

# A Decentralized Strategy for Cooperative Robot Exploration

Antonio Franchi Luigi Freda Giuseppe Oriolo Marilena Vendittelli  
Dipartimento di Informatica e Sistemistica  
Università di Roma “La Sapienza”  
Via Ariosto 25, 00185 Roma, Italy  
{franchi,freda,oriolo,venditt}@dis.uniroma1.it

**Abstract**—We present a decentralized cooperative exploration strategy for mobile robots. A roadmap of the explored area, with the associate safe region, is built in the form of a compact data structure, called *Sensor-based Random Graph*. This is incrementally expanded by the robots by using a randomized local planner which automatically realizes a trade-off between information gain and navigation cost. Connecting structures, called *bridges*, are incrementally added to the graph to create shortcuts and improve the connectivity of the roadmap. Decentralized cooperation and coordination mechanisms are used so as to guarantee exploration efficiency and avoid conflicts. Simulations are presented to show the performance of the proposed technique.

## I. INTRODUCTION

Exploration of unknown environments is one of the most challenging problems in robotics. This task typically requires a mobile robot to cover an unknown area while learning, at the same time, a model of the environment or locating a given object. A wide range of applications are possible including automated surveillance, search-and-rescue operations in hostile areas, map building and planetary missions.

In general, many advantages result from the use of a multi-robot system [1], [2]. In exploration, a team of robots typically reduce the time required to complete the task. If a map is to be acquired, the redundant information provided by multiple robots can be also used to increase the final map accuracy and the quality of the localization [3]. In order to achieve these objectives, some sort of task decomposition and allocation are required. In practice, strategies to conveniently distribute robots over the environment should be accurately devised in order to reduce the occurrence of spatial conflicts [4] and actually reap the benefits of a multi-robot architecture. Clearly, communication plays a crucial role in achieving a cooperative behavior with improved performance [5].

In most exploration strategies, the boundary between known and unknown territory (the *frontier*) is approached in order to maximize the information gain. For the multi-robot case, a pioneering work in this direction is [6]: the robots merge the acquired information in a global gridmap of the environment, from which the frontier is extracted and used to plan the individual robot motions. While this basic scheme lacks an arbitration mechanism preventing robots from approaching the same frontier region, in [7] it is proposed to negotiate robot targets by optimizing a utility function which takes into account the information gain of a particular region, the cost

of reaching it and the number of robots currently heading there. In [8], the utility of a particular frontier region from the viewpoint of relative robot localization (and hence, of the accuracy of map merging) is also considered. In the incremental deployment algorithm of [9], robots approach the frontier while retaining visual contact with each other. An interesting multi-robot architecture in which robots are guided through the exploration by a market economy is presented in [10], whereas [11] proposes a centralized approach which uses a frontier-based search and a bidding protocol assign frontier targets to the robots.

This paper presents a randomized strategy for cooperative robot exploration which is an evolution of the method presented in [12]. The basic tool used in that work is the *Sensor-based Random Tree* (SRT), a compact data structure representing a roadmap of the explored area [13], [14]. In particular, in [12], the robots of the team build a roadmap of the environment in the form of a *forest* of SRTs. At first, each robot expands its own tree, then, it enters in a supportive phase helping other robots in building their SRTs. No connections between the trees are created during the expansions. Hence, to allow the supportive phase, an ad hoc structure connecting the roots must be added.

In the present work, the robots of the team cooperatively build a map of the environment in the form of a *single* graph, called *Sensor-based Random Graph* (SRG). In this case, the separation of the algorithm in two phases is overcome since, at all times, the robots expand the same graph with the possibility of moving towards any local node which is safely reachable.

As in [12], each node of the SRG contains a collision-free robot configuration and the Local Safe Region (LSR) perceived from that location; an arc between two nodes represents a collision-free path between the two configurations. In particular, in [12], the trees represent exploration paths actually traveled by the robots. In the SRG-based exploration, new structures, called *bridges*, are added to these exploration paths to join ‘directly’ connectable nodes. The bridges improve the capability of the roadmap to capture the connectivity of the free area of the environment. Essentially, they act as shortcuts which can be traveled by the robots.

Each robot of the team builds a portion of the SRG by using a randomized local planner which privileges the frontier of the current LSR, i.e., the directions that lead from the LSR

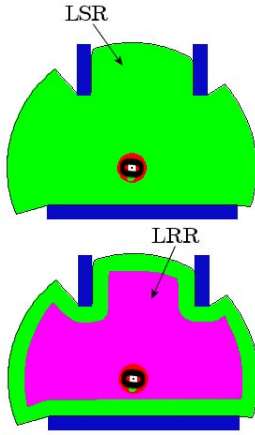


Fig. 1. The LSR and the LRR of the robot.

to unexplored areas. This mechanism automatically realizes a trade-off between information gain and navigation cost when choosing the next robot configuration.

Each robot of the team participates in building the SRG, taking into account the presence of other robots through the computation of its local frontier, and planning its motion toward areas which appear to be unexplored by itself as well as the rest of the team. In addition to this cooperation mechanism, there is a simple coordination algorithm that guarantees safe collective motion. A key feature of the proposed strategy is that it is completely decentralized and can be implemented with a limited communication range.

The paper is organized as follows. Our working assumptions are stated in Sect. II. The SRG data structure is discussed in Sect. III. Then, in Sect. IV, we describe the basic steps of the exploration algorithm running on each robot. Simulation results in different environments are reported and discussed in Sect. V. Possible extensions of the present work are mentioned in the concluding section.

## II. PROBLEM SETTING

The cooperative SRG-based exploration method is presented under the following assumptions.

- 1) All the robots in the team are identical.
- 2) The robots move in a planar workspace, i.e.,  $\mathbb{R}^2$  or a connected subset of it.
- 3) Each robot is a disk whose configuration  $q$  is the position of the disk center<sup>1</sup>.
- 4) Each robot knows its configuration.
- 5) The robots are equipped with a sensory system which provides the *Local Safe Region*  $LSR(q)$ , a (possibly conservative) description of the free space surrounding the robot at  $q$ . The LSR is a star-shaped<sup>2</sup> subset of  $\mathbb{R}^2$ , whose maximum radius is bounded by the robot perception range  $R_p$ .

<sup>1</sup>Hence, the configuration space of each robot is a copy of the workspace with the obstacles grown so as to allow for the robot size [15].

<sup>2</sup>This means that  $LSR(q)$  is homeomorphic to the closed unit disk and a line segment from the robot center to any point in  $LSR(q)$  is completely in  $LSR(q)$ .

6) Each robot can broadcast within a communication range  $R_c$  the information stored in its memory (or relevant portions of it) at any time. The robot ID number is included in the heading of any transmission. The robot is always open for receiving communication from other robots inside  $R_c$ .

Many of these assumptions are only taken for simplicity and can be relaxed. First, the robots do not need to be identical: for example, the sensory systems may very well differ in nature and perception range. The assumption of planar workspace is obviously not restrictive: 3D worlds are perfectly admissible as long as the sensory system allows the reconstruction of a *planar* LSR for planning the robot motion. Assumption 3 is only taken for ease of presentation: the proposed method is readily applicable to robots with arbitrary shape. As for Assumption 4, it can be eliminated by incorporating a localization module in the SRG method as described in [16]. Obviously, Assumption 5, and in particular the star-shaped hypothesis, is consistent with the physics of most common sensors, i.e., range-finders, but it also applies to more sophisticated perception techniques (e.g., vision). See the concluding section for additional comments.

## III. THE SENSOR-BASED RANDOM GRAPH

Each node of an SRG represents a collision-free configuration  $q$  together with the associated Local Safe Region  $LSR(q)$ . An arc between two nodes represents a collision-free path between the two configurations.

Other fundamental data associated to each node of the SRG are: the *Local Reachable Region*  $LRR(q)$ , which is the set of all the configurations which can be reached from  $q$  with the robot staying within  $LSR(q)$  (see Fig. 1); the *local frontier*, i.e., the portion of the LRR boundary leading to unexplored areas; the *node weight*, which is equal to the local frontier length.

Clearly,  $LRR(q)$  and  $LSR(q)$  coincide if the robot is a point. When the robot disk has a finite size and is adequately controllable (i.e., STLC),  $LRR(q)$  is obtained from  $LSR(q)$  by 1) erosion [17] in which the robot disk is the structuring element

2) extraction of the connected component which contains  $q$ .

In this framework, the *Safe Region* SR of a robot is the union of all the LSRs collected in its graph. Correspondingly, the *Reachable Region* RR of a robot is the connected component containing its current configuration in the erosion of its SR. While the SR represents the explored area in the workspace, the RR represents the explored and reachable region in the configuration space.

The SRG is incrementally built by extending the structure in the most promising direction via a biased random mechanism. A local coordination strategy takes into account other robots in order to maximize the information gain and guarantee collision avoidance.

Note that, the  $i$ -th robot of the team has its own ‘representation’  $SRG_i$  of the SRG, either acquired directly or through communication with other robots. The SRG cooperatively built

```

SRG BASED EXPLORATION
1   build and wait GPA
2   perceive LSR, compute LRR
3   update SRGi    %information received from other robots is used here
4   build GEA
5   extract local frontier of LRR
6   if local frontier is non-empty
7     choose a target on local frontier and plan a safe path to it
8   else
9     compute forward trees of adjacent nodes
10    update local frontiers and weights of nodes in the forward trees
11    if a forward tree has non-empty forward frontier
12      set as target the adjacent node with the greatest weighted forward frontier length
13    else
14      terminate exploration: homing, exit from the algorithm
15  if |GEA| > 1
16    check feasibility of the robot paths in the GEA
17    if there are unfeasible paths
18      choose a master in the GEA    %the master assigns a target to each robot in the GEA
19      receive target from master
20  move to target

```

Fig. 2. A pseudocode description of the SRG-based exploration algorithm running on the  $i$ -th robot.

by the team of robots is the union of all the SRG <sub>$i$</sub> 's and is a distributed representation of the environment.

#### IV. COOPERATIVE SRG-BASED EXPLORATION

The exploration algorithm running on the  $i$ -th robot is shown in Fig. 2. The basic steps therein reported describe the exploration cycle. We first give a quick commentary of these steps, and then discuss their structure in some detail.

At each iteration of the algorithm, the robot uses all the available information (partly gathered on its own and partly acquired through communication with other robots) to identify the *Group of Engaged Agents* (GEA), i.e., the other agents of the team with which cooperation and coordination are necessary. This is achieved by first building the *Group of Pre-engaged Agents* (GPA), i.e., the robots which are candidate to belong to the GEA, and synchronizing with them (line 1, see Sect. IV-A for further details).

Next, the robot gathers data through its sensory system and builds the current LSR and the associated LRR (line 2).

At this point, the SRG <sub>$i$</sub>  is updated (line 3, see Sect. IV-B). A new node, containing the current configuration, the new LSR and LRR, is added to the SRG <sub>$i$</sub> . An arc is created between the previous and the new node. New information received from other robots is merged in the graph. The robot merges the SRG <sub>$i$</sub>  with the SRG <sub>$j$</sub> 's of all the other robots in its communication range. At this point, the robot creates *bridges* between 'directly' connectable nodes.

The actual GEA can now be built (line 4). Next, the robot computes the local frontier of the LRR (line 5, see Sect. IV-C). If this is not empty, the robot generates a random configuration contained in the current LRR and directed towards the local

frontier (line 7, see Sect. IV-D). A collision-free path reaching the new configuration is planned inside the current LRR.

If the local frontier of the current node is empty, the robot needs to locate a promising region on its SRG <sub>$i$</sub>  where unvisited frontiers still remain. To this end, it computes the forward trees of its adjacent nodes and their associated forward frontiers (line 9–11, see Sect. IV-E). In particular, a *forward tree* of an adjacent node is a minimum spanning tree which is rooted in that node and does not contain paths passing through the current node. The *forward frontier* associated to a forward tree is the union of all the local frontiers of its nodes. At this point, if one of the adjacent nodes has a non empty forward frontier, the robot selects as target the adjacent node with the greatest weighted forward frontier length (line 12, see Sect. IV-E for further details).

After the target is selected, the robot checks if the GEA is composed only of itself. If this is the case, the robot directly moves to its target. Otherwise, the prospective paths of the robots in the GEA are checked for mutual collisions, and are accordingly classified in feasible and unfeasible paths (line 16). If there are unfeasible paths, a coordination phase takes place locally. A master is selected in the GEA: this may either confirm or modify the current target of the robot. In particular, the robot's move may be simply forbidden by resetting the target to its current configuration. Last, the target is received from the master (line 19) and then robot moves towards it (line 20).

The main loop is repeated until the termination condition is verified (line 14), i.e., when the current local frontier and the forward frontiers of the adjacent nodes are empty. This means that the robot is unable to further expand the graph (no

frontiers remain) and therefore it has to go back to its starting position (*homing*).

In the above exploration algorithm, only perception, planning and motion functionalities are made explicit. A parallel communication thread runs on each robot, not described here for compactness [18]. In principle, two approaches are possible. In the first, which is chosen for our presentation, each robot continuously broadcasts all its knowledge, including that derived from other robots: this means that the relevant information for cooperation and coordination with other agents inside the communication range  $R_c$  is immediately available to the robot. The second solution, aimed at reducing bandwidth consumption, would be to establish robot-to-robot communications only when needed.

#### A. GPA/GEA construction

At the start of the algorithm, the robot is stationary and needs to identify other robots whose LSRs may overlap with its own, in order to cooperate (optimize the exploration) and coordinate (avoid conflicts) with them. The other robots may be stationary as well (in this case, their targets coincide with the current configuration) or moving towards a target; hence, a synchronization phase is needed.

We say that two robots are *GPA-coupled* if the distance between their targets is at most  $2R_p$ , i.e., twice the perception range. The GPA of the robot is then built by grouping together all the robots to which it can be connected through a chain of GPA couplings ( see Fig. 3, left). To achieve synchronization, the GPA is computed and updated until all its members are stationary. In particular, the robot exits from this synchronization phase when all the members of its GPA are stationary. Let  $T$  be the maximum time required to reach the target<sup>3</sup>, the upper bound of the waiting time is  $(N - 1)T$ , where  $N$  is the number of robots of the team.

The communication range  $R_c$  clearly plays a role in the GPA construction. Since the maximum distance between the robot and any other robot with which it is GPA-coupled is  $3R_p$  (the other robot may still be moving to its target, which however cannot be farther than  $R_p$  from the current configuration), it is sufficient to assume  $R_c \geq 3R_p$  to guarantee that the GPA accounts for all the robots that are candidate to belong to the GEA.

Once the robot is synchronized with its GPA, has perceived the LSR and updated its SRG, it builds the GEA, i.e., the robots with which cooperation and coordination are actually necessary. If we define two robots to be *GEA-coupled* when their LSRs overlap, the GEA of the robot (see Fig. 3, right) is composed by all the GPA robots to which it can be connected through a chain of GEA couplings. Synchronization guarantees that all the GPA robots are still when the GEA is computed. The GEA is *symmetric*, i.e., it is the same for all robots in the group.

The GEA is a cornerstone of our method, as it identifies a group of robots that, in view of their vicinity, spontaneously

<sup>3</sup> $T$  is clearly bounded. In fact, the target is at a distance at most  $R_p$ , since it is always contained in the current LSR.

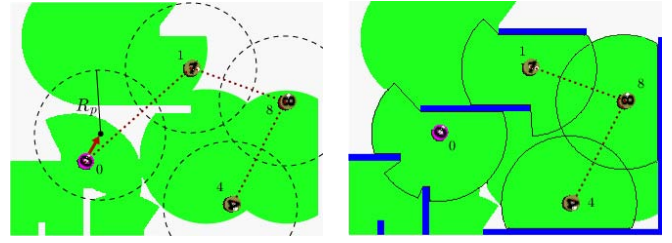


Fig. 3. An example of GPA/GEA construction *Left*: The GPA of robot 1 consists of robots 0,1,4,8: robot 0 is still moving towards its target point, while robots 1,4,8 are stationary. The perception areas of the robots (prospective in the case of robot 0) overlap in pairs. *Right*: Once the LSR have been computed, only robots 1,4,8 belong to the GEA of robot 1 since their LSRs overlap in pairs.

agree to cooperate and coordinate with each other on a temporary basis. A remarkable feature of the GEA is that it can be built with a limited communication range ( $R_c \geq 3R_p$ ).

#### B. SRG update

At each iteration of the algorithm the  $i$ -th robot updates its  $SRG_i$ . At first, new information received from other robots is added to the graph. In particular, when the  $i$ -th robot has communicated with  $j$ -th robot, the  $SRG_i$  and the  $SRG_j$  are merged. Obviously, this is done in an efficient way by communicating only parts of the graphs which have not been received yet. Then, additional structures, called *bridges*, are created to improve the connectivity of the roadmap. These essentially act as useful shortcuts which can be traveled by the robots.

A bridge is a triplet arc-node-arc. It is placed between a pair of nodes  $v$  and  $w$  only if and only if two conditions are satisfied: *i*) the graph distance between  $v$  and  $w$  is greater than a certain threshold; *ii*) the robot can move from  $v$  to  $w$  remaining in  $LSR(v) \cup LSR(w)$ . The first condition is aimed at improving the graph connectivity without significantly increasing its complexity

Clearly, this requires that  $v$  and  $w$  belong to the same connected component of the erosion of  $LSR(v) \cup LSR(w)$ . The middle node of the bridge is placed inside this region, and in particular so as to fall in the intersection  $LSR(v) \cap LSR(w)$ .

In particular, when  $v \in LRR(w)$  (and  $w \in LRR(v)$ ), the bridge degenerates in a single arc which represents the path from  $v$  to  $w$  entirely contained in  $LRR(v) \cap LRR(w)$ .

#### C. Local frontier extraction

Once the  $i$ -th robot has updated its  $SRG_i$ , it computes its current local frontier. This is the portion of the current LRR boundary leading to regions of the configuration space which appear unexplored according to the available information. To this end, the robot uses all the information stored in its memory and received through present and past communications.

To find promising configurations in the current LRR, its boundary is divided in *obstacle*, *free* and *frontier* arcs (see Fig. 4). Obstacle arcs correspond to configurations in which the robot touches a locally detected obstacle. Free arcs fall within the interior of the current Reachable Region of the

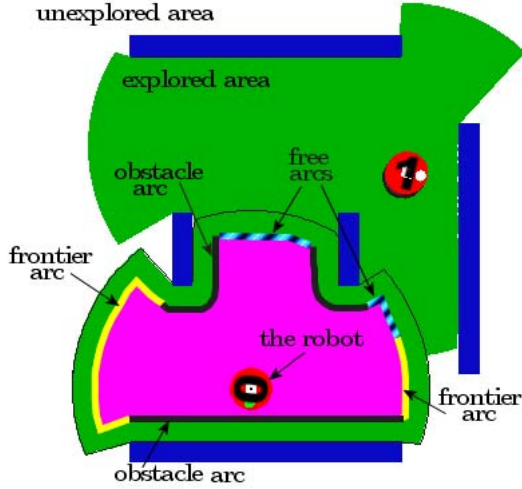


Fig. 4. Frontier, free and obstacle arcs in the Local Reachable Region LRR.

robot. Any arc of the current LRR boundary which is neither obstacle nor free is a frontier arc and, by construction, identifies the boundary between the current LRR and unexplored regions of the configuration space.

The above definition, according to which a frontier arc is computed from the  $SRG_i$ , which is built using all the information received from other robots, implements a simple decentralized cooperation mechanism aimed at optimizing the exploration performance of the team. Such mechanism is inherently local and contingent because it relies on communication between the robots.

#### D. Generating a new target configuration

If the local frontier of the current LRR is non-empty, a local planner generates a random configuration in the current LRR and directed towards the local frontier.

The local frontier is in general the union of disjoint frontier arcs  $f_1, f_2, \dots, f_m$ . At first, one of the local frontier arcs is selected with a probability proportional to its length. Let  $f$  be the selected frontier arc and assume it is represented by parametric equations in which the parameter  $s$  is the arc length. Let  $L$  be the length of the arc  $f$  so that  $s \in [0, L]$ . The target configuration is selected on  $f$  by generating  $s$  on the basis of a normal distribution with mean value  $L/2$  and standard deviation  $\sigma = L/6$ .

At this point, a collision-free path to the target can be planned. Note that, the controllability of the robot guarantees that any target configuration in the LRR can be reached by a feasible path which stays in the LRR.

Thanks to the synchronization phase performed by building and waiting the GPA of the robot, all the robots in its GEA plan at the same time, and therefore the cooperation mechanism intrinsic to the local frontier definition is enforced on all the GEA. This 'agreement of intents' is realized without any centralized decision module.

#### E. Forward trees and forward frontiers

When the current node has an empty local frontier, the robot has to move towards a promising region where unvisited frontiers still remain. To this end, it computes the forward frontiers of its adjacent nodes and then it selects as next destination the node with the greatest weighted forward frontier length.

In particular, let  $v$  be the node corresponding to the current configuration of the robot in its  $SRG_i$   $\mathcal{K}$ . Let  $\mathcal{K} \setminus v$  be the graph obtained from  $\mathcal{K}$  removing  $v$  and all its attached arcs. A forward tree and a forward frontier can be associated to each node  $v_a$  adjacent to  $v$ .

The *forward tree*  $\mathcal{N}(v_a, v)$  of an adjacent node  $v_a$  is a minimum spanning tree rooted in  $v_a$  and contained in  $\mathcal{K} \setminus v$ . In particular,  $\mathcal{N}(v_a, v)$  contains all the nodes of  $\mathcal{K}$  which are reachable from  $v_a$  without passing through  $v$ .

The *forward frontier* of  $\mathcal{N}(v_a, v)$  is the union of the local frontiers of all its nodes. The corresponding *weighted forward frontier length* is computed as follows:

$$F(v_a, v, \mathcal{N}(v_a, v)) = \sum_{w \in \mathcal{N}(v_a, v)} \frac{1}{\overline{vv_a} + l(v_a, w)} p(w)$$

where  $w$  denotes the generic node of the tree  $\mathcal{N}(v_a, v)$ ;  $\overline{vv_a}$  is the length of the arc between  $v$  and  $v_a$ ;  $l(v_a, w)$  is the length of the shortest path between  $w$  and  $v_a$  computed on  $\mathcal{N}(v_a, v)$ ;  $p(w)$  is the length of the local frontier of  $w$ .

Note that, in the previous computation, the weight  $p(w)$  of a node  $w$  is divided by the length of the shortest path which brings the robot from the current node  $v$  to  $w$  passing through  $v_a$ . In particular, for  $w = v_a$ ,  $l(v_a, w)$  is zero and  $p(v_a)$  is divided by  $\overline{vv_a}$ .

#### F. GEA path feasibility check

Although the local frontier of the robot cannot belong to the LSR of another robot of the GEA (see Fig. 4), the two prospective paths may still intersect. Figure 5 is an example of the conflict that may arise.

Hence, let  $\mathcal{G}$  be the subset of robots in the GEA. The prospective paths of the robots of  $\mathcal{G}$  are verified whether they are all simultaneously feasible<sup>4</sup> or not. To this end, all the pairs of paths that intersect with each other are identified, and the corresponding robots stored in the GEA *unfeasible subset*  $\mathcal{G}_u$ . The remaining robots are the GEA *feasible subset*  $\mathcal{G}_f$ . The complexity of this check is  $O(|\mathcal{G}|^2)$ .

#### G. Coordination

If the subset  $\mathcal{G}_u$  of robots with unfeasible paths is non-empty, a coordination phase takes place locally. At first, a *master robot* within  $\mathcal{G}$  is elected. This can be accomplished in many ways through a deterministic procedure known by all the robots; for instance, the robot with the higher ID number can be chosen. Two cases are then possible:

1) If the robot is the master, it builds the vector of targets  $\mathcal{Q}_{\mathcal{G}}$  which collects the target configurations received from the

<sup>4</sup>Collision check is performed in configuration space, not taking into account the possibility of velocity scaling along the paths to avoid collisions. Paths that intersect are not allowed in our approach.

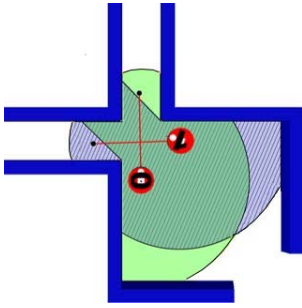


Fig. 5. The prospective paths of robots belonging to the GEA may intersect.

GEA robots. Then, it rearranges this vector so as to obtain a feasible collective motion. Here, rearrange may mean either simply accepting/resetting the target of a robot to the current configuration (i.e., authorizing/forbidding the move) or adding a third option, i.e., changing it to a new target. Correspondingly, we have devised two strategies, i.e., *organization via arbitration* and *organization via replanning* (see below).

2) If the robot is not the master, it waits for until the receipt of a specific signal from the master.

The final operation is to retrieve and return the robot's (possibly modified) own target from  $\mathcal{Q}_G$ .

1) *Organization via arbitration*: This strategy implements a simple arbitration mechanism on  $\mathcal{G}$ . All the robots contained in the feasible subset  $\mathcal{G}_f$  are allowed to move (their target configuration is left unchanged). The robots in the unfeasible subset  $\mathcal{G}_u$  are not allowed to move (their target is reset to the current configuration) with the exception of a single one whose motion is authorized (by construction, this strategy is guaranteed not to produce conflicts).

The selection of the authorized robot in  $\mathcal{G}_u$  may be done on the basis of various criteria. The one we have used chooses randomly one of the robots (if any) whose local frontier is empty: these are robots whose target is either their parent node or one of their children nodes. This strategy is motivated by the fact that, if their move is not authorized, such robots will have to wait for their path to become clear, as they cannot change their target (as opposed to robots whose local frontier is non-empty, to which the random planner may propose a different target in the next cycle).

An antithetical criterion would be to choose randomly among the robots in  $\mathcal{G}_u$ , using a probability proportional to the extension of their local frontier.

2) *Organization via replanning*: This strategy tries to modify the targets of the robots in  $\mathcal{G}$  so as to maximize the number of simultaneous feasible moves. This may be done by formalizing the problem as follows.

Consider the set of targets  $\mathcal{Q}_G$  associated to the GEA  $\mathcal{G}$ . Two targets in  $\mathcal{Q}_G$  are called *compatible* if they can be reached by the corresponding robots with paths that do not intersect. Let  $G$  be the *compatibility graph* associated to  $(\mathcal{G}, \mathcal{Q}_G)$  and defined as the indirect graph whose nodes represent the robots in  $\mathcal{G}$  and whose arcs join pairs of nodes with compatible targets. A *maximum clique* of  $G$  is a complete subgraph of

$G$  with maximum cardinality, corresponding to a maximum subset of robots with compatible targets. The identification of a *maximum clique* is a well-known NP-complete problem in the context of the graph theory [19], [20].

The ideal objective of the replanning strategy is to modify the set of targets  $\mathcal{Q}_G$  so as to maximize the cardinality of the associated maximum clique(s), with the constraint that the target of each robot is either accepted, changed to another configuration towards the local frontier of the robot (if this is non-empty) or to the current robot configuration (the move is not authorized). This is a very complex problem whose solution would require the computation of maximum cliques as a subproblem. To find a satisfactory solution in a given amount of time, we have adopted a randomized search technique, performed by the master as a sequential game with complete information. A description of the game is given in [18].

## V. SIMULATIONS

In this section we present some comparative simulations to evaluate the performance of the SRG-based method with respect to the SRT-based method in [12]. The simulations are performed in Move3D [21], a software platform developed at LAAS-CNRS and dedicated to motion planning<sup>5</sup>. The exploration team is composed of a varying number of MagellanPro robots carrying a 360° laser range finder, with a perception range of 4 times the robot diameter.

The performance of the methods are evaluated in terms of *exploration time* (the time required by the last robot of the team to return home) and *traveled distance* (the average the distance traveled by each robot). These values are respectively expressed as a percentage of the values obtained with a team composed by a single robot using an SRT-based method. Environment coverage is not reported because it was complete in all our simulations. In view of the randomized nature of our method, numerical results for each scenario are averaged over ten simulation runs.

Simulations are performed in a garden-like environment. The communication range is limited to  $R_c = 3R_p$ . Two nodes are candidate to be connected by a bridge if their graph distance is at least 3 arcs. Coordination is achieved through organization via replanning. We consider two possible initial deployments of the team. In the first, the robots are initially scattered in the environment (as if they had been parachuted). In the second, more realistic for environments with a single main entrance, the exploration is started with the robots grouped in a cluster.

Figure 6 shows the exploration progress of SRG-based exploration with a team of 8 robots and a scattered initial deployment. Green areas represent the region so far explored. The robots are represented by red circles with the ID number. The perception locations are marked by black points. Yellow segments represent paths traveled by the robots during exploration. Bridges are depicted in blue and may or not have

<sup>5</sup>Move3D is at the origin of the product KineoWorks currently marketed by the company Kineo CAM ([www.kineocam.com](http://www.kineocam.com)).

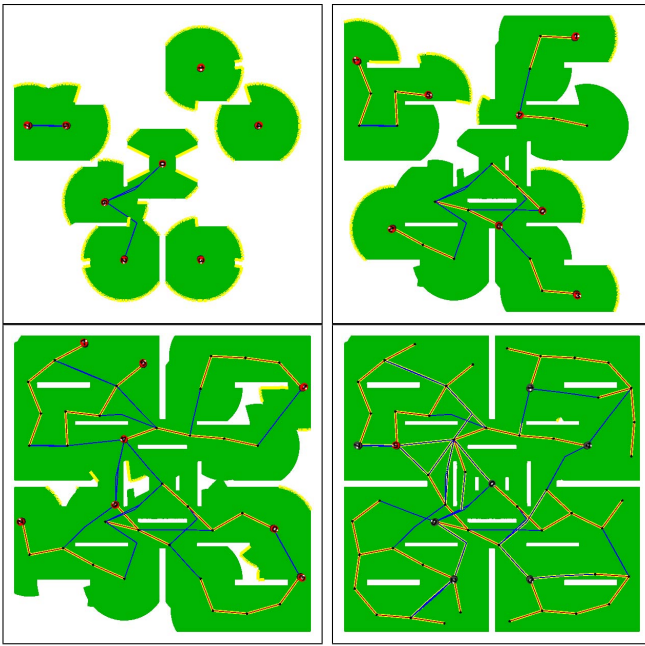


Fig. 6. SRG-based garden exploration with scattered start.

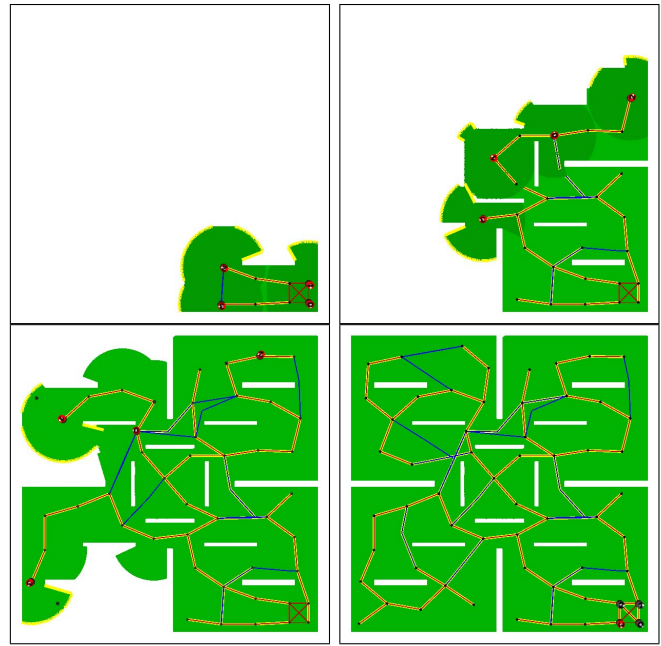


Fig. 8. SRG-based garden exploration with clustered start.

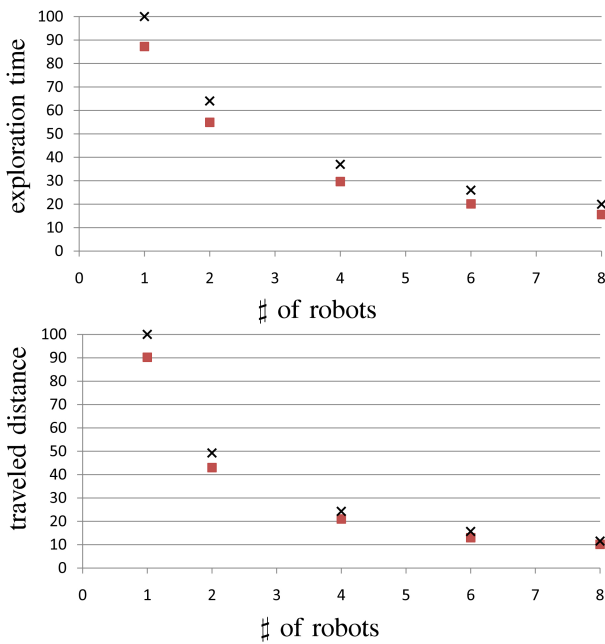


Fig. 7. SRG-based garden exploration with scattered start: exploration time (above) and traveled distance (below) with teams of different cardinality. Results for SRG-based (squares) and SRT-based (crosses) explorations are shown.

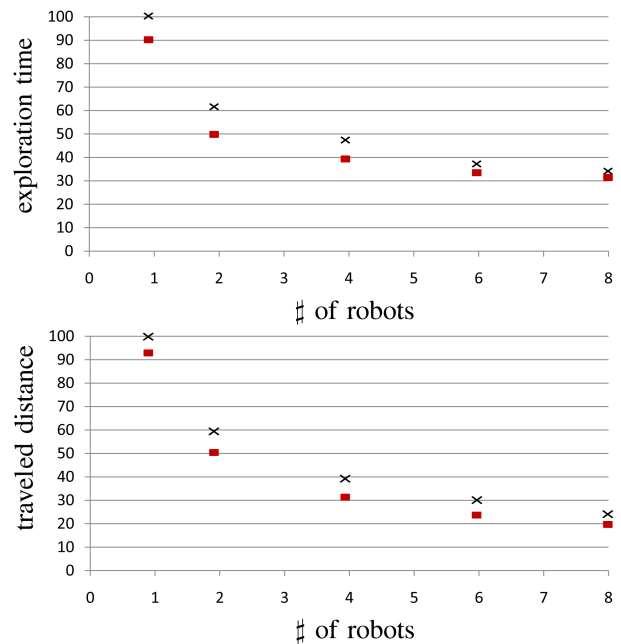


Fig. 9. SRG-based garden exploration with clustered start: exploration time (above) and traveled distance (below) with teams of different cardinality. Results for SRG-based (squares) and SRT-based (crosses) explorations are shown.

been traversed by the robots. Exploration time and traveled distance for teams of different cardinality are shown in Fig. 7. Average results are shown for both the SRG-based (squares) and the SRT-based (crosses) method. As the number of robots in the team increases, the exploration time quickly decreases and tends asymptotically to zero (consider that an increment in the number of evenly deployed robots corresponds to a decrement of the individual areas they must cover, until no

motion at all is necessary). A similar behavior is observed for the traveled distance. The performances of the two methods for a scattered start are similar as the number of robots increases. This is due to the scattered initial deployment, which leads each robot to perform most of the exploration ‘on its own’. As a result, the bridges present in SRG are rarely traversed and the performance of the SRG-based method does not improve significantly over the SRT-based method.

Figure 8 shows a typical exploration progress of SRG-based exploration with a team of 4 robots and a clustered initial deployment. Figure 9 summarizes the performance of both SRG and SRT-based methods for teams of different cardinality. In this case the exploration time asymptotically tends to a nonzero value, which approximately represents the time required by a single robot to perform a roundtrip between the cluster center and the farthest point in the environment. Instead the average distance traveled by each robot still tends to zero. The results show that, on the average, the SRG-based method introduces a significant improvement in both the exploration time and the traveled distance.

Additional simulations, including movies, are available at <http://www.dis.uniroma1.it/~labrob/research/multiSRG.html>.

## VI. CONCLUSIONS

In this paper we have presented a randomized strategy for cooperative robot exploration which is an evolution of the method introduced in [12]. While in that work the roadmap of the environment is built in the form of a forest of disjoint trees, in the present work a single graph, called Sensor-based Random Graph (SRG), is expanded by the robots. This compact data structure is incrementally enhanced by adding bridges in order to improve its connectivity.

A simple and very efficient decentralized cooperation mechanism is inherent in the method. This simply consists in an appropriate definition of the local frontier, by which each robot plans its motion towards areas that appear to be unexplored by the rest of the team on the basis of the available information. Local coordination guarantees that the collective motion of the team is feasible from the collision viewpoint. Simulation results have shown the satisfactory performance of the method, even in the case of limited communication range.

In our view, the main feature of the proposed method is that no central supervision is needed, and no task decomposition/allocation is performed. The selection of exploration actions by each robot is spontaneous and made on the basis of the available information. This is expected to provide robustness with respect to possible robot breakdowns as well as communication failures.

Future work will address several aspects, i.e., the integration of a localization module along the lines of [16], the extension to heterogeneous robots, and an experimental validation so as to test the robustness of the proposed method with respect to sensors and communication failures. Also, we intend to explore the possible relevance of the small-world effect to our work, especially to the speed of cooperative search [22].

## REFERENCES

- [1] Y. Cao, A. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.
- [2] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, pp. 375–397, 1996.
- [3] I. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 7–40, 2001.
- [4] D. Goldberg and M. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," in *14th AAAI/9th IAAI*, 1997, pp. 637–642.
- [5] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994.
- [6] B. Yamauchi, "Frontier-based exploration using multiple robots," in *2nd Int. Conf. on Autonomous Agents*, 1998, pp. 47–53.
- [7] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. on Robotics and Automation*, vol. 1, no. 3, pp. 376–386, 2005.
- [8] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 3232–3238.
- [9] A. Howard, M. Mataric, and S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 2849–2854.
- [10] R. Zlot, A. Stenz, M. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *2002 IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 3016–3023.
- [11] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *17th AAAI/12th IAAI*, 2000, pp. 852–858.
- [12] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "A randomized strategy for cooperative robot exploration," *2007 IEEE Int. Conf. on Robotics and Automation*, pp. 768–774, 2007.
- [13] G. Oriolo, M. Vendittelli, L. Freda, and L. Trosso, "The SRT method: Randomized strategies for exploration," in *2004 IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 4688–4694.
- [14] L. Freda and G. Oriolo, "Frontier-based probabilistic strategies for sensor-based exploration," in *2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3892–3898.
- [15] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer Academic Publishers, 1991.
- [16] L. Freda, F. Loiudice, and G. Oriolo, "A randomized method for integrated exploration," *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.
- [17] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [18] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "A randomized strategy for cooperative robot exploration," DIS, Università di Roma "La Sapienza", Tech. Rep., 2006.
- [19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman, 1983.
- [20] F. Harary, *Graph Theory*. Reading, MA: Addison-Wesley, 1994.
- [21] T. Simeon, J.-P. Laumond, and F. Lamiroux, "Move3d: A generic platform for path planning," in *4th Int. Symp. on Assembly and Task Planning*, 2001, pp. 25–30.
- [22] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, 2007.