Guaranteed-Performance Multi-Robot Routing under Limited Communication Range

Alejandro R. Mosteo, Luis Montano, and Michail G. Lagoudakis

Abstract A common assumption made in multi-robot research is that robots operate in a fully networked environment, where any two robots can communicate reliably at any time. However, in real-world missions over large geographical areas robots have a limited communication range and signal propagation conditions may be unpredictable due to obstacles or interferences. In such cases, communication limitations must be taken explicitly into account. In this paper, we present a multi-robot routing algorithm for dense environments under limited communication range. We focus on the aspect of careful mission planning in order to guarantee continuous connectivity and mission completeness, while offering a guarantee on performance under a variety of team objectives. The proposed algorithms are demonstrated in a large-scale domain using realistic simulations on the Player/Stage platform.

1 INTRODUCTION

Many application domains involve tasks of visiting places of interest using robots: soil or atmospheric sampling, sensory inspection, object delivery, etc. This problem of routing a team of robots over appropriate paths with the goal of visiting all target locations (corresponding to specific tasks) is known as *multi-robot routing*. A solution to this problem guarantees that all locations are visited, while some team performance measure of cost is optimized. It is known that solving multi-robot routing optimally is an NP-hard problem for many interesting team objectives [4]:

————————————————

Alejandro R. Mosteo and Luis Montano

Instituto de Investigación en Ingeniería de Aragón, University of Zaragoza, 50018 Zaragoza, Spain, e-mail: {amosteo,montano}@unizar.es

Michail G. Lagoudakis

Department of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania, Crete, Greece e-mail: lagoudakis@ece.tuc.gr

minimizing total mission cost (energy), minimizing total mission time (makespan), and minimizing average service time (latency).

There is a growing trend to drop the assumption that the robots of the team are able to freely communicate since, in many interesting real-world applications, network connectivity must be explicitly considered and preserved. Furthermore, while some domains may tolerate temporary losses of communication between team members, in others it is mandatory to maintain connectivity at all times: monitored surveillance, rescue missions, tightly coordinated cooperation, etc. These include domains where uninterrupted contact to a stationary monitoring base or command center is commonly required.

Mobile robots typically use an ad-hoc wireless connection to communicate. Maintaining full connectivity between team members does not necessarily imply that each robot communicates directly with all other robots, but rather that any robot can reach any other robot either directly or by relaying messages through some other robot(s). Therefore, it is required at all times that a spanning tree of network links over all robot locations exists. Adding such limited communication constraints to multi-robot routing makes the problem even harder.

In this paper we present an approach tailored to these concerns which provides a routing solution suitable for dense environments and prevents network splits with minimal assumptions on the signal propagation. In effect, each target is eventually served by a group of robots without breaking their connectivity requirement to a stationary base. We study algorithms that yield efficient routes and we prove that the quality of these routes is within a factor from the optimal ones using graph-theoretic arguments. The proposed approach is validated in a large realistic scenario using the Player/State robot simulator.

The paper is organized as follows: Section 2 reviews the issue of limited connectivity and how other researchers have addressed it. Section 3 provides a formal definition of the multi-robot routing problem. Section 4 describes the proposed strategy, whose theoretical properties and quality bounds are proved in Section 5. Empirical results from simulation are shown and discussed in Section 6. Finally, we discuss future work and conclude in Section 7.

2 LIMITED CONNECTIVITY

The limited connectivity issue is a growing concern in multi-robot research. One common approach is to include link quality in the motion [7] or task generation functions. This idea is not readily applicable to general-purpose field or service robotics, because the tasks are provided by external sources and may be totally unrelated to those generated by the team in order to preserve communication. These approaches are, furthermore, prone to local minima problems where some robots become trapped by obstacles, threatening mission completion. To work around this problem, some proposals [10] require temporary network splits to allow for escape routes, and therefore cannot guarantee uninterrupted connectivity.

General approaches suitable for general-purpose service robotics, where tasks are not tied to a particular problem domain and uninterrupted connectivity is an explicit requirement, are scarce. In [3], a line-of-sight constraint is applied, but actual link quality is not used. Other approaches rely heavily on assumptions about the signal decay function. However, there is growing evidence and concern [9] suggesting that such models can badly misrepresent the behavior of real signals, leading to temporary connectivity losses or outright algorithm failures.

Real signals in populated scenarios exhibit a complex behavior caused by reflections, multi-paths, scattering and interferences, among other factors [11]. This uncertainty directed us towards reactive solutions where the actual signal is measured *in situ*, instead of relying on a predictive model. Along these lines, in the past we proposed several algorithms [6] that operate on top of a signal-constrained navigation model, albeit only in an obstacle-free environment. An alternative proposal experiments with a robot that deploys stationary relays when necessary to extend its range of operation [8] in out-of-sight scenarios, focusing mainly on hardware issues.

In this work we extend ideas found in both directions of this recent work to provide a general solution for field robots in scenarios with arbitrarily shaped obstacles. Reactive operation based on measured signal enables us to drop most assumptions on the signal propagation model. In practice, we only require that the decay function is reasonably continuous, meaning that, between two consecutive measurements of signal quality made by a moving robot, the signal will not skip a controlled zone that serves as a buffer between the safe and the forbidden zones (Fig. 1). As a bonus, our approach in this work is readily applicable to tethered robots [5] and teams of robots.
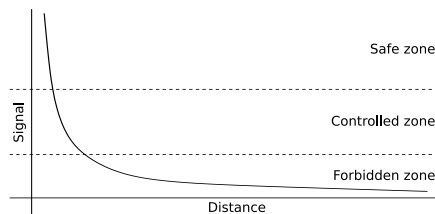

Fig. 1 Typical degradation of the radio signal quality as a function of the distance between transmitter and receiver. A link is broken when signal quality drops to the forbidden zone.

3 MULTI-ROBOT ROUTING

The considered *multi-robot routing* problem is formally specified by a set of robots, $R = \{r_1, r_2, \ldots, r_n\}$, a set of targets, $T = \{t_1, t_2, \ldots, t_m\}$, the locations of both robots and targets on the two-dimensional plane, and a non-negative cost function $c(i, j)$, $i, j \in R \cup T$, which denotes some abstract cost of moving between locations i and j (e.g., distance, energy, time, etc.). We assume that the robots are identical; therefore the same cost function applies to all robots. Typical cost measures are travel distance, travel time, or energy consumption between locations. In this work, we additionally assume that the cost function satisfies the triangle inequality. This as-

sumption holds as long as the robots follow the cheapest obstacle-free path when moving between locations.

The mission begins with all robots located at some base location and completes when all targets are visited and the robots return to base. The problem of multi-robot routing is to find an allocation of targets to robots and a path for each robot that begins at the base, visits all targets allocated to it, and ends back to the base, so that the mission is completed and a team objective is minimized. Three intuitive team objectives [4], corresponding to minimizing energy, makespan, and latency respectively, are:

MINSUM: Minimize the sum of the robot path costs over all robots.

MINMAX: Minimize the maximum robot path cost over all robots.

MINAVE: Minimize the average target path cost over all targets.

The *robot path cost* of a robot is the sum of the costs along its entire path. The *target path cost* of a target t is the total cost of the path traversed by the designated robot from the base up to target t along its path. The three team objectives above can be formally expressed as

$$\min_{\mathscr{A}} \sum_j RPC(r_j, A_j), \quad \min_{\mathscr{A}} \max_j RPC(r_j, A_j), \quad \min_{\mathscr{A}} \frac{1}{m} \sum_j CTPC(r_j, A_j),$$

where $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$ is a partition of the set of targets, so that targets in A_i are allocated to robot r_i, $RPC(r_i, A_i)$ denotes the *robot path cost* for robot r_i to visit all targets in A_i starting from and finishing to the base, and $CTPC(r_i, A_i)$ denotes the *cumulative target path cost* of all targets in A_i, again, if robot r_i visits all targets in A_i starting from the base.

4 THE CONNECTTREE **ALGORITHM**

Our generic strategy for multi-robot routing under limited communication range in dense environments is based on the following ideas. Robots move together as a whole towards a target, monitoring the network link to the base; once this link falls under the safety threshold and into the controlled zone, one robot is stopped at that point to act as a relay, and the rest continue advancing while monitoring now their link to the stopped relay robot. Using this pattern repeatedly, the robots can reach far away from the base without ever breaking the link to the base. Note that the robots do not need to know their communication range; the decision to stop a robot as a relay is made dynamically as dictated by the locally measured signal quality.

Once the target is visited, the robots are allowed to retreat from the target only along the path they used to reach it to avoid potential deadlocks due to the density of obstacles and the connectivity constraint. This retreat mirrors the deployment, and a relay can only start retreating when all moving robots reach the relay location. In practice, the set of paths used by the robots is a tree rooted at the base station (Fig. 2).

Note that two paths to different targets may share a portion in the upper levels of the tree. Therefore, after visiting a target, robots do not need to retreat all the way to the base to move on to the next target. Instead, it is sufficient to retreat up to a common ancestor point between the two paths. Thus, during the entire process, critical network links lie within the tree and the relative robot ordering along any path remains the same when going towards a target and when retreating back. This would allow the use of this approach in conveniently tethered robots, as well as to a single robot capable of deploying and collecting static relays. Alg. 1 shows the pseudo-code for the CONNECTTREE algorithm.

It can be argued that it is more MINSUM-cost efficient to begin moving only one robot, and move relays one by one on demand once a weak signal is detected between any robot pair. This scheme is not recommended for completeness purposes; it involves two moving endpoints in the network links, instead of just one, increasing the risk of sharp network dropouts. Moreover, it does not improve MINMAX and MINAVE.

Fig. 2 A possible path tree visiting four goals. Any depth-first traversal gives minimum backtracking and thus mission timespan. However, 3-2-1-4 would minimize average cumulative cost.

There are two key subroutines affecting performance in the CONNECTTREE algorithm. *AllocateFrom* (line 2) decides the next target to visit. One advantage of using a tree structure is the fact that any depth-first traversal will have minimum backtracking along the branches, and thus gives the least MINSUM and MINMAX cost for the given tree. We used depth-first traversals in this work, while future research may specifically address traversal orderings for the MINAVE objective. The most influential subroutine, however, is *BuildTree* (line 1), which builds the tree routes of the robots. Depending on the depth of the tree from the base, the robots may or may not be able to traverse it in its entirety. If the depth exceeds the maximum range they can reach collectively, the remaining target(s) along that branch will be abandoned. To avoid such potential failures, one may use a straightforward scheme: construct a star-like tree where each target is attached directly to the root (base) using the least-cost path between them. Despite its simplicity and the potential inefficiency of the resulting tree, this scheme guarantees that the mission will be accomplished, if it can be accomplished. No target will be abandoned, unless it is beyond the ability of the team to reach it. We refer to this simple variation of the CONNECTTREE algorithm as STAR.

Further improvements studied in this paper focus solely on tree construction with the goal of achieving provably good performance. It should be clear that the total cost of the constructed tree is indicative of the team performance during the mission. Additionally, the depth of the tree plays a key role in the success of the mission. Therefore, it makes perfect sense to try to keep the total cost and the depth of the tree low. Towards this end, we suggest three different ways of building such trees based on solving a Traveling Salesman Problem (TSP), finding a Minimum Spanning Tree (MST), and finding a depth-limited spanning tree, respectively.
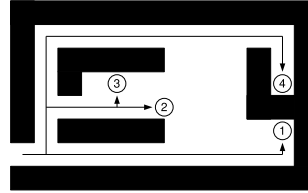

Algorithm 1 Description of the CONNECTTREE strategy. Variable i defines the range $i..n$ of moving robots, while $1..i-1$ represents the stopped relays. r_0, for simplicity, denotes also the Base. $\omega_{i,j}$ denotes link quality between robots i, j (or Base and r_j for $\omega_{0,j}$). UT is the upper threshold of the controlled zone. Variations are denoted by $s \in S = \{\text{STAR}, \text{TSPT}, \text{MST}, \text{DLST}\}$

Inputs: Base, T, n, s

 1: PathTree ← BuildTree(s, Base, T) // *Construct the initial tree of paths.*
 2: CurrentTarget ← AllocateFrom(PathTree) // *Choose first target.*
 3: $i \leftarrow 1$ // *Initially, all robots move.*
 4: **while** PathTree $\neq \emptyset$ **do** // *Main mission loop, targets remaining.*
 5:    **while** ¬Reached(CurrentTarget) **do** // *Advancing towards target.*
 6:       StepTowards($who \Rightarrow \{r_i \dots r_n\}, where \Rightarrow$ CurrentTarget)
 7:       **if** $\omega_{i-1,i} \leq UT$ **then** // *Link exhausted, r_i must stop.*
 8:          **if** $i = n$ **then** // *Was r_i our last available robot?.*
 9:             PathTree ← BuildTree(s, Base, $T \cap$ RemainingTasksIn(PathTree))
10:             **exit** // *Tree rebuilt, backtrack to new path.*
11:          **else** // *Stop r_i to act as relay.*
12:             StopRobot(i); $i \leftarrow i + 1$
13:          **end if**
14:       **end if**
15:    **end while**
16:    **if** Reached(CurrentTarget) **then**
17:       PathTree ← PathTree $\setminus$ CurrentTarget // *Remove reached target.*
18:    **end if**
19:    **if** PathTree $\neq \emptyset$ **then** // *Select next target from remaining ones.*
20:       NextTarget ← AllocateFrom(PathTree)
21:       BackPoint ← CommonAncestor(CurrentTarget,NextTarget)
22:       // *Deepest common node in the retreat path and the new path.*
23:    **else**
24:       NextTarget ← BackPoint ← Base // *Only remains the comeback to base.*
25:    **end if**
26:    **while** ¬Reached(BackPoint) **do** // *Retreating from current target.*
27:       StepTowards($who \Rightarrow \{r_i \dots r_n\}, where \Rightarrow$ BackPoint)
28:       **if** Reached(r_{i-1}) **then** // *Retreat starts for deepest stopped relay.*
29:          $i \leftarrow i - 1$
30:       **end if**
31:    **end while**
32:    CurrentTarget ← NextTarget // *Start advancing to next target.*
33: **end while**

TSPT: The first proposed algorithm is based on computing or approximating an optimal open TSP tour, which visits all targets beginning from the base exactly once, but does not return to the base. Despite the NP-hardness of the problem, there are several efficient algorithms and heuristics for finding optimal or near-optimal TSP

tours for real-world problems in reasonable time. An open TSP tour is a degenerate tree with just a single branch, therefore CONNECTTREE can easily traverse it. However, its depth may be well beyond the reaching ability of the team and the mission may fail even though the robots might have the ability to accomplish it. To guarantee mission completeness, we propose an iterative replanning scheme: As soon as an unreachable target is detected during tree traversal, a new open TSP tour is constructed from the base to the remaining non-visited targets. The robots will retract back, in the worst case all the way to the base, to traverse the new tree. The first target on the new tree will be connected to the base with the least-cost path, therefore at least one target will be visited at each replanning and the mission will progress steadily. Under this replanning scheme, the mission will always be completed, if it can be completed. We refer to this variation of the CONNECTTREE algorithm as TSPT (TSP tree).

MST: The second proposal is based on finding a minimum spanning tree (MST) over all target nodes and the base. Using Prim's algorithm with the base as the seed node, a minimum spanning tree is easily constructed in polynomial time and can be traversed by the CONNECTTREE algorithm. Even though its total cost is kept low, the depth can be arbitrary, but certainly no more than the depth of the TSP tree. Again, to guarantee mission completeness, we adopt the same replanning scheme. As before, the mission will always be completed, if it can be completed. We refer to this variation of the CONNECTTREE algorithm as MST.

DLST: Our last proposal is based on finding a depth-limited spanning tree with the goal of minimizing the possibility of a costly replanning step. If the robots have some estimate of the maximum depth D they can collectively reach, then we can construct a spanning tree whose depth will not exceed D. This is achieved by a simple modification to Prim's algorithm: at each step we only consider those edges between the partial tree and the remaining nodes that do not result in a tree depth greater than D. In a situation of abandoned targets due to an optimistic estimate of D, we proceed with the replanning scheme as above. We refer to this variation of the CONNECTTREE algorithm as DLST (Depth-Limited Spanning Tree).

In summary, all four variations of the CONNECTTREE algorithm are safe (network integrity is maintained at all times) and complete (mission is successfully accomplished) for any number of robots, any number of targets, and any communication range per robot, unless the mission is a failure by design (some target is out of range even using the optimal path with all robots).

5 PERFORMANCE BOUNDS

From a graph-theoretical point of view, the problem of finding the "best" tree to be traversed by CONNECTTREE is equivalent to a minimum Steiner tree problem [2]: given a weighted graph $G = (V, E, w)$ and a subset $U \subseteq V$, find a tree in G that interconnects all vertices in U with the least cost. Notice the difference to an MST problem over U; a Steiner tree may contain additional vertices not in U that help

reduce the cost. The Euclidean Steiner tree problem considers the interconnection of $|U|$ points on the plane with the minimum cost using any intermediate points on the plane. In our case, the Steiner tree would be rooted at the base with all targets at the leaves. The minimum Steiner tree problem is known to be NP-hard, therefore we abandoned the idea of finding an optimal Steiner tree in favor of approximation algorithms.

In this section, we analyze the proposed algorithms and relate their performance to a baseline performance determined as the MINMAX cost (or the MINSUM cost; they are the same) of a closed TSP tour of a single robot that begins at the base, visits all targets, and returns to the base without any communication range limits. We refer to this ideal solution as CTSP (Closed TSP) and we define the performance ratio of any algorithm as the ratio of its performance according to some metric over the baseline performance. Our main results are stated below.

Theorem 1. The performance ratio of the TSPT and MST variations of CONNECT-TREE without replanning is at most $2n$ for the MINSUM, at most 2 for the MINMAX, and at most 2 for the MINAVE objectives.[1]

Proof. Under our tree traversal conventions, the mission cost according to the MIN-MAX metric of any variation will be exactly twice the total cost of the underlying tree. Clearly, the open TSP tour cannot be worse than the closed TSP tour and the MINMAX performance of TSPT is exactly twice the cost of an open TSP tour. Therefore, MINMAX(TSPT) $\leq$ 2 MINMAX(CTSP) and the MINMAX performance ratio of TSPT is bounded by 2. Similarly, it is known that in metric spaces, like ours, the total cost of a minimum spanning tree does not exceed the cost of a closed TSP tour. The MINMAX performance of the MST variation is exactly twice the cost of a minimum spanning tree. Therefore, MINMAX(MST) $\leq$ 2 MINMAX(CTSP), and the MINMAX performance ratio of MST is also bounded by 2. The MINSUM cost of any variation cannot be determined precisely, however it will be within a factor n of the MINMAX metric (all robots traverse the entire tree in the worst case), that is MINSUM(X) $\leq n$ MINMAX(X). Similarly, the MINAVE metric for any variation will not exceed the MINMAX metric (almost all targets are placed at the leaf of the tree visited last in the worst case, therefore each target is visited just before the end of the mission), that is MINAVE(X) $\leq$ MINMAX(X). The remaining bounds are now obvious.

Theorem 2. The performance ratio of the TSPT and MST variations of CONNECT-TREE with replanning is at most $2nm$ for the MINSUM, at most $2m$ for the MIN-MAX, and at most $2m$ for the MINAVE objectives.

Proof. Each planning or replanning of a tree eliminates at least one target, therefore in the worst case m trees will be build. For each of these m trees the previous bounds hold, therefore it suffices to include a factor of m.

---

[1] As of this time, we have not been able to obtain bounds on the performance of DLST, but we are confident they exist.

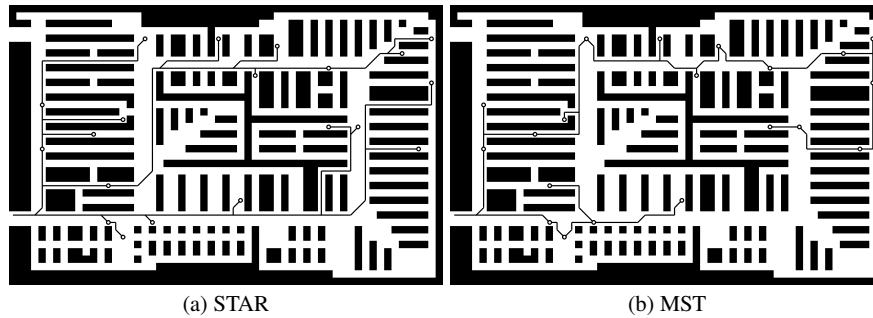(a) STAR   (b) MST

Fig. 3 Simulation scenario with overlaid path trees. Twenty targets are shown for clarity, although fifty targets were actually used in simulations.

Despite the fact that these results represent a worst-case, pessimistic analysis that yields loose bounds, it is striking that the additional cost introduced by the limited communication range cannot really grow arbitrarily. If the need for replanning is avoided, then performance is guaranteed to be very close to the ideal CTSP performance. Our experimental results demonstrate that performance in practice is well below the theoretical bounds.

6 SIMULATION RESULTS

The simulation map is a $135m \times 86m$ realistic scenario (Fig. 3), modeled after a shopping mall close to our lab, where obstacles are the product stands. This scenario is interesting, not only because of the amount, size and density of obstacles, but also because of the large central obstacle: Its particular configuration and its relative position to the deployment point would trap robots in deadlock situations (which cannot happen in our approach) when using methods based on local utility functions. A collaboration agreement has been reached with the owners of the shopping mall to conduct real experimentation on-site in the near future. Visiting randomly distributed targets in such a scenario may represent tasks of object inspection or delivery, service requests, or sample collection missions, for example.

The simulation environment was the Player/Stage robot simulator [1] which offers realistic mobile robot kinematics. The robot team consisted of nine simulated Pioneer 3AT robots, one of which served as the communication base and stayed at the initial point of deployment. High-level path planning was achieved by applying standard A^* search to an 8-vicinity grid of fixed size over the mission area. Paths obtained this way were used for constructing trees over the grid.

There were $n = 8$ mobile robots and $m = 50$ targets in the mission. The targets were uniformly randomly distributed over the free cells of the map. The communication range was modeled as a circular area of fixed radius around the robot. The value of this radius was not known to the robots at execution time. Instead, they
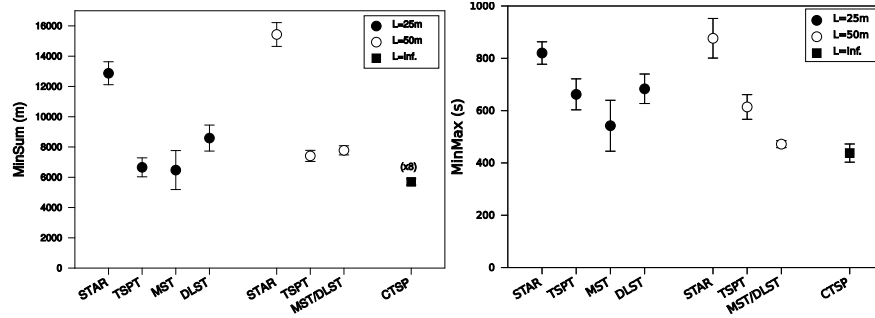
Fig. 4 Left: MINSUM evaluation. Right: MINMAX evaluation.

queried a function that returned a simulated measurement of link quality ($\omega_{i,j}$ in Alg. 1) derived from this radius.

More specifically, the radius length L was set to the values of $L = 25m$ and $L = 50m$, typical of wifi transmitters. The shortest range is barely enough to reach the farthest places in the map, thus making the abandonment of targets a significant concern, and was chosen to gain information on performance under extreme conditions. The longest range, on the contrary, provided insight on the improvements that can be gained, when we are not too tight on resources. For the DLST variant, we set $D = nL$. Finally, the CTSP cost of the ideal closed TSP
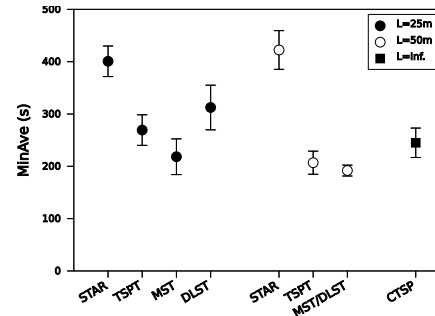

Fig. 5 MINAVE evaluation.

tour was measured to serve as a comparison basis with respect to the theoretical bounds; CTSP was obtained using a single robot without any network restrictions. Ten runs were performed for each algorithm and each value of L, using a different seed for each run; nevertheless, the same set of seeds was used across the different algorithms, so that they are compared on the same basis.

Fig. 3 shows two initial trees for 20 targets only (for clarity). It can be easily seen that the STAR tree is more balanced than the MST tree; the MST tree eventually might require a replanning to complete the mission.

Figs. 4 to 5 show mean values and the 95% confidence intervals for each of the three performance metrics in all cases. We observed that DLST with $L = 50$ generated the same solutions as MST. This means that the branch depth limit was not hit, and thus both are conflated in the figures. We highlight the following observations:

• Traversal of the STAR tree requires a noticeable additional amount of backtracking, which results in the worst performance in all cases.

• MST is the best performer for MINMAX and MINAVE. Of special interest is the fact that it outperforms DLST for $L = 25$, despite DLST making use of the D

limit. This is a consequence of the pessimistic prediction of DLST: paths along the branches of the tree are usually longer than the Euclidean distance between stopped relay robots (the signal propagates in straight lines). Consequently, DLST discarded branches during tree construction that were in fact valid.

• In the MINSUM case, we observed that TSPT and MST performed comparably well. Together with the previous point, this makes MST the best choice amongst the tested strategies.

• Comparing the performance of the algorithms to the ideal CTSP baseline performance (shown only in Fig. 4–Right), we see that all instances with known bounds are well below these bounds. Thus, the typical performance is noticeably better; the worst case represents rather pathological cases.

• Note that the MINSUM performance is specially affected in the CONNECTTREE strategy; we could say that it is its "Achilles' heel" and the price it pays for network integrity. Since all robots not acting as static relays are moving together all the time, this causes the MINSUM cost to grow by a factor of n in the worst case.

• To allow for a fair comparison, we have plotted the actual MINAVE cost of CTSP in Fig. 5 instead of the baseline cost. Similarly, in Fig.4, the baseline CTSP cost would correspond to a single robot performing the closed route. However, to make the comparison more meaningful, in this figure the CTSP cost is n times that of the single robot.

• We did not run the algorithms with $L = \infty$ because no replannings were necessary for $L = 50$. Consequently, we would have observed the same MINMAX and MINAVE costs, and an even higher MINSUM cost.

7 CONCLUSION

In this paper we studied the problem of multi-robot routing under communication constraints, when network integrity is a primary concern and dense obstacles make the modeling of accurate signal propagation too difficult. The proposed strategy relies on the careful construction of a tree of paths, rooted at a monitoring station, that reaches all targets. Robots traverse this tree in a network-safe manner which requires only local link information. A valid target may require any number of robots (even all) to be serviced, even using the optimal route; all such targets are guaranteed to be serviced.

We focused on tree construction algorithms backed with theoretical bounds on their performance with respect to a baseline performance of an optimal closed TSP tour cost in a network unrestricted environment. Four algorithms were studied: a simple star tree of least-cost paths from the base to each target, an optimal open TSP tour, a minimum spanning tree, and a depth-limited spanning tree.

Simulations in a large realistic scenario empirically demonstrated improvement over the simple star solution, and identified the minimum spanning tree (MST) as the best choice for our network safe approach. We also observed that the empirical performance ratios are well below the worst case bounds.

In future work, with the adoption of predictive signal models, parallelization of task execution may be achieved, without relinquishing the inherent network safeness of the CONNECTTREE approach. The theoretical analysis may also be extended to bound actual performance in terms of the truly optimal performance for each metric.

References

[1] Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Int. Conf. on Adv. Robotics, pp. 317–323 (2003)

[2] Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. Elsevier (1992)

[3] Kalra, N., Ferguson, D., Stentz, A.: A generalized framework for solving tightly-coupled multirobot planning problems. In: Int. Conf. on Robotics and Automation, p. 3359 (2007)

[4] Lagoudakis, M.G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., Jain, S.: Auction-based multi-robot routing. In: Robotics: Science and Systems Conf. Cambridge, USA (2005)

[5] Marques, C., Cristoao, J., Lima, P., Frazao, J., Ribeiro, I., Ventura, R.: Raposa: Semi-autonomous robot for rescue operations. In: Int. Conf. on Intell. Robots and Systems, pp. 3988–3993 (2006). DOI 10.1109/IROS.2006.281836

[6] Mosteo, A.R., Montano, L., Lagoudakis, M.G.: Multi-robot routing under limited communication range. In: Int. Conf. on Robotics and Automation (2008)

[7] Mostofi, Y.: Communication-aware motion planning in fading environments. In: Int. Conf. on Robotics and Automation (2008)

[8] Pezeshkian, N., Nguyen, H.G., Burmeister, A.: Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations. In: IASTED Int. Conf. on Robotics and Applications. DARPA/ITO, Germany (2007)

[9] Quin, L., Kunz, T.: On-demand routing in MANETs: The impact of a realistic physical layer model. Lecture Notes in Computer Science (2003)

[10] Rooker, M.N., Birk, A.: Communicative exploration with robot packs. Lecture Notes in Artificial Intelligence pp. 267–278 (2006)

[11] Sridhara, V., Bohacek, S.: Realistic propagation simulation of urban mesh networks. Computer Networks **51**(12), 3392–3412 (2007). DOI doi:10.1016/j.comnet.2007.01.035