:** Total exploration time (TT) under ideal communication conditions. Both approaches show a decreasing trend of TT as the fleet size increase. Nevertheless, the fact that the performance improvements are decreasing suppose the existence of a limit on the benefit of robots adding.

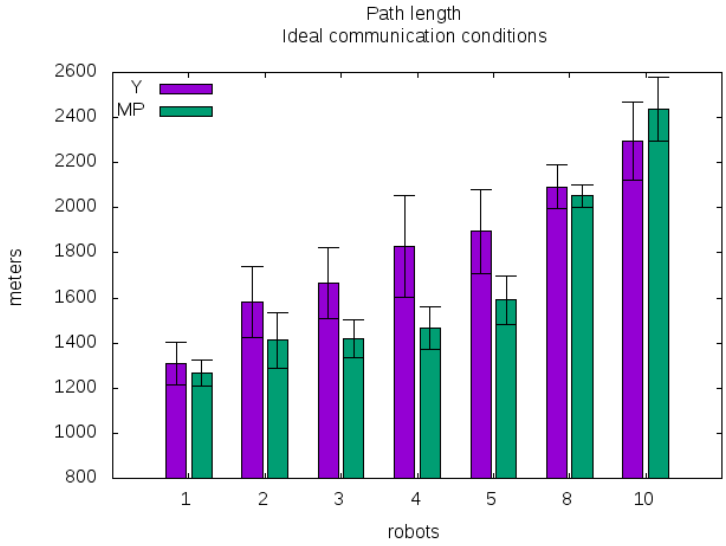



**Figure 4.5:** Path length (PL) under ideal communication conditions. The trend of PL is upward in both cases.

peculiar wall distribution all together seem to make the crowding unavoidable for the *MinPos* approach, causing a severe worsening on its *PL* performance.

Finally, it is interesting to observe the over-sensing-cell phenomenon, because, by observing the amount of rework done by the fleet during exploration tasks, it also gives a good measure of the system efficiency.

In this case we start the analysis pointing that in an ideal world –with perfect communications, perfect sensing and instantaneous actions– there would be no place for over-sensing. Nevertheless, in real-world communications and sensing systems are not perfect but, more important, all actions take time. Even the ones which do not involve motion like sense, compute, communicate actions need some window time to be executed. Therefore, many things can happen simultaneously, e.g. sensing actions conducted on the same objects. In such a case, two or more robots might report the discovery of the same cells.

In conclusion, even under ideal communication conditions, it is possible to register some level of over-sensing, and this level is unavoidable because of the parallel nature of the system. However, it is equally interesting to analyse the over-sensing results: *i*) when the fleets are obeying different policies. *ii*) to have a baseline against which the results obtained under non-ideal communication conditions may be compared.

Backing to the experiments, during the simulation runnings we verify that the most significant over-sensing record is mainly generated at starting steps when all robots are very close to each other (recall that all robots start from the same corner of the scenario, see Table 4.1) and, in consequence, its sensing scopes overlap each other, significantly. In Fig.4.6 the robot placement setup at the starting time is shown.

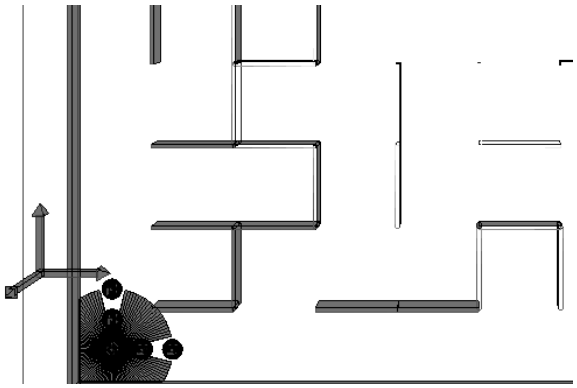


**Figure 4.6:** Robot placement setup at starting time. Robots are represented by black dots. The sensing scope of the robot placed right in the corner is represented by a grey area where it is possible to see the laser aces and the obstruction caused by some teammates. Robots are placed from the corner along the $x$ and $y$ axes. As the fleet size increase, new robots are placed next following the row of robots on each axis, alternately.

Conversely, after this initial period, the robots overlap each other less frequently, and hence the *OSR* remains almost unchangeable over time, in both approaches. Despite this, minor differences may be highlighted. Due to a better fleet distribution on the terrain –that decreases the probability of simultaneous sensing events, the fleet makes slightly less rework under *MinPos* approach than under *Yamauchi* approach (see Fig.4.7).

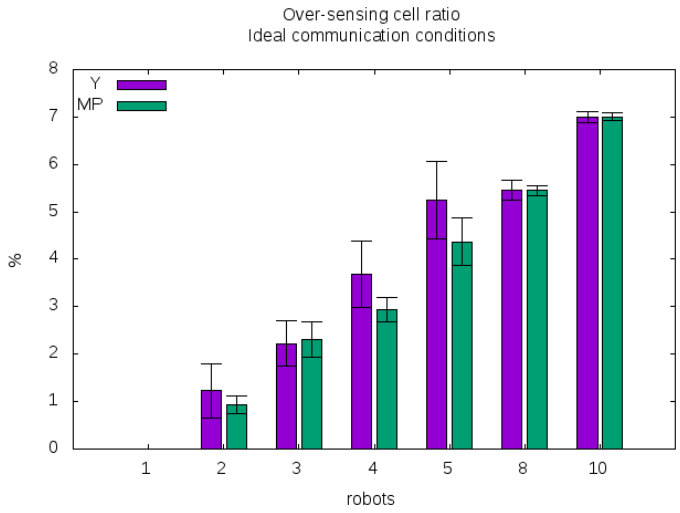



**Figure 4.7:** Over-sensing ratio (OSR) under ideal communication conditions. Shows how as fleet size increase the trend of OSR is upward as well. This is expected since the more robots sensing the environment the higher the probability of simultaneously sensing the same cells.

### 4.3.3 Conclusions

Concerning the maze scenario, the conclusions of the section are: *i*) Regarding fleets integrated with at most five robots, the *MinPos* approach is clearly advantageous (outperforming the *Yamauchi* approach in all assessed figures of merit). *ii*) The benefits of employing the *MinPos* approach are severely affected when the fleet increase beyond five robots, decreasing quickly or even disappearing when it is about eight robots.

## 4.4 AAMO assessment

This section aims to study the impact of using different *HO-Threshold* values on the performance of the proposed *AAMO* approach when the fleet is asked to explore an environment under non-ideal communication conditions. Moreover, these results are compared with the one achieved by other approaches like *Yamauchi* and *MinPos* –when they are subject to non-ideal communication conditions too, and also with an *event-based-connectivity* strategy that does make all efforts in favour of connectivity (regardless the total exploration time).

This last comparison is namely important because the performance of this kind of strategy may serve as an upper bound on the connectivity level over time and the total exploration time as well. To do so, typically two strategies (based on different connection requirements, see Sec.1.2.1) can be considered: the ones which force the robots to be connected only on task-arrival time (kind of event-based connectivity) or the ones which force the robots to keep always connected –even during the path traversal periods (continuous connectivity). In the former, the robots are forced to select only between tasks which location would not

cause isolation on arrival –regarding the current task assignment of the fleet. Nevertheless, it does not take into account the connectivity level along the path between the current robot location and the location of the task under consideration. Conversely, the latter imposes stronger restrictions on the fleet mobility in order to guarantee connectivity at all times.

Consequently, depending on the application field the latter strategy would be recommended but is more complex to implement than the former. On the contrary, the former allows a simpler implementation but could lead to a lower level of connectivity along the exploration. Concerning this document, a connectivity-at-task-arrival-time based strategy is used for comparison purposes.

Besides, it is also important to highlight that, despite *Yamauchi* and *MinPos* assume ideal communication conditions, neither of both approaches needs to be modified or adapted in order to properly run under non-ideal communication condition. Nevertheless, in the *MinPos* case, some major degradation is expected because of the following working hypothesis are not guaranteed anymore: all robots share the same map and know the position of the other fleet members, at all times. This could lead to incoordinations that, in turn, would harm the dispersion strategy on which the approach is strongly based. Conversely, in the *Yamauchi* case, the level of expected degradation is fewer due to the coordination level between robots is fewer as well. Robots only try to avoid going to the same task simultaneously.

### 4.4.1 Collected data

In order to conduct the assessment and comparison stated above, at least ten realistic software-in-the-loop simulations were executed on the *Maze* scenario presented in Fig. 4.1. All collected data is presented in Table 4.3 and are organised obeying the following scheme. The columns refer to (from left to right): **F**igure of **M**erits (FM), *Approach*, where *Y*, *MP*, *EbC* and *AAMO:HO-Th* stand for ***Y****amauchi*, ***M****in****P****os*, ***E****vent-****b****ased* ***C****onnectivity*[4] and ***A****uto-****A****daptive* ***M****ulti-****O****bjective:****H****uman****O****perator-****Th****eshold*, respectively; and fleet size $|R|$. The *HO-Threshold* values are 20m, 15m and 10m (equivalent to 66%, 50% and 33% of the communication range $c_i$, respectively). Besides, since the communication conditions are non-ideal all runnings only concern fleets integrated with multiple robots (explicitly avoiding the single robot case because the communication conditions do not make any difference in it).

### 4.4.2 Effectiveness assessment

We start the analysis highlighting that all implemented approaches –and particularly all *AAMO* instances– can adequately explore all the environments presented above in Section 4.1.1. Coherently, all instances achieve a high level of *CR* when exploring the *Maze* environment, as can be appreciated in Fig.4.8.

[4]Implemented by an *AAMO:*$\infty$ instance, as was mentioned above in Sec.3.2.4 of the previous chapter.

**Table 4.3:** *AAMO* Results obtained under non-ideal communication conditions on Maze environment.

| FM | Approach | $\|R\|$ = 2 | | 3 | | 4 | | 5 | | 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD |
| TT | Y | 1216 | 131.7 | **904** | 73.6 | **759** | 78.2 | 653 | 53.3 | 496 | 86.5 |
| | MP | 1201 | 118.2 | 945 | 95.4 | 801 | 90.2 | 683 | 57.9 | **491** | 29.2 |
| | EbC | 1751 | 131.0 | 1600 | 190.7 | 1339 | 291.7 | 1028 | 154.6 | 750 | 89.0 |
| | AAMO:20 | 1292 | 87.8 | 1100 | 88.1 | 913 | 94.9 | 767 | 104.0 | 661 | 108.2 |
| | AAMO:15 | 1222 | 73.6 | 1100 | 130.1 | 823 | 79.0 | 723 | 65.7 | 606 | 85.8 |
| | AAMO:10 | **1137** | 85.3 | 960 | 123.1 | 774 | 94.3 | **620** | 76.9 | 514 | 23.1 |
| PL | Y | 1592 | 144.2 | **1707** | 173.8 | **1842** | 190.0 | **1846** | 139.0 | **2216** | 290.7 |
| | MP | 1583 | 134.9 | 1744 | 177.7 | 1911 | 200.2 | 1960 | 173.8 | 2375 | 159.9 |
| | EbC | 2215 | 154.0 | 2929 | 323.5 | 3416 | 618.6 | 3181 | 415.1 | 3412 | 296.3 |
| | AAMO:20 | 1726 | 114.4 | 2086 | 222.5 | 2243 | 236.1 | 2394 | 252.8 | 2782 | 376.1 |
| | AAMO:15 | 1669 | 86.9 | 2056 | 207.8 | 2106 | 204.8 | 2263 | 219.2 | 2536 | 222.6 |
| | AAMO:10 | **1542** | 119.9 | 1859 | 225.6 | 1982 | 215.5 | 2007 | 221.7 | 2279 | 248.5 |
| CR | Y | 99.2 | 0.33 | 99.4 | 0.37 | 99.3 | 0.33 | 99.3 | 0.24 | 99.2 | 0.20 |
| | MP | 99.3 | 0.35 | 99.4 | 0.29 | 99.4 | 0.33 | 99.6 | 0.34 | 99.2 | 0.20 |
| | EbC | 99.1 | 0.04 | 99.0 | 0.05 | 99.1 | 0.09 | 99.1 | 0.11 | 99.2 | 0.30 |
| | AAMO:20 | 99.1 | 0.26 | 99.2 | 0.28 | 99.2 | 0.28 | 99.3 | 0.35 | 99.4 | 0.37 |
| | AAMO:15 | 99.2 | 0.35 | 99.3 | 0.32 | 99.3 | 0.31 | 99.2 | 0.12 | 99.4 | 0.39 |
| | AAMO:10 | 99.3 | 0.32 | 99.3 | 0.40 | 99.3 | 0.29 | 99.3 | 0.29 | 99.1 | 0.05 |
| OSR | Y | 13.39 | 6.72 | 20.21 | 17.71 | 28.95 | 19.00 | 23.28 | 13.94 | 6.44 | 1.28 |
| | MP | 20.49 | 13.22 | 28.72 | 18.85 | 30.46 | 19.77 | 27.98 | 18.51 | 6.10 | 0.87 |
| | EbC | 2.47 | 2.98 | 3.99 | 3.01 | 3.94 | 1.96 | 5.04 | 2.34 | 5.44 | 0.01 |
| | AAMO:20 | **1.85** | 1.57 | 2.68 | 1.12 | 3.53 | 2.29 | **4.56** | 1.46 | 5.92 | 0.83 |
| | AAMO:15 | 1.94 | 1.50 | **2.47** | 1.07 | **2.40** | 0.10 | 4.81 | 2.04 | **5.43** | 0.02 |
| | AAMO:10 | 2.47 | 2.98 | 3.99 | 3.01 | 3.94 | 1.96 | 5.04 | 2.34 | 5.44 | 0.01 |
| DLR | Y | 77.0 | 9.10 | 85.9 | 6.35 | 79.7 | 7.47 | 77.1 | 5.94 | 60.7 | 9.38 |
| | MP | 81.7 | 7.69 | 87.2 | 7.26 | 80.2 | 8.28 | 77.0 | 8.52 | 59.6 | 10.74 |
| | EbC | **14.5** | 2.60 | **30.9** | 3.83 | **29.0** | 9.02 | 41.8 | 11.16 | **39.4** | 8.64 |
| | AAMO:20 | 40.6 | 9.32 | 54.0 | 9.43 | 44.1 | 13.29 | 51.8 | 10.11 | 46.5 | 13.92 |
| | AAMO:15 | 45.6 | 12.41 | 53.9 | 15.36 | 55.4 | 14.85 | **39.9** | 11.22 | 47.8 | 12.30 |
| | AAMO:10 | 61.3 | 9.49 | 68.6 | 13.24 | 54.9 | 9.53 | 62.8 | 11.32 | 46.6 | 14.28 |
| MDLR | Y | 34.8 | 14.30 | 58.0 | 26.30 | 41.8 | 18.06 | 49.3 | 15.41 | 25.0 | 6.66 |
| | MP | 33.8 | 10.88 | 55.5 | 16.44 | 40.6 | 17.52 | 44.6 | 18.38 | 27.2 | 8.40 |
| | EbC | **3.3** | 0.39 | **6.6** | 1.70 | **6.7** | 2.99 | **12.7** | 6.90 | **11.0** | 3.84 |
| | AAMO:20 | 17.8 | 10.47 | 21.7 | 8.96 | 15.9 | 7.61 | 24.2 | 12.99 | 18.0 | 8.88 |
| | AAMO:15 | 19.2 | 6.49 | 22.2 | 9.65 | 24.4 | 13.17 | 15.2 | 6.62 | 17.4 | 4.36 |
| | AAMO:10 | 27.2 | 10.11 | 33.6 | 10.95 | 20.0 | 5.72 | 24.9 | 9.15 | 22.9 | 8.61 |

### 4.4.3 *AAMO vs* Baseline comparison

In relation to *TT*, as was expected in multi-robot systems, all *AAMO* instances benefit from adding robots to the fleet. This result can be seen in Fig.4.9a. Nevertheless, compared to the baseline results all *AAMO* instances show performance degradation (see Fig.4.9b).

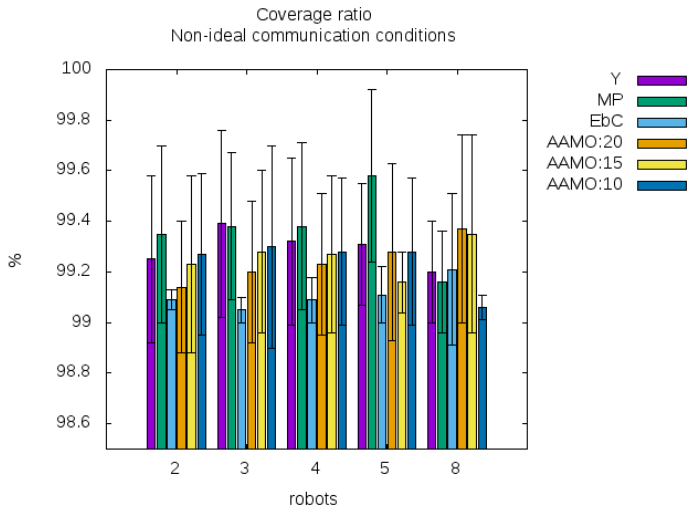



**Figure 4.8:** *AAMO* Coverage ratio. Regardless of how different are the *HO-Threshold* values, in all cases, the *AAMO* approach can cover more than 99% of the terrain.

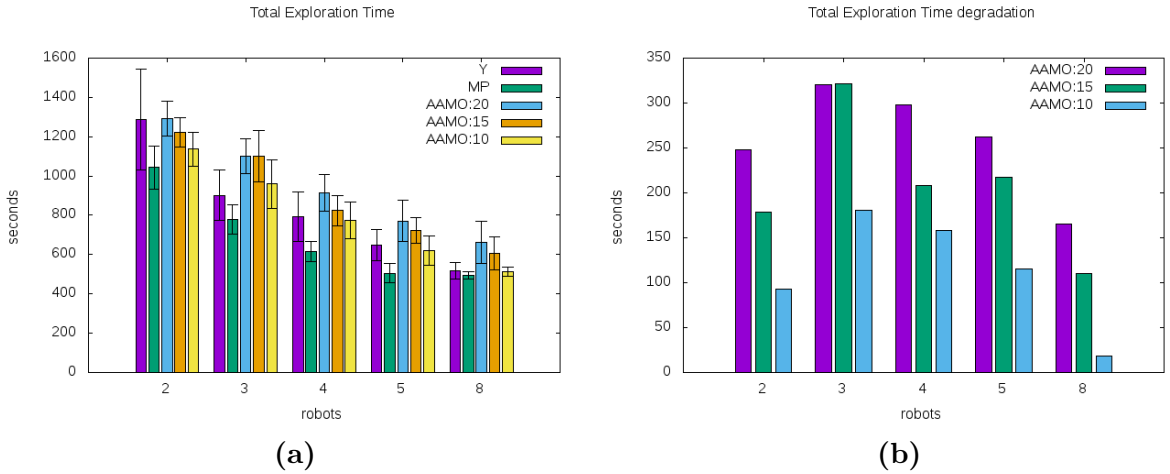
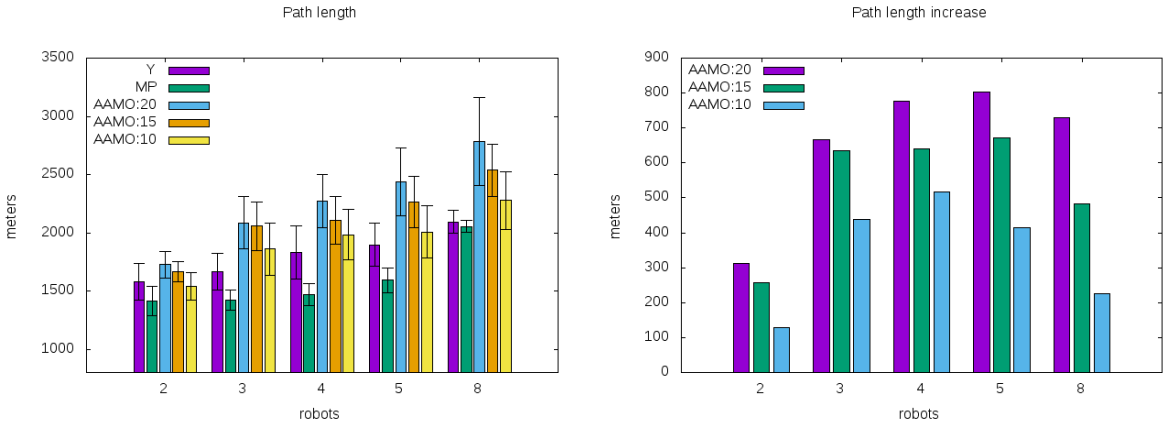



**Figure 4.9:** *AAMO* Total Exploration Time (TT) under non-ideal communication conditions and Degradation with respect to baseline results. (a) All *AAMO* instances show a decreasing trend of TT as the fleet size increase. The *Yamauchi* and *MinPos* approach results (coloured in purple and green, respectively) obtained under ideal communication conditions are placed together to make the comparison easier. (b) The degradation is expressed in terms of the difference between the TT achieved by each of the *AAMO* instances and the one achieved by the *MinPos* approach, for each fleet size.

The evidence indicates that the more efforts made in favour of connectivity (bigger HO-Threshold) the worst *TT*. In other words, not all *HO-Threshold* setup values produce the same level of performance degradation. Since the degradation of TT performance could be very problematic in many application fields, this subject is carefully analysed.

At first, the PL indicator can help to initially explain why the fleet spends more time under *AAMO* approach than under the *MinPos* approach, to explore the same environment.

In Fig.4.10a it is possible to observe the same behaviour as in the baseline (see Sec.4.3): larger fleets imply bigger PL; while in Fig.4.10b the difference between the corresponding total length of the paths traversed by fleets is shown.




**(a)** **(b)**

**Figure 4.10:** *AAMO* Path length (PL) under non-ideal communication conditions and Degradation with respect to baseline results. (a) An increasing trend of PL is shown by all *AAMO* instances as the fleet size increase. The *Yamauchi* and *MinPos* approach results (coloured in purple and green, respectively) obtained under ideal communication conditions are placed together to make the comparison easier. (b) The degradation is expressed in terms of the difference between the PL achieved by each of the *AAMO* instances and the one achieved by the *MinPos* approach, for each fleet size.

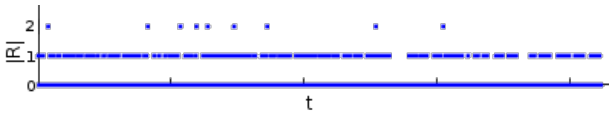
The similarity between Fig.4.9b and Fig.4.10b is remarkable and could explain to a large extent the origin of the TT degradation. Simply, under the *AAMO* approach, the robots are asked to invest some effort (traduced into a distance employing the HO-Threshold) in order to keep the fleet connected and hence it is logic to get a bigger PL as a result. Moreover, the trade-off between path and connectivity utility discussed in Sec.3.1 shows up through these results, reflecting that the price of connectivity is not being able to apply an optimal policy concerning path costs.

Nevertheless, there exists a small portion of the TT degradation that cannot be explained by the PL increasing. Therefore, the hypothesis assumed in the tractability analysis made at the end of Sec.3.3.2 are compared here with the simulation results in order to add a complementary explanation on the TT degradation. Furthermore, this TT degradation shows a parabolic trend as the fleet size increase, presenting a maximum in three-sized fleets, independently of the *HO-Threshold* values. Thus, the analysis will be conducted observing what happens when the fleet size does change but the *HO-Threshold* does not (in order to explain the shape of the curve or the relative values), and the opposite conditions are imposed in order to explain the absolute values.

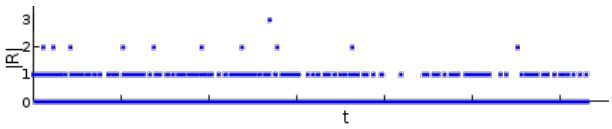
In any case, it is worth knowing that the *Task selection* algorithm is the most demanding software component in the software architecture of the robots. Hence, the overall performance of the multi-robot system is highly determined by the performance of this component. In turn,

–as was pointed out in Sec.3.3.2– its performance is strongly influenced by the amount of unassigned thresholded tasks $n = |T^{\mathrm{HO}}|$ and the number of unassigned robots in a connected component $m = |R^u|$ that are making a decision at the same time, in the following way: $|Ar^n_m| = \frac{n!}{(n-m)!} = \Pi^{n-1}_{m=0}(n-m) \rightarrow O(n^m)$. Therefore, the smaller $|T^{\mathrm{HO}}|$ and $|R^u|$ the faster the algorithm will run. Please, recall that $|T^{\mathrm{HO}}|$ is upper bounded by the number of unassigned tasks $|T^u|$.

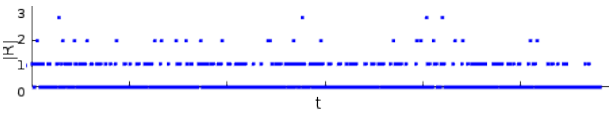
Firstly, from Figure 4.11 and Figure 4.12, it is possible to examine how $|R^u|$ and $|T^u|$ change along explorations depending on the fleet size. In all cases, both values show well-defined patterns that are easily identifiable. Concerning $|R^u|$ (see Fig.4.11) it is possible to state that in all *AAMO* instances –working in a fully asynchronous modality– the probability of two or more robots simultaneously running a decision making process is negligible. Thus the majority of time either none robot is making a decision or at most one robot is evaluating the available tasks.




**(a)** 3-Robot System




**(b)** 5-Robot System




**(c)** 8-Robot System

**Figure 4.11:** Maximum amount of unassigned robots $|R^u|$ in any connected component over time under different sized multi-robot systems. All images concern instances of *AAMO* set with HO-Threshold=15. Blue dots represent the $|R^u|$ (on average) that are simultaneously making a decision along the exploration.

Results obtained during simulations are summarised in Table 4.4 and show a behaviour that is consistent with this last statement independently of the fleet size. The low ratio of robot coincidences is remarkable (e.g. for 3-sized fleets, about 96% of the decision making moments have only one robot participating in it).

In conclusion, in practice, the worsening of the TT performance is apparently only related to the incidence of the *HO-Threshold* on the $|T^{\mathrm{HO}}|$ value. Next, this relation is carefully studied, and some answers are essayed.

The parabola described by the TT degradation values in Fig.4.9 suggests the presence of two factors impacting on this behaviour. One pressing the trend upwards and other in

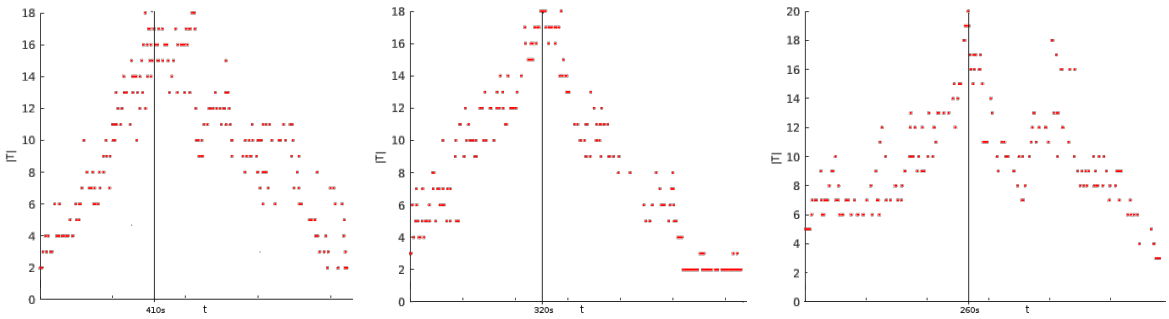
**Table 4.4:** *AAMO* Robot Coincidence on Decision Making moments.

| HO-Threshold | $\|R\|$ | Robot coincidences | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | | 2 | | 3 | | 4..$\|R\|$ | |
| | | AVE | StD | AVE | StD | AVE | StD | AVE | StD |
| AAMO:10 | 3 | 0.959 | 0.01 | 0.041 | 0.01 | $\simeq 0$ | $\simeq 0$ | n/a | n/a |
| | 5 | 0.895 | 0.02 | 0.097 | 0.02 | 0.011 | 0.01 | $\simeq 0$ | $\simeq 0$ |
| | 8 | 0.807 | 0.02 | 0.153 | 0.04 | 0.037 | 0.02 | $\simeq 0$ | $\simeq 0$ |
| AAMO:15 | 3 | 0.969 | 0.02 | 0.031 | 0.02 | $\simeq 0$ | $\simeq 0$ | n/a | n/a |
| | 5 | 0.929 | 0.02 | 0.068 | 0.03 | 0.003 | 0.01 | $\simeq 0$ | $\simeq 0$ |
| | 8 | 0.823 | 0.03 | 0.146 | 0.03 | 0.024 | 0.01 | $\simeq 0$ | $\simeq 0$ |
| AAMO:20 | 3 | 0.968 | 0.02 | 0.032 | 0.02 | $\simeq 0$ | $\simeq 0$ | n/a | n/a |
| | 5 | 0.917 | 0.02 | 0.080 | 0.02 | 0.003 | 0.01 | $\simeq 0$ | $\simeq 0$ |
| | 8 | 0.838 | 0.02 | 0.148 | 0.03 | 0.018 | 0.01 | $\simeq 0$ | $\simeq 0$ |

a counter sense. In the following two particular factors are analysed: the fleet size and the bounded condition of the environment. *i*) As the fleet size increase, robots make progress faster causing the enlargement of the $|T^{\text{HO}}|$ happens more quickly as well. Whether $|T^{\text{HO}}|$ rises, the task selection algorithm becomes slower, and thus the increase in the fleet size could explain the first increasing section of the trend. *ii*) In bounded environments, the multi-robot exploration systems typically show two mobility patterns that characterise, in turn, two different exploration stages: *1*) One is characterised by the dispersion of the fleet on the terrain. In such a stage, the new available tasks appear closer to each other, and its total amount $|T^u|$ is upward. *2*) On the contrary, the second exploration stage is characterised by the convergence of the fleet to the remaining unexplored zones starting when it is no longer possible to disperse the fleet until the end of the exploration. In such a stage, the new available tasks generally appear further to each other and its total amount $|T^u|$ is decreasing. Therefore, since the tasks $T^{\text{HO}}$ are the ones which are closer than a relative distance *HO-Threshold*, under the *AAMO* approach it is statistically less demanding for the robots to select a task during the last exploration stage than in the initial one.

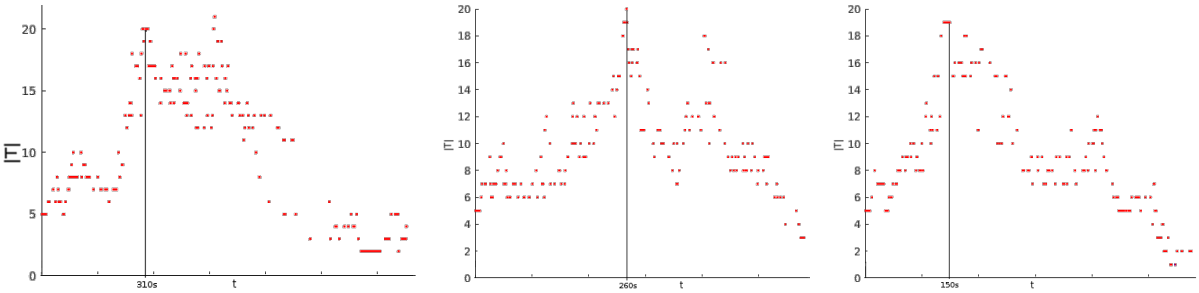
Additionally, either when the fleet size increase or the *HO-Threshold* decrease, the transition from the first to the second exploration stage is achieved faster. This fact can be corroborated in both Fig.4.12 and Fig.4.13. For instance, concerning Fig.4.12, the 3-Robot system spends about 410s to reach the end of the dispersion stage whereas the 5-Robot system and 8-Robot system spend about 320s and 260s, respectively. Likewise, from Fig.4.13, the AAMO:20 instance spend about 310s to reach the end of the dispersion stage whereas the AAMO:15 and AAMO:10 spend about 260s and 150s, respectively.

Hence, although the impact of the fleet size on the exploration stage transition appears to be higher than the one caused by the *HO-Threshold* value, both aspects contribute to reducing the task selection effort enabling robots to save time in the task allocation procedure anticipatedly.




**(a)** 3-Robot System **(b)** 5-Robot System **(c)** 8-Robot System

**Figure 4.12:** Amount of unassigned tasks $|T^u|$ over time for different sized multi-robot systems. All images concern instances of *AAMO* set with HO-Threshold=15. The maximum $|T^u|$ and the end of the dispersion stage are reached at the same time. Red dots represent the $|T^u|$ considered by robots (on average) along the exploration.




**(a)** HO-Threshold=20 **(b)** HO-Threshold=15 **(c)** HO-Threshold=10

**Figure 4.13:** Amount of unassigned tasks $|T^u|$ over time for different instances of the *AAMO* approach on 8-Robot systems. The maximum $|T^u|$ and the end of the dispersion stage are reached at the same time. Red dots represent the $|T^u|$ considered by robots (on average) along the exploration.

In conclusion, when the *AAMO* is executed in bounded environments, the addition of robots and the decreasing of *HO-Threshold* can almost entirely mitigate the worsening in the total exploration time performance. Please note that the performance degradation of AAMO:10 instances is almost null for eight-sized fleets.

From these promising results, in the following, all *AAMO* instances are compared with the other approaches concerning non-ideal communication conditions.

### 4.4.4 *AAMO* efficiency assessment

Firstly, concerning TT (see Fig.4.14), the evidence confirms two expected results: *i*) All approaches benefit from adding robots to the fleet. *ii*) Since it only takes care of connectivity, the EbC approach shows the worst performance. Additionally, all *AAMO* instances show competitive TT results even slightly outperforming other approaches in the case of *AAMO:10*.

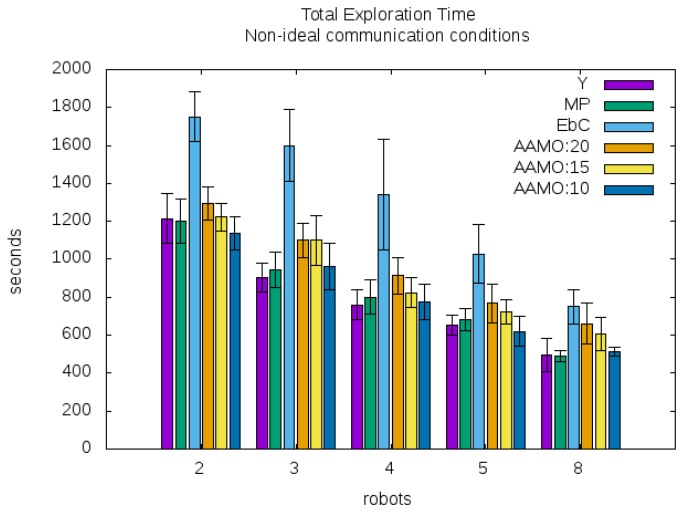



**Figure 4.14:** Total Exploration Time (TT) under non-ideal communication conditions.

Secondly, concerning the PL indicator (see Fig.4.15), the EbC approach present the worst performance, coherently. Likewise, all *AAMO* instances show competitive results too.
Besides, and as was pointed above, the TT and PL results show that the lack of ideal communication conditions negatively affects the *MinPos* approach more than the *Yamauchi* approach.

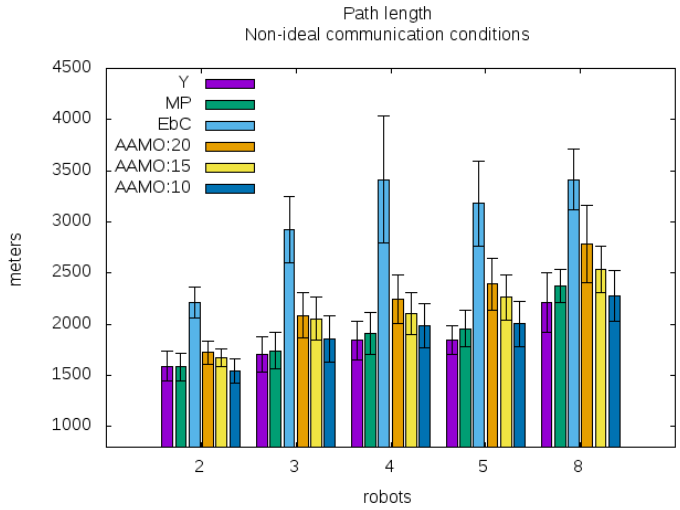



**Figure 4.15:** Path length (PL) under non-ideal communication conditions.

Up to this point, the *AAMO* approach has shown results as good as the *MinPos* approach. Next, the indicators related to connectivity are analysed in order to accurately assess the potential advantages of the *AAMO* approach in the presence of more realistic communication conditions.
The DLR indicator trend is shown in Fig.4.16. As can be seen, while the performance of the *MinPos* and *Yamauchi* approaches are the worst, the EbC performance is remarkably the best.

On the other side, the *AAMO* approach results represent a very good improvement concerning both *MinPos* and *Yamauchi* approaches. This chart reveals that our approach outperforms both Yamauchi and minPos approaches independently of the fleet size. Nevertheless, the smaller fleet, the greater outperforming. The explanation can arise correctly from intuition: when the environment is bounded, the probability of being disconnected tends to decrease as the fleet size increase. Therefore, the benefits of our approach tend to be smaller when the fleet size increases, although, it is always meaningful. Please note that even in the largest fleet size case, the DLR of *AAMO* represents an improvement of 20% compared to the corresponding *Yamauchi* or *MinPos*.

Furthermore, the relation between TT, DLR and *HO-Threshold* is noticeable. The more effort demanded by the human operator (higher threshold), the slower but higher connected the *AAMO* performs.

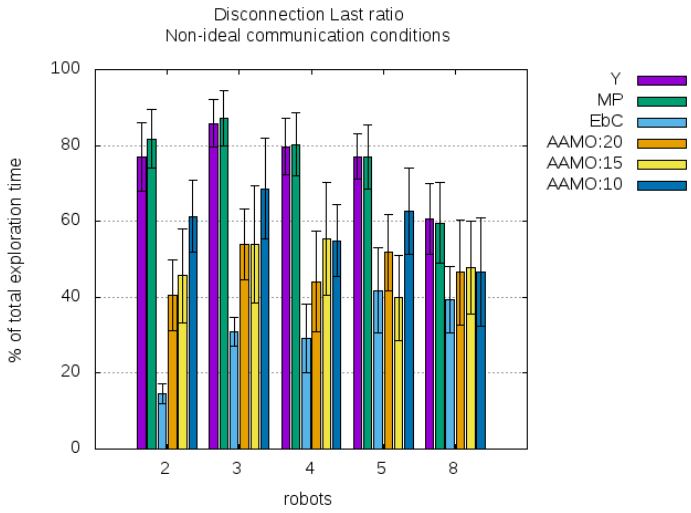



**Figure 4.16:** Disconnection Last Ratio (DLR) under non-ideal communication conditions. The bigger *HO-Threshold*, the smaller the DLR. This fact holds showing oscillatory behaviour as the fleet size increase.

Regarding the oscillation registered, it could suggest the existence of the following rational pattern. When fleet size is even, the more natural way to avoid isolation situations is keeping in pairs (connected with at least another teammate). Contrarily, when the fleet size is odd, not all robots can keep in pairs. In case the fleet has divided, at least one subgroup must be composed of three robots. Therefore, this oscillatory behaviour could hint at the fact that odd-sized fleets need to make little more effort to avoid robot isolation situations and are consequently subject to more significant DLR results as well.

To sum up, the results of TT, PL and DLR indicators are strongly influenced by the *HO-Threshold* values accordingly with the following relation: the bigger *HO-Threshold*, the longer TT, the larger PL and the smaller DLR, independently of the fleet size.

Likewise, it is interesting to analyse the DLR indicator and network topology together. This way it is possible to get a closer notion about the interaction between robots along the

exploration. Fig.4.17 is devoted to showing the number of connected components present in the network, averaged over time.
Please note that for the AAMO:20 instance –run on 2-Robot fleet– the DLR is about 40% (see Fig.4.16) coinciding with the percentage achieved by the 2CC of the same fleet size in Fig.4.17. In other words, the fleet holds a network composed of one single connected component during $60\%(100\% - 40\%)$ of total exploration time. Consistently, this is equivalent to say that during this portion of the time none robot has been disconnected.

Additionally and as a matter of fact, the chart shows that as the fleet size increase it is more difficult to keep the whole fleet connected: 1CC stack is decreasing in size as the fleet size increase. Nevertheless, it also shows that simultaneously with the adding of new robots, the fleet is more and more cohesive (in relative terms). This fact may be corroborated looking at the upper part of the chart where the stacks corresponding to the highest number of connected components are plotted. The following pattern can be observed: the number of connected components (given by $nCC$) increase slower than the fleet size $n$. Again, the fact that the *Maze* scenario is bounded may explain this phenomenon to a large extent.

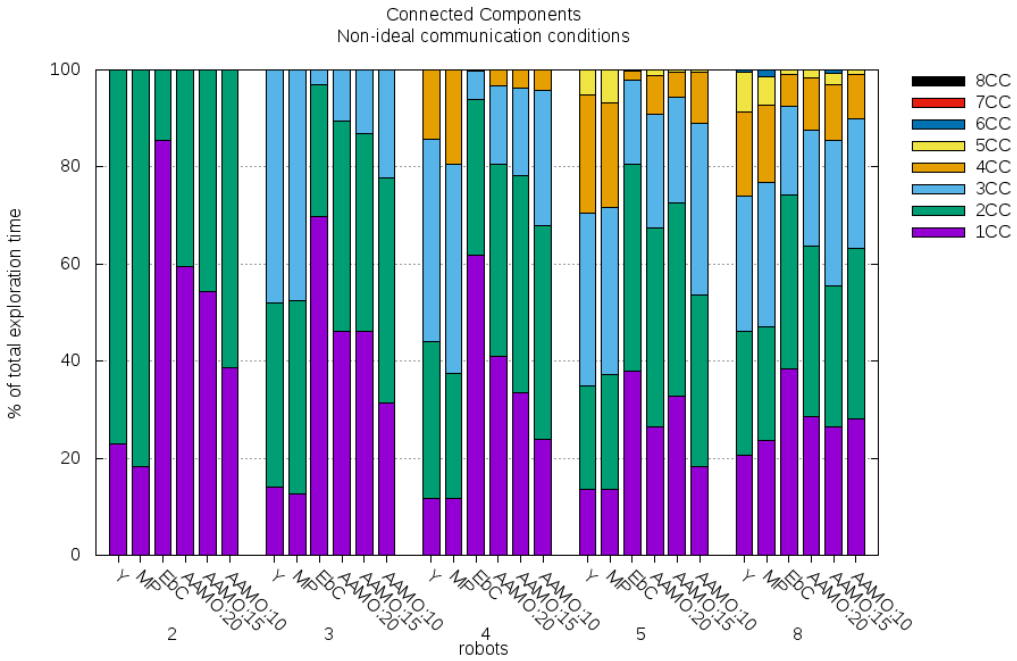



**Figure 4.17:** Network topology composition under non-ideal communication conditions averaged over time. Depending on the number of connected components and the fleet size, it is possible to study the existence of sufficient conditions to fall into isolation situations. For instance, for a 3-Robot fleet, the 2CC or 3CC topologies imply having at least one robot isolated while for a 5-Robot fleet this implication is related to 3CC, 4CC or 5CC topologies, and so on.

Even though all this information gives an approximated notion about how disconnected is the fleet (group perspective) along explorations, it is not enough to hint what is happening at the individual level. Thus, it is also interesting to study the worst case of the individual disconnections last. This way it is easier to evaluate both coordination capabilities (how long a

robot is unable to coordinate its actions with any other teammates) and risky situations (how long the fleet present single points of failure). Recall that the key motivations in considering communication constraints are strongly related with the rework avoidance: *i*) when robots are unconnected they have fewer possibilities to coordinate their actions hence they could visit the same regions unnecessarily. Hence, keep them connected is a way to favour efficiency. *ii*) in the presence of damages or inner failures the exploration strategy should take those events into account preventing the need for re-exploration.

In Fig.4.18 the trend followed by Maximum Disconnection Last Ratio MDLR indicator is depicted showing that the bigger *HO-Threshold*, the shorter disconnection periods and that the last of isolation situations is at most equivalent to half of the DLR values for every fleet size and *HO-Threshold* value as well. In other words, the isolation situations regard more than one single robot and this in turn, reveals that under the *AAMO* approach the robots often intent to rejoin each other.

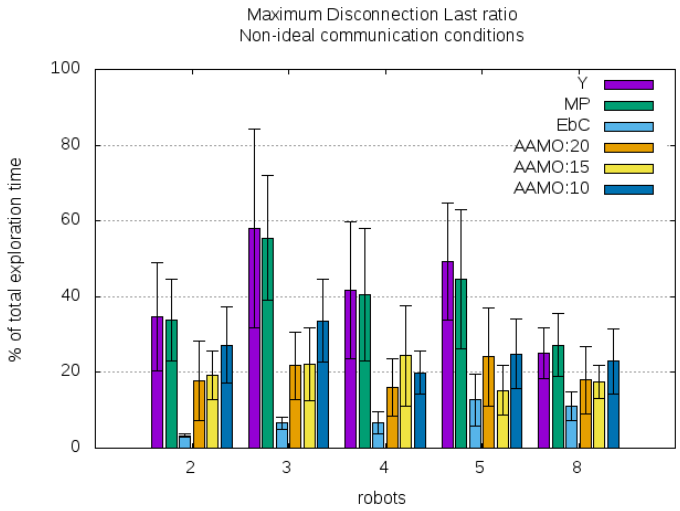



**Figure 4.18:** Maximum Disconnection Last Ratio (MDLR) under non-ideal communication conditions. MDLR shows the longest individual isolation period registered by some fleet member along the exploration. The trend is oscillatory following the same pattern as the DLR indicator.

At last but not least, it is worth to discuss the trend of OSR as the fleet size increase. The results obtained by the different *AAMO* instances are depicted in Fig.4.19. In Section 4.3 the OSR levels were achieved mostly thanks to simultaneous sensing actions, conversely, in this simulation runnings, the OSR achieve higher levels due to non-ideal communication conditions. As was expected, the more the mapping information of the robots are out-of-date to each other, the higher the OSR. However, in any communication conditions, the same upper bound is achieved. This suggests that the size and bounded condition of the *Maze* environment could be limiting the over-sensing phenomenon when fleet size increase beyond five robots.

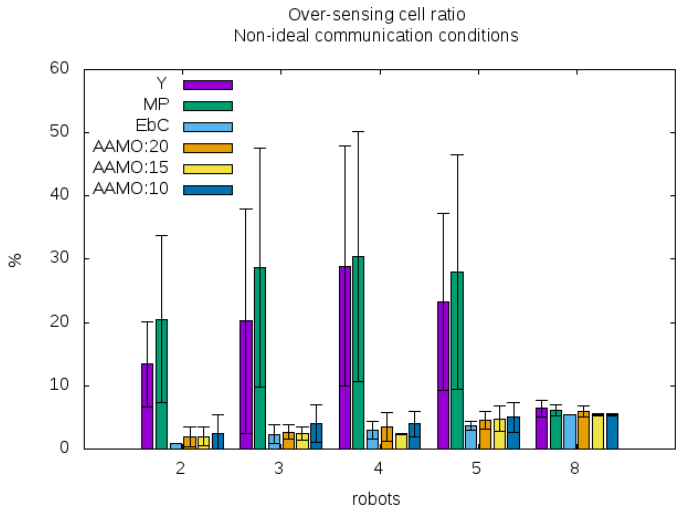



**Figure 4.19:** Over-sensing ratio (OSR) under non-ideal communication conditions. The *Yamauchi* and *MinPos* approach results (coloured in purple and green, respectively) obtained under ideal communication conditions are placed together to make the comparison easier.

### 4.4.5 Conclusions

Concerning the *Maze* scenario and the baseline stated in Sec.4.3, the conclusions of this section are: *i*) The *AAMO* approach can be employed as a strategy to coordinate multi-robot systems that are dedicated to exploration tasks. *ii*) As was expected, the *HO-Threshold* value directly impacts on the connectivity level that the fleet can hold during the mission. *iii*) Likewise, the relation between *HO-Threshold* values and the TT and DLR/MDLR indicators is the expected: the bigger *HO-Threshold* value, the worse TT performance but, the better DLR/MDLR ratios. *iv*) Although, all instances of the *AAMO* approach present TT degradation concerning the baseline, in any case, it is not significant due to the computation of the proposed task-to-robots distribution. *v*) All *AAMO* instances outperform the baseline concerning the DLR and MDLR indicators. *vi*) Except for DLR/MDLR, all instances of the *AAMO* approach outperform the *EbC* approach. *vii*) The topology of the fleet networks shown during exploration is consistent with the *HO-Threshold* values, for all *AAMO* instances.

To sum up, the *AAMO* approach shows effectiveness and flexibility (through the *HO-Threshold* setup) to tackle the multi-robot exploration problem. Particularly concerning the efficiency related to both completion time and connectivity level maintenance, the approach appears as an intermediate solution that presents much better TT performance than the most restrictive approach *EbC* and better connectivity level along exploration than the approaches that do not take care about communication issues.

From all expressed above a question come up: since the *AAMO* take care about multiple objectives, it would be possible to improve the *AAMO* proposal by considering multiple roles? In the affirmative case, it would be expected to get better DLR or MDLR performance without

degrading the TT performance?

Next, in Chapter 5 a novel dual-role based proposal is presented. This proposal is built on top of the *AAMO* by adding a new perspective where the robots are designated to pursue one single exploration objective at a time depending on its roles. Further below, in Chapter 6 the approach is assessed and compared with the *AAMO* results.

Chapter 5

# Dual Role Approach

Contents



In this chapter, a dual role strategy is described. This approach aims to perform exploration task achieving a better balance between time performance and the amount and last of disconnection periods among team members. The general idea consists in having robots exclusively playing the role of communication relays besides the explorer robots. Relay robots are expected to provide better connectivity conditions to the team, relaxing the mobility restrictions on the explorers (Nestmeyer et al. 2017; Rahman et al. 2017).
This approach leverages upon the task allocation mechanism previously presented in Chapter 3 adding a new layer on top, in charge of role assignment. Under this paradigm, explorer and relay robots have different goals. The explorers are expected to consider frontier locations in order to enlarge the global environment knowledge while relays should think in terms of locations that keep or enlarge the connectivity of the fleet. The so-called *Relay Placement - RP* problem is presented in Section 5.2. Next, a new signal-strength based approach to solve the *RP* is thoroughly described.

## 5.1 Role assignment problem

When considering the use of different roles in a robot team the way in which the roles are assigned is a key issue. Basically, there are two kinds of assignments: static or dynamic. Additionally, the fleet heterogeneity is another important aspect on this matter. In Table 5.1 some representative approaches of the possible combinations of these two aspects are summarised.

**Table 5.1:** Role assignment parameters.

| Heterogeneity \ Assignment | Static | Dynamic |
|---|---|---|
| Homogeneous | (Magán-Carrión et al. 2017)<br>(Krupke et al. 2015) | (Banfi et al. 2018)<br>(Nestmeyer et al. 2017)<br>(Banfi et al. 2016)<br>(Cesare et al. 2015) |
| Heterogeneous | (Rahman et al. 2017)<br>(Pralet and Lesire 2014)<br>(Pei and Mutka 2012) | [] |

Depending on the availability of relay robots (unlimited or limited amount of units) the assignment could deserve more attention. When availability is out of the question or simply the amount of robots is big enough to satisfy the mission requirements, a Homogeneous/Static setup require simpler algorithms than a Homogeneous/Dynamic setup. On the contrary, it demands more attention to resource management.

With the *static* assignment, the roles are typically set from the beginning of the mission and do not change anymore. This kind of assignment is easier to implement because it does not add complexity to the task selection algorithms: no role changing is required along the exploration (the opposite happens in the systems with dynamic role assignment). Nevertheless, this simplicity in the robot algorithms implicitly causes an overload on the robot delivering task (in charge of progressively release relay robots into the environment when it is needed).

When the multi-robot system is heterogeneous (typically, the explorer robots are equipped with better sensory systems than relays and relays are equipped with better communication devices than explorers), dynamic assignment could not be worth and commonly presents restrictions: e.g. role changes are only possible in one direction (explorers can eventually play the role of relays but not the opposite).

In the presence of limited resources the Homogeneous/Static setup is more sensitive to unexpected situations where the number of designated relays could become not enough compared with the connectivity requirements. Additionally, if there exist strong restrictions on connectivity the movements of the explorers should be more conservative –not spreading too much– avoiding to break links that relays could not support or repair. Conversely, the dynamic assignment strategy could be the key to properly react when facing these kinds of challenging cases. The amount of relays/explorers can go up and down accordingly with the current communication conditions. This leads to the need for adaptation, typically based on decision rules.

## 5.2 Relay placement problem

Given a fleet $R = R_e \bigcup R_r$ (defined as in Section 2.2) composed of explorer and relay robots, a region $E_{known}$ (as defined in (2.2)) known by all robots in $R$ at some instant $t$ and a set of candidate tasks $T$ (as defined in Section 2.4), the relay placement problem can be stated as the problem of finding the best locations where the set of relay robots $R_r$ can connect a set of candidate task locations following some optimisation criteria (e.g. throughput, bandwidth, redundancy). This general statement matches up with the so-called *Steiner minimum tree* problem (Magán-Carrión et al. 2017). From this, the influence of the availability of resources, the role assignment type and the application field must be carefully analysed.

Firstly, the influence of the application field where the multi-robot system will be applied over the optimisation criteria selection is notorious. To cite an example, in search and rescue missions not only connectivity is required but very good throughput and bandwidth. In these scenarios, the success of the mission strongly depends on the ability to recognise victims which is in turn supported by the video streaming capabilities of every explorer robot involved in the search.

Next, concerning the number of resources and the way the roles are assigned the analysis may be conducted by the study of the relation between $|T|$ and $|R|$. On the one hand, when $|T| <= |R_e|$, despite the explorer robots could be able to reach all candidate tasks, depending on the spatial distribution of the tasks $T$, the $|R_r|$ would be not enough to support the network connectivity requirements. In this cases, the systems with dynamic role assignment offer certain advantages because of having the possibility to increase the number of relays $|R_r|$ through changing the role of the idle explorers. Conversely, a static role assignment based system will be forced to constrain its exploration capabilities in order to comply with the connectivity requirements. On the other hand, when $|T| > |R_e|$, the set of explorers is not large enough to cover all candidate tasks and consequently the task locations to be connected will depend on the task selection finally made by explorers. Anyway, the number of task locations will be upper bounded by $|R_e|$. Once again, systems based on static role assignment could be too rigid to tackle the relay placement problem in certain circumstances.

In conclusion, regardless of the relation between $|T|$ and $|R|$, the minimum number and location of relays can be obtained solving the corresponding *Steiner minimum tree* problem. Nevertheless, depending on how big are $|T|$ and $|R_e|$ the complexity of computing the optimal solution could make it prohibitive. In the following, the task allocation scheme and the approach employed to compute a general solution of the relay placement problem is described.

## 5.3 Task allocation scheme

The allocation scheme concerns the distribution of tasks to robots. Under the dual-role modality described here in this Chapter, all robots coordinate their actions following exactly the scheme proposed in the AAMO approach (described in Section 3.3.1). By contrast, the

presence of different roles implies that the robots have different goals –considering a different set of tasks– depending on the role they are playing at each moment. Thus, while the explorer robots select its tasks employing the Alg.2 (presented in Chapter 3), the relay robots need to follow a different approach. Namely, the main difference comes from the fact that in the one case (explorers) the tasks rise directly from the exploration itself while in the other, the tasks –as the right places to perform the relay role– must be calculated. Therefore, the different nature of both underlying problems demands different solutions, consequently. Nevertheless, once the candidate places are computed, the task selection problem can be faced similarly.

The task selection algorithm will be described in Section 5.4. Besides, it is important to keep in mind that: *i*) After determining the best candidate locations where to place the relay robots, all the *Multi-Objective* machinery (introduced in Chapter 3) is employed to solve the task selection problem. *ii*) The solution proposed here does not take into account either whether the role assignment mechanism is static or dynamic nor the number of available relay robots in the fleet. Both things are considered as problems of a higher level of abstraction, and they are not addressed here in this document.

## 5.4 Relay placement approach

This Section is devoted to describing the strategy followed by relays to select its tasks in order to support as much as possible the network connectivity. In this strategy, the communication model (as defined in (2.1)) has a distinguished role. Both the current connectivity degree of the fleet and the most promising relay candidate locations are computed from this model. Besides, in order to avoid evaluating all possible candidates, a climbing-hill-like searching technique is employed to follow an ascending path on a surface derived from the corresponding signal strength map.

Firstly, given that a graph may induce the topology of the robot network and since the network –in many opportunities– could not be broken by any of the next movements of the explorers (remaining composed of one single connected component), the candidate locations may be chosen for both maintenance or repair purposes. The general idea is to identify the best places in the more promising regions of $E_{known}$ where would be more convenient to place a relay. In the first one –maintenance purpose, candidate regions are the ones which improve the connectivity of as many nodes as possible. In the second –support purpose, selected regions are the ones which contain places that can play the role of bridges among unconnected components of the network.

In Fig.5.1, representative scenes are depicted. Starting for Fig.5.1a, the first row depicts the robots present in the scene as well as its spatial distribution and the scope of the communication devices. Since there are no obstacles, the more promising region to place a relay coincide with the geometric region where the communication scopes overlap each other. However, in the presence of obstacles this is not the rule, see Fig.5.1b and Fig.5.1c. Instead of geometric approaches, the communication model (see Sec.2.3) gives a correct way to always visualise these areas, independently of the obstacle existence. In turn, the functions

$\max_{\forall j \in R} \Gamma_i(j) \mid i \in \mathbb{R}^2$ and $\Omega_i(j) \mid i \in \mathbb{R}^2 \wedge j \in R$ (defined in (2.1),(3.3) and depicted in second and third rows, respectively) show that the highest connectivity level is reached in the same regions where the strongest communication signal strength falls into a local minimum (corresponding to the whole valley in Fig.5.1a or to some specific minor areas of the valley in both Fig.5.1b and Fig.5.1c).

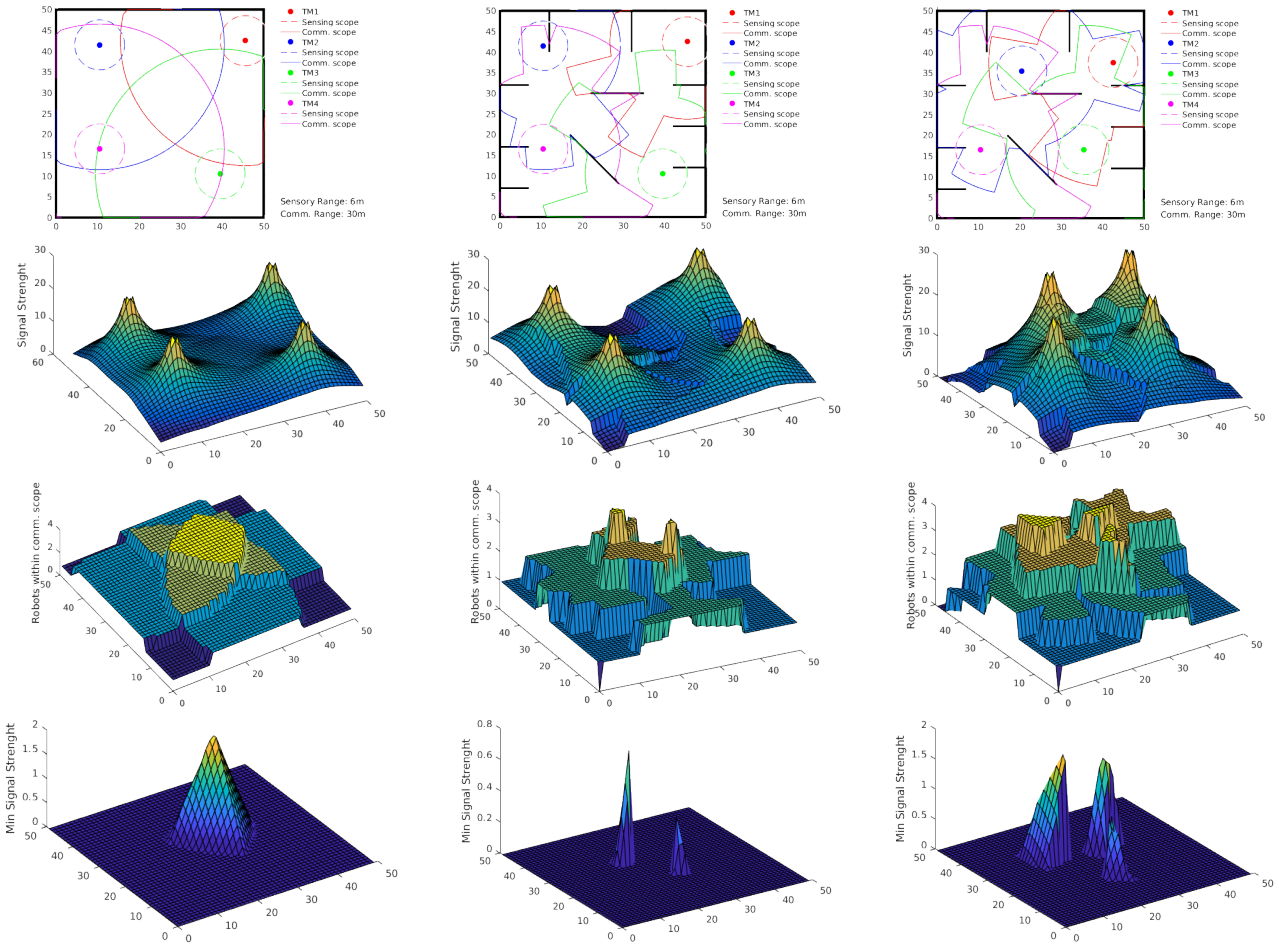



**Figure 5.1:** Relay placement regions. On each column, the scene, the strength of the strongest communication signal on each cell of the environment, the connectivity level reached on each cell and the weakest strength of the strongest communication signals on each cell, are depicted by rows, respectively. In the scene, robots are represented by coloured dots while solid black lines represent walls. Column (a) refers to a connected fleet of robots exploring an obstacle-free zone. Column (c) refers to the same configuration as in (a) but under stronger communication restrictions caused by the presence of several walls. Consequently, the fleet is unconnected. (b) refers to a different configuration of the fleet where despite the communication restrictions the fleet is still connected.

Unfortunately, neither of both functions by itself can support the calculus of the best relay placement. On the one hand, the connectivity level $\Omega_i(j)$ does not provide uniqueness meaning that, once on top, more than one cell offers the same highest connectivity level. Conversely, the signal strength perceived on these cells could be slightly different between

each other. Hence, from there on a second searching strategy should be followed to reach the cell where connectivity level and signal strength maximise simultaneously. On the other hand, descending on the strongest signal strength map ($\max_{\forall j \in R} \Gamma_i(j) \mid i \in \mathbb{R}^2$) from whichever transmitting sources to the referred valley zone is not guaranteed. Even when the right descending direction is chosen, the function does not offer any useful information to stop descending until the global minimum itself is reached.

Thus, an alternative function is defined to support the search of the best locations where place the relay robots in a straight way. As can be seen in the fourth row, from the $\min_{\forall j \in R} \Gamma_i(j) \mid i \in \mathbb{R}^2$ function is possible to climb to the top until the relative maximum is reached. This maximum value occurs precisely in the place where the connectivity level is the highest, and it is not possible to move aside without worsening the signal strength perceived from some transmitting source. Therefore, by analogy with game theory, the $\operatorname{argmax}_{\forall i \in \mathbb{R}^2} \min \Gamma_i(j)$ can be seen as a Pareto optimal point. In Fig.5.1b and Fig.5.1c, it is possible to see how good is the function in capturing the best candidate regions to go, from scenes representing both unconnected and connected fleets. Namely, in Fig.5.1b robot $TM1$ and $TM3$ (red and green dots, respectively) are disconnected from the rest, and the $\min \Gamma_i(j)$ function reveals the existence of two promising regions where to place a relay in order to reconnect both robots at once. The only difference between them would be related with signal strength obtained on the links, but both regions offer the possibility to reestablish the communication between the fleet members.

On the contrary, both Fig.5.1c and Fig.5.1a present a connected fleet of robots where the existence of unique or multiple regions is directly caused by the presence of walls and its attenuation effect on the communication scopes.

Finally, the procedure through which the network is maintained or repaired may be seen as a stack of algorithms where it starts from the top (more abstract) to bottom (more concrete) addressing different subproblems. Alg.3 describes in higher level terms the most important tasks a robot $R_i$ that plays the role of relay must solve to select it next target location: *i*) from the RP problem resolution, obtaining the best candidate cells where to go next. *ii*) from the best assignment problem resolution, choosing the best cell where to go next.

**Algorithm 3** Task selection algorithm.

1: **function** GETASSIGNMENT($i, R^*, E_{known}$)
   ▷ i stands for the position of a relay robot in the vector $R^*$.
   ▷ $R^*$ stands for the robots location vector.
   ▷ $E_{known}$ stands for the portion of the environment $E$ known by all robot in $R^*$.
2: $[R_e, R_r] \leftarrow R^*$
3: $assig \leftarrow getTargets(R_e)$ ▷ array of targets already assigned to explorer robots
4: $arrCCell \leftarrow getRelayPos(assig, E_{known})$ ▷ array of promising candidate cells
5: **return** $getAssignment(i, R_r, arrCCell, |R^r|, \infty, E_{known})$ ▷ the chosen cell
6: **end function**

Firstly, it is important to highlight that the input parameter $R^*$ specifically corresponds

to the locations of the teammates currently connected with the robot $R_i$. In Line 2, explorers and relays are split up into two subsets. Next, in Line 3, the set of explorer destinations are check –this way the relay robots can anticipate the position of the explorers in the close future. The set of candidate cells (free cells defined in Sec.2.1) is computed in Line 4. Please note that the size of this set is indicative of the minimum amount of relays needed to support the present connectivity network requirements. Hence, depending on the role assignment nature of the system this information could be used either to ask the base for support (static role assignment) or as an input in a dynamic-role-assignment procedure. Besides, the current locations of the other relay robots are willingly not taken into account. This decision pursues two goals: *i*) reducing the problem domain only $R_e$: the smaller the corresponding set of nodes in the graph the faster algorithm execution. *ii*) becoming the relay robots more reactive to changes in the fleet configuration. As a result, relay robots check and correct its position permanently.
Finally, in Line 5, the selection of a place in accordance with the other relay robots preferences is computed. To this end, the *getAssignment* function (see Alg.3) is used with *HO-Threshold*$= \infty$ favouring the tasks that provide connectivity over the rest. Nevertheless, given all cell candidates represent places where the relay will be connected with the fleet at highest connectivity level, this setting implies that the relays will order the cells by distances contributing to having the relay positioned as fast as possible.
Additionally, this algorithm only assumes that all robots in $R^*$ are connected and the fact that there exists at least one robot playing the relay role –the robot $i$, hence it is generic enough to be applicable on systems which have static or dynamic role assignment policies, indistinctly.

### 5.4.1 Relay placement algorithm

The Relay placement algorithm is sketched in Alg.4. Firstly, it is important to remark that the input parameter $T_e$ specifically corresponds to the future locations (location of assigned targets) of the explorer teammates $R_e$ currently connected with the relay robot $R_i$, which is supposed to be executing the algorithm. Secondly, the set of connected explorer robots is represented employing standard graph theory concepts. Particularly, the network topology is induced by a weighted complete graph, as follows: $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}, \mathcal{W})$ where the nodes $\mathcal{V}(t) = \{v \mid v \in [1 \dots |T_e|]\}$ represent the tasks already assigned to explorer robots $T_e$, $\mathcal{E} = \{i, j \mid (i, j) \in \mathcal{V} \times \mathcal{V}\}$ are the edges and $\mathcal{W} : \mathcal{E} \to \mathbb{R} \mid \mathcal{W} = \Gamma_i(d_j, N_w^j)$ represents the cost function. This way, weights are taken as the expected signal strength between each pair of nodes (the assigned task locations) at time $t$. Positive values of $\mathcal{W}$ represent operative links while negative values correspond to broken links.

In Line 2 the network topology is computed generating the set of connected components and the signal strength matrix as well. From the former is possible to know if the whole fleet is connected in one single connected component or not and what robot belongs to which component. Only operative links are considered as long as there are no disconnected robots. On the contrary, only broken links are considered in the presence of disconnections. On the other hand, the information of the signal strength matrix is used in Line 4 to compute the

**Algorithm 4** Relay placement algorithm.

1: **function** GETRELAYPOS($T_e, E_{known}$)
▷ $T_e$ stands for the vector of explorer robot assigned locations.
▷ $E_{known}$ stands for the portion of the environment $E$ known by all robots in $R_e$

2: $[binns, matSS] \leftarrow getConnComp(T_e, E_{known})$ ▷ connected components and signal strength matrix
3: $\mathcal{G}(t) \leftarrow matSS$ ▷ $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}, \mathcal{W})$
4: $mst \leftarrow getMST(\mathcal{G}(t), E_{known})$ ▷ minimum spanning tree
5: $k \leftarrow 1$
6: **if** $|binns| > 1$ **then** ▷ more than one connected component
7: **for** each $e = (e_1, e_2)$ in $mst.edges$ **do**
8: **if** $matSS(e_1, e_2) < 0$ **then**
9: $arrCell(k) \leftarrow getBestCell((e_1, e_2), mst.nodes, E_{known})$
10: $k++$
11: **end if**
12: **end for**
13: **else** ▷ only one connected component
14: **for** each $e$ in $sort(mst.edges, \text{“}descend\text{”})$ **do** ▷ Sorted in descending by cost
15: $arrCell(k) \leftarrow getBestCell(e, mst.nodes, E_{known})$
16: $k++$
17: **end for**
18: **end if**
19: **return** $arrCell$
20: **end function**

$mst$[1] corresponding to the graph $\mathcal{G}(t)$. Given that the edge weights are the signal strength between nodes, this tree delivers a handy hint about which edges will be weakened by the next movements of the explorers. Hence, it is possible to choose the most promising regions along operative or broken links depending on the current topology. After this step (sketched apart in Alg.5), the algorithm can follow two branches depending on the topology. If there exists more than one connected component (Lines 6-12) and coherently with the actions followed in Alg.5, only the minimum amount of edges corresponding to broken links are taken into account to compute new relay location candidates (Line 9). It is important to notice that the minimum amount of candidate locations is guaranteed since the procedure is carried out on the $mst$ previously computed in Line 4. Conversely, when in the presence of only one connected component (Lines 13-18) all edges are computed. In any case, one connected component or more, the result is ordered by edge-cost favouring the prioritisation of the cells located along the most promising edges. This solution is supported by the *getBestCell* function described in Section 5.4.3.

Next, the way the edges are selected following the principles of very well-known $mst$ procedures like *Prim* or *Kruskal* (Kleinberg and Tardos 2006) is carefully discussed in Section 5.4.2 and the way the best cells are chosen from the selected regions supported by the $\min \Gamma_i(j)$

[1]Minimum spanning tree (Kleinberg and Tardos 2006).

function is discussed in Section 5.4.3.

### 5.4.2 Minimum spanning tree calculus

Before starting, it is worth noticing that all the standard *mst* solutions assume that the edge weights were adequately set up. It implies that weights are strictly positive, zero weights could be employed to represent unconnected node pairs and the smaller weights will be preferred over the greater ones. The solution constitutes a spanning tree –reaches all nodes of the graph– in which the sum of edge costs is minimal. Thus, setting up the edge weights is a crucial task on this matter.

Algorithm 5 presents the way the *mst* is computed and can be seen as a wrapper of the traditional methods. It works modifying the weight of edges conveniently before applying the standard *mst* procedures. To do so, the network topology is analysed in order to implement different weight transformations. The cases are divided into three: *i*) fully connected graphs. *ii*) totally unconnected graphs. *iii*) the remaining cases.

**Algorithm 5** Minimum spanning tree algorithm.

1: **function** GETMST($\mathcal{G}(t), E_{known}$)
▷ $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}, \mathcal{W})$ stands for network topology at time $t$.
▷ $E_{known}$ stands for the portion of the environment $E$ where nodes $\mathcal{V}$ belong to at time $t$.
2: **if** $\mathcal{W}(e) > 0 \, \forall e \in \mathcal{E}$ **then** ▷ Fully connected graph
3: **return** $minSpanTree(\mathcal{G}(t))$
4: **else if** $\mathcal{W}(e) < 0 \, \forall e \in \mathcal{E}$ **then** ▷ Totally disconnected graph
5: **for** each $e$ in $\mathcal{E}$ **do**
6: $\mathcal{W}(e) \leftarrow abs(\mathcal{W}(e))$
7: **end for**
8: **return** $minSpanTree(\mathcal{G}(t))$
9: **else** ▷ There could be more than one connected component
10: **for** each $e = (v_1, v_2)$ in $\mathcal{E} \mid \mathcal{W}(e) < 0$ **do**
11: **if** $sScope(v_1, E_{known}) \bigcap sScope(v_2, E_{known}) = \emptyset$ **then** ▷ sScope returns the set of cells within the signal scope
12: $\mathcal{W}(e) \leftarrow \infty$
13: **else**
14: $\mathcal{W}(e) \leftarrow abs(\mathcal{W}(e) - \max \mathcal{W}) + 1$
15: **end if**
16: **end for**
17: **return** $minSpanTree(\mathcal{G}(t))$
18: **end if**
19: **end function**

Firstly, the fully connected graph is considered in Line 2. Recalling that the strongest the communication signal between two nodes the heaviest edge. Hence, this is the simplest case because the *mst* would return the minimum amount of edges needed to reach all nodes

prioritising the ones which represent the weakest links. Please note that these edges are processed in descending afterwards (see Line 14 in Alg.4).

By contrast, in Lines 4-8 totally unconnected graphs are considered. In this case, all weights are strictly negative. The smaller the weight, the more difficult –or even impossible– to reconnect. Therefore, taking the absolute value is enough to have a strictly positive set of weights accordingly ordered: the smaller weight, the easier to repair edge.

The third case is evaluated from Line 9. Note that there could be more than one connected components but the presence of negative edges is not a sufficient condition to ensure that. In other words, even in the presence of negative weights, the fleet could be fully connected. Moreover, assuming for an instant that negative weights are acceptable as an *mst* input, according to the position where each weight would be placed if they had been ordered in ascending, the *mst* computation would prioritise the edges which could represent insuperable disconnection situations over the ones where the placement of a relay could make a big difference. Thus, all edges that represent insuperable situations (see Lines 11-12) must be discarded while the weights of the remaining have to be coherently transformed. To this end, all preserved –positive and negative– weights become strictly positive following an ordered from the most to the least promising edge set. This implies that the negative values would be placed after the positive ones in an ascending arrangement, keeping its original relative order. Yet, the positive values should be reordered in the counter sense. This transformation is carried out in Line 14 behaving as follows: *i*) $w(e) = maxW \implies |w(e) - maxW| + 1 = 1$. So, the biggest positive weight becomes 1, and will be the smallest as well; *ii*) $0 \leq w(e) < maxW \implies 1 < |w(e) - maxW| + 1 \leq maxW + 1$. All positive weights are reordered in ascending; *iii*) $w(e) < 0 \implies |w(e) - maxW| + 1 > maxW + 1$. All negative weights keep the order but becoming positive and bigger than the biggest of the weights that were already positive.

In order to give a better understanding of this edge weight treatment, the procedure is applied in three similar instances of the problem as depicted in Fig.5.2. Despite the slightly different spatial distribution of the robots in both of three scenarios it is possible to observe that the similarities and differences of the resultant *mst* are remarkable.

Firstly, the connection between robots $TM_2$ and $TM_4$ is the strongest (closest and free of obstacles) in all scenes, and hence its weight is set to 1. In consequence, this edge has a high probability of being chosen to be included in each *mst*. By contrast, the weight of the weakest connection (between robots $TM_1$ and $TM_4$ in Fig.5.2a and Fig.5.2b or $TM_1$ and $TM_2$ in Fig.5.2c) is accordingly the highest. Besides, in all three cases is possible to see how the positive edges are reordered in ascending while the negative edges become positive keeping the original order between each other. On the other hand, regarding the differences, in column (a) the *mst* is composed only of the originally positive edges. This is absolutely expected given two facts: in the final arrangement, these edges weigh less than the originally negative and the network is connected. Consequently, there is no reason to include any of the originally negative edges. Contrarily, in both columns (b) and (c) the network presents some disconnection. Furthermore, since the relative weight between edges varies from one scene to the other, the resultant *mst* do not include the same set of edges.

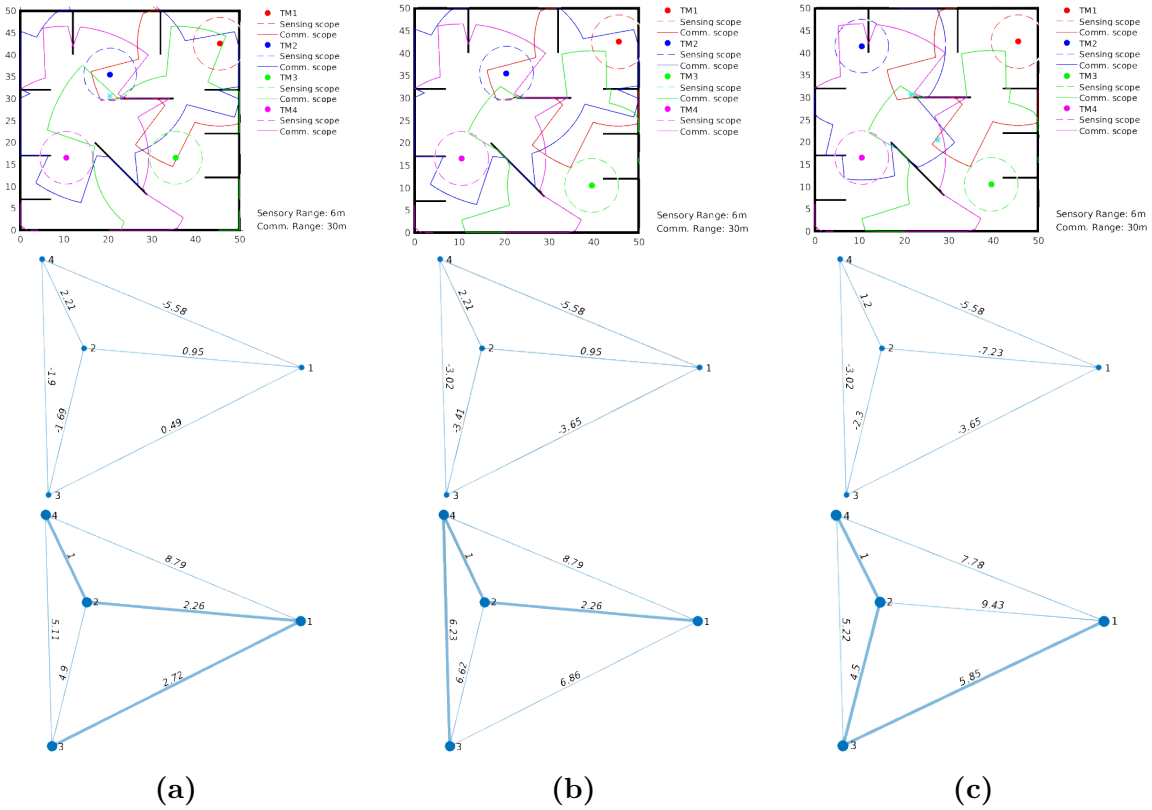



**Figure 5.2:** Reflects the main steps of the *mst* building procedure regarding the edge-weight treatment. The first row depicts the location of the tasks assigned to the explorer robot fleet whereas the corresponding graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}, \mathcal{W})$ and *mst* (bold edges) are depicted in the second and third rows, respectively. Each column represents different configurations of a 4-size fleet of explorer robots. Column (a) corresponds to a 1-connected-component explorer robot network. Column (b) corresponds to a 2-connected-component explorer robot network. Column (c) corresponds to a 3-connected-component explorer robot network.

In conclusion, this algorithm rearranges the weights of the edges in the graph –before invoking a traditional *mst* solution, in order to obtain a tree which edges represent the more promising regions where to place a relay robot along. Given that the extra effort added by the transformation task in the worst case only consists in one single round over the edges $\mathcal{E}$, the running time of this algorithm is asymptotically upper bounded by $m \log n$, with $m = |\mathcal{E}|$ and $n = |\mathcal{V}(t)|$.

### 5.4.3 Selection of Relay placement candidates

This section aims to explain in detail how the more promising cells are selected to be finally considered by the assignment procedure –in Line 5 of the Alg.3. The working assumption is that, given an edge, there exists a bundle of cells somewhere along it where would be possible to place a relay to obtain both the maximum connectivity level and signal strength

–compared with the close surroundings. The procedure operates iteratively, as a climbing-hill like searching process, choosing the best candidate so far and discarding all the rest on each step until either the candidate set is empty or it is no longer possible to overcome the current candidate. The mechanism is sketched in Alg.6.

**Algorithm 6** getBestCell.

1: **function** GETBESTCELL($(e_1, e_2), \mathcal{V}, E_{known}$)
▷ $(e_1, e_2)$ stand for the end points of a given edge.
▷ $\mathcal{V}$ stands for the node set of a tree.

2: $lstCell \leftarrow sScope(e_1, E_{known}) \bigcap sScope(e_2, E_{known})$ ▷ sScope returns the set of cells within the signal scope
3: **for** each $v$ in $\mathcal{V}$ **do**
4: **if** $v \neq e_1 \wedge v \neq e_2$ **then**
5: **if** $sScope(v, E_{known}) \bigcap lstCell \neq \emptyset$ **then**
6: $lstCell \leftarrow sScope(v, E_{known}) \bigcap lstCell$
7: **end if**
8: **end if**
9: **end for**
10: **if** $lstCell \neq \emptyset$ **then**
11: $bestCell \leftarrow rand(lstCell)$ ▷ Represents the bestCell found so far
12: $computed \leftarrow bestCell$ ▷ Represents the set of already computed cells
13: $maxSS \leftarrow \min_{\forall j \in \mathcal{V}} \Gamma_{bestCell}(d_j, N_w^j)$
14: $toCompute \leftarrow lstCell \bigcap (neighbours(bestCell) \setminus computed)$
15: $stop \leftarrow False$
16: **while** !$stop$ **do**
17: $ss \leftarrow \max_{\forall c \in toCompute} \min_{\forall j \in \mathcal{V}} \Gamma_{toCompute}(d_j, N_w^j)$ ▷ Signal strength value, see (2.1)
18: $c \leftarrow \arg\max_{c \in toCompute} \min_{\forall j \in \mathcal{V}} \Gamma_{toCompute}(d_j, N_w^j)$ ▷ Cell c where ss is reached
19: $computed \leftarrow computed \bigcup toCompute$
20: **if** ss>maxSS **then**
21: $bestCell \leftarrow c$
22: $maxSS \leftarrow ss$
23: $toCompute \leftarrow lstCell \bigcap (neighbours(bestCell) \setminus computed)$
24: **end if**
25: $stop \leftarrow ss \leq maxSS \vee toCompute = \emptyset$
26: **end while**
27: **return** $bestCell$
28: **else**
29: **return** $\emptyset$
30: **end if**
31: **end function**

To do so, from Lines 2-9, the communication model is used in order to build the initial instance of the candidate cells set (this result is analogous to the one depicted in the third row of Fig.5.1). After that, the searching process starts from a cell randomly chosen. In

general, the searching processes have to fix a bootstrap from which the search begins, and here stochasticity plays an important role in improving the performance. Recalling that the set structures –independently of the programming language– usually do not provide stochastic access to its elements, the performance of the climbing-hill like searching methods could be severely affected by a naive selection of the starting point. To see that clearer in our problem, let's consider a prominent initial candidate set as a bundle of cells in which the best candidate is located over one of the borders of the region. In such a case, if the starting cell corresponds to a cell located just in the opposite region with respect to the goal, the search will always iterate the maximum number of steps. Therefore, by randomising the selection of the bootstrap, it is possible to tackle these worst cases decreasing the searching time on average. Continuing with the algorithm steps, in Line 12 the set of already computed cells is updated. Line 13 stores the strength of the weakest communication signal perceived from the current cell. Line 14 expands the searching domain adding the cells that belong to both the candidate region and the closest neighbourhood of the current cell but have not been considered yet. Next, between Lines 16-26, the searching process is repeated until at least one of the stopping criteria is verified. Please note that in Line 27 the first condition reflects the fact that the $\min \Gamma_i(j)$ function enables to climb from the plain to the top monotonically.

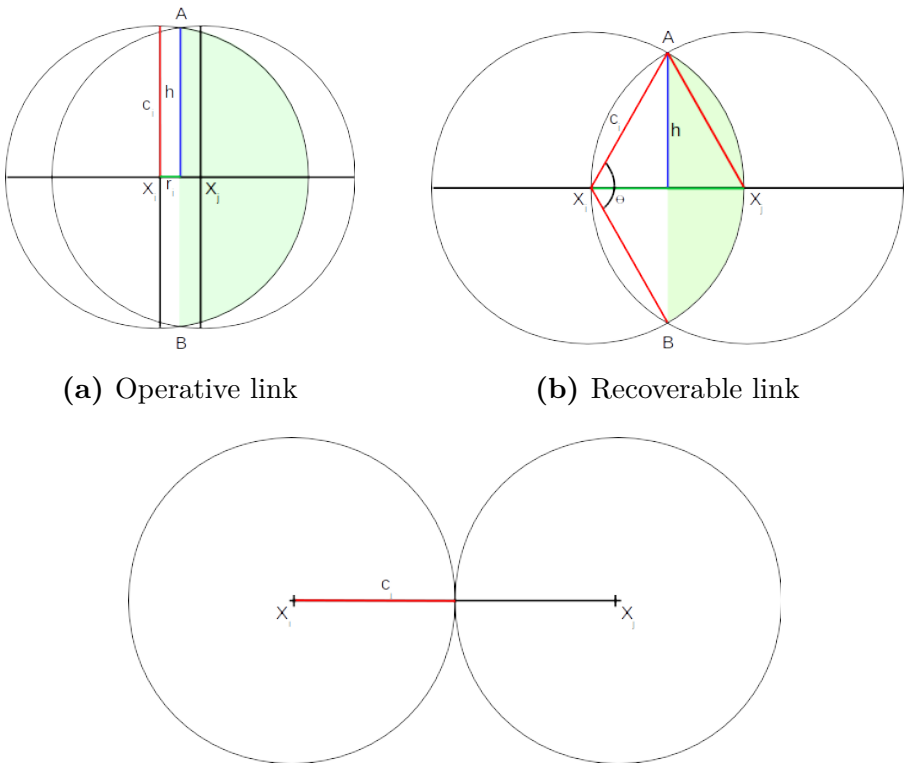
The running time of this searching algorithm mostly depends on two factors: the area under consideration (determined by the communication scope intersections) and the granularity of the underlying grid (the occupancy grid defined to model the environment in Chapter 2). Actually, both factors define the function domain: the set of cells computed between Lines 2-9. Thus, the level of incidence of both factors is analysed in the following, separately.

Firstly, it is worth noticing that by construction the area under study represents the region of the environment where the placement of a relay robot would lead to the highest level of connectivity between the relay itself and the other robots in the surroundings (measured regarding the number of connections). Besides, this area could concern three different situations such that each one may impact differently on the *getBestCell* running time. To see that, let's analyse which are the worst case in all three cases: *i*) an edge that represents an operative communication link. *ii*) an edge representing a disconnected pair of nodes when the communication link is recoverable. *iii*) an edge representing an irrecoverable communication link.
Following the corresponding worst cases depicted in Fig.5.3, the area of the overlapping regions in each worst case is derived as follows:

**Operative link (Fig. 5.3a)** Let $\bigcirc X_i c_i, \bigcirc X_j c_i$ be circles with centres $X_i, X_j$ and radius $c_i$, then taking $h = c_i$ the expression for the area $A(\cdot)$ of communication scope intersection is given by:

$$\begin{aligned} A(\bigcirc X_i c_i \bigcap \bigcirc X_j c_i) \quad &\leq \frac{A(\bigcirc X_i c_i)}{2} + \frac{A(\bigcirc X_j c_i)}{2} - 2 \cdot 2 r_i c_i = \\ &= A(\bigcirc X_i c_i) - 4 r_i c_i = \pi c_i^2 - 4 r_i c_i \end{aligned} \tag{5.1}$$

where the double of the robot radius $2r_i$ represents the minimum distance between whichever two robots.




**(a)** Operative link **(b)** Recoverable link

**(c)** Irrecoverable link

**Figure 5.3:** getBestCell worst cases. In both of three situations, the worst case is given by the maximum overlapping area between regions within the communication signal scope $c_i$ and is achieved in the absence of obstacles and the presence of only two robots. This area is decreasing conforming the distance between robot locations grows from $r_i$ (almost full overlapping) to $2c_i$ (no overlapping). (a) Connected worst-case link. (b) Represents an unconnected link that can be recovered with the placement of a relay. The overlapping area achieves its maximum when robots are precisely at a distance equal to the range of the communication devices. (c) Represents the case where the distance between robots is strictly greater than $2c_i$ and hence the placement of only one relay robot in between is not enough to recover the communication link.

**5.3b Recoverable link** Let $\bigcirc X_i c_i, \bigcirc X_j c_i$ be circles with centres $X_i, X_j$ and radius $c_i$, let $\triangle AX_iB, \triangle AX_jB$ be equilateral triangles such that $A(\triangle AX_iB) = A(\triangle AX_jB)$, then by *Pythagoras* $\left(\frac{c_i}{2}\right)^2 + h^2 = c_i^2$ holds and thus $h = \frac{\sqrt{3}c_i}{2}$ and the expression for the area $A(\cdot)$ of communication scope intersection is given by:

$$A(\triangle AX_iB) = 2\frac{\frac{c_i}{2}h}{2} = \frac{\sqrt{3}c_i^2}{4}$$

$$\bar{AB} = 2h = \sqrt{3}c_i$$

$$\theta = \angle AX_iB = \angle AX_jB = \frac{2\pi}{3} \implies A(\diamond AX_iB) = A(\diamond AX_jB)$$

$$A(\diamond AX_iB) = \pi c_i^2 \cdot \frac{\theta}{2\pi} = \frac{c_i^2\theta}{2} = \frac{\pi c_i^2}{3}$$

$$
\begin{aligned}
A(\bigcirc X_i c_i \bigcap \bigcirc X_j c_i) &= 2 \cdot (A(\diamond A X_i B) - A(\triangle A X_i B)) = \\
&= 2 \cdot c_i^2 \left( \frac{\pi}{3} - \frac{\sqrt{3}}{4} \right) = c_i^2 \left( \frac{2\pi}{3} - \frac{\sqrt{3}}{2} \right)
\end{aligned} \tag{5.2}
$$

**5.3c Irrecoverable link** Let $\bigcirc X_i c_i, \bigcirc X_j c_i$ be circles with centres $X_i, X_j$ and radius $c_i$, the expression for the area $A(\cdot)$ of communication scope intersection is given by:

$$A(\bigcirc X_i c_i \bigcap \bigcirc X_j c_i) = 0$$

From these results and since the *getBestCell* algorithm running time is a function of the number of cells to be evaluated, and this, in turn, depends on the intersection of the regions within the communication scope, the edges that represent operative communication links offer a higher difficulty in the worst case than the others. Furthermore, the number of cells for a given area is closely related with the grid granularity. As a matter of fact, the higher precision of the grid the larger number of cells, and in consequence the longer running time. On the contrary, when the grid is coarser, the same area is represented by fewer cells.

Assuming that in mobile robotics the motion planning algorithms should be safe, the occupancy grid granularity is lower bounded by $2r_i$-sized side cells (the smallest cell capable of containing a whole robot inside). This would lead to the number of cells in the worst case equal to $(\pi c_i^2 - 4 r_i c_i) \frac{1}{(2 \cdot r_i)^2} = \frac{\pi}{4} \left( \frac{c_i}{r_i} \right)^2 - \frac{c_i}{r_i}$.

Nevertheless, the way the cells belonging to an specific area are considered can make a big difference. Therefore, it is interesting to analyse how the climbing-hill search strategy may play a significant role in improving the performance, independently of the area size. In any regular grid, without any pruning policy, the growth rate (in cells) when going from any cell to another is given by the series $[c_1, c_2, c_2 + 8, ..., c_{i-1} + 8, ...]$ where $c_1 = 1, c_2 = 8, i \in [3..\infty)$. Hence, for any $k$-cell separated pair of starting and goal cells, the brute force algorithm evaluates $c_1 + c_2 + \sum_{i=3}^{k} c_i = c_1 + c_2 + \sum_{i=3}^{k} (8(k-2) + c_2) = c_1 + c_2 + (k-2)c_2 + 8 \cdot \sum_{i=3}^{k} (k-2) = c_1 + c_2 + (k-1)c_2 + 8(k-2)^2 = 1 + 8(k^2 - 3k + 3) = 8(k^2 - 3k) + 25$ cells in the worst case. On the contrary, the pruning policy used in Line 22 of the *getBestCell* algorithm provides a five-times linear growth factor evaluating in consequence only $1 + 8 + 5(k-2) = 5k - 1$ cells in the worst case. Both growth rates can be analysed from Fig.5.4.

In conclusion, a brute force searching algorithm would lead to a $\frac{\pi}{4} \left( \frac{c_i}{r_i} \right)^2 - \frac{c_i}{r_i}$ running time algorithm but, regarding the growth rate of the climbing-hill strategy derived before and since $k = \frac{\overline{AB}}{cellSide} = \frac{2c_i}{2r_i}$ represents the maximum distance –in cells- between the starting and the goal cells (see Fig.5.3a), the *getBestCell* running time is upper bounded by $5k - 1 = 5\frac{c_i}{r_i} - 1$, instead.

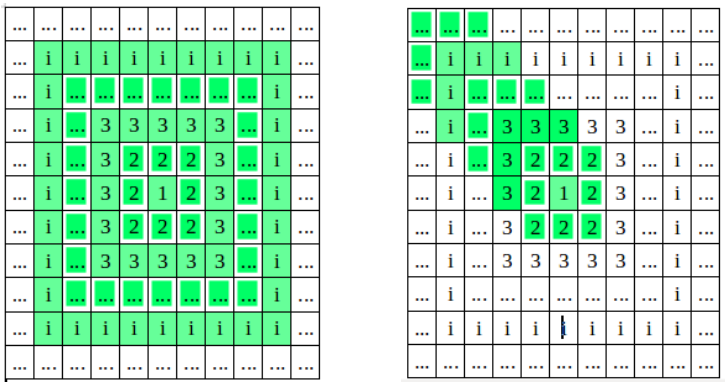



**Figure 5.4:** Grid exploration strategies. Green cells denote already evaluated cells. (a) Using the brute force search, in the worst case, every ring of cells is wholly evaluated before passing to the next. (b) Using a climbing-hill search, in the worst case, only five new cells are evaluated each time (except for the first and second rings). Actually, the worst case is reached when the best next candidate cell is placed in any corner of the next ring of cells; otherwise, there would be only three or four new cells to be evaluated.

### 5.4.4 Computational tractability analysis

In this section, the computational tractability of the *getRelayPos* algorithm is analysed. This analysis is based on the notion that an algorithm's worst-case running time on inputs of size $n$ grows at a rate that is asymptotically upper bounded by some function $f(n)$.

To this end and following a standard procedure on this matter (Kleinberg and Tardos 2006), a $T(n)$ function is defined as the worst-case running time of the *getRelayPos* algorithm on inputs of size $n$. Thus, $f(n)$ is called to be a function that for sufficiently large $n$, the function $T(n)$ is bounded above by a constant multiple of $f(n)$. More precisely, $T(n)$ is $O(f(n)) \iff \exists k > 0 \wedge n_0 \geq 0 \mid \forall n > n_0 \implies T(n) \leq k \cdot f(n)$.

Directly from Alg.4 it is possible to state that: $T(n) = T_1(n) + T_2(n) + T_3(n)$, where $T_1(n), T_2(n), T_3(n)$ are the worst-case running times of the algorithm *getConnComp* (Line 2), the algorithm *getMST* (Line 4) and the statements between Lines 6-18 of the corresponding *getRelayPos* algorithm, respectively.

As was described before in Section 5.4.1, *getConnComp* computes a weighted complete graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}, \mathcal{W})$, where $\mathcal{V}(t) = \{v \mid v \in [1 \ldots |T_e|]\}$ are the nodes, $\mathcal{E} = \{i, j \mid (i, j) \in \mathcal{V} \times \mathcal{V}\}$ are the edges and $\mathcal{W} : \mathcal{E} \to \mathbb{R} \mid \mathcal{W} = \Gamma_i(j, E_{known}(t), t)$ represents the cost function, from which is possible to know both the topology of the network and the cost matrix. The inputs are $T_e$ representing the task assignment of the explorer robots $R_e$ and $E_{known}$ representing the portion of the environment already explored by the fleet. This latter parameter is useful to compute the wall attenuation effect on the communication between every pair of robots, not affecting the running time. Conversely, the former parameter is closely related to the size of the resultant graph and then is directly responsible for the running

time of the algorithm. Hence, it is possible to state that: $n = |R_e| = |T_e| = |\mathcal{V}(t)| \implies |\mathcal{E}| = \frac{|\mathcal{V}(t)|(|\mathcal{V}(t)|+1)}{2} \implies T_1(n)$ is $O\left(n^2\right)$.

Concerning the *getMST* algorithm, the worst-case running time was previously derived in Section 5.4.2. Thus, $T_2(n)$ is $O\left(n^2 \log n\right)$.

Finally, the analysis of the worst-case running time corresponding to the statements between Lines 6-18 deserve more attention. First of all, the statements deal with two disjoint situations: the graph has only one connected component or not. Both branches operate similarly: iterating on certain edges of the *mst* and computing the corresponding *bestCell* for each one. Despite the fact that in the former case only the positive cells of the *mst* count whereas the negative edges are taken into account in the latter, in the worst case (e.g. chain like formation) both cases could have to deal with a *mst* composed of $(|\mathcal{V}(t)| - 1) = n - 1$ edges. Therefore, in the worst case, the *getBestCell* function could be applied the same times in any branch, indistinctly.

Recalling that the *getBestCell* algorithm is asymptotically upper bounded by the expression $5\frac{c_i}{r_i}$, $T_3(n)$ is $O\left(5\frac{c_i}{r_i}(n-1)\right)$, consequently.

In conclusion, when this proposed relay placement approach is followed the worst-case running time algorithm $T(n)$ is asymptotically upper bounded by $\max\{n^2, n^2 \log n, 5\frac{c_i}{r_i}(n-1)\} = n^2 \log n$ and for that reason $T(n) = O\left(n^2 \log n\right)$. This way an instance of the *Steiner tree* problem is tackled employing a polynomial time approximation algorithm instead of an exponential $n$ running time optimal solution (Garey, Graham, and Johnson 1977).

## 5.5 Closing statements

In this chapter, a novel dual role based approach for multi-robot exploration missions was introduced in detail. All related problems were carefully studied as well as the computational complexity of the proposed solutions. Besides, since the approach does not make strong working assumptions, it should be flexible enough to apply to systems which have static or dynamic role assignment policies, indistinctly.

Next, in Chapter 6 this approach is assessed and compared with some *AAMO* instances in order to verify its feasibility and to quantify the impact of using roles on top of the *AAMO* underlying proposal as well.

Chapter 6

# Dual Role Results

Contents



This chapter is devoted to assessing the *Dual Role* approach introduced in Chapter 5. Besides, it is important to compare its results to some *AAMO* instances in order to identify performance improvements or worsenings caused by the introduction of roles.

## 6.1 Simulation setup

All simulations were conducted over *MORSE* physics simulator[1] using *ATRV*-like robots equipped with laser range sensors. The more relevant simulation parameters have been shown in Chapter 4 in Table 4.1.
It is important to precise that except for the *Communication range* that depends on the device, the rest of communication factors were taken from (Bahl and Padmanabhan 2000) in attention to their strong dependency on the materials present in the environment. The values of *HO-Threshold* considered here are 20 and 10 (equivalent to 66% and 33% of the communication range $c_i$, respectively).
Besides, since the communication conditions are non-ideal all runnings only concern fleets integrated with multiple robots (explicitly avoiding the single robot case because the communication conditions do not make any difference in it).
In all simulations localisation and low-level motion control are taken for granted.

[1] `www.openrobots.org/morse/doc/stable/morse.html`

### 6.1.1 Scenarios

The terrain covers an 80x80 $m^2$ flat surface with static obstacles (e.g. walls, including the outer perimeter) following a maze-like spatial distribution. This *Maze* scenario was initially proposed as a benchmark in (Yan et al. 2015) and has been presented in this document in Chapter 4 (see Fig.4.1c).

### 6.1.2 Relay Robot Architecture

In this section, the software architecture of a robotic relay agent is presented. In Fig.6.1 the main components are depicted.

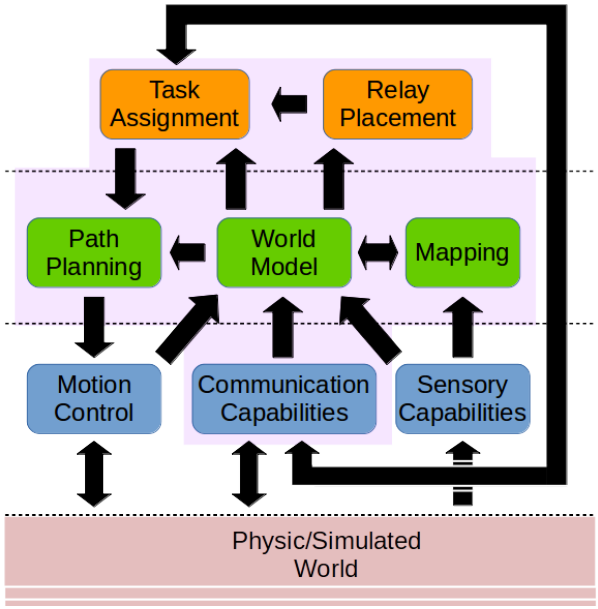



**Figure 6.1:** Relay Agent Architecture. First and second layers are equal to the ones in Fig.4.2. The third layer includes Relay Placement related algorithms. Components on the shadowed zone were totally developed during this work.

Please note that the unique difference between the explorer and relay robots architecture resides in the third layer and is related with the *Relay Placement* component (first and second layers are exactly the same as these of the explorer agents described above in Section 4.1.2). Namely, this component –as its name suggests– is in charge of implementing the *Relay placement* algorithm introduced in Sec.5.4.1. To do so, it uses the information provided by the *World Model* component (e.g. location of the tasks already assigned to explorer robots and the map of the portion of the environment already known by the robot fleet). Once the relay placement problem is solved, the results are passed to the *Task Assignment* component in order to determine the best distribution of places to relay robots.

## 6.2 Figure of merits

The performance of the approach is assessed in terms of the same indicators presented in Sec.4.2 of Chapter 4.

## 6.3 Dual Role assessment

This section aims to study the impact of assigning different roles to robots on the performance of an *AAMO*-driven multi-robot system when it is asked to explore an environment under non-ideal communication conditions. For simplicity reasons, a static role assignment was adopted. So that two instances of the *Dual Role (DR)* approach are considered, *DR:10:1R* and *DR:10:2R*. In both *DR* setups, the *HO-Threshold* value is set to 10 due to the *AAMO:10* approach has shown the best TT and PL performance (see Chapter 4). This way, it is expected to keep the best of both worlds: TT and PL performance supported by the *HO-Threshold* value and good connectivity during the exploration mission through the relay presence.
In one setup, *DR:10:1R*, there exists only one relay robot while in the other, *DR:10:2R*, there exists two of them. Nevertheless, since in the 2-sized fleet is useless to have any relay robot, both setups are only applied from 3-sized fleets on. Besides, while the *DR:10:1R* is applied in all remaining cases, the latter is applied only in the largest fleet regarding *i*) the relay role only pursues connectivity goals. *ii*) all experiments consists of constant fleet sizes. Consequently, in any case, having a relay implies not having an explorer and would be a source of considerable worsening in the small size fleets case.

The assessment of *DR* approach concerns both the analysis of their different performance indicators when varying the fleet size and its comparison against the results obtained by pure *AAMO* driven fleets.

### 6.3.1 Collected data

At least ten realistic software-in-the-loop simulations were executed on the *Maze* scenario (see Fig.4.1c in Chap.4). All collected data is presented in Table 6.1 and are organised obeying the following scheme. The columns refer to (from left to right): **F**igure of **M**erits (FM), *Approach*, where *DR:10:nR* and *AAMO:HO-Th* stand for ***D**ual **R**ole:10* composed of one or two ***R**elay* robots and ***A**uto-**A**daptive **M**ulti-**O**bjective:**H**uman**O**perator-**Th**reshold*, respectively; and fleet size $|R|$.

### 6.3.2 Effectiveness assessment

Starting with the effectiveness, the CR indicator shows that the DR approach can adequately accomplish the *Maze* exploration –regardless neither the fleet size nor the number of relay

**Table 6.1:** *DR* Results obtained under non-ideal communication conditions on Maze environment.

| FM | Approach | $\|R\|$ = 2 | | 3 | | 4 | | 5 | | 6 | | 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD | AVE | StD |
| TT | DR:10:1R | N/A | N/A | 1217 | 76.8 | 1083 | 127.4 | 804 | 90.1 | 719 | 93.5 | 633 | 69.4 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | 941 | 99.3 | 751 | 47.4 | 619 | 52.9 |
| | AAMO:20 | 1292 | 87.8 | 1100 | 88.1 | 900 | 98.7 | 767 | 104.0 | 721 | 96.2 | 661 | 108.2 |
| | AAMO:10 | **1137** | 85.3 | **960** | 123.1 | **774** | 94.3 | **620** | 76.9 | **587** | 53.4 | **514** | 23.1 |
| PL | DR:10:1R | N/A | N/A | 2479 | 146.3 | 2666 | 256.1 | 2440 | 247.4 | 2357 | 234.6 | 2590 | 310.6 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | 2813 | 386.8 | 2443 | 168.0 | 2634 | 188.7 |
| | AAMO:20 | 1726 | 114.4 | 2086 | 222.5 | 2243 | 236.1 | 2394 | 252.8 | 2222 | 281.5 | 2782 | 376.1 |
| | AAMO:10 | **1542** | 119.9 | **1859** | 225.6 | **1982** | 215.5 | **2007** | 221.7 | **1999** | 220.6 | **2279** | 248.5 |
| CR | DR:10:1R | N/A | N/A | 99.3 | 0.41 | 99.2 | 0.26 | 99.1 | 0.26 | 99.2 | 0.26 | 99.2 | 0.30 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | 99.2 | 0.23 | 99.2 | 0.18 | 99.1 | 0.10 |
| | AAMO:20 | 99.1 | 0.26 | 99.2 | 0.28 | 99.2 | 0.28 | 99.3 | 0.35 | 99.3 | 0.37 | 99.4 | 0.37 |
| | AAMO:10 | 99.3 | 0.32 | 99.3 | 0.40 | 99.3 | 0.29 | 99.3 | 0.29 | 99.3 | 0.23 | 99.1 | 0.05 |
| OSR | DR:10:1R | N/A | N/A | 4.16 | 3.08 | 3.67 | 1.93 | 4.84 | 1.91 | 4.80 | 1.44 | 6.48 | 2.05 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | **3.53** | 0.19 | 4.86 | 1.26 | 5.56 | 0.28 |
| | AAMO:20 | **1.85** | 1.57 | **2.68** | 1.12 | **3.53** | 2.29 | 4.56 | 1.46 | 4.83 | 1.52 | 5.92 | 0.83 |
| | AAMO:10 | 2.47 | 2.98 | 3.99 | 3.01 | 3.94 | 1.96 | 5.04 | 2.34 | **4.49** | 0.91 | **5.44** | 0.01 |
| DLR | DR:10:1R | N/A | N/A | 65.1 | 11.66 | 56.5 | 16.62 | 51.5 | 15.76 | 59.1 | 13.16 | 52.8 | 19.05 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | **45.0** | 12.85 | 52.8 | 15.73 | 49.8 | 7.92 |
| | AAMO:20 | **40.6** | 9.32 | **54.0** | 9.43 | **44.1** | 13.29 | 51.8 | 10.11 | 48.4 | 7.60 | **46.5** | 13.92 |
| | AAMO:10 | 61.3 | 9.49 | 68.6 | 13.24 | 54.9 | 9.53 | 62.8 | 11.32 | **47.4** | 11.07 | 46.6 | 14.28 |
| MDLR | DR:10:1R | N/A | N/A | 29.6 | 8.03 | 20.6 | 11.53 | 24.2 | 10.01 | 22.8 | 14.62 | 21.7 | 11.38 |
| | DR:10:2R | N/A | N/A | N/A | N/A | N/A | N/A | **23.1** | 13.32 | 17.7 | 5.21 | **17.2** | 7.99 |
| | AAMO:20 | **17.8** | 10.47 | **21.7** | 8.96 | **15.9** | 7.61 | 24.2 | 12.99 | **16.2** | 4.77 | 18.0 | 8.88 |
| | AAMO:10 | 27.2 | 10.11 | 33.6 | 10.95 | 19.9 | 5.72 | 24.9 | 9.15 | 24.8 | 13.81 | 22.9 | 8.61 |

robots, achieving high levels of coverage. In Fig.6.2 the series of CR values corresponding to different fleet sizes are shown.

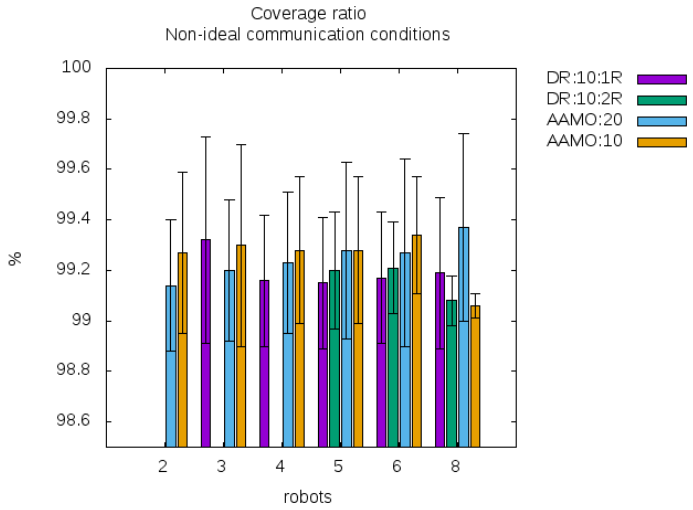



**Figure 6.2:** *DR* Coverage ratio (CR).

### 6.3.3 Efficiency assessment

Regarding the efficiency, all charts allow comparing the performance of the *DR* instances from two different perspectives. One refers to the comparison between the results obtained by $n$-sized fleets driven by the *AAMO* instances and the results obtained by $n$-sized fleets composed of $n_e$ explorer robots and $n_r$ relay robots driven by the *DR* approach, given that $n = n_e + n_r$. This perspective will be referred to as the *Substitution* perspective.
The other one, referred to as *Addition* perspective, corresponds to the comparison between the results obtained by $n_e$-sized fleets driven by the *AAMO* instances and the results obtained by fleets composed of $n_e$ explorer robots plus $n_r$ relay robots driven by the *DR* approach.

Concerning TT, the DR approach also benefits from adding robots to the fleet (as was expected). This result can be seen in Fig.6.3. The trend of TT is monotonically decreasing in both *DR* instances.
From the *substitution* perspective, both *AAMO* instances overcome the *DR:10:1R* instance on the smaller fleets showing that the substitution of one explorer by one relay robot implies some completion time degradation. Nevertheless, concerning 6-sized and 8-sized fleets, the *DR:10:1R* instance outperforms the *AAMO:20* instance. Similarly, the *DR:10:2R* instance presents completion time degradation concerning the results of the 5-sized and 6-sized fleets but outperforms both the *AAMO:20* and the *DR:10:1R* instances concerning the results of the 8-sized fleets. Thus, it would be possible to claim that, on bounded environments, the worsening caused by the substitution of explorer robots by relay robots may be compensated by the higher efficiency achieved by explorers when the fleet size is big enough.

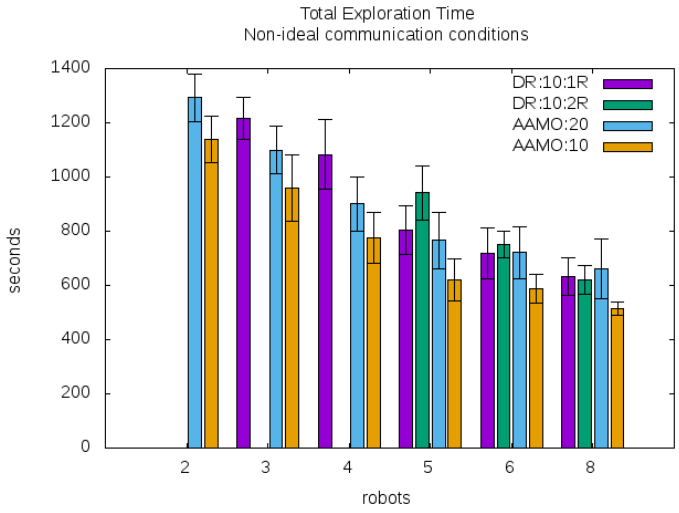



**Figure 6.3:** Total exploration time (TT).

On the other hand, when comparing TT from the *Addition* perspective, the contribution of relay robots is more evident. About the *AAMO:20*, one may claim, for instance, that having one relay and two explorers is worse than having three explorers but is better than having only two explorers. Furthermore, this relation holds for all fleet sizes and both *DR* instances. Note that under this additive perspective, the results achieved by the *DR:10:2R* instance are

better than the *AAMO:20* instance and even better than the *DR:10:1R*. For instance, having three explorer robots plus two relay robots is better than having three explorers plus only one relay –concerning the addition of a second relay robot, and also better than having three explorers without any relay robot –concerning the addition of two relay robots. Similarly, the *DR:10:2R* instance results slightly better or equal to the *AAMO:10* instance.
This last statement might sound obvious, but it perfectly could happen differently. The presence of relay robots would help to support the fleet connectivity but could negatively affect the mobility of the explorers too. Fortunately, it seems not to be the case here. The addition of relay robots allows the explorers to be more efficient, probably due to the fewer effort demanded to keep themselves connected. Thus, concerning the TT results, it is possible to conclude that when there is no need for substitution (unlimited or simply large availability of robots), the *DR* approach is clearly advisable.

With respect to the PL indicator (see Fig.6.4), the results obtained when the fleets are driven by the *DR* approach show a similar pattern to the one concerning the results obtained under the *AAMO* approach but more prominent. It consists of a first increasing stage, a decreasing stage in between and another increasing stage at the end of the PL trend.

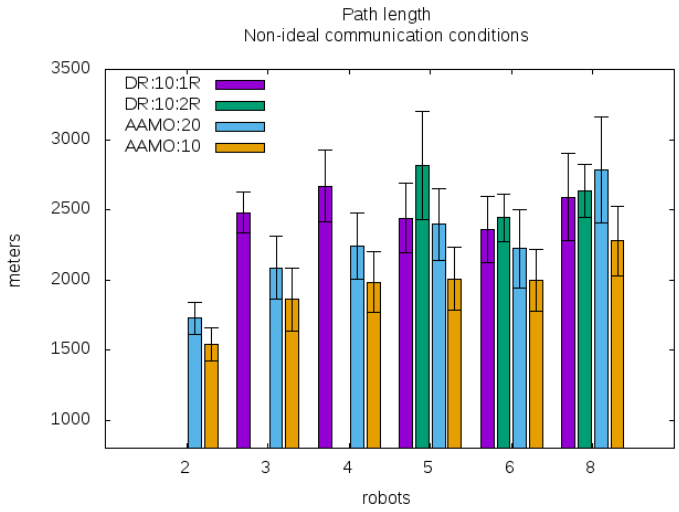



**Figure 6.4:** Path length (PL).

The results corresponding to the smaller fleets are noticeable and could suggest that the smaller fleet size, the more frequently the relay has to change its location, traversing longer distances to keep the fleet connected, which is probable and would explain the high PL values concerning 3-sized and 4-sized fleets and the degradation with respect to the *AAMO* instances from both *Substitution* and *Addition* perspectives as well. This hypothesis was confirmed after checking that the distance travelled by the relay robot is equivalent to the degradation presented in both 3-sized and 4-sized fleets.

The PL indicator is particularly important because it indirectly speaks about energy usage. Hence, these results could be advising on how troublesome could be these fleet compositions, regarding energy supplies. On the contrary, in 6-sized and 8-sized fleets –from both

perspectives– the relay robot relocations could somewhat imply shorter distances. Namely, the 8-sized fleets driven by both *DR* instances outperform the *AAMO:20* instance. In conclusion, the *DR* approach seems to be advisable only for large fleets. Concerning its deployment, the energy supply issues would deserve special attention.

Next, in order to analyse how the whole fleet is affected by disconnections, the DLR results are shown in Fig.6.5. Firstly, the results obtained under the *DR:10:1R* instance shows a particular pattern. A kind of zig-zag with two decreasing sections. The first one until 5-sized fleets and the other from 6-sized fleets.

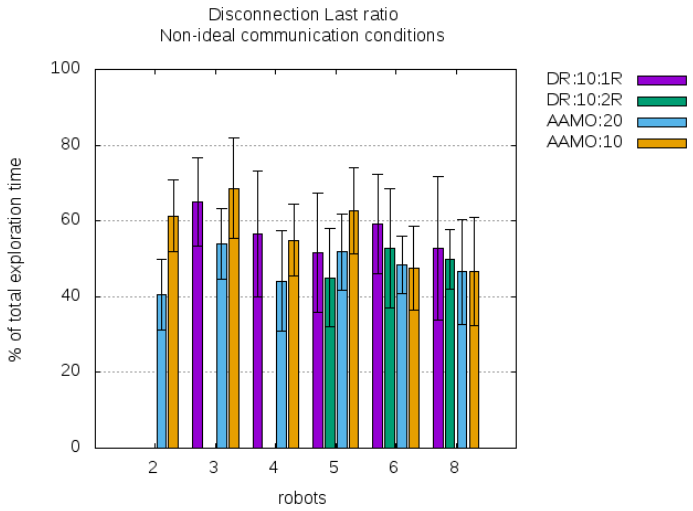



**Figure 6.5:** Disconnection Last rate (DLR).

From the *Substitution* perspective, in the first section (3, 4 and 5-sized fleets) the *DR:10:1R* instance outperforms or resembles the performance of the *AAMO:10* instance while is outperformed by the *AAMO:20* instance, regardless the fleet size. This suggests that the influence of a bigger *HO-Threshold* is enough to make the fleet more cohesive compared to the performance caused by a lesser *HO-Threshold* plus the presence of only one relay robot.
Concerning the first results obtained under the *DR:10:2R* execution (5-sized fleets), both *AAMO* instances are outperformed. Conversely, the *DR:10:2R* instance presents severe degradation when executed by 6-sized or 8-sized fleets while both *AAMO* instances continuously enhance its performances from 5-sized fleets. This kind of abrupt loss of performance is not possible to be explained just observing the information contained in this chart and, for that reason, will be analysed jointly with the network topology and the MDL indicator as well.

Concerning the *Addition* perspective and referred to the first section of the chart, the results obtained by the smaller *DR:10:1R* driven fleets are better than or similar to the one obtained under the *AAMO:10* instance, suggesting that the addition of one relay robot enhance the connectivity level of these fleets.

In order to get a closer notion about the interaction held between robots along the exploration, in the following the DLR indicator and the network topology are analysed together.

To this end, Fig.6.6 is devoted to showing the number of connected components present in the network, for different fleet sizes averaged over time.

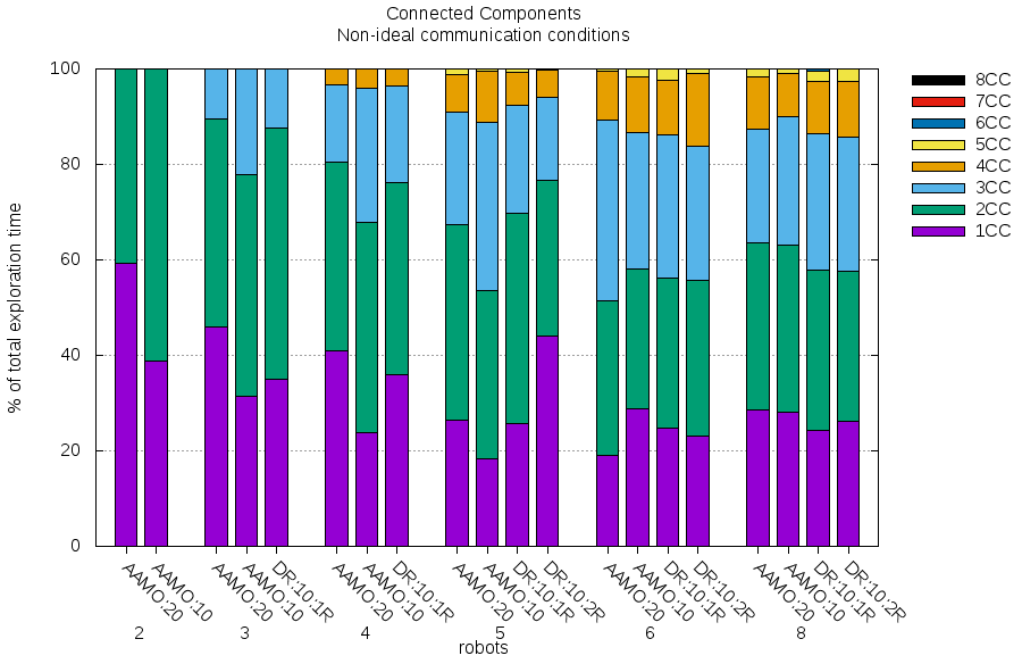



**Figure 6.6:** *DR* Network topology composition averaged over time on different-sized fleets. Depending on the number of connected components and the fleet size, it is possible to study the existence of sufficient conditions to fall into isolation situations. For instance, for a 3-Robot fleet, the 2CC or 3CC topologies imply having at least one robot isolated while for a 5-Robot fleet this implication is related to 3CC, 4CC or 5CC topologies, and so on.

Firstly, note that this information reveals complementary aspects. Taking as an example the *DR:10:1R* instance on a 5-Robot fleet, note that the DLR is about 50% while the percentage of TT that the network topology is composed either of 1CC or 2CC is higher than the DLR. This fact evidence that during the 2CC composition periods the fleet split up into 4-sized and 1-sized robot subgroups, leaving one robot isolated. This reveals an opportunity to improve the behaviour of the fleet when being under such conditions (that are also corroborated in fleets of other sizes).

The first part of the chart (smaller fleets) shows that *DR:10:1R* produce a slight improvement on the network topology compared to the *AAMO* instances to which the 1CC stack is monotonically decreasing in size as the fleet size increase. It is worth noticing that under the *DR:10:1R* instance the 1CC stack is always higher than the corresponding *AAMO:10*, even becoming very similar to the one corresponding to *AAMO:20* on the 5-Robot fleet.

Another insight is obtained by looking at the upper part of this chart where the stacks corresponding to the highest number of connected components are plotted. It reveals that the addition of new robots does not necessarily imply that the network topology is less cohesive. The number of connected components $nCC$ increase slower than the fleet size $n$. Although this behaviour may be explained in a large extent by the fact that the *Maze* is not an unbounded

scenario, it is remarkable that the network topology of the *DR:10:1R* driven fleets are at least as cohesive as the ones of the *AAMO:20*. Regarding this aspect, the addition of one relay compensates the smaller *HO-Threshold*.

Continuing with the network topology analysis, it is possible to observe that the fleets of size 3,4 and 5 present until 3CC, 4CC and 5CC network topologies, respectively. Hence, in each case, during a certain period, all robots have been disconnected. Although these periods are the shortest (the CC placed on top are the smallest), the fact that even the relay robot has been unconnected really stands out. This leads to considerate that sometimes the relay robot must move away from the fleet –traversing part of the path unconnected due to the spatial distribution of walls– in order to relocate itself in a better place. In Fig.6.7 an example scene is depicted.

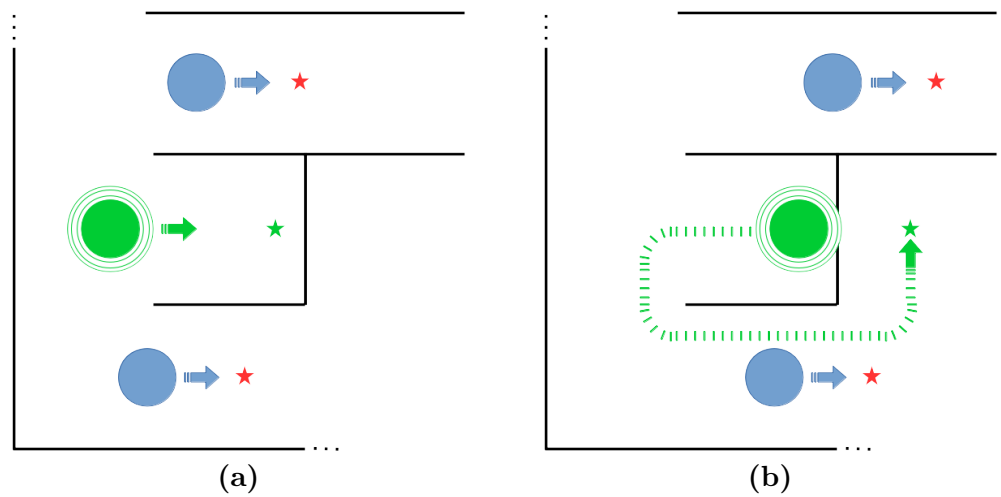



**Figure 6.7:** Relay disconnection situations. In the scenes, there are two explorer robots (blue circles) and one relay robot (green circle). Current targets are denoted by red and green stars, respectively. The assignment is denoted by arrows. (a) Robots are currently connected in the same connected component. Relay robot can select a new place keeping the fleet connected in the close future despite the explorer movements. (b) The assignment of explorer robots demands the relay robot to plan a path during which the fleet split up circumstantially into three connected components.

Finally, because the network topologies of both *DR* and *AAMO* approaches are very similar concerning 6-sized and 8-sized fleets, it is possible to apply this previous assumption to explain the gap of DLR performance too. This way, one would say that the sudden worsening of the DLR results obtained on 6-sized and 8-sized fleets when driven by any of the *DR* instances is caused by adverse movements of the relay in its eagerness of support the fleet connectivity. In order to check this hypothesis, the DLR of the relay robots was measured during 6-sized and 8-sized fleet explorations validating the explanation given above.

Concerning the individual isolation level, the trend of MDLR is shown in Fig.6.8. On the one hand, the *DR:10:1R* shows a kind of oscillatory pattern resembling the patterns presented by the *AAMO* instances. On the contrary, the *DR:10:2R* shows a decreasing trend.
From the *Substitution* perspective the *DR:10:1R* resembles or even outperforms the *AAMO:10*

instance. On the other hand, the *DR:10:2R* resembles or even outperforms both *AAMO* instances.
From the *Addition* perspective both *DR* instances outperform both *AAMO* only when the even-sized fleets results are compared to the one corresponding to fleets which size is immediately smaller, consistently with the oscillatory patterns of the *AAMO* instances.
Besides, from both *Substitution* and *Addition* perspectives, having two relay robots is better than having only one.

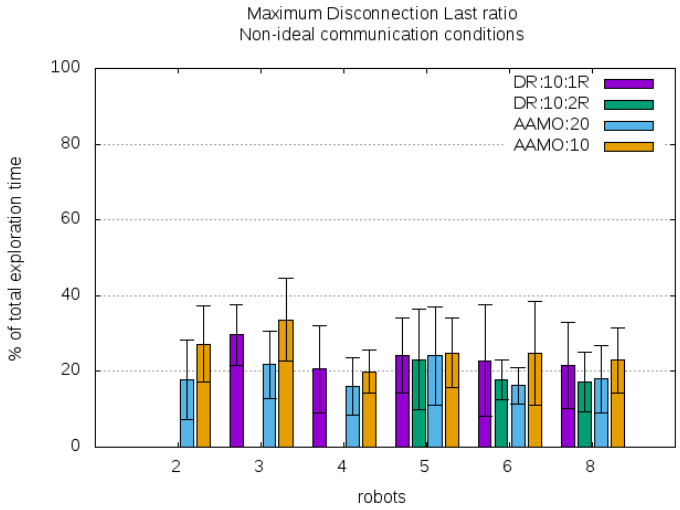



**Figure 6.8:** Maximum Disconnection Last Ratio (MDLR).

In relation to the OSR indicator (see Fig.6.9), both *DR* instances show similar results to the very good results achieved by the *AAMO* instances.

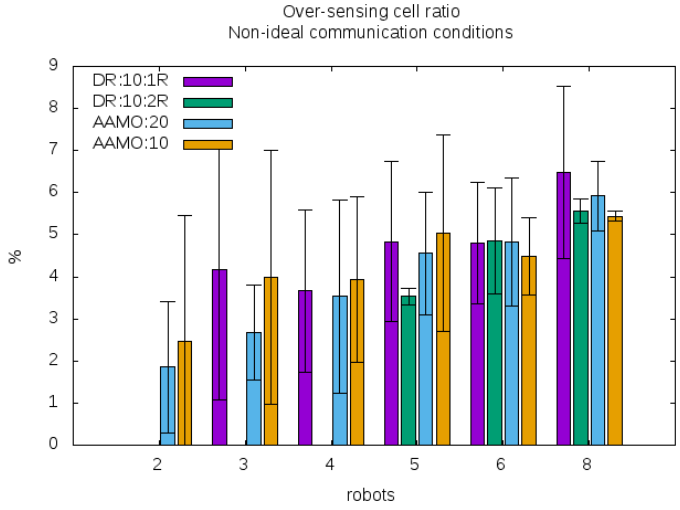



**Figure 6.9:** *DR* Over-sensing cell ratio (OSR) under non-ideal communication conditions.

### 6.3.4 Conclusions

Concerning the *Maze* scenario, the conclusions of this section are: *i*) The *DR* approach has demonstrated to be effective in tackling the multi-robot exploration problem. *ii*) Concerning the performance on completion time and connectivity aspects, the *DR:10:2R* instance has shown better indicators especially on large fleets. *iii*) Furthermore, when there is no need for substitution (unlimited or simply large availability of robots) the *DR* approach is advisable *iv*) On the contrary, when the *DR:10:1R* is employed to drive small fleets some potential risks on energy usage have been detected. *v*) Additionally, on several fleet sizes, both instances of the *DR* approach have shown significant worsening on one or other indicator that also deserves some attention.

To sum up, the *DR* approach has demonstrated to be a valuable alternative to enhance the promising results shown by the *AAMO* approach. Unfortunately, the use of a static role assignment mechanism could have affected the potential benefits of adding roles on top of the *AAMO* approach. From the previous analysis and after having identified some particular situations, there is room to claim that a dynamic role assignment mechanism would be more suitable for supporting connectivity without worsening the completion time performance. This kind of assignment not only would contribute to keeping the presence of relay at a minimum number (assigning this role only when it is strictly needed) but also would help to choose the best candidates for playing the role itself (e.g. assigning the role to best placed robots in each time). An example scene is depicted in Fig.6.10.

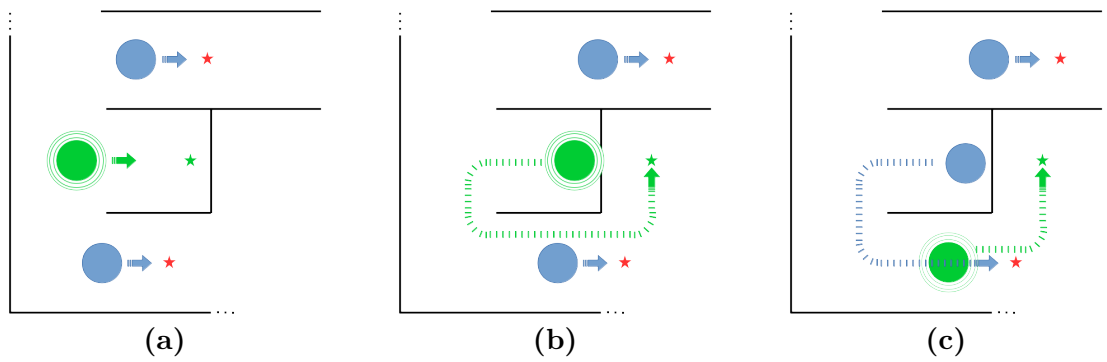



**Figure 6.10:** Relay reassignment situations. In the scenes, there are two explorer robots (blue circles) and one relay robot (green circle). Current targets are denoted by red and green stars, respectively. The robots-to-tasks assignment is denoted by arrows. (a) Robots are currently connected in the same connected component. Relay robot can select a new place keeping the fleet connected in the close future despite the explorer movements. (b) Static Role Assignment. The shortest path between the relay robot location and the best place to support the fleet connectivity, makes it move away from the fleet, splitting it up circumstantially into three connected components. (c) Dynamic Role Assignment. The assignment of relay role can change whenever it is needed in order to support exploration and connectivity at once.

Next, in Chapter 7 the thesis is concluded, and some possible future directions are advised.

Chapter 7

# Conclusions

The exploration problem is a fundamental subject in autonomous mobile robotics. In this thesis, the multi-robot exploration problem in communication-restricted environments has been addressed. It consists of exploring initially unknown environments with a fleet of autonomous mobile robots subject to constrained communication conditions.

The survey on multi-robot exploration –summarized in Chapter 1– highlights the main contributions as well as several drawbacks in the state-of-art approaches. It suggests that, for example, there is room to enhance the event-based connectivity strategies, particularly the connectivity-regaining policies that are strongly related to the definition of both reconnection frequency and meeting points.

This chapter aims to describe the major contributions of this work on this subject and to discuss perspectives and future work directions as well.

## 7.1 Contributions

The main contributions of this work refer to two novel decentralized and fully asynchronous proposals to address the multi-robot exploration problem.

### 7.1.1 Task assessment problem

The proposed *Auto-Adaptive Multi-Objective (AAMO)* approach follows a multi-objective assessment strategy where the tasks under consideration are assessed regarding two objectives: the cost associated with the corresponding shortest path and the connectivity level each task location can offer to robots at arrival time. The multi-objective strategy is implemented employing a weighted sum that trades travelling cost off for connectivity levels. Up until here, all these concepts are quite standard being present in several state-of-art approaches. Nevertheless, in this work: *i*) connectivity awareness ability is given to the robots by modelling attenuation effects that commonly affect the communication signal strength. *ii*) the weights of these potentially conflicting objectives are derived from formal analysis instead of a training stage, making the system more adaptable to different environments. *iii*) the human operator is asked to use his application-field expertise to play a part in the task assessment process by setting a distance threshold until which the tasks that preserve or enlarge connectivity are

preferred over the rest. All this leads to a more flexible system where the robots can deal with communication constraints adjusting the weights of each objective independently of any scenario in a more intuitive manner and saving a lot of training time too.
All existence and correctness proofs conducted on the task selection procedure support the fact that the robots are always capable of auto-adapting the objectives weights in order to select the tasks accordingly with the human-operator criterion. In conclusion, this task assessment approach may be very advantageous considering its ease of deployment.

### 7.1.2 Task allocation problem

Concerning the tasks-to-robots distribution algorithm all previous proposals explicitly avoid the combinatorial blow-up of allocation complexity using different heuristics. Nevertheless, heuristic-based approaches make assumptions that cannot be verified at all times. In consequence, when the heuristic fails the robots choose suboptimal distributions.
Taking into account this limitation and since the number of possible distributions depends on both the number of available tasks and the number of robots making a decision, the proposal here presented computes optimal distributions based on more general assumptions such as *i*) robots can implicitly coordinate their actions. *ii*) asynchronism may keep the number of simultaneous decision making at small values. *iii*) pruning the furthest tasks (out of the scope defined by the human operator –*HO-Threshold*) does not prevent the computation of optimal tasks-to-robots distribution.
Consequently, the robots can compute the same optimal tasks-to-robots distribution in a short time as long as they consider a subset of the available tasks (the tasks within the *HO-Threshold*). Implicit coordination is reached following a proper decentralized approach. Asynchronism is taken as a natural way to achieve efficiency: avoiding waiting times (robots do not wait for others) and making locally optimal decisions linearly computable most of the time.

### 7.1.3 Connectivity maintenance problem

While all event-based connectivity approaches consist in the execution of regaining-connectivity actions in the presence of specific events (e.g. typically disconnections, whenever it happens or periodically after a certain amount of time), the *AAMO* approach integrates a less restrictive connectivity strategy where the robots are motivated but not compelled to regain connectivity. When selecting their targets, the robots are always considering the opportunity cost of keeping connected or regaining connectivity, implicitly. Furthermore, in reconnection cases, the task location becomes the meeting point itself eliminating the need for rendezvous policy implementation and, maybe more important, avoiding deviations from natural paths. This way, the policy is utterly transparent to the eyes of the external observer: every time it is possible to explore and keep or enlarge connectivity level the robots will choose this option. On the contrary, when it is not, they merely behave guided by a pure path-cost exploration. Particularly concerning the efficiency related to both completion time and connectivity level

maintenance, the approach is capable of decreasing the last of disconnection periods without a noticeable degradation of the completion exploration time, appearing as an intermediate solution that presents much better completion time performance than the most restrictive event-based connectivity approaches and better connectivity level along exploration than the approaches that do not take care about communication issues.
In conclusion, in application fields where strong communication requirements do not condition the mission, this approach represents a proper option for coping with real communication constraints –always present in practice, being more fault tolerant and still having good performance, all at once.

Additionally, the proposed *Dual Role (DR)* approach has shown good results demonstrating to be a valuable alternative to enhance the promising results shown by the *AAMO* approach. Assigning different roles –explorer and communication relay, helps to improve the benefits of the *AAMO* approach, for instance, when large fleets are employed. Unfortunately, the use of a static role assignment mechanism during the experimental tests could have affected the benefits of adding roles on top of the *AAMO* approach.

### 7.1.4 Relay placement problem

Based on the communication model, a novel polynomial-time relay placement approach for multi-robot exploration missions was introduced in detail. All related problems were carefully studied as well as the computational complexity of the proposed solution. Besides, since the approach does not make strong working assumptions, it should be flexible enough to apply to systems which have static or dynamic role assignment policies, indistinctly.

## 7.2 Future research directions

In this thesis, novel approaches have been explored in order to address the multi-robot exploration problem. New research questions have arisen along this stage leaving, as a result, several opportunities to improve the developed system.

### 7.2.1 Indoor Environments

Although the environments employed in simulations are proposed as benchmarks, it would be beneficial to check the validity and performance of the proposed approaches on a broader variety of scenarios. Large office-like environments would be interesting to put the system on more realistic situations like mapping buildings where larger fleets could be employed too.

### 7.2.2 Heterogeneous fleets

Since the robot model defined can support robots with different characteristics, exploiting heterogeneity could be an auspicious research direction. Integrating a fleet with heterogeneous robots (e.g. different in size, sensory and motion capabilities) could enhance the skills of the fleet. For instance, given their greater mobility, UAVs could help the fleet to keep connected by playing the relay role, while small terrestrial robots could be the key to get into access-restricted spaces. This last case can be easily implemented by adapting the *path utility* function (3.2) making the utility of the inaccessible *tasks* equal to zero, as follows:

$$\overline{\Delta} = \overline{d} - \underline{d}$$
$$\overline{d} = \max \left\| X_i, T_j \right\|_{sp}, \forall j$$
$$\underline{d} = \min \left\| X_i, T_j \right\|_{sp}, \forall j$$
$$\Delta_i(T_j) = \begin{cases} \left\| X_i, T_j \right\|_{sp} - \underline{d} & \\ \left\| X_i, T_j \right\|_{sp} = \min\limits_{\substack{wp_k \in E_{known} \forall k \in \{1 \ldots e\} \\ wp_1 = X_i, wp_e = T_j}} \sum_{k=1}^{e-1} \left\| wp_k - wp_{k+1} \right\|_2 & \text{if accessible} \\ \overline{\Delta} & \text{otherwise} \end{cases}$$

### 7.2.3 Dynamic *HO-Threshold*

The identification of different exploration stages (dispersion and convergence, see Sec.4.4.2 in Chapter 4) leads to consider the advantage of varying the *HO-Threshold* value during the exploration in order to make the response of the system more suitable for the specific needs of each stage. The transition between them is remarkable and could be detected by the fleet automatically. Therefore, the fleet would be able to auto-adapt the human-operator criterion building its own criterion on-line. In such a case, the human operator could be asked to set a range for the distance threshold, instead of a single value.

### 7.2.4 Dynamic role assignment

From some situations identified during the *DR* approach assessment, there is room to claim that a dynamic role assignment mechanism would be more suitable for supporting connectivity without worsening the completion time performance. This kind of assignment not only would contribute to decreasing the presence of relay (assigning this role only when it is strictly needed) but would help to choose the best candidates for playing the role itself (e.g. assigning the role to best placed robots).

### 7.2.5 Open systems

Open systems are characterized as the multi-agent systems where agents can join or leave the group after the startup of the system. In the context of a multi-robot system guided by a

dual-role approach (with dynamic or static role assignment) it is interesting to think of having the possibility of delivering new robots from the base station (e.g. when more explorers or relay robots are needed) or taking them in from the environment (e.g. when they are no longer needed or in case of battery demise).

### 7.2.6 Reproducible research

During this work, accessing the implementation of other approaches represented a great difficulty. Actually, the ones to which the new proposals were compared, were implemented from scratch by the author. Besides, only in few cases, the reports concerns benchmark scenarios. Thus, achieving a fair comparison represent a challenge itself. Due to this, making the code and the testbeds presented here in this document available could be valuable for other researchers.

### 7.2.7 Dual Role comparison

Related with these previous research directions, it would be also very important to implement some state-of-art dual-role approaches in order to conduct a comparative assessment (on the same environments, using the same robotic units, measuring the same indicators, with the same setups, etc) between the *DR* approach here introduced and approaches of the same nature.

### 7.2.8 Cooperative Path Planning

It is well known that every autonomous mobile robotic system needs some collision avoidance mechanism in order to move around in the environment in a safe way. In multi-robot systems, other teammates themselves may become obstacles in the path planned by a robot. Despite all robots should be able to avoid collisions, they can get stuck anyway (e.g. when falling into deadlocks). Since this situations are typically costly, preventing them is advisable. In such cases, provide the fleet with a cooperative path planning mechanism may redound in high travelling performances and in decreasing the risk of damage due to collisions between robots. To this end, robots should share the planned paths between each other in order to try to avoid collisions as much as possible explicitly.

### 7.2.9 Experiments on real fleets

At last but not least, executions on real systems are also planned. Despite the goodness of any simulator, many important details escape from its scope. Besides, the proposed approaches are designed to serve as solutions for real-world applications so that it is imperative to verify its feasibility in real scenarios. Regarding the equipment availability of the involved laboratories, the candidate platforms would be either IRobot or KheperaIII units.

# Bibliography


Amigoni, Francesco, Jacopo Banfi, and Nicola Basilico (2017). "Multirobot Exploration of Communication-Restricted Environments: A Survey." In: *IEEE Intelligent Systems* 32.6, pp. 48–57 (cit. on pp. 6, 7, 22, 49, 50).

Bahl, Paramvir and Venkata N Padmanabhan (2000). "RADAR: An In-Building RF-based User Location and Tracking System." In: *Proc. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc.* 00.c, pp. 775–784. arXiv: 1106.0222 (cit. on pp. 20, 46, 87).

Banfi, Jacopo et al. (2016). "Asynchronous multirobot exploration under recurrent connectivity constraints." In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5491–5498 (cit. on pp. 6, 15, 70).

Banfi, Jacopo et al. (2018). "Strategies for coordinated multirobot exploration with recurrent connectivity constraints." In: *Autonomous Robots* 42.4, pp. 875–894 (cit. on pp. 15, 70).

Bautin, Antoine and Olivier Simonin (2012). "MinPos : a Novel Frontier Allocation Algorithm for Multi-robot Exploration." In: *ICIRA* 7507 (cit. on pp. 5, 6, 40, 45).

Bautin, Antoine, Olivier Simonin, and François Charpillet (2011). "Towards a communication free coordination for multi-robot exploration." In: *6th National Conference on Control Architectures of Robots, CAR2011* (cit. on p. 40).

— (2013). "SyWaP: Synchronized wavefront propagation for multi-robot assignment of spatially-situated tasks." In: *2013 16th International Conference on Advanced Robotics, ICAR 2013* (cit. on pp. 27, 49).

Burgard, W. et al. (2005). "Coordinated multi-robot exploration." In: *IEEE Transactions on Robotics* 21.3, pp. 376–386 (cit. on pp. 3–5).

Caccamo, Sergio et al. (2017). "RCAMP: A resilient communication-aware motion planner for mobile robots with autonomous repair of wireless connectivity." In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 2010–2017. arXiv: 1710.09303 (cit. on p. 22).

Cavalcante, Rodolfo C., Thiago F. Noronha, and Luiz Chaimowicz (2013). "Improving combinatorial auctions for multi-robot exploration." In: *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6 (cit. on p. 5).

Cesare, Kyle et al. (2015). "Multi-UAV exploration with limited communication and battery." In: *Proceedings - IEEE International Conference on Robotics and Automation* 2015-June.June, pp. 2230–2235 (cit. on pp. 13, 14, 70).

Couceiro, Micael S. et al. (2014). "Darwinian swarm exploration under communication constraints: Initial deployment and fault-tolerance assessment." In: *Robotics and Autonomous Systems* 62.4, pp. 528–544 (cit. on pp. 12, 13).

Derbakova, Anna, Nikolaus Correll, and Daniela Rus (2011). "Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems." In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3863–3868 (cit. on p. 11).

Elfes, Alberto (1989). "Using Occupancy Grids for Mobile Robot Perception and Navigation." In: *Computer* 22.6, pp. 46–57 (cit. on pp. 4, 19).

Fink, Jonathan, Alejandro Ribeiro, and Vijay Kumar (2013). "Robust Control of Mobility and Communications in Autonomous Robot Teams." In: *IEEE Access* 1, pp. 290–309 (cit. on p. 22).

Frey, Brendan J. and Delbert Dueck (2007). "Clustering by passing messages between data points." In: *Science*. arXiv: 1401.2548 (cit. on p. 22).

Garey, M. R., R. L. Graham, and D. S. Johnson (1977). "The Complexity of Computing Steiner Minimal Trees." In: *Society for Industrial and Applied Mathematics - SIAM* 32.4, pp. 835–859 (cit. on p. 85).

Gerkey, B. P. (2004). "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems." In: *The International Journal of Robotics Research* 23.9, pp. 939–954 (cit. on p. 4).

Hollinger, Geoffrey A. and Sanjiv Singh (2012). "Multirobot coordination with periodic connectivity: Theory and experiments." In: *IEEE Transactions on Robotics* 28.4, pp. 967–973 (cit. on pp. 5, 11, 40).

Jensen, Elizabeth A, Ernesto Nunes, and Maria Gini (2014). "Communication-Restricted Exploration for Robot Teams." In: *Workshop on Multiagent Interaction without Prior Coordination*. Association for the Advancement of Artificial Intelligence (AAAI). Québec, Canada, pp. 15–21 (cit. on p. 12).

Keidar, Matan and Gal A. Kaminka (2012). "Robot Exploration with Fast Frontier Detection: Theory and Experiments." In: *International Conference on Autonomous Agents and Multiagent Systems*, pp. 113–120 (cit. on p. 4).

— (2014). "Efficient frontier detection for robot exploration." In: *The International Journal of Robotics Research* 33.2, pp. 215–236 (cit. on p. 4).

Kleinberg, Jon and Eva Tardos (2006). *Algorithm design*. arXiv: 1439880182 (cit. on pp. 76, 84).

Korsah, G Ayorkor, Anthony Stentz, and M Bernardine Dias (2013). "A comprehensive taxonomy for multi-robot task allocation." In: *The International Journal of Robotics Research* 32.12, pp. 1495–1512 (cit. on pp. 4, 5).

Krupke, Dominik et al. (2015). "Distributed cohesive control for robot swarms: Maintaining good connectivity in the presence of exterior forces." In: *IEEE International Conference on Intelligent Robots and Systems* 2015-Decem, pp. 413–420. arXiv: 1505.03116 (cit. on pp. 17, 70).

Kuhn, H. W. (1955). "The Hungarian method for the assignment problem." In: *Naval Research Logistics* 2, pp. 83–97 (cit. on p. 5).

Laëtitia Matignon, Laurent Jeanpierre and Abdel-Illah Mouaddib (2012). "Coordinated Multi-Robot Exploration Under Communication Constraints Using Decentralized Markov Decision Processes." In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (cit. on pp. 11, 12).

Le, Van Tuan et al. (2009). "Making networked robots connectivity-aware." In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3502–3507 (cit. on p. 10).

MacQueen, James (1967). "Some Methods for classification and Analysis of Multivariate Observations." In: *5th Berkeley Symposium on Mathematical Statistics and Probability 1967* (cit. on p. 22).

Magán-Carrión, Roberto et al. (2016). "Optimal relay placement in multi-hop wireless networks." In: *Ad Hoc Networks* 46, pp. 23–36 (cit. on p. 15).

Magán-Carrión, Roberto et al. (2017). "A Dynamical Relay node placement solution for MANETs." In: *Computer Communications* 114.April, pp. 36–50 (cit. on pp. 15, 70, 71).

Michael, Nathan et al. (2009). "Maintaining Connectivity in Mobile Robot Networks." In: *Springer Tracts in Advanced Robotics (STAR)* 54.54, pp. 117–126 (cit. on p. 10).

Mosteo, Alejandro R., Luis Montano, and Michail G. Lagoudakis (2008). "Multi-robot routing under limited communication range." In: *2008 IEEE International Conference on Robotics and Automation*, pp. 1531–1536 (cit. on pp. 9, 10).

Nestmeyer, Thomas et al. (2017). "Decentralized simultaneous multi-target exploration using a connected network of multiple robots." In: *Autonomous Robots* 41.4, pp. 989–1011. arXiv: `1505.05441` (cit. on pp. 14, 15, 69, 70).

Pal, Anshika, Ritu Tiwari, and Anupam Shukla (2013). "Communication constraints multi-agent territory exploration task." In: *Applied Intelligence* 38.3, pp. 357–383 (cit. on p. 49).

Pei, Yuanteng and Matt W. Mutka (2012). "Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration." In: *IEEE International Conference on Robotics and Automation*, pp. 1551–1556 (cit. on pp. 13, 14, 70).

Pham, Viet-cuong and Jyh-Ching Juang (2013). "A multi-robot, cooperative, and active slam algorithm for exploration." In: *Int. Journal of Innovative Computing, Information and Control* 9.6, pp. 2567–2583 (cit. on pp. 5, 12).

Pralet, Cédric and Charles Lesire (2014). "Deployment of Mobile Wireless Sensor Networks for Crisis Management: a Constraint-Based Local Search Approach." In: *International Conference on Principles and Practice of Constraint Programming*, pp. 870–885 (cit. on pp. 13, 70).

Rahman, Md Mahbubur et al. (2017). "Relay vehicle formations for optimizing communication quality in robot networks." In: *IEEE International Conference on Intelligent Robots and Systems* 2017-Septe, pp. 6633–6639 (cit. on pp. 15, 16, 69, 70).

Rooker, Martijn N. and Andreas Birk (2007). "Multi-robot exploration under the constraints of wireless networking." In: *Control Engineering Practice* 15.4, pp. 435–445 (cit. on pp. 9, 10).

Satici, Aykut C. et al. (2013). "Connectivity preserving formation control with collision avoidance for nonholonomic wheeled mobile robots." In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5080–5086 (cit. on p. 49).

Sheng, Weihua et al. (2006). "Distributed multi-robot coordination in area exploration." In: *Robotics and Autonomous Systems* 54.12, pp. 945–955 (cit. on p. 5).

Simmons, Reid et al. (2000). "Coordination for Multi-Robot Exploration and Mapping." In: *Aaai*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp. 852–858 (cit. on pp. 3, 4, 9).

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press (cit. on p. 49).

Valentin, Luis et al. (2014). "Motion strategies for exploration and map building under uncertainty with multiple heterogeneous robots." In: *Advanced Robotics* 28.17, pp. 1133–1149 (cit. on p. 6).

Vazquez, Jose and Chris Malcolm (2004). “Distributed Multirobot Exploration Maintaining a Mobile Network.” In: *IEEE International Conference on Intelligent Systems* (cit. on pp. 8, 9).

Wurm, K.M., C. Stachniss, and W. Burgard (2008). “Coordinated multi-robot exploration using a segmentation of the environment.” In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1160–1165 (cit. on pp. 3, 5, 42).

Yamauchi, Brian (1998). “Frontier-based exploration using multiple robots.” In: *Proceedings of the second international conference on Autonomous agents AGENTS 98*, pp. 47–53 (cit. on pp. 4, 22, 45).

Yan, Zhi et al. (2015). “Metrics for performance benchmarking of multi-robot exploration.” In: *IEEE International Conference on Intelligent Robots and Systems* 2015-December, pp. 3407–3414 (cit. on pp. 3, 48, 88).

Yuan, Jing et al. (2010). “A cooperative approach for multi-robot area exploration.” In: *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 1390–1395 (cit. on p. 4).

Zlot, R. et al. (2002). “Multi-robot exploration controlled by a market economy.” In: *Proceedings 2002 IEEE International Conference on Robotics and Automation* 3.May, pp. 3016–3023 (cit. on p. 5).

**Résumé** — Le problème d'exploration est un sujet fondamental de la robotique mobile autonome qui traite la réalisation de la catographie complète (*mapping*) d'un environnement précédemment inconnu. Il y a plusieurs scénarios où l'achèvement de l'exploration d'une zone est une composante principale de la mission à accomplir. Par exemple: l'exploration planétaire, la reconnaissance, la recherche et le sauvetage, l'agriculture, le nettoyage des lieux dangereux, comme champs de mines et des zones radioactives. D'autre part, la communication sans fil joue un rôle important dans les stratégies multi-robot collaboratives. Malheureusement, la supposition ou l'exigence de communication stable, ou encore, la connectivité continue, peuvent être compromises dans des scénarios réels. Dans cette thèse, deux nouvelles approches abordent le problème d'exploration multi-robot d'environnements, en considérant une communication restreinte. D'abord, une stratégie multi-objectif auto-adaptative est proposée pour diriger la sélection de tâches en tennant compte de la performance d'exploration et du niveau de connectivité. Deuxièmement, deux rôles – l'explorateur et le relais de communication – sont considérés pour améliorer la stratégie de sélection de tâche précédente. Basé sur le modèle de communication, une nouvelle approche de placement de robot relais pour des missions d'exploration multi-robot est présentée en détail. Comparé avec d'autres approches de l'état de l'art, les deux approches proposées dans cette thèse sont capables de diminuer la durée de périodes de déconnexion sans dégradation considérable sur temps d'exploration.

**Mots clés :** Missions d'exploration, systèmes coopératifs, systèmes multi-robot coordonnés, environnements avec communication restreinte.

**Abstract** — The exploration problem is a fundamental subject in autonomous mobile robotics that deals with achieving the complete coverage of a previously unknown environment. There are several scenarios where completing exploration of a zone is a main part of the mission, e.g. planetary exploration, reconnaissance, search and rescue, agriculture, cleaning, or dangerous places as mined lands and radioactive zones. Wireless communication plays an important role in collaborative multi-robot strategies. Unfortunately, the assumption or requirement of stable communication and end-to-end connectivity may be easily compromised in real scenarios. In this thesis, two novel approaches to tackle the problem of multi-robot exploration of communication constrained environments are proposed. At first, an auto-adaptive multi-objective strategy is followed in order to support the selection of tasks regarding both exploration performance and connectivity level. Secondly, two roles –explorer and communication relay– are considered in order to improve the benefits of the previous task selection strategy. Based on the communication model, a novel polynomial-time relay placement approach for multi-robot exploration missions is introduced in detail. Compared with others, the proposed approaches are capable of decreasing the last of disconnection periods without a noticeable degradation of the completion exploration time.

**Keywords:** Exploration missions, Cooperative systems, Multi-robot coordinated systems, Communication restricted environments.

---

Aerospace Vehicles Design and Control Department (DCAS),
10 Avenue Edouard Belin 31400 Toulouse
Facultad de Ingeniería (InCo),
J. H. y Reissig 565 11300 Montevideo