# Multi-robot exploration under communication constraint: a disCSP approach

Arnaud Doniec, Noury Bouraqadi, Michael Defoort
Univ Lille Nord de France, F-59000 Lille, France
EMDouai, IA, F-59508 Douai, France
{doniec,bouraqadi,defoort}@ensm-douai.fr

Van Tuan Le, Serge Stinckwich
GREYC, UMR 6072 (CNRS, Université de Caen)
UMI UMMISCO 209 (IRD/UPMC/IFI-MSI)
lvtuan@gmail.com
serge.stinckwich@ird.fr

May 3, 2010

**Abstract**

Exploration of an unknown environment is one of the major applications of Multi-Robot Systems. Many works have proposed multi-robot coordination algorithms to accomplish exploration missions based on multi-agent techniques. Some of these works focus on multi-robot exploration under communication constraints. This paper is an extract of [4] in which we propose an original way to formalize and solve this issue. Our proposal relies on distributed constraint satisfaction problems (disCSP) which are an extension of classical constraint satisfaction problems (CSP). Compared to other works, our proposal is fully distributed and guaranties the exploration of an unknown environment with maintenance of connectivity between all the members of a robots' team.

## 1 Introduction

Multi-robot systems (MRS) consist in a set of autonomous mobile robots which collaborate to perform a mission. This collaboration is allowed by communication abilities which usually rely on radio communication technologies. For example, Mobile Ad Hoc Networks (MANET) are frequently employed to support explicit communications in MRS: each robot becomes a node of the network and is able to send, receive and relay data to other robots. In order to have an accurate and efficient collaboration between robots, each node (robot) of the network has to be reachable at each instant. This implies that, in addition to their collaborative task, the robots have to perform an extra task which consists in maintaining the network connectivity.

One of the major applications of multi-robot systems is the exploration of an unknown environment. It consists, for a fleet of robots, in navigating while incrementally constructing a map of the environment. In this application, the robots have to collaborate to:

- spread out on the ground (and consequently: speed up the exploration and use less energy),
- keep in touch with each other in order to exchange partial maps and share areas that have not been explored yet.

Therefore, a good collaboration scheme for multi-robots exploration has to conciliate these two antagonist constraints.

## 2 Multi-robot exploration related works

The multi-robot exploration issue has been addressed in the literature using different approaches and has been originally initiated by: [9] and [14].

In [14], robots try to build and use a global map for the exploration. The postulate of this work is the following: to speed up the exploration, robots have to gain new information about the environment. Therefore, they have to move towards the boundary between open space and unexplored area. In practice, some robots try to get closer to the boundary but without real cooperation with the rest of the fleet. Nevertheless, this article states the concept of *frontier-based exploration* which has inspired many other works.

Main improvings of the frontier-based exploration have consisted in introducing coordination between robots. This has firstly been performed in a centralized way. In [11], the authors propose to assign a target destination to each robot so that the expected information gain over time will be maximized. Since the computation of the optimal solution is intractable in practice, they proposed an heuristic based on bids construction. Each robot estimates the utility and the cost to travel towards various locations. A central server receives all the bids and assigns a location target to each robot taking possible overlaps in the coverage of the ground into account.

More recently, Rooker and Birk [8] also proposed a centralised coordination ensuring that, during the exploration, no robot will loose the connection with the rest of the fleet. To achieve this goal, a central entity collects the current positions of all the robots and generates a set of future configurations for the fleet (i.e. the future possible positions of the robots). Due to the high number of available combinations, all configurations can not be considered but only a limited number of them. Among this number of generated configurations, the central entity chooses the best one according to an utility function. This function gives penality when the evaluated future position is occupied by an obstacle or when it puts a robot out of the communication range with the other robots.

Clearly, centralized coordination algorithms dedicated to frontier-based exploration seem difficult to employ in real applications. They are not fault tolerant since the entire system will fail whether the central entity fails. Moreover, for large scale MRS, the use of an entity which centralizes all data accounts for a bottleneck in terms of decision computing and communication.

To avoid these drawbacks, some works have investigated multi-robot coordination in a distributed way. In [13], the authors propose an exploration algorithm based on the selection of different behaviors: avoiding obstacles, maintaining network connectivity, exploring around the frontier. This selection takes into account the current network condition which is known by each robot thanks to periodically exchanged messages. To achieve the connectivity of the network, each robot analyses the topology of the network and makes distinction between a simple articulation and a bridge. A simple articulation is a link whose deconnection does not imply the loss of connectivity in the network. A bridge is a link whose deconnection creates two unconnected subnetworks.

Many other works use multi-agent bidding algorithms to achieve multi-robot coordination. An example of such an approach can be found in [10]. The authors introduce a bid calculation allowing robots to find their best target locations. The bid calculation is based on a ponderated sum of three elements: the potential information gain of the target location, the distance from the current position of the robot to the target location and a nearness measure intended to characterize the ability to maintain communication links with other robots. To perform coordination, robots periodically broadcast their best bids to all other robots within the same subnetwork. At the end of a constant time, the robot who provides the best bid is declared as winner and is allowed to move towards the target. This process restarts for all the remaining robots.

The use of distributed bidding algorithms does not guaranty the maintenance of connectivity. Indeed, the introduction of the nearness measure in the bid calculation is just an heuristic which tends to make robots stay close to each other. Robots exhibit certain clustering behaviors during the exploration but deconnections can occur and lead to the partition of the network. The authors explicitely mention this problem and talk about inconsistency between maps of the disjoint subnetworks.

The aim of this article is to provide a fully distributed algorithm for exploration with maintenance of connectivity. In the following, we present an original way to formalize and solve this issue considering distributed constraint satisfaction problems.

# 3 DisCSP for multi-robot coordination

## 3.1 Distributed Constraint Satisfaction Problems (disCSP)

Constraint satisfaction is a classical and powerfull tool in artificial intelligence [2] whose traditional applications concern planning, scheduling, placement, logistics and so on. A constraint satisfaction problem (CSP) can be viewed as a triplet $(X, D, C)$ in which: $X$ is a finite set of variables, each variable $x_i \in X$ is associated to a finite domain $dom(x_i) \in D$ and relates to a finite set of constraints $C$. Solving a CSP requires to find for each variable $x_i$ a value in $dom(x_i)$ which is consistent with $C$ (i.e. which does not violate any constraint of $C$).

The concept of distributed CSP has been introduced to formalize and solve naturally distributed decision problems [16] which generally deal with a set of data, shared out among many sites and whose centralization is often impossible.

A disCSP $(X, D, C, A)$ is an extension of the triplet $(X, D, C)$ where $A$ is a finite set of agents $\{A_1, A_2, \ldots, A_p\}$ in which each $A_k(1 \leq k \leq p)$ owns a subset of $X$: $X_{A_k}$ with $\bigcap_{A_k \in A} X_{A_k} = \varnothing$ and a subset of $C$: $C_{A_k}$. From the point of view of agent $A_k$, variables of set $X_{A_k}$ are called ”owned variables” whereas the set $X \setminus X_{A_k}$ refers to the ”foreign variables”.

Most of algorithms to solve disCSP are distributed and asynchronous. To execute such algorithms, an agent has to be able to send messages to any other agents of its accointance set[1]. As the algorithms are asynchronous, delays can occur when messages are exchanged through the communication network. Nevertheless, we assume that when two agents exchange successive messages, the sending order is preserved when receiving. For more details about disCSP algorithms, readers can consult [16].

## 3.2 Modelling the multi-robot exploration problem as a disCSP

To expose our proposal, we consider the traditional hypothesis made in works related to multi-robot exploration:

- each robot, during its movement, updates its map with the new areas discovered,
- periodically, the robots exchange their map so that they can own a global map and know the position of the frontier,
- each robot is able to know its own position and the other robots' ones.

To express the multi-robot exploration as a distributed constraint satisfaction problem, we have to discretize the space of actions of robots. The movement of a mobile robot is usually the result of a combination of lateral and longitudinal accelerations. To simplify our speech, we consider 8 possible directions corresponding to the cardinal points of a compass: go to the north, go to the north-east, go to the east, go to the south-east, etc.

Considering this discretization, each robot has to find a decision among these 8 possibilities such that three requirements are fulfilled:

1. its future location does not break the connectivity of the network,
2. its future location does not induce overlaps between the sensor ranges,
3. its future location allows to discover new unexplored areas.

To evaluate if a movement will break or not the connectivity of the network, a robot has to be aware about it. In practice, a robot has to identify among all these direct connections the one that allows to access to the rest of the network (the others being able to be broken if necessary). Our algorithm presented in [6] allows the robots to obtain such information: based on exchanged

[1] In MAS, the term accointance refers to the set of agents with which an agent can exchange messages knowing their address and id

messages in the network, each robot constructs a table containing the id of access robots with which it has to maintain a direct connection in order to stay in touch with a reference node[2].

Thus, the requirement 1. can be expressed as a constraint defined by an inferiority test between two values: the distance between the future positions of the two robots and the communication range (Figure 1).

To maximize the discovery of new unexplored areas, a robot has to be closed to the frontier. The requirement 3. can be expressed as a specific ordering between the 8 cardinal positions such that the first cardinal position allows to have the lowest distance between the robot and the frontier, the second cardinal position allows to have the second lowest distance, etc. To enforce the efficiency of the exploration by reducing overlaps (requirement 2), we can also impose that the distance between the future positions of any two robots of $A$ have to be superior to the sum of their sensor ranges.

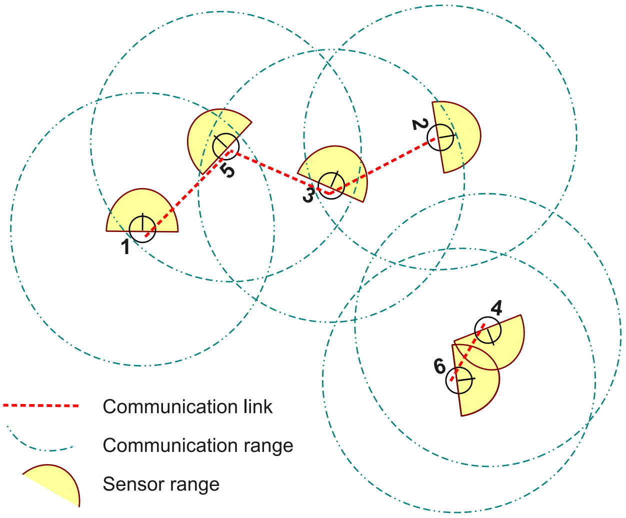

Figure 1: Connectivity constraint between two robots

This previous statement can be expressed as the following disCSP:

- $A = \{A_1, A_2, \ldots, A_p\}$ denotes a fleet of $p$ robots exploring an unknown environment and sharing a common map of already explored areas.
- $X = \{x_1, x_2, \ldots, x_p\}$ is composed of variables storing the next heading of each robot of $A$.
- $D = \{dom(x_1), \ldots, dom(x_p)\}$ with $dom(x_i)(1 \leq i \leq p)$ is the set of all 8 cardinal directions that a robot $A_i$ can choose to plan its next movement. The domain is ordered by the following relation: $v_1 \preccurlyeq v_2 \equiv dist(fp(A_i, v_1), frontier) < dist(fp(A_i, v_2), frontier)$ with $(v_1, v_2) \in dom(x_i)^2$.

[2] When all robots maintain a communication path towards a reference node, then the connectivity of the network is assured.

- $C = C_1 \cup C_2$ where:

$$\begin{aligned} C_1 = \ & \{\forall A_i \in A, \ \ \exists A_j \in AR_{A_i}, \\ & \qquad dist\,(fp(A_i, x_i), fp(A_j, x_j)) < cr\} \\ C_2 = \ & \{\forall A_i \in A, \ \ \forall A_j \in A \setminus A_i, \\ & \qquad 2sr < dist\,(fp(A_i, x_i), fp(A_j, x_j))\} \end{aligned}$$

with:

- $sr$ the sensor range of a robot[3],
- $dist(p_1, p_2)$ the euclidian distance between the position $p_1$ and the position $p_2$,
- $fp(A_i, x_i)$ the future position of $A_i$ considering its future direction $x_i$ and its current vector speed,
- $cr$ the communication range of a robot.

To explore an unknown environment, the robots of set $A$ have to periodically solve this disCSP in order to be able to choose a heading compatible with the requirements previously introduced. We present the resulting algorithm in the following.

## 3.3 Multi-robot exploration algorithm based on distributed constraint reasoning

The proposed multi-robot exploration algorithm consists in repeating the following sequence until the end of the mission:

1. update maps and connectivity tables for each robot,
2. construct the disCSP based on connectivity tables and current positions of robots,
3. order the value of each domain taking the distance to the frontier into account,
4. solve the disCSP to obtain future directions of each robot,
5. operate the movement of each agent during a fixed time period.

The mission is finished when there is no more area to explore. To be able to detect this state, each robot has to own a global view of the environment. Clearly, this implies that robots build this global view by exchanging periodically their local map. In pratice, our algorithm can be run concurrently with a process realizing simultaneous localisation and mapping (SLAM) [5].

[3] To simplify our speech, is made the assumption that all robots have identical sensor capacities. This assumption is also made for the communication range.

**algorithm** *explore*

```
1: begin
2:   while $frontier$ exists do
3:     update map and connectivity table
4:     $X_{A_i} \leftarrow \{x_i\}$
5:     $C_{A_i} \leftarrow \{dist(fp(A_i, x_i), fp(A_j, x_j)) < cr\}_{A_j \in AR_{A_i}}$
6:     $C_{A_i} \leftarrow C_{A_i} \cup \{2\ sr < dist(fp(A_i, x_i), fp(A_j, x_j))\}_{A_j \in A \setminus A_i}$
7:     $heading \leftarrow \{N, NE, E, SE, S, SW, W, NW\}$
8:     $\bar{H} \leftarrow \varnothing$
9:     for each $h \in heading$ do
10:      if there is an obstacle towards $h$ then
11:        $\bar{H} \leftarrow \bar{H} \cup \{h\}$
12:      end
13:    end
14:    $H \leftarrow heading \setminus \bar{H}$
15:    $dom(x_i) \leftarrow \{v_a \in H / \forall (v_a, v_b) \in H^2, a < b \Rightarrow v_a \preccurlyeq v_b\}$
16:    $dom(x_i) \leftarrow dom(x_i) \cup \bar{H}$
17:    solveDisCSP
18:    move towards direction $x_i$ during $dt$ seconds
19:  end
20: end
```

Figure 2: Distributed exploration algorithm

Figure 2 details the exploration algorithm run by robot $A_i$. The algorithm stops when there is no more unexplored area as expressed at line 2. Line 3 concerns the updates of the map and the connectivity table of $A_i$ using the algorithm presented in [6].

From lines 4 to 16, $A_i$ constructs its part of the disCSP. At line 5, $A_i$ is aware about its own connectivity with the rest of the fleet and is able to initialize its set $C_{A_i}$ with the constraints of connectivity maintenance. At line 6, $A_i$ completes its set $C_{A_i}$ with the constraints of non overlap between sensor ranges.

Lines 7 to 16 concern the ordering of the domain of $x_i$. Among the set of all possible headings $\{N, NE, E, SE, S, SW, W, NW\}$, the robot $A_i$ selects the one which brings him closer to the frontier. The order relation $\preccurlyeq$ compares the distances to the frontier from two future positions. The future position of a robot can be calculated using the current speed and heading of the robot and the position of known obstacles.

Considering the situation of Figure 3, the strict application of relation $\preccurlyeq$ between $W$ and $SE$ gives $W \preccurlyeq SE$ since, in term of the euclidian distance, $fp(A_i, W)$ is closer to the frontier than $fp(A_i, SE)$. Obviously, the frontier is not directly reachable from $fp(A_i, W)$ because of the obstacle. To take into account such a situation, the set of all possible headings is splited into two subsets: the set $\bar{H}$ whose headings lead into obstacles and the set $H$ which contains the remained free headings.

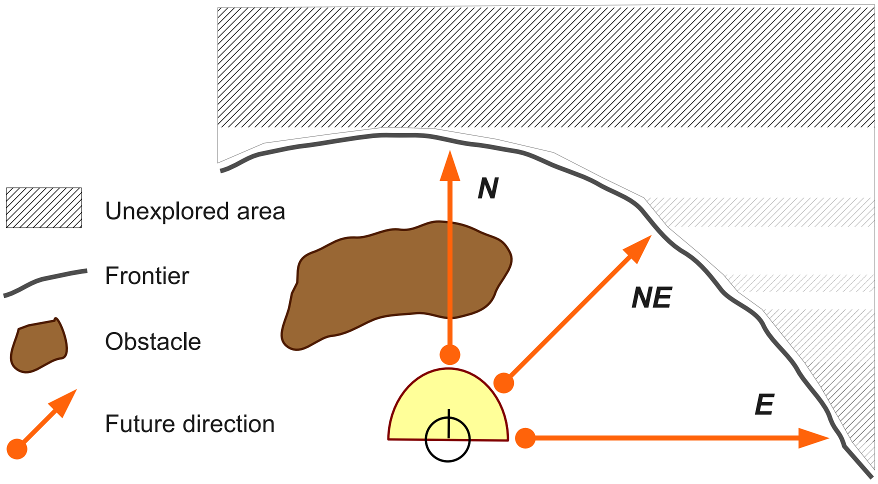

Figure 3: Future possible positions according to current speed and obstacles position

At line 15, the domain of $x_i$ is initialized with the values of $H$ ordered with the relation $\preccurlyeq$. In some cases, the only way to maintain connectivity is to choose a direction towards an obstacle. Consequently, the values of the set $\bar{H}$ are added at the end of the domain $dom(x_i)$(line 16). In the example, such an ordering policy gives $\{N, NE, S, E, SW, SE, NW, W\}$ as the domain for the variable $x_i$.

The disCSP is solved at line 17. Here any disCSP solving algorithm can be used such as ABT [15], AWS [16], ABT$_{not}$[1], etc. Finally, the movement of the robot is operated at line 18. To take the obstacle into account in a finer manner, a navigation algorithm can be used [3].

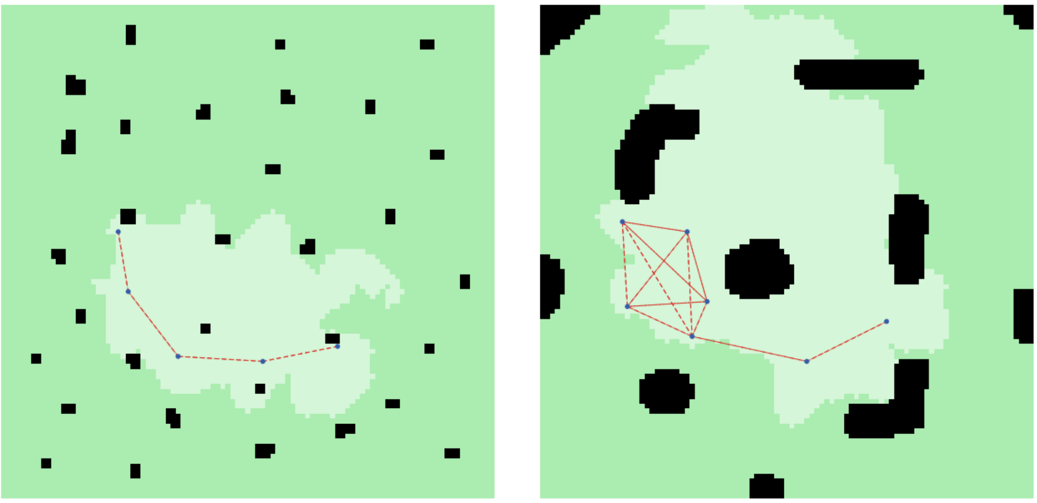

Figure 4: Examples of environments with different complexity and density of obstacles

# 4 Implementation and Validation

## 4.1 Implementation details

To evaluate the feasibility and the relevance of our approach, we have implemented the exploration algorithm in NetLogo. NetLogo is a multi-agent programmable modeling environment [12] which allows to prototype quickly systems of situated agents evolving in a two dimensions world. Our considered environments are closed and modeled as a grid with $100 \times 100$ cells. Each cell can be empty, occupied by a robot or an obstacle, explored or unknown.

Many parameters of the robots like the sensor range, the communication range and $dt$ can be tuned. Many data are recorded at each simulation: exploration duration, travelled distance by each robot, number of exchanged messages by robots. In addition, specific data about disCSP solving are available: number of ”nogood” messages, number of ”ok” messages, number of cycles, number of checked constraints. The disCSP solving algorithm used for our experiments is ABT whose implementation has been provided by [7] in community models of NetLogo.

## 4.2 Simulations

In our simulations, we have considered environments with different levels of complexity depending on: the shape of the obstacles (concave vs convexe), the size of the obstacles (compared to the size of one robot), the density of the obstacles, the minimum distance between two close obstacles. Figure 4 gives two examples of possible environments.

At the beginning of each simulation, the robots are initially aligned in the bottom right corner of the environment with a distance between them inferior to the communication range. For our experiments, we have considered two different communication ranges: a short one and a large one. Figures 5 and 6 are examples of obtained results respectively for a simple and a complex environment. They present the average of exploration durations according to the number of robots deployed in the environment.

Our different experimentations show that the typology of the environment has a great impact on the duration of the exploration: a ”simple” environment is faster explored than a ”complex” one. Moreover, for closed environments, adding robots improves the speed of the exploration to a certain limit: when robots are too many, they interfer with each other and spend more time avoiding each other than exploring.

Clearly, a large communication range also helps to improve the exploration: robots are less constrained in their movements and can cover a larger area of exploration. On the other hand , we have noticed that initial positions of robots in the environment have a minor influence on the exploration.

Figure 7 presents the average of exchanged messages for different sizes of robots’ team to accomplish a full exploration. When the number of robots increases, the size of the disCSP grows and consequently more messages are required to find a solution. Meanwhile, the increase of the number of exchanged messages is balanced by the fact that the duration of the exploration is reduced with more robots. Let us remind that we have used in our prototype the simplest algorithm

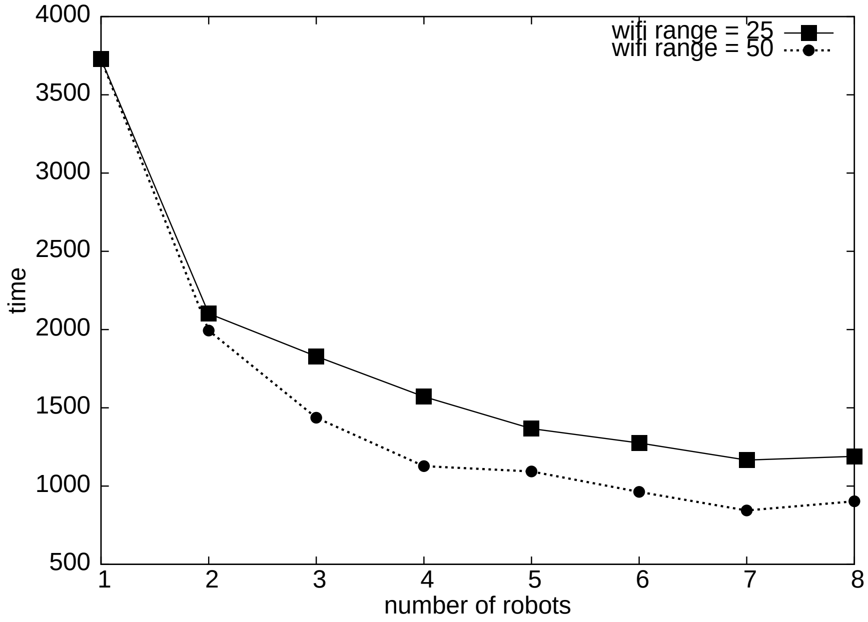

Figure 5: Experiments with a ”simple” environment

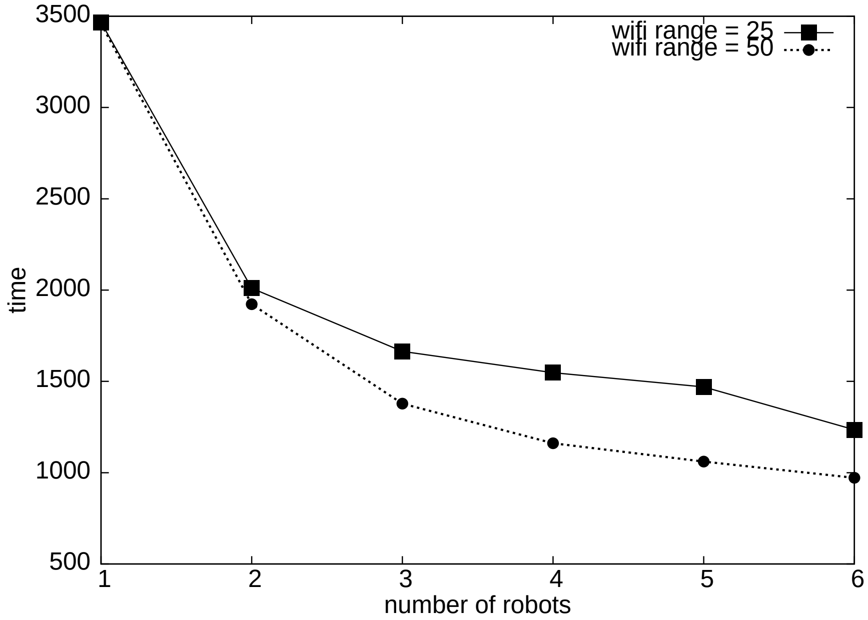

Figure 6: Experiments with a ”complex” environment

(ABT) for solving the disCSP. Using a more sophisticated algorithm will help to reduce the number of exchanged messages at each decision step.

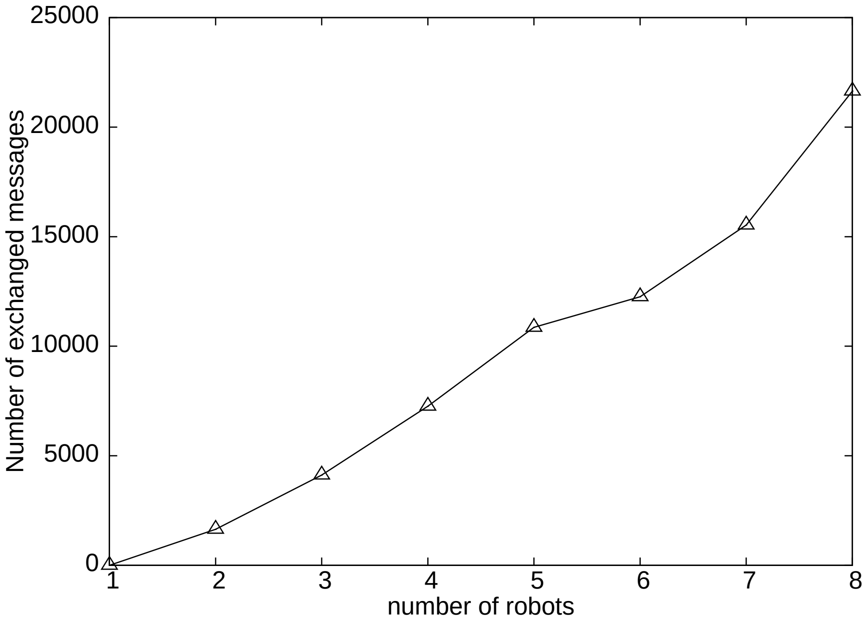

Figure 7: Number of exchanged messages during an exploration

# 5 Conclusion

This paper proposes an original way to deal with multi-robot exploration under limited communication range. From a robot point of view, such a task consists in choosing its best decision such as it can explore unknown areas and maintain connectivity with the rest of its fleet. To state this coordination problem, we use a well known formalism in artificial intelligence: the distributed constraint satisfaction problems (disCSP). Based on this foundation, a new distributed algorithm for the coordination of robots is introduced. Using its own connectivity awareness, each robot constructs a part of the disCSP by adding constraints with the rest of the fleet. The distributed resolution of the disCSP gives the future direction of each robot.

Simulations give interesting results: reasonable number of exchanged messages, decrease of the exploration duration when the number of robots grows, robust connectivity maintenance, easier deadlock detection. Next step of this work will consist in implementing our approach on real robots.

# References

[1] C. Bessiere, I. Brito, A. Maestre, and P. Meseguer. Asynchronous backtracking without adding links: A new member in the ABT family. *Artificial Intelligence*, 161:7–24, 2005.

[2] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[3] M. Defoort, J. Palos, A. Kokosy, T. Floquet, and W. Perruquetti. Performance-based reactive navigation for nonholonomic mobile robots. *Robotica*, 27(2):281–290, 2009.

[4] A. Doniec, N. Bouraqadi, M. Defoort, V-T. Le, and S. Stinckwich. Distributed constraint reasoning applied to multi-robot exploration. In *21st International Conference on Tools with Artificial Intelligence (ICTAI)*, New Jersey, USA, 2009.

[5] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I and II. *IEEE Robotics & Automation Magazine*, pages 99–117, 2006.

[6] Van Tuan Le, Noury Bouraqadi, Victor Moraru, Serge Stinckwich, and Arnaud Doniec. Making networked robot connectivity-aware. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3502–3507, 2009.

[7] I. Muscalagiu, J. M. Vidal, and H. Jiang. Random binary CSPs using asynchronous backtracking. NetLogo Models, http://discsp-netlogo.fih.upt.ro/, 2007.

[8] M. N. Rooker and A. Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.

[9] H. Shatkay and L. P. Kaelbling. Learning topo-logical maps with weak local odometric information. In *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 920–929, 1997.

[10] W. Sheng, Q. Yang, J. Tan, and N. Xi. Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54:945–955, 2006.

[11] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[12] S. Tisue and U. Wilensky. Netlogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of the Agent 2004 Conference*, 2004.

[13] J. Vazquez and C. Malcolm. Distributed multirobot exploration maintaining a mobile network. In *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, volume 3, pages 113–118, 2004.

[14] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents (Agent'98)*, 1998.

[15] M. Yokoo. Distributed constraint satisfaction for DAI problems. In *Proceedings of the 10th International Workshop on Distributed Artificial Intelligence.*, 1990.

[16] M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems*. Springer, 2001.