Multirobot Persistent Patrolling in Communication-Restricted Environments

Marta Romeo, Jacopo Banfi, Nicola Basilico, and Francesco Amigoni

Abstract In multirobot patrolling, a team of robots is deployed in an environment with the aim of keeping under observation a set of locations of interest. In several realistic mission scenarios, only human operators sitting at a base station are able to assess the situation on the basis of data sent by robots. Examples include watching pictures or video streams to detect intruders and correlating measurements to detect leaks of contaminants. We assume that a communication infrastructure is available only in some regions of the environments, from where messages can be exchanged with a sufficient bandwidth between the robots and the base station. In this paper, we first extend the classical multirobot persistent patrolling framework and the related idleness evaluation metrics to such environments with a limited number of "communication zones". Then, we present some centralized and distributed patrolling strategies tailored for this communication-restricted framework. Finally, we evaluate their performance using ROS/Stage simulation.

1 Introduction

In the last few years, a lot of effort has been devoted in the autonomous robotics community to the study of effective patrolling strategies in different problem settings and under different assumptions [1–4, 9, 10, 13, 15]. In the basic formulation of the *multirobot persistent patrolling problem*, a team of robots is deployed in an environment represented as an undirected weighted graph, with the aim of minimizing the *idleness* of each location (vertex) to be patrolled, defined as the time elapsed since that location has received the visit of a robot [10]. Evaluation metrics

Marta Romeo, Jacopo Banfi, and Francesco Amigoni
Politecnico di Milano, Milano, Italy,
e-mail: marta.romeo@mail.polimi.it,{jacopo.banfi,francesco.amigoni}@polimi.it

Nicola Basilico
University of Milan, Milano, Italy, e-mail: nicola.basilico@unimi.it

related to this notion, such as average and worst-case idleness, are introduced and investigated for different patrolling strategies and in a number of different conditions [4, 9, 10, 13, 15].

Some realistic mission scenarios challenge an (often implicit) assumption made in the above formulation, namely that a single robot is able to assess the situation of the visited vertices solely on the basis of the data it collected. In these cases, data must be reported to a base station, acting as a mission central room, where human operators can assess the situation and take appropriate countermeasures. For instance, think about the presence of an intruder or about a leak of contaminants, when the robots might not have the physical capabilities required to detect the problem and intervene to solve it, because operators need to watch pictures or video streams and to correlate measurements. In this paper, we assume that communication between the robots and the base station can only use existing communication infrastructures, such as tactical networks or cellular networks, that are characterized by the fact that only some regions of the environment are covered by a sufficiently strong signal. The presence of such "communication zones" in the environment to patrol was recently investigated in [1], where a centralized inspection tour for a team of robots is calculated with the aim of minimizing the time lag between the visit of a location and its report to the base station.

In this paper, we propose some strategies for multirobot persistent patrolling in communication-restricted environments characterized by the presence of a limited number of communication zones. More precisely, we extend the classical idleness evaluation criteria for the communication-restricted patrolling framework introduced in [1]. Then, we present some patrolling strategies for such a framework. One of these strategies is centralized and optimal, while the other two are distributed. Although non optimal in general, these last strategies are less computationally demanding and are particularly suited for coping with unpredictable events (e.g., temporary unavailability of communication links). The proposed patrolling strategies are implemented within a ROS-based architecture, and ROS/Stage simulations are run to analyze and compare them in a number of different settings. The results show that a centralized planning scheme is not actually needed to obtain good performance and that both the proposed distributed strategies perform well.

2 Related work

The study of multirobot patrolling strategies is nowadays a rather lively sub-field of autonomous robotics, especially from the theoretical point of view (although some practical systems have been proposed [9]). Approaches can be broadly divided in two categories: those considering an adversarial and those considering a non-adversarial setting. In the former category, patrolling strategies are developed considering a specific model of rational intruder [2, 3], while, in the latter category, the patrolling strategies are in general agnostic about the particular events to be monitored, preferring to maximize the frequency of visit of some locations that

need to be kept under constant observation [4, 6, 9, 10, 13, 15] (hence the name "persistent patrolling"). In this paper, we concentrate on the last category of works. To the best of our knowledge, there are few works in the literature considering communication constraints in robotic patrolling and in the more general context of robotic information-gathering missions. For example, in [11], the authors consider a generic multirobot routing task, in which robots are required to maintain, by means of local and multi-hop transmissions, a global communication infrastructure while visiting some target areas at minimum traveling cost. More recently, [16] considers the same general setting, but with the additional requirement of planning robots' paths while maintaining a given minimum end-to-end data rate between robots.

Another problem setting is investigated in [7], where robots plan informative paths under the constraint of regaining multi-hop connectivity after a fixed number of time steps. The assumption on communication infrastructure we make in this paper is rather different from that adopted in these works, because we do not rely on a multi-hop network to deliver data to a base station, but on an existing infrastructure that is available only in some areas. More precisely, motivated by recent military and civil case-studies [12, 17], in [1] we have introduced a patrolling framework modeling the presence of an existing communication infrastructure (for example, a tactical network, a wireless network, or a cellular network) that robots could exploit to reliably communicate with a base station. (This setting is similar to that investigated in [5], but in the context of static target search.)

In this paper, we investigate further along this line through the adoption of the same communication model. However, instead of considering a patrolling mission consisting of a *single* inspection tour computed offline (as in [1]), we focus on *persistent* patrolling, in which vertices need to be visited multiple times in order to keep them under constant observation.

3 Problem definition

A team $R = \{1, \ldots, m\}$ of robots must persistently patrol some locations of a given environment while sending reports to, and exploiting information provided by, a base station (BS). Robots can communicate with the BS by exchanging messages through a pre-existing communication infrastructure, which is only accessible from some regions of the environment. To formalize this setting, we represent the environment by means of an undirected graph $G = (V, E)$ in which vertices represent all the locations robots can visit, while edges encode physical connections between them (without considering self-loops). Each edge $(v_i, v_j) \in E$ is associated with a positive number $d(v_i, v_j)$ representing the physical distance between v_i and v_j in the environment. We assume that our graph represents a physical environment. For this reason, we impose that the distance $d(v_i, v_j)$ associated to any physical edge $(v_i, v_j) \in E$ is not greater than that of any other (v_i, v_j)-path in G. Vertices V are used to represent both locations to patrol and communication zones. Accordingly, following the notation introduced in [1], each vertex in V is preassigned to up to two

types: m, if the vertex needs to be persistently monitored (patrolled), and c, if the vertex allows to exchange messages with the BS. We denote with $V_m, V_c \subseteq V$ the two subsets of m-type and c-type vertices, respectively. (Notice that V_m and V_c may share a non-empty intersection.) Robots move on the graph G by traversing the physical edges E: once they reach a m-type vertex, they accomplish a (instantaneous) patrolling visit at it; once they reach a c-type vertex, they are allowed to communicate with the BS. At the moment, we do not assume any limit on the communication from a c-type vertex to the BS. In Section 5, we consider errors in the transmission of messages.

In order to define the objective of the patrolling, we now generalize the classical evaluation metrics related to the notion of *idleness* [4] to our communication-restricted framework. The persistent patrolling task unfolds across an arbitrary long time horizon T in which time evolves in discrete steps $\{1,\dots,T\}$. The mission undertaken by a robot r is represented by a *walk* on the graph G of the form $w_r = (v_0, v_1, \dots, v_k)$, where v_0 denotes the robot's starting vertex, v_k denotes the last vertex visited before T, and a generic v_i denotes the i-th vertex visited by the robot. (Notice that, being w_r a walk on G, the same vertex can appear multiple times in w_r.) For each walk w_r, we define a function $t_r : \{1,\dots,k\} \to \{1,\dots,T\}$ mapping the i-th visited vertex by robot r to the time step in which the visit takes place. In particular, each $t_r(i)$ is determined by the total distance traveled until the i-th vertex visited in w_r ($\sum_{j=0}^{i-1} d(v_j, v_{j+1})$) and by robot r's (possibly varying) speed along that portion of the walk. Now, for any $i \in \{1,\dots,k\}$ such that $v_i \in V_m$, let $v_{\bar{c}}, \bar{c} \geq i$ be the first vertex after the i-th vertex visited s.t. $v_{\bar{c}} \in V_c$, and call $c(i) = \bar{c}$. Following this notation, we define robot r's *reporting time* of visit i to the BS (only for vertices v_i s.t. $v_i \in V_m$) as $t_r(c(i))$, assuming that this quantity will be infinite if the visit is never reported before the patrolling task ends at T. (Notice that $t_r(c(i)) = t_r(i)$ iff $v_i \in V_c \cap V_m$.)

Contrarily to many other patrolling settings, it may be possible in our framework that the BS receives reports containing outdated information. This happens when there exist two robots $r, r' \in R$ (not necessarily different), two walks $w_r = (v_0, v_1, \dots, v_k), w_{r'} = (v_0, v_1, \dots, v_{k'})$, and two visits $i \in \{1,\dots,k\}, j \in \{1,\dots,k'\}$ s.t. $v_i = v_j$, $t_r(i) < t_{r'}(j)$, and $t_r(c(i)) \geq t_{r'}(c(j))$. In such a case, we say that the i-th visit of robot r is *outdated*, in the sense that the corresponding report to the BS will contain outdated information. We are now ready for introducing metrics related to *communication idleness*.

We define the *instantaneous communication idleness* of vertex $v \in V_m$ at time $\bar{t} \in \{1,\dots,T\}$ as:

$$IC_v(\bar{t}) = \bar{t} - \bar{t'}, \tag{1}$$

where $\bar{t'} \leq \bar{t}$ denotes the timestep of the last not outdated report of vertex v to the BS not after timestep $\bar{t}$. We assume that the initial situation of all the vertices to be monitored is known, and hence that $IC_v(\bar{t}) = \bar{t}$ if there exists no visit to v that has been communicated (by any robot) until time $\bar{t}$. Notice that, according to our definition, in case $V_m = V_c = V$ no visit is considered outdated, and the notion of instantaneous

communication idleness collapses into the notion of idleness classically adopted in multirobot persistent patrolling (see, e.g., [4]).

The *instantaneous graph communication idleness* at time $\bar{t}$ is defined as the average instantaneous communication idleness among the vertices to be patrolled:

$$IC_G(\bar{t}) = \frac{\sum_{v \in V_m} IC_v(\bar{t})}{|V_m|}. \tag{2}$$

In this paper, we are mainly interested in studying patrolling strategies allowing to minimize the *average graph communication idleness* at the end of the mission, formally defined as:

$$\overline{IC}_G = \frac{\sum_{\bar{t} \in \{1,\ldots,T\}} IC_G(\bar{t})}{T}. \tag{3}$$

Another interesting performance indicator we will investigate is the *worst-case graph communication idless* at time T, WIC_G, which can be informally defined as the largest instantaneous vertex communication idleness encountered through the patrolling task. Average and worst-case idleness for the single vertices can be defined in a similar way.

4 Communication-aware patrolling strategies

In this section, we present three different multirobot patrolling strategies (plus a random baseline one) suitable for the patrolling setting introduced above, aimed at minimizing the average graph communication idleness $\overline{IC}_G$. First, we present a strategy in which the BS performs the planning phase by computing centralized optimal visit plans for all the robots. Then, we present two distributed strategies in which robots query the BS about the current situation of the environment in terms of vertices idlenesses and teammates current commitments (plans) and then calculate their plans.

4.1 Globally optimal strategy

The *globally optimal strategy* (G-OPT) is an optimal, centralized, and offline strategy working as follows. At the beginning of the patrolling mission, the BS computes (using brute force) the best set of joint robot walks $\mathbf{w}^* = (w_1^*, w_2^*, \ldots, w_m^*)$ spanning the whole task horizon T and minimizing $\overline{IC}_G$; then, it communicates the plans to the robots (that are assumed to be initially placed in possibly different starting vertices belonging to V_c), which can then start to execute them. The pseudo-code of the planning algorithm of G-OPT is shown in Algorithm 1.

About the computational complexity of such a planning scheme, notice that the evaluation of $\overline{IC}_G$ for a candidate set $\mathbf{w}$ of joint robots' walks requires pseudo-

polynomial time $O(mT)$ in general, while the number of joint walks of length not exceeding T on G grows exponentially with the size of the input: therefore, this strategy is suitable only for small problem settings. Moreover, it is evident that the walks obtained by adopting this planning approach will be actually optimal w.r.t. the minimization of $\overline{IC}_G$ (if the errors in the predictions of their execution can be assumed to be negligible).

Algorithm 1 G-OPT planning algorithm

```
W ← {enumerate all joint walks of m robots on G with maximum length T}
for all w ∈ W do
    simulate the concurrent execution of w and compute the corresponding IC̄_G
end for
w* ← {the best w obtained that minimizes IC̄_G}
return w*
```

We remark that an optimal strategy could also be obtained by formalizing the problem as a Mixed Integer Linear Program, in the spirit of [1]. However, with respect to [1], such an approach would require the additional modelling of the possibility of multiple passages through the same vertices at any time step along the whole mission horizon, making the MILP overly-complicated.

4.2 Individually optimal strategy

The *individually optimal strategy* (I-OPT) can be thought as the distributed and online version of G-OPT. When adopting this strategy, robots iteratively compute new portions of their own walks as soon as they reach a new c-type vertex, querying the BS about (a) the current m-type vertices instantaneous communication idlenesses and (b) their teammates intentions, expressed in terms of portions of the walks they are currently following. More specifically, let $\bar{t}$ be a generic timestep in which a robot r reaches a c-type vertex, and let $\hat{w}_1, \ldots, \hat{w}_{r-1}, \hat{w}_{r+1}, \ldots, \hat{w}_m$ the portions of walks its teammates are going to follow from step $\bar{t}$. (Notice that such portions of walks may be outdated, as they depend on the previous arrivals of the robots to c-type vertices.) Robot r computes a new portion of its walk $\hat{w}_r$ on G of estimated travel time H (the planning horizon) as the walk minimizing the average graph communication idleness between $\bar{t}$ and $\bar{t}+H$, while taking into account the walks followed by its teammates (if a teammate has communicated a plan that ends before $\bar{t}+H$, it is assumed to remain still at the last vertex of the communicated plan). If H is sufficiently small, a complete evaluation of all the possible walks of length H can be completed in reasonable time; otherwise, a sampling strategy needs to be adopted. The pseudo-code of this planning scheme is reported in Algorithm 2.

Algorithm 2 I-OPT planning algorithm for robot r at time $\bar{t}$

$\hat{w}_1,\ldots,\hat{w}_{r-1},\hat{w}_{r+1},\ldots,\hat{w}_m \leftarrow$ {query the BS about the walks followed by other robots}
query the BS about instantaneous communication idlenesses of vertices V_m
$W_r \leftarrow$ {enumerate/sample walks from r's current position along $[\bar{t},\bar{t}+H]$}
for all $w_r \in W_r$ **do**
  $\hat{\mathbf{w}} \leftarrow (\hat{w}_1,\ldots,\hat{w}_{r-1},w_r,\hat{w}_{r+1},\ldots,\hat{w}_m)$
  simulate the execution of $\hat{\mathbf{w}}$ and compute the corresponding $\overline{IC}_G$ in $[\bar{t},\bar{t}+H]$
end for
$\hat{w}_r \leftarrow$ {the best w_r obtained that minimizes $\overline{IC}_G$ in $[\bar{t},\bar{t}+H]$}
return $\hat{w}_r$

4.3 A simple reactive strategy

The *reactive strategy* (RE) is still distributed, but does not involve long-term planning up to horizon H as the previous one; instead, it is inspired by the Cognitive Coordinated strategy studied in [4,10]. (This strategy empirically showed good performance when considering the classical notion of idleness.) By adopting this strategy, robots iteratively choose a new m-type vertex $\hat{v}$ to reach according to a heuristic in which vertices with the highest instantaneous communication latency have higher priority, and, in case the chosen vertex does not belong also to V_c, subsequently move to the closest c-type vertex in order to communicate back data relative to $\hat{v}$ and to obtain from the BS the data needed for computing a new plan.

More specifically, when a robot r arrives at the c-type vertex selected for communicating the data relative to the previous $\hat{v}' \in V_m$ at a generic time step $\bar{t}$, it queries the BS about (a) the current m-type vertices instantaneous communication idlenesses and (b) their teammates intentions, expressed in terms of portions of walks currently being followed (as in the previous strategy). It then decides to reach the m-type vertex $\hat{v}$ displaying the highest communication idleness that is currently not in the portion of walk followed by any other robot (if each vertex will be visited by at least one robot, choose one vertex randomly). The path from the current vertex of robot r to $\hat{v}$ is computed on G, so that, as a side-effect of the choice, additional m-type and c-type vertices can be visited (in the latter case, the data relative to the visited m-type vertices are sent to the BS). In case $\hat{v} \notin V_c$, the path is augmented with the shortest path connecting $\hat{v}$ to the closest c-type vertex. The pseudo-code each robot runs for implementing this strategy is shown in Algorithm 3. Clearly, the algorithm runs in linear time w.r.t. the number of graph vertices if the shortest paths composing $\hat{w}_r$ are already available (notice that they can be pre-computed at the beginning of the mission).

In the next section, we will investigate the behavior of these strategies by also comparing them against a baseline *random strategy* (RANDOM) in which each robot chooses randomly a new vertex of G to reach, regardless of its current idleness and membership to V_m and/or V_c.

Algorithm 3 RE planning algorithm for robot r at time $\bar{t}$

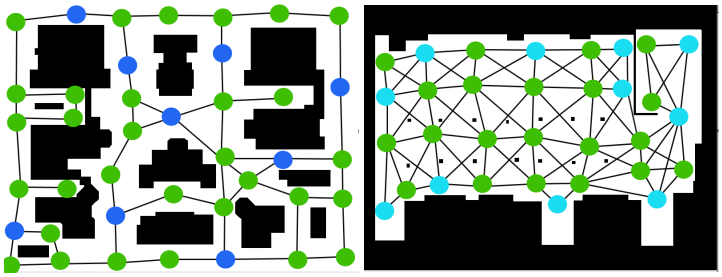
$\hat{w}_1, \ldots, \hat{w}_{r-1}, \hat{w}_{r+1}, \ldots, \hat{w}_m \leftarrow$ {query the BS about the walks followed by others}
query the BS about instantaneous communication idlenesses of vertices V_m
$\hat{v} \leftarrow \arg\max_{v \in V_m, v \notin \hat{w}_i, i \in R \setminus \{r\}} IC_v(\bar{t})$
compute $\hat{w}_r$ as the shortest path on G from r's current position to $\hat{v}$
if $\hat{v} \notin V_c$ **then**
  let $\hat{v}_c$ be the c-type vertex closest to $\hat{v}$
  augment $\hat{w}_r$ with the shortest path on G from $\hat{v}$ to $\hat{v}_c$
end if
return $\hat{w}_r$

5 Experimental evaluation

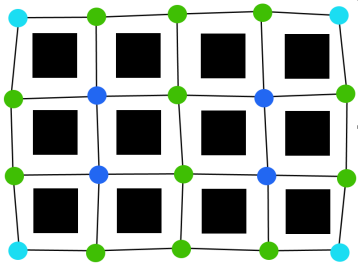
We validated the proposed patrolling strategies in simulated, yet realistic, scenarios within the ROS/Stage framework [14, 18]. To enrich the Stage simulator with regions providing a communication link with a BS, we implemented an additional ROS node called *Communication Server* (CS) in charge of handling the communication between robots and BS. The CS node receives all the messages from the robots (BS) and appropriately forwards them to the BS (robots), considering robots locations in the environment. We selected three representative environments, represented as 800x600 pixels bitmap images, upon which we constructed the patrolling graph $G = (V, E)$ by randomly selecting vertices sufficiently far from the obstacles, manually pruning some of them, and by adding edges only between vertices in line-of-sight to obtain a reasonable topological representation of the environment. In all the environments, communication vertices are chosen manually. The three selected environments are (see Fig. 1):

- the Leonardo campus of the Politecnico di Milano (Poli – approx. size 200 m $\times$ 150 m), characterized by the presence of several buildings and discretized in 40 m-type vertices, 9 of which also belong to V_c;
- the "acapulco_convention_center" of the Radish dataset repository [8] (Open – approx. size 80 m $\times$ 60 m), characterized by a large hall and discretized in 32 vertices, 11 of which only belong to V_c and the remaining 21 only belong to V_m;
- an imaginary grid-block environment similar to those used in [4, 13] (Grid), discretized in 20 vertices, 4 of which only belong to V_c, 12 only to V_m, and the remaining 4 to both V_m and V_c.

Each simulated robot is equipped with a controller allowing to choose between one of the proposed patrolling strategies, and it is assumed to be able to perfectly localize itself in the environment (by using its true position as given by Stage). Given a waypoint to reach, path planning is performed by means of the A* algorithm run on a grid-based discretization of the environment, followed by a simple path-smoothing phase. Robots move at a maximum speed of 10 px/s, avoiding each other by means of a simple behavioral rule according to which the robot with highest ID is given precedence to pass.



(a) Poli (b) Open



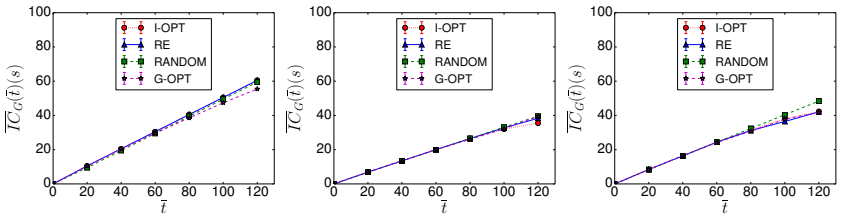
(c) Grid

Fig. 1: The three environments. Green, blue, and light-blue vertices define the sets $V_m \setminus V_c$, $V_m \cap V_c$, and $V_c \setminus V_m$, respectively.

Each run is repeated 5 times in order to cope with the non-deterministic outcome of the experiments due to the message exchange protocol adopted by ROS (graphs report averages over the runs and corresponding standard deviation bars). For each team size, we initially place the robots in the corners of the environments, and we keep fixed their initial positions across the different runs. The planning horizon of I-OPT is kept fixed at $H = 150$ seconds, except where indicated otherwise, as this value allows to perform a complete walk enumeration in reasonable time in all the environments. (Notice that this is possible since robots' movements along the edges consume a significant amount of time steps.) All the simulations are performed on a laptop equipped with an Intel P7450 processor and 2 GB of RAM.

We start by reporting in Fig. 2 the average idleness values obtained in the three environments by the four proposed patrolling strategies (G-OPT, I-OPT, RE, and RANDOM) through short mission durations T with 2 robots. Such durations are chosen as the ones in which G-OPT is able to calculate a complete plan in the same amount of time required by I-OPT (a few seconds); notice that the durations are not equal for the different environments, as they depend on the complexity of the planning graphs G (which follows from the topology of the environments, as well as from the selected discretization method). All the performance curves remain very close until the end of the mission, where the random baseline starts to behave slightly worse than all the other strategies in the Poli and Grid environments. These results show that I-OPT and RE are able to perform as well as G-OPT on short mis-

sion horizons and justify their employment on longer mission durations for which G-OPT is not a viable choice. Notice that, in the Open environment, I-OPT behaves slightly better than G-OPT towards the end. This is due to several factors, such as robots interfering with each other while executing their paths, and a non-perfect simulation of the concurrent walks execution during the planning phase.



(a) Poli (b) Open (c) Grid

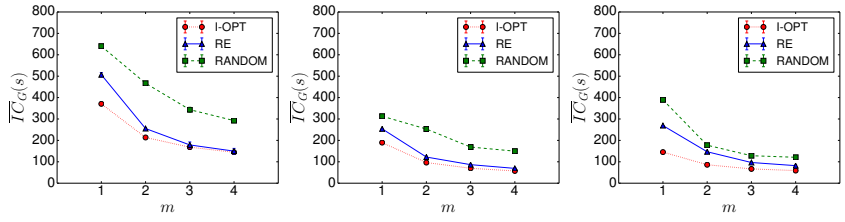
Fig. 2: Comparison of G-OPT, I-OPT, RE, and RANDOM in short mission durations with 2 robots.

We now focus on $T = 30$ minutes patrolling missions (the average idleness values tend to stabilize after this amount of time), and examine the impact of different team sizes on the average and worst-case idleness values obtained by I-OPT, RE, and RANDOM. Fig. 3 shows the average idleness values obtained in the three environments by the three strategies for a number of robots m varying from 1 to 4. In all the environments and for all the values of m, RE outperforms RANDOM and it is in turn outperformed by I-OPT. These results show that the increased planning complexity of the three strategies is immediately reflected in their effectiveness. Increasing the value of m leads to an advantage for all the strategies, but the performance gain becomes smaller. (For larger graphs with similar structures, we argue that the performance should follow the same decreasing trend for a fixed ratio $|V|/m$.) If we now look at the worst-case idleness values (Fig. 4), we can see that I-OPT is outperformed by RE in Poli and Open. This is not surprising, as RE is designed to always lead the robots towards the vertex currently displaying the highest idleness, while I-OPT plans without considering worst-case idleness values. Again, larger team sizes lead to lower worst-case idleness values in all the environments and for all the strategies, with the only exception of RANDOM in Open, where there is always at least one vertex whose situation is never reported to the BS.

To conclude, we report in Fig. 5 the results obtained with different parameters on the Poli environment for 3 robots. Fig. 5(a) shows the average idleness values obtained for different planning horizons H in I-OPT. As expected, a planning horizon too short or too long leads to worse performance than a balanced one: in the former case, the plan is obtained quickly, but it is not effective; in the latter case, the longer time spent during planning is not compensated by its effectiveness. A way to select a good value for H is to start from a value that allows to reach every node from any node (closely related to the diameter of G) and, then, decrease it if com-

puting strategies takes too long. Fig. 5(b) investigates the impact of varying the ratio $|V_c|/|V|$, while keeping each vertex also in V_m. For $|V_c|/|V| = 0.05$ I-OPT behaves as badly as RANDOM. The reason is that, in this case, there are only two c-type vertices, placed at two opposite corners of the map: therefore, a planning horizon of $H = 150$ seconds always leaves out from the plans some m-type vertices that are too far. For $|V_c|/|V| = 0.1, 0.2$ the performance is substantially similar for I-OPT and RE, and augmenting the ratio to 0.4 does not lead to substantial improvements for RANDOM and I-OPT, while RE is somehow biased by the presence of several communication vertices since it is more rare that the obtained walk need to be augmented to reach a c-type vertex, implying that less m-type vertices are visited as a side-effect.

Finally, we focus on a realistic case in which we simulate the temporary unavailability of a communication link. Faults are assumed to happen independently from each other as follows: once a robot reaches a c-type vertex, with probability p_f the expected communication link will not be present (due, e.g., to a temporary network congestion), while with probability $1 - p_f$ the robot and the BS will be able to communicate as before. To deal with these unexpected events, we employ a simple recovery procedure in which the robots decide to move towards the closest c-type vertex. In Fig. 5(c), we report results on how performance of I-OPT in the Poli environment degrades with increasing p_f (note that the recovery procedure is independent of the chosen patrolling strategy). Our simple recovery procedure is effectively able to mitigate the impact of communication failures up to 25%, while for higher values of p_f the performance worsens less gracefully, but not dramatically.

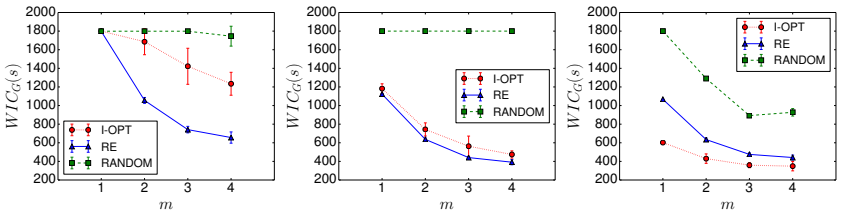


(a) Poli (b) Open (c) Grid

Fig. 3: Average idleness values of I-OPT, RE, and RANDOM for different team sizes for a 30 minutes mission.

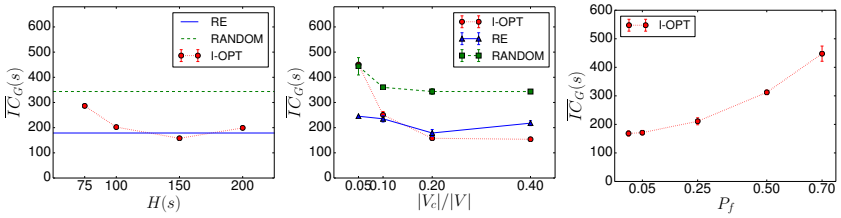
6 Conclusions

In this paper, we extended the classical multirobot persistent patrolling framework and the related idleness evaluation metrics in order to account for the presence of a limited number of communication areas and we presented some patrolling strate-



(a) Poli (b) Open (c) Grid

Fig. 4: Worst-case idleness values of I-OPT, RE, and RANDOM for different team sizes for a 30 minutes mission.



(a) Varying H in I-OPT. (b) Varying the ratio $|V_c|/|V|$. (c) Varying p_f.

Fig. 5: Different parameters in the Poli environment (m=3).

gies tailored for such a setting. Experimental results show the effectiveness of the proposed patrolling strategies in the optimization of our idleness-based evaluation criteria in different environments and settings.

As future works, we intend to adapt the theoretical analysis of cyclic strategies of [4] (relative to the classical worst-case idleness metric) to our framework and to validate our multirobot system also on real robots.

References

1. Banfi, J., Basilico, N., Amigoni, F.: Minimizing communication latency in multirobot situation-aware patrolling. In: Proc. IROS, pp. 616–622 (2015)
2. Basilico, N., Carpin, S.: Online patrolling using hierarchical spatial representations. In: Proc. ICRA, pp. 2163–2169 (2012)
3. Basilico, N., Gatti, N., Amigoni, F.: Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In: Proc. AAMAS, pp. 57–64 (2009)
4. Chevaleyre, Y.: Theoretical analysis of the multi-agent patrolling problem. In: Proc. IAT, pp. 302–308 (2004)
5. Dames, P., Kumar, V.: Cooperative multi-target localization with noisy sensors. In: Proc. ICRA, pp. 1877–1883 (2013)
6. Elmaliach, Y., Agmon, N., Kaminka, G.: Multi-robot area patrol under frequency constraints. Ann Math Artif Intel **57**(3-4), 293–320 (2009)

7. Hollinger, G., Singh, S.: Multirobot coordination with periodic connectivity: theory and experiments. IEEE T Robot **28**(4), 967–973 (2012)
8. Howard, A., Roy, N.: The robotics data set repository (Radish). http://radish.sourceforge.net/ (2003)
9. Iocchi, L., Marchetti, L., Nardi, D.: Multi-robot patrolling with coordinated behaviours in realistic environments. In: Proc. IROS, pp. 2796–2801 (2011)
10. Machado, A., Ramalho, G., Zucker, J.D., Drogoul, A.: Multi-agent patrolling: an empirical analysis of alternative architectures. In: International Workshop on Multi-Agent Systems and Agent-Based Simulation, pp. 155–170 (2002)
11. Mosteo, A., Montano, L., Lagoudakis, M.: Guaranteed-performance multi-robot routing under limited communication range. In: Proc. DARS, vol. 8, pp. 491–502 (2009)
12. Ochoa, S., Santos, R.: Human-centric wireless sensor networks to improve information availability during urban search and rescue activities. Inform Fusion **22**, 71–84 (2015)
13. Portugal, D., Rocha, R.P.: Multi-robot patrolling algorithms: examining performance and scalability. Adv. Robotics **27**(5), 325–336 (2013)
14. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
15. Santana, H., Ramalho, G., Corruble, V., Ratitch, B.: Multi-agent patrolling with reinforcement learning. In: Proc. AAMAS, pp. 1122–1129 (2004)
16. Stephan, J., Fink, J., Kumar, V., Ribeiro, A.: Hybrid architecture for communication-aware multi-robot systems. In: Proc. ICRA, pp. 5269–5276 (2016)
17. Tortonesi, M., Stefanelli, C., Benvegnu, E., Ford, K., Suri, N., Linderman, M.: Multiple-uav coordination and communications in tactical edge networks. IEEE Commun Mag **50**(10), 48–55 (2012)
18. Vaughan, R.: Massively multiple robot simulations in stage. Swarm Intelligence **2-4**(2), 189–208 (2008)