2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Multi-Robot Routing under Limited Communication Range

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

Abstract— **Teams of mobile robots have been recently proposed as effective means of completing complex missions involving multiple tasks spatially distributed over a large area. A central problem in such domains is multi-robot routing, namely the problem of coordinating a team of robots in terms of the locations they should visit and the routes they should follow in order to accomplish their common mission. A typical assumption made in prior work on multi-robot routing is that robots are able to communicate uninterruptedly at all times independently of their locations. In this paper, we investigate the multi-robot routing problem under communication constraints, reflecting on the fact that real mobile robots have a limited range of communication and the requirement that connectivity must remain intact (even through relaying) during the entire mission. We propose four algorithms for this problem, all based on the same reactive framework, ranging from greedy to deliberative approaches. All algorithms are tested in various scenarios implemented using the Player-Stage robot simulation environment. Our results demonstrate that effective multi-robot routing can be achieved even under limited communication range with moderate loss compared to the case of infinite communication range.**

I. INTRODUCTION

Recent research progress in robotics has allowed the use of robot teams in various real-world applications, such as search-and-rescue missions, area exploration, and field demining. Teams of robots can be advantageous over single robots; they offer greater flexibility through dynamic team coordination and reorganization, greater efficiency through parallel task execution, and greater reliability through resource redundancy.

A central problem in teams of mobile robots relates to locomotion coordination, when the tasks comprising the mission are spatially distributed over a geographical area. The problem of routing robots over available paths between target locations (corresponding to specific tasks) is known as *multi-robot routing*. The objective is to find a route for each robot, so that each target location is visited exactly once by exactly one robot (no waste of resources), all target locations are eventually visited by some robot (mission completeness), and the entire routing mission is accomplished successfully in the best possible way (optimization of performance).

Most prior work on multi-robot routing, implicitly or explicitly, makes the assumption that the robots of the team are able to communicate at all times independently of their present location. This assumption holds true when the mission is extended over a small geographical area, for example, inside a building. However, in many interesting real-world applications missions are extended over large geographical areas, where network connectivity cannot be taken for granted. Mobile robots typically use a wireless connection to communicate with the other team members; it is, therefore, natural to assume that the communication range of each robot extends to a circular area around its current location up to a certain radius; any communication outside this area is not possible. 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 the minimum spanning tree over all robot locations has no edge longer than the maximum communication radius.

In this paper, we take limited communication constraints explicitly into consideration during the planning of routes. Such an approach is deemed necessary by the fact that each target may not be reachable independently by a single robot, without the support of other robots acting as relays. In effect, each distant target is eventually served by a group of robots, resulting in complicated allocation schemes. We propose four algorithms for multi-robot routing under limited communication. In this preliminary work on this problem, we focus solely on reactive allocations, whereby robot routes are built incrementally one target location at a time with the possibility of dynamically changing target allocations. However, in all cases, the resulting allocations allow the robots to accomplish a routing mission without breaking their connectivity requirement. This is guaranteed by an underlying motion-control mechanism based on virtual spring forces that keeps the robots together. We further test and compare all algorithms against each other in realistic multi-robot routing scenarios with varying degrees of communication range.

The remainder of the paper is organized a follows: In the next section, we review related work. Section III defines the multi-robot routing problem and Section IV discusses the issue of limited connectivity and the base mechanism that ensures network connectivity at all times. Section V describes the algorithms we propose; these algorithms are empirically evaluated in Section VI. Finally, we discuss future work in Section VII and conclude.

II. RELATED WORK

Communication constraints add a new level of complexity to the task allocation problem, however they bring the multi-

*Part of this research was conducted during a four-month research visit of A. R. Mosteo to the Technical University of Crete.

A. R. Mosteo and L. Montano are with the Instituto de Investigación en Ingeniería de Aragón, University of Zaragoza, 50018 Zaragoza, Spain {amosteo,montano}@unizar.es

M. G. Lagoudakis is with the Department of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania, Crete, Greece lagoudakis@ece.tuc.gr

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

robot routing problem closer to reality. Basic approaches opportunistically take advantage of network connectivity when available [1], but do not explicitly avoid network splits. To do so, a further possibility is to dictate task *generation* besides task allocation. In exploration, for example, goals may be decided as the result of cost functions that depend on signal quality [2]. This is difficult to carry over to more general applications, where tasks are provided by external sources and, thus, cannot be created based on system preferences.

Approaches where task generation is not controlled and connectivity is an explicit requirement are scarce. In the past a behavioral approach has been proposed, where connectivity maintenance is addressed, but is not guaranteed [3]. Other approaches rely on assumptions about the signal decay function [4] or the line-of-sight view [5]. However, it is known [6] that such models can badly misrepresent the real behavior of the signal. This may lead to failures in algorithms [7] or temporary connectivity losses.

The reactive allocation method proposed in this paper builds on an approach that treats connectivity as a strong, inviolable constraint. At the time of this writing we are not aware of other works which combine such solid network requirements with arbitrary routing tasks.

III. MULTI-ROBOT ROUTING

A *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 all 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 in either direction (e.g., distance, energy, time, etc.). Robots are assumed to be identical, therefore the same cost function applies to all of them. Typical cost measures, such as travel distance, travel time, or energy consumption between locations satisfy these assumptions in any typical environment.

The objective of multi-robot routing is to find an allocation of targets to robots and paths for all robots, so that all targets are visited and a team objective function is minimized[1]. In general, a team objective is expressed as

$$\min_{\mathscr{A}} f\big(g(r_1, A_1), \ldots, g(r_n, A_n)\big) \ ,$$

where function g measures the performance of each robot, function f measures the performance of the team, and $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$ is a partition of the targets, such that targets in A_i are allocated to robot r_i. In this paper, we consider three intuitive team objectives [8]:

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 r is the sum of the costs along its entire path, from its initial location to the last target on its path. The *target path cost* of a target t is the total cost of the path traversed by robot r (the unique robot assigned to visit t) from its initial location up to target t along its path.

The three team objectives above can be expressed in terms of our generic team objective structure. Let $RPC(r_i, A_i)$ denote the *robot path cost* for robot r_i to visit all targets in A_i from its current location. Similarly, let $CTPC(r_i, A_i)$ denote the *cumulative target path cost* of all targets in A_i, again, if robot r_i visits all targets in A_i from its current location. Then, the three team objectives can be expressed as

$$\begin{aligned}
\text{MINSUM} &: \quad \min_{\mathscr{A}} \sum_j RPC(r_j, A_j), \\
\text{MINMAX} &: \quad \min_{\mathscr{A}} \max_j RPC(r_j, A_j), \\
\text{MINAVE} &: \quad \min_{\mathscr{A}} \frac{1}{m} \sum_j CTPC(r_j, A_j).
\end{aligned}$$

Solving the multi-robot routing problem optimally under any of the above objectives is NP-hard [9]. Therefore, several researchers have focused on developing algorithms which deliver good allocations in practically efficient time.

IV. LIMITED CONNECTIVITY

A typical assumption made in multi-robot routing is that robots are able to communicate uninterruptedly at all times as they move, independently of their locations. In this paper, we investigate the multi-robot routing problem under communication constraints reflecting on the fact that real mobile robots have a limited range of communication and the requirement that connectivity must remain intact (even through relaying) during the entire mission.

Our routing algorithms are designed to work over an underlying mechanism that addresses solely the connectivity problem. This mechanism guarantees that, for reasonably continuous signal decay functions, the robotic team will at all times form a connected MANET[2] with real time traffic capabilities and optimal signal quality. This is enforced by giving higher priority to the coordinated motion subsystem over motion requests coming from other modules, such as task allocation. In essence, while the task allocation module is allowed to assign a target location to every robot at all times, the robots may fail to move towards it, if MANET maintenance requires it. Thus, task allocation must take this fact into account and guarantee that mission completion is possible without breaking the connectivity constraints or causing deadlock equilibrium states.

Contrary to other proposed solutions, where assumptions made on the signal qualities (line of sight, fixed radius coverage) may be violated at execution time if reality does not match them, our routing approach offers the advantage that connectivity constraints are never violated. Furthermore, our task allocation module has been designed with minimal and reasonable assumptions on real signal quality. Our only assumption dictates that, for two robots able to communicate directly, a third one between them is also able to talk to either of them. As long as this assumption holds, mission

[1] Although we assume that robots are not required to return to their initial locations, our algorithms and results apply also to the case of closed tours. Similarly, they apply to maximization of a utility function.

[2] Mobile Ad-hoc NETwork.

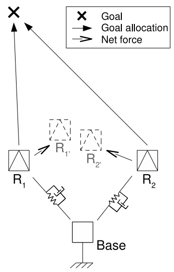

Fig. 1. The robots are initially unable to reach the goal. Net forces will bring them near or align them with the base. At that point one of them will be able to move forward. Robots forming a chain have maximum reachability.

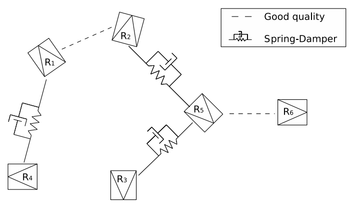

Fig. 2. An example of a MANET with virtual springs over links of low quality pertaining to the maximal quality spanning tree. Note that any single robot is not necessarily within the communication range of all the other robots. The robot clusters for task allocation derived from the connected components over springs are $\{R_1, R_4\}, \{R_2, R_3, R_5\}, \{R_6\}$.

completion is guaranteed for all goals within reach. In the worst case, a single chain is formed by all robots (Fig. 1).

A. Connectivity enforcement

Connectivity is enforced by a cooperative navigation system developed at the University of Zaragoza [10], [11]. Robots forming a MANET are allowed to move only in ways that do not break network connectivity. This is achieved by continuous monitoring of all robot-to-robot signal qualities (done as part of the real time networking protocol RT-WMP [10]) and building a virtual route along the spanning tree of maximum quality. The objective is to enable communication between any two robots at all times through this spanning tree.

Mobility is governed by the simulation of a physical spring-damper mesh model (Fig. 2), henceforth referred to as *spring*. Any link in the network spanning tree that falls below a safe quality level (Fig. 3) becomes a *virtual spring* that exerts attracting force between the robots that are about to cause a network split. As long as the link is above the safety level, no spring and, thus, no attracting force between the two robots appears. Robots move in reaction to the sum of forces exerted by springs upon them. Springs may appear because of insufficient link quality, but also because of goals, which act with a fixed attracting force, and because of obstacles, which exert a repelling force. These forces are translated into velocities that match the robot real capabilities (including non-holonomicity). This model ensures (a) smooth, jerk-free robot motion, (b) MANET connectivity maintenance, as there is always a spanning tree covering the entire network, and (c) maximal freedom of movement, as the spanning

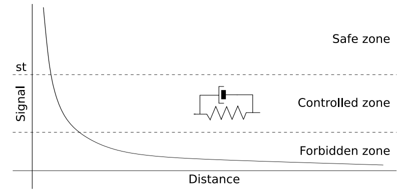

Fig. 3. Typical degradation of the radio signal quality as a function of the distance between transmitter and receiver. When quality falls below a safety threshold (st), a spring is activated to prevent network disconnection.

tree contains the minimum number of required links (and, ultimately, springs) to maintain a connected graph.

B. Task allocation strategy

We now describe the principal traits of our allocation strategy. In general, our strategy is based on assigning tasks to *clusters of robots*, consisting of robots linked (at any hop distance) by the controlled network links (springs) of the spanning tree (Fig. 2). All robots in the same cluster are assigned to the same task. This allocation guarantees that robots within a cluster do not exert conflicting forces upon each other towards different directions, which could cause deadlock equilibria. Instead, they are all trying to move towards the same goal, which guarantees that eventually they will from a chain with maximum reachability (Fig. 1).

Additionally, we define as *execution timespan*, the arbitrary time elapsed between the consecutive completion of two tasks. A robot supercluster (S-cluster henceforth) is a set formed by those clusters that, at any point during an execution timespan, have become temporarily linked by a spring. At the beginning of each execution timespan, each cluster defines a trivial S-cluster. As springs appear during each execution timespan, S-clusters of larger size are formed.

By furthermore imposing the same task to all robots in a S-cluster, it is guaranteed that cyclic allocations do not happen as clusters change. Since S-clusters can only grow in size within an execution timespan, at least one task among those assigned to the robot clusters forming the S-cluster will not be abandoned; in the extreme case, it would be a single goal assigned to all robots in the team. Should this happen, either all the robots can reach the goal (by forming a chain) and finish the current execution timespan, or the task is too far to be completed and shall be discarded from the mission. S-clusters are reset at the end of every execution timespan to allow for higher team throughput. The accompanying video clip offers examples of cluster formation during execution.

In summary, our task allocation strategy is characterized by the following properties: (a) it is *reactive*, as it reallocates tasks whenever the spring mesh changes, (b) it *degrades gracefully*, as it guarantees execution of at least one task at each time, and (c) it is *complete*, as it eventually completes the whole mission.

V. ALLOCATION ALGORITHMS

While the basic strategy remains the same, the key step of assigning pending tasks to S-clusters may be tailored to fit different needs according to the team objective. This is advantageous because we can retain the properties of the basic strategy, while using this step as a *swappable* allocation algorithm. This paper discusses four allocation algorithms, focusing on their multi-robot routing properties.

A. Greedy Allocation

The Greedy Allocation algorithm is the simplest approach to the allocation problem. In our implementation, the closest task to any robot is chosen and is propagated/allocated to all its S-cluster mates. This process is repeated until all robots have a goal. This Greedy Allocation algorithm is the only one presented herein that does not make any attempt at long-term, global planning. It serves as the comparison baseline for distributed algorithms.

B. TSP-Based Allocation

The motivation behind this algorithm is to avoid robot spreading, which leads to spring appearance and performance degradation. As a first step, a single-robot TSP[3] solution is computed. (In this work, the problem instance sizes permit the use of and optimal solver [12]; otherwise we would use any of the many known heuristics for good approximate solutions.) Let SC_t be the current S-clusters count at time t. At each reallocation, the first SC_t goals in the global TSP plan are allocated to the SC_t S-clusters using the well-known Hungarian method. This way, tasks are consumed by the team in the order dictated by the global TSP plan.

C. Clock Allocation

Our experiments with the two previous algorithms show that many times the robots naturally follow a sweeping behavior while maintaining an approximate abreast formation, as will be seen in Section VI. The Clock Allocation algorithm tries to explicitly induce this behavior, by precomputing a plan in which tasks are ordered by their angle in polar coordinates, starting at the closest one to any robot. The origin is placed at the middle of the working area, where the base is located. This plan is subsequently allocated to S-clusters and targets are consumed as explained in the TSP-Based Allocation algorithm.

D. Auction Allocation

With this algorithm we aim at influencing directly one of the three metrics presented in Section III. Drawing from past literature on auction-based multi-robot routing methods [9], we use auctions to preplan the order of tasks, according to an appropriate bidding rule [8] that relates to the desired metric.

While, in principle, we could use an auction to generate a single plan, like TSP-Based Allocation, this algorithm goes further and builds several smaller plans to be executed in parallel by different S-clusters according to a schedule geared towards optimizing the desired metric. This parallelism must be carefully exploited. Some tasks may be so distant that it would be impossible to reach them in parallel; an attempt would merely force the degradation of the team into a single S-cluster and completion of only one task, while others would be postponed.

In order to tackle this issue, the auction algorithm attempts to predict the number of robots needed to reach each goal from the initial (base) location. This prediction is based on the assumption that signal quality remains satisfactory within a known distance L, which when exceeded causes a spring to appear, if no other network route exists. With d denoting the goal distance to the base, tasks are classified in sets defined by the predicted number $N = \left\lfloor \frac{d}{L} \right\rfloor + 1$ of robots required to reach them from the base. It should be noted that this prediction, based on expected spring length, does not invalidate the properties of the basic task allocation strategy; it merely acts as a guess, which can only degrade performance, if wrong.

Tasks are auctioned sequentially set by set, with increasing $N = 1, 2, \ldots, n$, where n is the number of robots. For each set, at most $S_N = \left\lfloor \frac{n}{N} \right\rfloor$ plans can be carried out simultaneously in parallel for reaching targets in that set, since otherwise we would need more robots than available. Initially, for $N = 1$, the auction will result in the formation of up to n partial parallel plans. From those, only up to $n/2$ plans will participate in the auction set for $N = 2$. In general, the number of partial plans eligible to win tasks during the auction decreases with each farther set of tasks; within an auction set N, as soon as S_N different partial plans from the previous set win tasks, the remaining partial plans are finalized. Only one plan will survive for the tasks in auction set $N = n$. By construction, these plans can be carried out in parallel during execution; as short plans are completed, robots are freed up to join another robot cluster assigned to a longer plan, and so on, until all robots form a single cluster assigned to the longest plan.

VI. EXPERIMENTAL RESULTS

A. Simulation setup

In this preliminary study, we consider multi-robot missions where robots are deployed from a single base point and mission tasks cover a circular geographical area around this point. A single robot (or a station) remains at the central point and serves as the communication base, whereas all other robots can freely move around the area constrained only by the connectivity requirement. In this study, we only consider uniformly random distribution of tasks and no obstacles in the physical environment, representing exploration, mapping, or sample collection scenarios over a large terrain. As our goal is to study the feasibility of multi-robot routing under communication constraints, these choices represent an attempt to filter out bias coming from structured task distribution and/or specific obstacle layout.

Our simulation environment is based on the Player/Stage robot simulator [13] which offers realistic mobile robot dynamics. Our robot team consists of simulated Pioneer 3AT robots, one of which is always serving as the communication base and stays at the central point of deployment. Even though there are no physical, static obstacles in the environment, our robots are equipped with obstacle avoidance

[3]Traveling salesman problem.

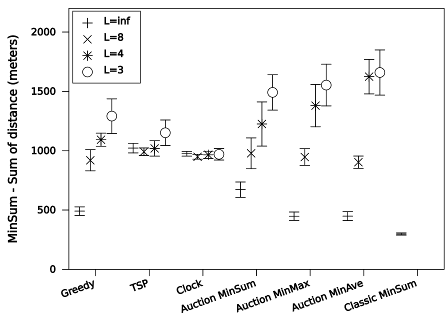

Fig. 5. Total distances traveled by all robots for each algorithm. Mean values and the 95% confidence intervals are shown in each case.

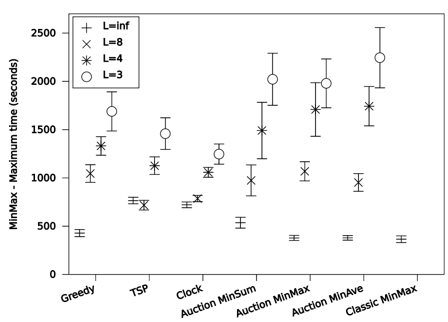

Fig. 6. Total mission completion time for each algorithm. Mean values and the 95% confidence intervals are shown in each case.

capabilities to dynamically avoid collisions with each other. For any given mission, our software measures the actual team performance with respect to all metrics studied in this paper.

More specifically, in our experimental setup, there are $n = 8$ mobile robots and $m = 100$ targets. The targets are uniformly randomly distributed over a circular area with a radius of 24 units. We study four communication ranges between the robots: (a) $L = \infty$, where there is no constraint and any robot can reach any target, (b) $L = 8$ units, where at most 3 robots are needed for reaching each target, (c) $L = 4$ units, where at most 6 robots are needed to reach the most distant targets, and (d) $L = 3$ units, where all 8 robots are necessary for reaching the most distant targets.

B. Simulation results

Performance for each of the proposed algorithms is measured in terms of the criteria described in Section III. These metrics are (a) total distance traveled by all robots combined (MINSUM), (b) mission timespan (MINMAX), and (c) average task completion time (MINAVE). In addition, for comparison purposes, we assessed the performance of classical multi-robot routing with auctions in the absence of communication constraints.

Fig. 4 shows two snapshots of each algorithm execution. Greedy has a characteristic spreading-out behavior, while TSP-Based and Clock exhibit the sweeping motion already described. Four parallel plans generated by Auction (with the MINSUM bidding rule) are visible at startup.

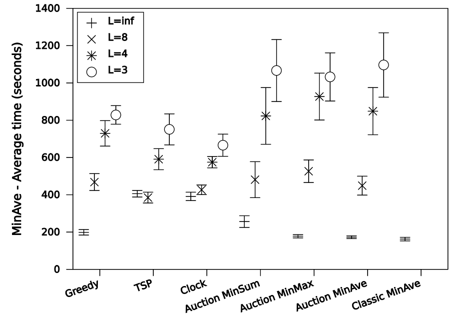

Fig. 7. Average task completion time for each algorithm. Mean values and the 95% confidence intervals are shown in each case.

Fig. 5 to Fig. 7 show the average performance of each algorithm with respect to each metric. We observe that TSP-Based and especially Clock are the most insensitive to the connectivity range L, with almost no penalty for $L = 8$. On the other hand, Greedy and Auction for all bidding rules quickly degrade with decreasing L. The plans built by Auction (using any of the three bidding rules) are difficult to adhere to in our reactive setup, because spring appearance often disrupts the predicted task execution order.

We notice that good performance under no constraints ($L = \infty$), for example in the Greedy and Auction case, does not necessarily carry over when communication is limited. TSP-Based and Clock are worse in the absence of constraints, but degrade at a lesser rate than the rest of the algorithms as constraints come into play. This is due to the characteristic sweeping pattern these algorithms induce. Furthermore, to its favor, Clock is computationally negligible ($O(m \log m)$), unlike TSP-Based or Auction ($O((n+m)^3)$).

The good performance of the sweeping behaviors suggests that partial parallel sweeps may improve the obtained results. In contrast, the parallel plans of Auction require more precise control over the spring spanning tree configuration to be of better use. We intend to explore this venue in the future.

In terms of objectives, MINMAX and MINAVE behave similarly, which has been observed also in other works on robot routing without constraints. This is due to the influence of maximum time on the average time. It is worth noting that there is little impact on MINSUM for all tested values of L in the case of sweeping behaviors like TSP-Based and Clock; this fact may be useful in scenarios where predicting power consumption is important.

VII. CONCLUSION

We have studied the problem of multi-robot routing under limited communication range. We have presented several algorithms for reactive task allocation when network connectivity is an inviolable constraint. Our proposal guarantees mission completion in open spaces and is customizable by means of swappable algorithms in order to optimize preferred performance metrics. We have studied common metrics used in multi-robot routing problems, such as team energy

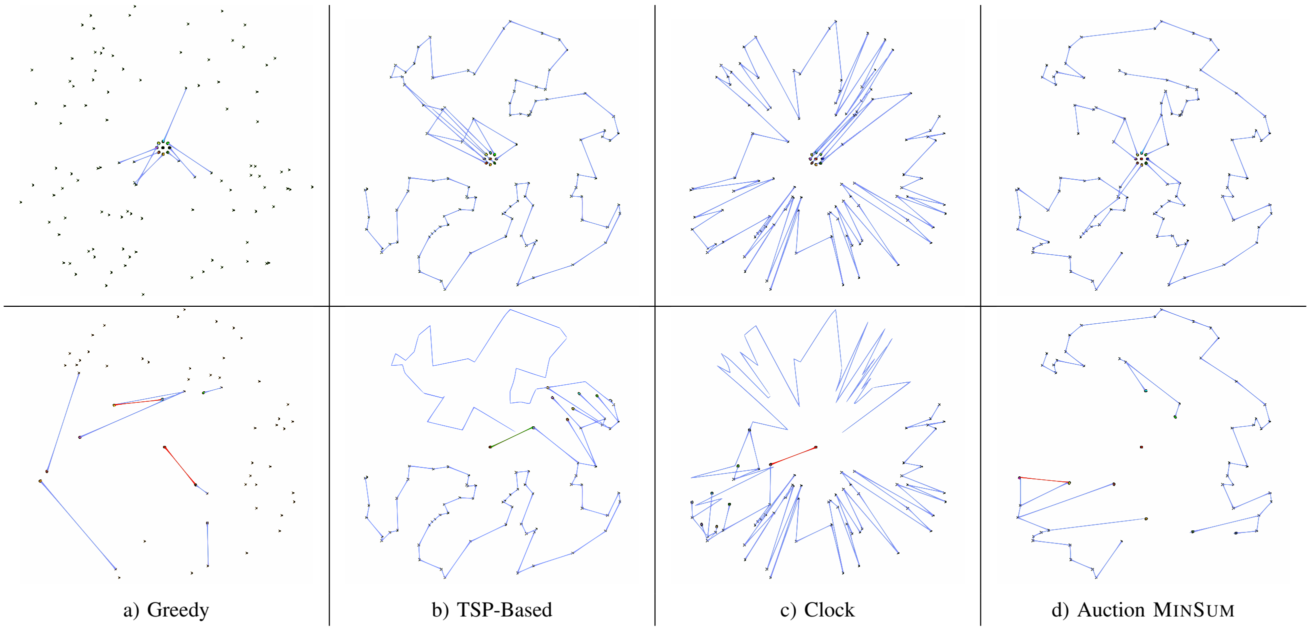

Fig. 4. Two snapshots of each algorithm in action, at initial (top) and intermediate (bottom) mission execution. Lines indicate task ordering.

consumption, mission timespan, and average task completion time. We have used realistic simulations (Player/Stage based) in order to test our algorithms and evaluated these metrics in scenarios with uniformly distributed random tasks around a central, static base. The penalty inflicted by the communication constraints is within one order of magnitude in all studied cases, and increases in most occasions with the inverse of the communication range. These findings show that effective multi-robot routing can be achieved even under limited communication range with moderate loss compared to the case of infinite communication range.

We have identified weaknesses and strengths of our algorithms with respect to these metrics and possible venues for improvement. In particular, the TSP-Based and Clock algorithms exhibit small sensitivity to the actual communication range, which makes them appropriate for situations where the actual communication range is not known in advance. However, our attempts at using auctions to optimize particular metrics did not achieve competitive results; further work is needed in order to take full advantage of the ability to schedule robot clusters in parallel. Some runs of the auction algorithm indicate that improvements over the TSP-based and Clock ones are possible, as auctions take into account explicitly the team performance metric, as well as the distribution of targets. Nevertheless, to exploit successfully this advantage some degree of control over the topology of the underlying connectivity spanning tree is necessary.

Beyond the improvement modifications suggested by the presented experiments, future work will address realistic considerations, such as completeness in the presence of complex obstacles (*culs-de-sac*, large obstacles), structured and dynamic task distribution (areas with clustered targets, dynamically generated tasks), support of multiple static or even mobile bases, and providing MANET support to mobile uncontrolled units (e.g. human teams in a disaster scenario).

VIII. ACKNOWLEDGMENTS

This work was partially supported by the Spanish project DPI2006-07928, the European project IST-1-045062-URUS-STP, a CAI/CONSI+D Europa grant awarded to A. R. Mosteo, and a Marie Curie International Reintegration Grant MIRG-CT-2006-044980 awarded to M. G. Lagoudakis.

REFERENCES

[1] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–378, 2005.

[2] M. N. Rooker and A. Birk, “Communicative exploration with robot packs,” *Lecture Notes in Artificial Intelligence*, pp. 267–278, 2006.

[3] A. Wagner and R. Arkin, “Multi-robot communication-sensitive reconnaissance,” in *ICRA*, 2004, pp. 4674–4681.

[4] J. Vazquez and C. Malcolm, “Distributed multirobot exploration maintaining a mobile network,” in *IEEE International Conference on Intelligent Systems*, 2004, pp. 113–118.

[5] N. Kalra, D. Ferguson, and A. Stentz, “A generalized framework for solving tightly-coupled multirobot planning problems,” in *ICRA*, 2007.

[6] I. Chlamtac, M. Conti, and J. J.-N. Liu, “Mobile ad hoc networking: imperatives and challenges,” *Ad Hoc Networks*, vol. 1, no. 1, 2003.

[7] L. Quin and T. Kunz, “On-demand routing in MANETs: The impact of a realistic physical layer model,” in *ADHOC-NOW*, 2003, pp. 37–48.

[8] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig, “The generation of bidding rules for auction-based robot coordination,” in *Proceedings of the 3rd International Multi-Robot Systems Workshop*, March 2005.

[9] M. Lagoudakis, V. Markakis, D. Kempee, P. Keskinocak, S. Koenig, C. Tovey, A. Kleywegt, A. Meyerson, and S. Jain, “Auction-based multi-robot routing,” in *Robotics: Science and Systems*, 2005.

[10] D. Tardioli and J. L. Villarroel, “Real time communications over 802.11: RT-WMP,” in *Mobile Ad-Hoc and Sensor Systems*, 2007.

[11] D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel, and L. Montano, “Enforcing network connectivity in robot team missions,” 2007, under review.

[12] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, “Concorde TSP solver.” [Online]. Available: http://www.tsp.gatech.edu/concorde.html

[13] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *International Conference on Advanced Robotics*, 2003, pp. 317–323.