IROS 2015 Workshop on On-line decision-making in multi-robot coordination (DEMUR'15)
http://robotics.fel.cvut.cz/demur15/

COMMUNICATION-CONSTRAINED MULTIROBOT EXPLORATION: SHORT TAXONOMY AND COMPARATIVE RESULTS

Jacopo Banfi[a,*], Alberto Quattrini Li[b], Nicola Basilico[c], Francesco Amigoni[a]

[a] *Politecnico di Milano, Milano, Italy*
[b] *University of South Carolina, Columbia, SC*
[c] *Università degli Studi di Milano, Milano, Italy*
[*] corresponding author: jacopo.banfi@polimi.it

Abstract. Exploration of initially unknown environments is a fundamental task in several robot applications, like search and rescue and map building. In the literature, many works propose exploration strategies that allow multiple mobile robots to autonomously select the next observation locations in order to incrementally discover the environment in an efficient way. The effects of realistic communication between robots have been considered by some papers that account for communication constraints in the design of exploration strategies. However, these approaches have not been comparatively assessed yet. In this paper, we briefly review the relevant literature of communication-constrained exploration strategies and present an experimental quantitative comparison of some of them.

Keywords: Multirobot exploration, Communication constraints, Exploration strategies.

1. Introduction

Exploration of initially unknown environments through the deployment of multirobot systems is an effective technique for many applications, including map building [1] and search and rescue [2]. A large number of exploration strategies have been proposed in the literature. Such variety of approaches triggered the need of a focused experimental comparison, with the aim of identifying the most significant methods and of exposing their strengths and weaknesses. Examples of works addressing this need include, but are not limited to, [3, 4].

Most of the literature on exploration strategies simply ignores communication constraints, as it is often assumed that all the robots and, possibly, a coordinating base station (BS) can always communicate with each other [5, 6]. However, such an assumption is not guaranteed in many applications and robots should appropriately move to ensure communication. Consider, for example, a search and rescue scenario: usually, a fixed BS is present and should be able to continuously communicate with the robots, so that human rescuers can watch the video stream of the robot on-board cameras to find victims. Another example is environmental monitoring, where robots collect data samples and send them to a BS. In this case, communication constraints are softer, as it is not required a continuous connectivity between robots and BS, provided that the collected data are downloaded at some time. With such applications in mind, in this paper, we consider a scenario in which robots should be connected to a BS to be able to deliver information. A number of recent works, such as [7–9], present exploration strategies complying with communication constraints.

In this paper, we experimentally compare a significant sample of exploration strategies that consider different communication constraints. The comparison is performed by using a robotic simulator that focuses on the communication aspects [10]. The main goal of our work is to derive some insights on the strengths and the weaknesses of such methods and to better understand how the communication constraints quantitatively affect the exploration performance, for helping the future development of better exploration strategies (integrating the results from [11], where the effects of different communication models on exploration are assessed).

This paper is structured as follows. Section 2 provides a brief survey on communication-constrained exploration. Section 3 formalizes the exploration problem. Sections 4 and 5 describe the methods we compare. Section 6 discusses our experiments and results, while Section 7 concludes the paper.

2. A short taxonomy

The problem of online multirobot exploration in presence of a fixed BS to which the gathered information is relayed has been tackled by different authors in different flavors. Most of the works are built upon the seminal paper of Yamauchi [12] on multirobot frontier-based exploration, where the idea is to have robots moving towards the boundaries between free known areas and unexplored space without any communication constraint.

A first way to intend the communication constraints is to maintain all the robots *continuously* connected to the BS, either directly or in a multi-hop fashion: this could be useful, for instance, in situations where real-time image streaming is important (e.g., in search and rescue). This problem has been studied in [13] and [7]. The algorithm proposed in [13] constructs a connected exploration tree in which the robots are organized in explorers and link stations: explorers are placed at the leaves of the tree, while the link stations are the inner nodes and ensure the connectivity of the BS (the root) with the explorers. In [7], the authors devise a local search method where the utility of the team configuration is computed in terms of distances from the closest frontiers: a configuration that does not allow full connectivity is highly penalized and is never chosen by the algorithm.

Another way in which the communication constraints can be intended is to ensure global connectivity only at the *deployment locations* of the robots. This is motivated by the fact that, typically, new information is gathered at the robots' goal locations, and robots can get disconnected while travelling to them. In [14], the authors study the problem of mobile sensors placement for maximizing the coverage of an unknown area while keeping each node connected to a BS via a multi-hop mutual-visibility constraint. The algorithm proceeds by sequentially deploying nodes after having selected the best goal locations by means of selection policies. A more recent work related to communication nodes deployment has been presented by Stump et al. [15]. Here, a set of agents is assumed to be already present in an environment (e.g., exploring), and two problems are faced: (i) finding a deployment of relay nodes which ensures global connectivity between each agent and the BS (again stated in terms of a mutual visibility constraint) and (ii) given the current deployment and new locations agents should reach, find the redeployment which minimizes the robots' traveling time. The former problem is reduced to the computation of a Minimum Steiner Tree with the agents' locations as terminal set, while the latter is solved by means of a (generally sub-optimal) dynamic programming algorithm. Finally, in [8], the authors propose a method for online multirobot exploration that ensures, besides full connectivity from the frontiers to the BS, a sufficient bandwidth for the transmission of data on the relay chain. This is achieved by splitting the problem in three sub-problems (explorers placement, relays placement, and path generation) which are solved as variations of known combinatorial optimization problems.

A third way in which the communication constraints have been thought is *periodic reconnection*: robots may be allowed to explore several regions in autonomy, but must be able to communicate their discoveries to the BS under a more or less strict regime. The work proposed in [16] considers a general mission scenario in which the robots must *synchronously* regain global connectivity with the BS after a fixed time interval. The authors prove the inapproximability of the problem and propose a heuristic algorithm based on planning robots' paths in turns, choosing the best path from a pool of samples according to a utility function which, in an exploration context, may be related to the information gain of the path. In [17] and [9], the authors consider periodic connectivity as an *asynchronous* condition that, although desired, is not put in the form of a real constraint, being only the result of an emerging behavior of the algorithms. Specifically, [17] proposes the so-called Role-Based exploration, in which robots are allowed to explore without taking into account communication limits. Rendezvous points, where it is known that explorers can communicate to the BS (possibly through relays), allow asynchronous updates of the environment map on the BS. In [9], the behavior of the robots is regulated by a utility function which considers the amount of information a robot has not yet delivered to the BS and the supposed amount of information known by the BS. Tuning a parameter, the mission planner is able to specify strategies ranging from a completely greedy exploration, with no returns to the BS, to an exploration ensuring the maximum update frequency to the BS. Asynchronous connectivity is also ensured by the works of [18] and [19]. The former proposes a behavior-based architecture, which is tested in scenarios with increasing prior information about the environment. The latter, although not explicitly considering a fixed BS, is able to achieve full exploration of an unknown environment with an architecture which relies on a small set of behaviors and messages exchanged between robots and dropped beacons. In both these two last works, an appropriate behavior regains connectivity with other robots when it is lost.

3. Problem formulation

Regardless of how the communication constraints are intended, the general problem setting we consider in this paper can be formalized in terms of the following common elements:

- A two-dimensional, continuous, and bounded environment $\mathrm{Env} \subset \mathbb{R}^2$. The interior points of the environment can belong to obstacles of arbitrary shape, whose set is denoted by Env_o, or can belong to the free space denoted by $\mathrm{Env}_f = \mathrm{Env} \setminus \mathrm{Env}_o$.
- A fixed BS, placed in Env.
- A set of m mobile robots moving in Env, equipped with sensors that allow covering a 180° area centered at the current heading of the robot, with finite range, and able to perceive free space and outer boundary of obstacles (e.g., a laser range scanner). From any position, a robot can perceive the surrounding environment and update a map that keeps track of the portion of the environment it has discovered. We denote by A the whole set of agents, composed by the m robots and the BS.

- A communication mechanism (e.g, line-of-sight or distance-based), which allows the robots and the BS to communicate between each other, either directly or in a multi-hop fashion. In this work, we assume that the bandwidth along a communication link is always sufficient to communicate all the information.

Informally, the two required (possibly conflicting) objectives of the problem are: (i) exploring the whole environment as fast as possible (or, more generally, ensuring the maximum coverage at any time instant) and (ii) relaying the newly gathered information to the BS as often as possible. In the next section, we will give a precise formalization of two kinds of communication constraints with a BS, denoted as "hard" and "soft", and we will describe the methods we selected for comparison that allow a team of robots to explore an unknown environment under such constraints.

4. Hard communication constraints

As new information is typically obtained at the frontiers between known and unknown space, it is natural to adopt the concept of frontier-based exploration [12] in the definition of the hard communication constraints with a BS:

Communication constraints are said to be hard if (i) when a robot acquires some information at some location, it must be able to forward it to the BS from that same location, and (ii) before any new plan is computed, the whole team (robots and BS) must be globally connected.

We do not explicitly formulate the stronger hard constraint such that the robots are required to be connected at *any* time during exploration. This, however, can be seen as a special case of our definition.

We now provide a formal setting in which to model the exploration process. We represent the environment as a graph $G = (V, E)$, where V represents (a discretized version of) the locations of Env that must be explored, and the set E encodes the shortest paths whose length is $d(v_i, v_j)$ that the robot should traverse between pairs of vertices $v_i, v_j \in V$. In addition to this, we consider a set C of communication links. This set specifies which communications can be performed on the graph G. More formally, if $(v_i, v_j) \in C$, then a robot at v_i can communicate with a robot at v_j and *vice versa.* We assume that the communication mechanism is such that communication links are based on the physical connections between vertices. More formally, $C \subseteq E$, where C can be computed from E (a typical mechanism is to assume that, if the distance between two adjacent vertices in G is shorter than some threshold, then a communication link is available between them). The graph G is initially unknown to the robots that must discover it in an incremental way by taking joint perceptions through a sequence of discrete epochs. At a generic epoch t, we denote with G^t the portion of the graph known by the BS at t (this graph can be defined as a subgraph of G where sets E^t and C^t are restricted versions of E and C such that only the vertices explored within epoch t, denoted as V^t, are included). We denote as *team deployment* at epoch t the set $Q^t = \{b, q_1^t \dots q_m^t\}$ where $b \in V^t$ denotes the fixed location of the BS and $q_i^t \in V^t$ denotes the location (vertex) occupied by robot i in G^t *before* taking any perception (here, we ignore the robots' orientation). Once each robot reached its deployment location specified in Q^t, perceptions are taken and, assuming that each robot can forward its sensing data to the BS, graph G^{t+1} is computed at the BS. Such graph can be obtained by merging all the new perceptions made by those robots occupying frontier vertices, i.e., those vertices representing locations in V^t at the boundaries between known and unknown environment. We will typically refer to a frontier vertex as f, to which a utility value $u(f)$ is associated. This value can be derived for each f by keeping an additional grid map representing the explored environment (not to be confused with the graph G^t), and measuring the number of adjacent grid cells placed at the frontier between known and unknown space forming a cluster with center in f; such a value can also be seen as an information gain estimate since frontiers that have a higher number of cells are more likely to provide new information than those with a lower number of cells. Formally, the new information from perceptions can be defined in the following way. For a vertex $v \in V^t$, let us denote with $N(v)$, $\mathcal{E}(v)$, and $\mathcal{C}(v)$ the set of vertices adjacent to v in G, the set of physical edges of E incident to v in G, and the set of communication links of C incident to v in G, respectively. Then, we can express the transition from G^t to G^{t+1} with the following simple rules (for a generic agent a): $V^{t+1} = V^t \bigcup \underset{a \in A}{\cup} N(q_a^t)$, $E^{t+1} = E^t \bigcup \underset{a \in A}{\cup} \mathcal{E}(q_a^t)$, and $C^{t+1} = C^t \bigcup \underset{a \in A}{\cup} \mathcal{C}(q_a^t)$.

Given the above setting, the hard communication constraints can be expressed as a feasibility requirement for any team deployment. At any epoch t, once graph G^{t+1} has been computed by integrating in G^t perceptions at deployment Q^t, the feasible deployments Q^{t+1} are those for which the subgraphs of G^{t+1} induced by Q^{t+1} are connected with respect to C^{t+1}. In such deployments Q^{t+1}, each robot can communicate (directly or along a multi-hop route) with the BS whose location is fixed at b in any deployment: therefore, it is natural to interpret the problem in a centralized way by assuming that exploration is dictated by some planning process residing at the BS.

4.1. Formulation as an Integer Linear Program

We now formalize the model allowing to compute the optimal team deployment Q^{t+1} given the newly perceived graph G^{t+1} and the current deployment Q^t. In doing so, we define measures of costs and gains for deployments. Costs can be defined with no uncertainty, being related to the transition from the previous deployment to the new one, as the cumulative distance traveled by the robots to reach their assigned positions (in most indoor environments, it is a reasonable assumption to assume the time needed by robots to rotate negligible). Gains, on the other side, can only be estimated since they depend on the amount of new information that can be discovered. As customarily done, we assume to have some estimated measure of the information gain achievable by a perception from some vertex v and we denote it as $g(v)$. For non-frontiers vertices we will have $g(v) = 0$. However, as we have already discussed, the hard communication constraint requires, in general, to assign robots to some of the non-informative vertices in order to comply with the connection requirements. Assuming to search for an optimal solution with respect to a given trade-off between costs and gains, we can formulate such a problem with an integer linear program (ILP). The model is based on the *directed* graph derived by G^{t+1} by doubling each undirected communication edge, except for the set of edges adjacent to the BS, for which we keep only the outgoing arcs. Therefore, with a slight abuse of notation, C^{t+1} will denote in this section the set of all directed arcs derived as described above.

We employ three sets of binary decision variables from which we can derive the solution of the planning problem at epoch t after the graph G^{t+1} has been computed, namely the deployment Q^{t+1}:

- z_{av} for an agent $a \in A \setminus \{BS\}$ and a vertex $v \in V^{t+1} \setminus \{b\}$; it takes value 1 if and only if $q_a^{t+1} = v$;
- y_v for a vertex $v \in V^{t+1} \setminus \{b\}$; it takes value 1 if and only if $\exists a \in A : q_a^{t+1} = v$;
- x_{ij} for each directed arc $(v_i, v_j) \in C^{t+1}$; it takes value 1 if and only if for some $v_i, v_j \in V^{t+1}$ it holds that $i \in Q^{t+1}$ and $j \in Q^{t+1}$.

We denote with $\mathcal{C}^{\pm}(v)$ the arcs entering(+)/leaving(-) vertex $v \in V^{t+1}$ and with $\delta^{\pm}(S)$ the directed cuts induced by the set of vertices $S \subseteq V^{t+1}$. Finally, we denote with α the parameter used to tune the trade-off between costs and gains. The ILP model reads as follows:

$$\text{maximize} \sum_{a \in A} \sum_{v \in V^{t+1}} (g(v) - \alpha d(q_a^t, v)) z_{av} \tag{1}$$

subject to

$$\sum_{a \in A \setminus \{BS\}} z_{av} = y_v \qquad \forall v \in V^{t+1} \setminus \{b\} \tag{2}$$

$$\sum_{v \in V^{t+1} \setminus \{b\}} z_{av} = 1 \qquad \forall a \in A \setminus \{BS\} \tag{3}$$

$$\sum_{(i,j) \in \mathcal{C}^-(v)} x_{ij} = y_v \qquad \forall v \in V^{t+1} \setminus \{b\} \tag{4}$$

$$\sum_{(i,j) \in \delta^-(S)} x_{ij} \geq y_v \qquad v \in S, b \notin S, \ \forall S \subseteq V^{t+1} \tag{5}$$

The objective function maximizes the cumulative information gain robots can get from a joint perception, while accounting for the cost as the total traveled distance. The parameter α can be chosen so that the maximum cost spent by a robot for traveling is never bigger than the minimum achievable information gain, thus encoding an order of preference for the two objectives. To obtain such a property we adopt the following value for α:

$$\alpha = \frac{\min_{v \in V^{t+1}} \{g(v)\} - \epsilon}{m \cdot \max_{v \in V^{t+1}, a \in A} \{d(q_a^t, v)\}}$$

where ϵ is a sufficiently small constant. Constraints (2) ensure that each vertex is occupied by at most one robot; constraints (3) enforce each robot to be placed in exactly one vertex; constraints (4) enforce that every selected vertex has exactly one predecessor on its path to the BS; finally, constraints (5) enforce the fact that if a robot is placed in a vertex v, there must exist a connected sequence of communication links leading to the BS. Note that the number of constraints (5) is exponential in the size of the input. Therefore, to solve the model optimally, we employ a Branch & Cut approach similar to that used in [20] for solving the Steiner Tree Problem. The idea is to gradually introduce violated inequalities (5) as soon as the solution of a new LP relaxation is available (the problem of recognizing such violated constraints is solvable in polynomial time).

4.2. Adapting a dynamic programming method

The second method we consider for comparison is based on the work by Stump et al. [15] that uses dynamic programming to compute a deployment trying to minimize the total traveled distance and, at the same time, keeping a communication route between a BS and some of the assigned target locations. The algorithm receives as input:

- a graph G where vertices V represent possible robot locations and communication links C represent direct communication links between locations;
- the current team deployment Q^t and the current communication topology C^t; here, the communication topology is not intended as the set of available communication links (like we introduced above), but as the minimal tree of links which is actually used to forward transmissions;
- a pre-assigned robot-frontier assignment for a subset (possibly empty) of robots.

The output of the method is a new team deployment Q^{t+1} where the locations of pre-assigned robots are fixed and other robots' positions form a connected configuration with the BS that minimizes the traveling cost. Relay robots are assigned to intermediate known vertices and the movement and communication costs of these assignments are computed according to:

$$Q(V^t, C^t, Q^t, C^{t+1}, Q^{t+1}) = \sum_{a \in A} d(q_a^t, q_a^{t+1}) + \mu\Big(\sum_{(v_i, v_j) \in C^{t+1}} w(q_a^t, q_a^{t+1})\Big) \quad (6)$$

where C^{t+1} is the new communication topology to be used for transmissions from robots on the frontiers to the BS; Q^{t+1} is the new team deployment; $d(v_i, v_j)$ is, just like we formulated in the previous section, the distance from vertex v_i to vertex v_j; and $w(v_i, v_j)$ represents the communication cost incurred when sending data from vertex v_i to vertex v_j using communication topology C^{t+1}. Finally, μ is a parameter for trading off between redeployments where the robots move as little as possible (small μ) and redeployments where the robots end up as close together as possible (large μ). This method cannot be adopted as it is for our exploration setting, as we do not consider any pre-assignment of robots to frontiers but we instead include it in the solution we search for. To this end, we adapted the method proposed in [15] by computing assignments according to a simple greedy algorithm that maximizes the utility of selecting frontier vertex f for robot a. The utility function is defined similarly to what done in [9] as $u(f)/d^2(q_a^t, f)$, where $u(f)$ is the utility value associated to the frontier vertex f, presented above, and $d(q_a^t, f)$ is the distance between q_a^t and f. If such joint assignment admits a connected communication topology when all robots are assigned to frontiers then it is adopted. Otherwise, the robot-frontier assignment with the lowest utility is removed and the algorithm of [15] is run again until a deployment with a connected communication topology is found.

4.3. A local search method

The method presented by Rooker and Birk in [7] is based on a local search approach on a grid-represented environment. In our experiments we assume that each grid cell corresponds to a graph vertex and cell adjacencies correspond to physical adjacencies (set E) in our graph. Given the current team deployment Q^t, the method works according to the following steps:

1) a set of candidate team deployments $Q_1^{t+1}, Q_2^{t+1}, \ldots, Q_k^{t+1}$ is randomly generated from Q^t in this way: each Q_i^{t+1} $(1 \leq i \leq k)$ contains a random location for each robot $a \in A$ (thus m locations), such that it is either $q_a^t \in Q^t$ (remaining in that vertex) or is adjacent to q_a^t (moving to an adjacent vertex);
2) each generated deployment is associated with a utility $U(Q_i^{t+1})$, and the one maximizing $U()$ is chosen as the next deployment; then the process restarts from the previous step.

After some preliminary experiments, we slightly modified the utility function used in [7] to improve the quality of the obtained solutions. The utility we adopted is defined, with a slight override of notation, as $U(Q_a^{t+1}) = \sum_{a \in A} U(q_a^{t+1})$ where:

$$U(q_a^{t+1}) = \begin{cases} -M & \text{if } q_a^{t+1} \text{ not feasible,} \\ u(f^*_{q_a^{t+1}})/d^2(q_a^{t+1}, f^*_{q_a^{t+1}}) & \text{otherwise.} \end{cases} \quad (7)$$

The rationale is the following. An assigned location q_a^{t+1} is not feasible if it leads to a collision with an obstacle or if reaching it will cause a communication loss with the BS. In such cases, a large penalty $-M$ is applied. Otherwise, the utility of such an assigned location is related to $f^*_{q_a^{t+1}}$, namely the frontier location closest to q_a^{t+1} ($u()$ and $d()$ have the same meaning of the previous sections). This method inevitably suffers of local minima problems. We dealt with them by setting a maximum number of iterations allowed to compute a single deployment: when this threshold is surpassed, the frontier with the least utility is removed from the set of available frontiers and the algorithm repeats.

5. Soft communication constraints

The soft communication constraints can be defined as:

Communication constraints are said to be soft if the communication between the BS and the robots, despite being a desired condition, needs not to be maintained on a regular basis.

According to this definition and recalling what said in the previous section, a robot may explore more than one frontier before being able to flush information to the BS through a relay link. We now give a brief overview of the method presented in [9].

5.1. Utility-based exploration

In this method, the mission planner sets a threshold parameter $r \in [0, 1)$ which represents the trade-off between a greedy exploration behavior ($r = 0$) and the willingness of the robots to return to the BS to communicate their discoveries ($r \to 1$). Following the same notation of [9], let InfBase_i be the information robot i believes the BS holds at the current time. This can be obtained either by direct communication with the BS or by exchanging a message with a robot which has communicated to the BS more recently than i. Define now InfNew_i to be the new information of robot i (the one that has not already been sent to the BS). If two agents i and j meet, and j is closer to the BS than i, InfNew_i and InfNew_j are updated as $\text{InfNew}_i = \emptyset$, $\text{InfNew}_j = \text{InfNew}_i \cup \text{InfNew}_j$: this is done in order to reduce the amount of redundant information delivered to the BS. At any control step, a robot decides to return to a location where it is able to communicate with the BS if:

$$\frac{|\mathrm{InfBase}_i|}{|\mathrm{InfNew}_i| + |\mathrm{InfBase}_i|} < r \tag{8}$$

where the absolute value denotes the area of the region. When (8) is not satisfied, the robot a greedily chooses a frontier according to the same utility function defined for the previous methods, namely $u(f)/d^2(q_a^t, f)$.

6. Simulation activity

To perform replicable tests under controlled conditions, we use a robot simulator. Although perfect localization and sensor data are assumed, we selected MRESim [10], because it focuses on the communication aspects and it has been also used to test the methods in [17] and [9].

The simulated robot is a differential drive robot, like a P3AT, equipped with a laser range scanner with a maximum range of 100 pixels (cells), a 180° FOV, and an angular resolution of 1°. The environment is represented as an occupancy grid and the communication is based on a limited line-of-sight model (range of 200 pixels). Note that the sensor and communication ranges could correspond to 5 m and 10 m, respectively, considering 5 cm a pixel.

The exploration strategies presented in the previous sections are evaluated in three environments where robots start from fixed locations (see Figure 1): all the three environments are taken from the MRESim repository and are representative of different scenarios. Their size is about 800 by 600 pixels, thus resulting in about 40 by 30 m. We consider teams of 2, 4, 6, and 8 robots. We define an experimental setting as an environment (office, open, or maze), a number of robots (2, 4, 6, or 8), an exploration strategy (OptHard of Section 4.1, Stump adaptation of Section 4.2, Rooker method of Section 4.3, and Utility of Section 5.1; for the latter we consider values of 0.1, 0.5, and 0.9 for r). For each experimental setting, we execute 5 runs of 500 simulation cycles each, since there could be some situations in which the robot cannot find any feasible path (due to some error approximation in integrating the sensory data into the grid map) and it recovers by taking a random movement. We slightly modified the Utility method, so that, as soon as the remaining number of simulation cycles approaches those needed for returning to the BS to make reports, robots choose to come back. Robots' default speed was set to 4 pixels/cycle: therefore, assuming a realistic speed of 20 cm/s, each simulation cycle corresponds to approximately 1 second.

The performances are assessed by measuring the traveled distance (the mean over the robots), the time that the robots are not in communication with the BS (the mean over the robots), the percentage of area discovered by the robots and known to the BS, and the replan time for deciding the next location(s) to reach. For reasons of space, we will report only the results obtained without including the replanning time along the simulation cycles, because we want to focus on the *goodness* of the online decisions undertaken by the different strategies. Indeed, hardware platforms or software implementations could have a strong impact on the replanning time and affect the overall team performance. All the experiments were run on a Linux machine with 2.7 GHz i5-4310M CPU and 8 GB RAM, while the ILP models are solved with the GUROBI solver [21].

Figure 2 shows the results for the office environment. We can observe that the traveled distance and the explored area are relatively low for the OptHard, Stump, and Rooker methods compared to the Utility method (regardless of the ratio parameter r). This can be explained by the fact that the first three methods ensure hard communication constraints between the BS and the robots, while the latter adopts just a soft constraint. Indeed, looking at the time not in communication, the values are low for the former methods, while they are high for Utility method.

Comparing the methods ensuring hard constraints, the Stump method performs always better than the Rooker one in terms of explored area, while OptHard only slightly overcomes Stump. The reason is twofold. First, we are solving an online problem: since the information gain can only be estimated, an optimal model given in terms of information gain does not ensure to actually obtain the best possible (offline) deployment. Second, we experimented the optimal ILP model by giving total precedence to the information gain: therefore, robots could travel a lot of distance to reach a deployment which is perhaps only slightly better than another "closer" to the current one. We leave to future work the exploration of different trade-offs. Focusing on the Rooker method, its main weakness regards the fact that it can end up in some local minima and take some time to escape from them: however, this is the only method able to guarantee continuous communication.

Now, let us look at how performance changes when varying the parameter r of the Utility method. In terms of explored area and travelled distance, the difference is not statistically significant (e.g., in the office environment, considering 8 robots with Utility0.1 and Utility0.9 the p-values are 0.3632 and 0.5097 for the explored area and traveled distance, respectively). In most of the cases, there is a slight decrease on the time in which robots do not communicate with the BS with increasing r. However, this difference is not statistically significant (e.g., for the same setting in the previous example, p-value$= 0.9059$). This can be justified by the fact that once the robots communicate the new information they return to explore the environment retraversing and reaching areas where they cannot communicate with the BS.

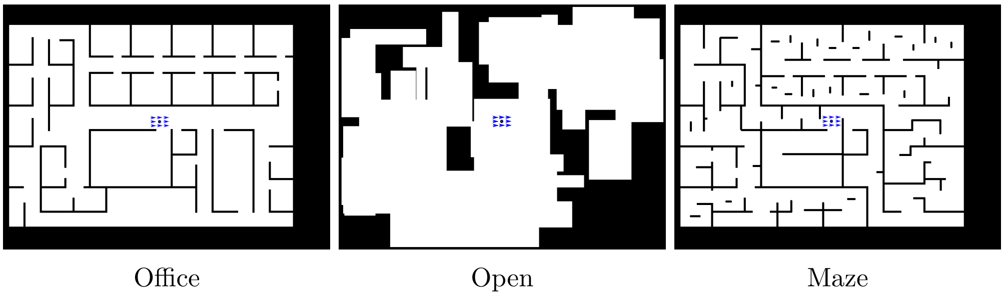

FIGURE 1. Test environments, size 800 by 600 pixels.

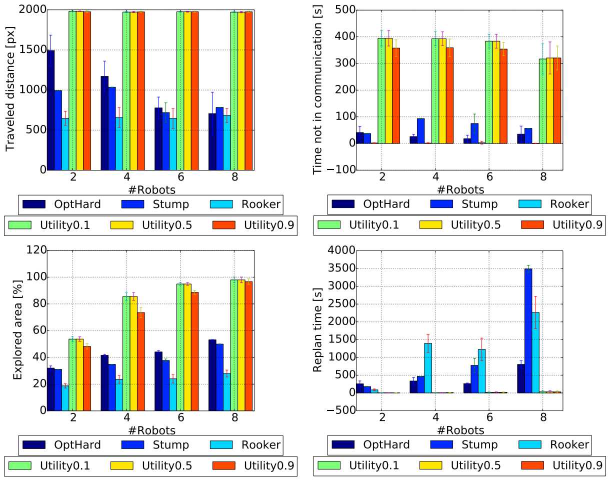

FIGURE 2. Results (average and standard deviation) for the office environment.

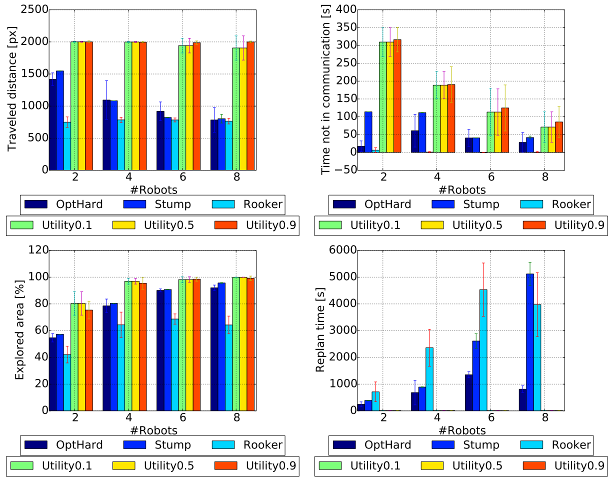

FIGURE 3. Results (average and standard deviation) for the open environment.

Increasing the number of robots provides a significant advantage in terms of explored area (e.g., in the office environment, with ILP method and 2 or 8 robots, p-value$< 10^{-9}$) and traveled distance for hard-constraints methods (e.g., for the same setting, p-value$= 0.0007$). Looking at the replanning time, we noticed that Rooker and Stump methods slow down with the number of robots, while Utility method, being basically reactive, is almost instantaneous. Stump's method, despite being a heuristic, needs in general more time than OptHard method. This is due to the fact that the reduced size of the exploration graph (never more than 100 vertices) is such that the Branch & Cut approach used for solving the ILP is very effective.

Figure 3 presents the results for the open environment. This environment seems easier to explore than the previous one, as the Utility method is able to attain almost 100% of explored area from 4 robots. Also, the OptHard method reaches quite good results (e.g., more than 90% for 6 and 8 robots). Compared to the OptHard, Stump always performs slightly better in this environment (considerations similar to that of the previous environment hold). Again, the Rooker method is the worst in terms of explored area, but, w.r.t. the office environment, the performances are more than doubled. This can be explained by the fact that the robots are able to perceive big portions of the environment with just one perception. The time not in communication significantly decreases increasing robots for the Utility methods (e.g., in the open environment, with Utility0.5 and 2 or 8 robots, p-value$< 10^{-9}$): this is due to the presence of few environmental obstacles, so that there are greater chances of being able to form a multi-hop bridge to the BS.

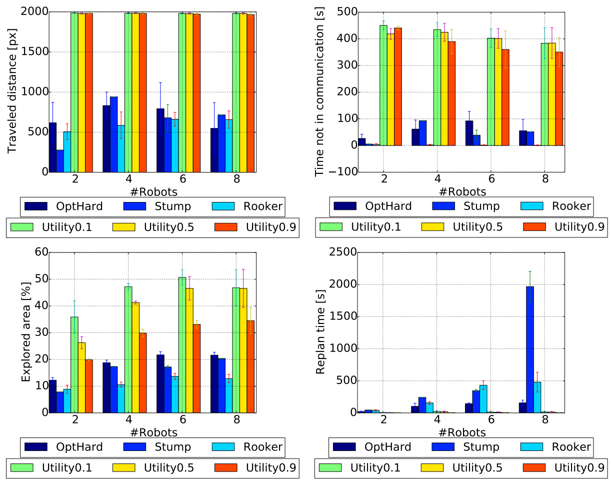

FIGURE 4. Results (average and standard deviation) for the maze environment.

Figure 4 shows the results for the maze environment. Here, the trends are significantly different w.r.t.

the previous environments in explored area and replanning time. First, the best combination of number of robots/method is only able to reach an average of 50% explored area. Second, choosing $r = 0.9$ for the Utility method seems to be a very bad choice, since the reduction in the explored area is only slightly compensated by a reduction in the time not in communication. Third, choosing $r = 0.1$ for the Utility method and 4 robots provides almost the best performance: in this complicated environment, the Utility method does not scale well with the number of robots. Focusing on the replanning time, we can notice that the time needed by Rooker method is dramatically reduced w.r.t. the previous environments: since the exploration proceeds more slowly, the path planning algorithm (A^*) is always able to provide the distance to the frontiers of each candidate configuration within a short time.

7. Concluding remarks

In this paper, we have provided a tentative taxonomy of communication constraints for multirobot exploration. We have presented an optimal formulation for the exploration problem with hard communication constraints. Then, we experimentally compared four exploration strategies with different communication constraints that mobile robots can employ in mapping unknown environments. This experimental analysis led to some useful quantitative insights on the strengths and the weaknesses of the different approaches. In particular, the simulations we performed expose the tradeoffs between communication constraints and performance. Our results can help to better assess the impact of communication requirements in exploration tasks and to predict performance variations when choosing different constraining methods.

The results presented in this paper constitute only a first step toward a comprehensive assessment of different exploration strategies with communication constraints. For instance, more environments, more exploration strategies, and more communication models need to be evaluated to draw stronger conclusions that could be useful for the design of better exploration strategies.

Acknowledgements

The authors thank Victor Spirin for providing the source code of the simulator and his method and Mattia Ornaghi for the help in implementing the exploration strategies.

References

[1] S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*, pp. 1–35. Morgan Kaufmann, 2002.

[2] S. Tadokoro. *Rescue Robotics*. Springer, 2010.

[3] F. Amigoni. Experimental evaluation of some exploration strategies for mobile robots. In *Proc. ICRA*, pp. 2818–2823. 2008.

[4] M. Julia, A. Gil, Ó. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Auton Robot* **33**(4):427–444, 2012.

[5] A. Quattrini Li, R. Cipolleschi, M. Giusto, F. Amigoni. A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings. *Auton Robot* 2015. To appear, DOI:10.1007/s10514-015-9480-x.

[6] K. Wurm, C. Stachniss, W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. IROS*, pp. 1160–1165. 2008.

[7] M. Rooker, A. Birk. Multi-robot exploration under the constraints of wireless networking. *Control Eng Pract* **15**(4):435–445, 2007.

[8] Y. Pei, M. Mutka, N. Xi. Connectivity and bandwidth-aware real-time exploration in mobile robot networks. *Wirel Commun Mob Comput* **13**(9):847–863, 2013.

[9] V. Spirin, S. Cameron, J. de Hoog. Time preference for information in multiagent exploration with limited communication. In *Proc. TAROS*, pp. 34–45. 2013.

[10] J. de Hoog, S. Cameron, A. Visser. Role-based autonomous multi-robot exploration. In *Proc. COGNITIVE*, pp. 482–487. 2009.

[11] G. Tuna, K. Gulez, V. Gungor. The effects of exploration strategies and communication models on the performance of cooperation exploration. *Ad Hoc Netw* **11**(7):1931–1941, 2012.

[12] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. AGENTS*, pp. 47–53. 1998.

[13] P. Mukhija, K. Krishna, V. Krishna. A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint. In *Proc. IROS*, pp. 4806–4811. 2010.

[14] A. Howard, M. Matarić, G. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Auton Robot* **13**(2):113–126, 2002.

[15] E. Stump, N. Michal, V. Kumar, V. Isler. Visibility-based deployment of robot formations for communication maintenance. In *Proc. ICRA*, pp. 4498–4505. 2011.

[16] G. Hollinger, S. Singh. Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE T Robot* **28**(4):967–973, 2012.

[17] J. de Hoog, S. Cameron, A. Visser. Autonomous multi-robot exploration in communication-limited environments. In *Proc. TAROS*, pp. 68–75. 2010.

[18] R. Arkin, J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *Proc. AMC*, pp. 455–461. 2002.

[19] E. Jensen, E. Nunes, M. Gini. Communication-restricted exploration for robot teams. In *Multiagent Interaction without Prior Coordination Workshop at AAAI*. 2014.

[20] T. Koch, A. Martin. Solving steiner tree problems in graphs to optimality. *Networks* **32**(3):207–232, 1998.

[21] Gurobi optimizer reference manual. `http://www.gurobi.com`, 2015.