Effects of Communication Restriction on Online Multi-robot Exploration in Bounded Environments

Elizabeth A. Jensen and Maria Gini

Abstract In multi-robot exploration for real-world applications, the amount of communication available in a given environment can vary greatly, and has significant effects on the performance of the robot team. We compare algorithms which use three communication models, from unlimited global communication to unplanned, semi-random encounters between robots; and we evaluate each algorithms' performance with different sized robot teams in three environments. We provide simulation results showing how the algorithms perform under optimal conditions, and discuss how real-world situations impact performance.

1 Introduction

As robot capabilities improve and their costs decrease, they become more viable and appealing for use in real-world applications [4, 12]. Multi-robot exploration and coverage systems are of particular interest for applications ranging from disaster relief to deep sea salvage expeditions to lunar missions. While each of these scenarios varies in urgency and danger, they all share the common goal of reaching full coverage—leaving no accessible part of the environment unexplored [11]—from a starting point in which the environment is unknown. With the environment unknown prior to the robot exploration, there can be no advance planning, and instead decisions on where to explore must be made within the environment. While these decisions are well studied for single robot systems [4, 7], using multiple robots adds complexity in how the robots coordinate and what and how necessary information is shared amongst them.

E. A. Jensen (✉) · M. Gini
University of Minnesota, Minneapolis, MN, USA
e-mail: jense924@umn.edu

M. Gini
e-mail: gini@umn.edu

N. Correll et al. (eds.), *Distributed Autonomous Robotic Systems*,
Springer Proceedings in Advanced Robotics 9,
https://doi.org/10.1007/978-3-030-05816-6_33

One of the key components of multi-robot coordination, and a major contributor to the complexity of the system, is communication, which can be assumed to be global and unlimited in simulation, but is much more variable and less reliable in a real-world scenario. This is particularly true of disaster scenarios, in which communication systems fail [12, 29]. Though temporary systems are often quickly set up, these don't necessarily penetrate the rubble of collapsed buildings, so algorithms for multi-robot exploration in these situations must be able to accommodate the limited and variable communication available to them. There is a growing body of research on algorithms that limit communication [8], but there are many algorithms which assume global or unlimited communication with effective approaches that can be adapted to work in communication restricted situations. We previously developed algorithms that operate under significant communication restrictions [15, 16], but must acknowledge that limiting communication slows the exploration, which does not mesh well with the urgency of the situation. In order to improve the coverage time of our algorithms, we compared performance with two algorithms that use vastly different communication and movement models. We then developed modified versions of our algorithm to incorporate certain features without relaxing our communication restrictions. We show that these modifications allow our algorithm to perform on par with the comparison algorithms in simulation, and that the modifications are also still functional in real-world scenarios through physical robot experiments.

While communication plays a large role in how multi-robot exploration algorithms perform, there are additional factors that impact that performance. In particular, we observe that the structure of the environment is important. We focus on bounded environments, which include any environment with a perimeter the robots can discern. If the environment is a large open arena or gym, with no obstacles, the robots' movement patterns can have a huge impact on the speed of coverage. With randomly placed large obstacles, the flow of exploration changes and the number of robots has a larger effect on the time to full coverage. In a highly structured office environment, with closely spaced doorways and branching points, the number of robots has more impact on the speed of coverage than the pattern of movement. We use the same algorithms to show the changes in exploration paths in each environment using different sized teams.

The primary contribution of this paper is an analysis of the effect that the amount of available communication has on online multi-robot coverage algorithm performance. We compare three algorithms with different communication models in three types of environments using multiple performance metrics. We show how performance can be improved, without sacrificing coverage or relaxing communication restrictions, through changes to movement patterns, by developing two new modifications to our algorithm. We provide simulation results for all algorithms to support the conclusions of the analysis. We also show that the modified algorithms exhibit the same performance improvements in physical robot experiments as in simulation.

2 Related Work

Multi-robot systems have become more popular over the years, as the cost and size of components have decreased. They also have advantages over single robot systems in terms of cost, efficiency and robustness [12]. While a multi-robot system is often composed of robots that are individually less capable than a single robot system, the inherent redundancy of the multi-robot system makes it more robust [7], and thus more accommodating of a changing and dangerous environment. This robustness to individual and partial failures is particularly important for unknown environments, in which attrition is more likely [24].

Many multi-robot exploration and coverage algorithms were not explicitly designed to handle the variability of real-world situations, but they do handle many of the necessary aspects, and are often used for comparison, particularly in simulations. This results in a need to assess the algorithms' applicability for real-world scenarios to make the comparisons meaningful. Online coverage algorithms often include expectations that a large (sometimes unlimited) number of robots are available, that global maps are available or created during the exploration, and that communication is available throughout the environment. These requirements cannot be satisfied in all real-world scenarios, and algorithms used in such situations must take these limitations into account.

The expectation of having a large number of robots available usually stems from the type of coverage desired. Gage [11] proposed three types of coverage—blanket, barrier, and sweep. Blanket coverage provides simultaneous coverage of the entire area; the goal of such algorithms is to maximize the area covered [17, 20]. Barrier coverage sets up a perimeter around an area to keep track of changes on the borders. Many early coverage algorithms focused on surveillance, and thus used blanket and barrier coverage, but they then settled for incomplete coverage when robot numbers were low [3, 29]. Using sweep coverage instead, the robots make a pass over the environment and ensure every point has been seen by at least one robot, but don't stay in any one location, instead moving progressively through the environment [26, 28], sometimes doing a patrol or repeated coverage [1, 10] to provide additional information in a dynamic environment. Using sweep coverage, a small team can provide complete coverage, without having to be concerned with the number of robots available in that situation [4, 15, 33].

To coordinate the team, redundancy and robustness lead to distributed rather than centralized systems, though centralization on a small scale has advantages. For example, in Stump et al. [31], a single robot is used as a base station, while the other robots form a communication bridge as they move. Similarly, in Rekleitis et al. [26] one robot acts as a stationary beacon for another robot, thus reducing odometry error of the moving robot. These approaches assume sophisticated sensors, which are expensive and not always available. The use of such sensors does enable the centralized approach to create a global map, which can be used to direct the movements of the robots [5, 25, 34]. Some work assumes that if the robots split up they will be able to make perfect maps of their explorations and it will be trivial to merge these when the team regroups [14, 18, 33], but this is difficult in practice [6, 30].

The use of maps is generally infeasible in a disaster scenario, and maps are not available for the scenarios in which the task is to explore a new environments. While many coverage algorithms assume that the environment is known, in part because they are used for surveillance, in our scenarios, there are no pre-existing maps. Thus, any algorithm that depends on having a map in order to complete the exploration will have difficulty achieving complete coverage. In addition, it is preferable to use sophisticated sensors for detecting points of interest in the environment, rather than for navigation or coordination, as that can be done with less powerful sensors [27].

The last restriction common to real-world scenarios, but often unlimited in online coverage algorithms, is communication range and bandwidth [2]. Nearly all centralized systems assume that individual robots can communicate directly with the central controller [27]; and algorithms that create global maps assume that the robots communicate globally [33]. Since communication systems are often down in the aftermath of a disaster [12, 24], or aren't available deep under water or on another planet, we must focus instead on achieving coverage with limited communication [8]. Most robots can provide some means of local communication themselves, such as Wi-Fi [17, 29, 32] or line-of-sight methods [10, 16], which can be used to direct the exploration.

In a disaster scenario, we want the robots to complete exploration as quickly as possible, and minimizing repeated coverage of any area in the initial exploration pass helps reduce exploration time. However, repeated coverage after full exploration can be useful to monitor the environment [10, 22]. With a small team of robots, it is common to use beacons to mark explored areas [4, 15]. This is inspired by animal behavior [3, 21, 23]. These beacons enable the robots to reduce repeated coverage and explore larger environments without having to create maps. Though the beacons may be destroyed or moved, they are more easily replaced than robots and they greatly improve efficiency [4, 16] using only local communication.

Though we are considering a broader range of situations than just disaster scenarios in this paper, the related restrictions narrow the scope of relevant metrics of algorithm performance. Yan et al. [35] produced an in-depth evaluation of metrics for multi-robot exploration. We measure each algorithms' performance using four metrics from previous research—total distance traveled, time to coverage, time to return (total time taken from start until robot returns to start location), and rate of coverage.

3 Algorithms

Yan et al. [35] also discussed the difficulties inherent in assessing and comparing multi-robot algorithms, due the high number of variables involved in each system. These variables include the type, size, capability, and number of robots, centralized versus distributed systems, as well as many facets of the environment. With our focus on comparing the impact of communication on performance, we selected distributed algorithms intended for use in bounded environments using ground robots which achieve complete coverage for comparison. We can thus be more confident that differences in performance are influenced by communication.

3.1 Rolling Dispersion Algorithm (RDA)

Our algorithm, RDA [15], is a distributed exploration algorithm for a group of robots in an unknown environment. Two of its key requirements are that the robots stay together as a group by utilizing the communication signal intensity, and that the robots carry and deposit beacons to mark explored areas or the return path. Marking the return path is uncommon among exploration or coverage algorithms, but it was required in order to ensure that information makes it back to human operators waiting outside the environment, as well as providing direction for the robots to exit when the exploration is complete, for use in later situations. Returning to the entry location also helps us provide guarantees that the exploration is complete.

In RDA, all robots are either explorers, which move into the frontier as long as they can still communicate with at least one other robot, or sentries, which guard intersections and way-points when the communication strength wanes. There is an initial dispersion phase, in which all but one robot (at the entrance) move away from each other as much as possible, trying to achieve maximum area coverage while remaining connected. This can cause issues when there are too many robots at the entrance, as the robots navigate around each other or move down the same path for some distance together, until intersections are found. Once the robots have reached maximum dispersion, however, the control mechanism shifts to Depth-First-Search (DFS), completing a path of exploration before moving to other paths, but may require some robots to abandon their frontier or sentry location (leaving behind a beacon to bring the explorers back later) in order to complete that path. Beacons are also used to mark loops and prevent repeated exploration.

RDA was designed for search and rescue applications, which included minimal communication, limited team size, unknown environments and fault-tolerance. While it provides complete coverage of the environment, and success even with attrition, it is not quick to complete exploration, and information gathered may not be passed to the outside until the robots return to the entrance. The constraints provide a solid foundation, but we determined that we could modify RDA to make it more efficient. We looked at other algorithms to determine what features would be most beneficial in our modifications, without having to relax our communication requirements, since those are the basis of RDA's full coverage guarantee.

3.2 Multi-robot Depth-First-Search

Multi-Robot Depth-First-Search (MR-DFS) presented by Brass et al. [4] is similar to RDA, in that both are based on DFS, but MR-DFS implements DFS with no communication requirements between robots except at branching points in the graph or tree. It is assumed that the distance along an edge between vertices is uniform and can be traversed in a single time-step. At any vertex, the robots divide evenly among the edges. The robots leave beacons at each vertex to inform subsequent robots of which direction the earlier robots have gone, so that the following robots can take different paths. This does assume that the robots can determine and convey the information about what direction they have gone, and that the beacons can pass on that directional information correctly. This differs from RDA's use of beacons, since it requires directional information, and is less robust if things shift. An edge is not considered covered until it has been traversed at least once in each direction, but once marked as covered, the robots do not traverse it again, except as a way to return.

MR-DFS allows the robots to split up and completely lose contact with one another. This leads to parts of the environment being unnecessarily explored multiple times, or robots traveling down a path where help might be needed, only to arrive after all the paths have been fully explored, leading to inefficiencies in distance traveled and energy consumption. In contrast, RDA is slower to reach full coverage because the robots must stay in communication with each other, but travel less distance overall, because they only travel paths when required. Areas that are redundantly covered due to the need for multiple robots to explore long, multi-branching paths are redundantly covered in both algorithms. MR-DFS's lack of communication range constraints is appealing for real-world applications, but the redundant exploration slows its rate of coverage in certain environments.

3.3 BoB

Viet et al. [33] present BoB, which is a multi-robot algorithm to achieve online complete coverage using a combination of the boustrophedon motion plan, in which the robots move along straight paths and then double back right beside the original path, much like an ox plowing a field (from which the motion plan takes its name), and backtracking using Greedy A*. The boustrophedon motion ensures that the robots cover everything in their area, and the backtracking allows them to quickly move to an open frontier when they reach an end point in their area. It does rely on the robots being able to globally communicate their locations and create maps of their areas as they explore, but the map is not needed in advance, making it a good comparison alongside RDA and MR-DFS. Though Viet et al. [33] discuss using only local interaction, their algorithm requires each robot to know the location of all other robots at each decision point, to make the Greedy A* backtracking possible. Global communication is not as easily achieved in real-world scenarios such as deep sea salvage operations, or search and rescue in the aftermath of an earthquake, but the motion pattern and use of multiple starting locations make it the fastest to achieve full coverage.

3.4 Modifications to RDA

As a result of reviewing the previous research, particularly algorithms that use very different forms of communication, we developed two modifications to RDA to improve its performance. MR-DFS allows the robots to move further apart than in RDA, because there is no direct communication over distance between robots. Thus, our first extension was to increase the communication range, creating RDA-EC (extended communication). In many environments, the effect of this modification is only that the robots can move further apart before needing to call for additional assistance. However, because the extended communication range also means that the robots move outside of sensor range of each other, we also added to the intersection detection rules. RDA-EC uses the obstacle detection sensor range in addition to considering the communication range to detect intersections, because in environments with wide open areas the robots were missing small parts of the environment when using only the communication range. An additional drawback to the extended communication range is that it requires a stronger communication device, and more power to run that device, but we kept the range short enough to remain on the inexpensive end of the communication device market.

We found the multiple starting locations in BoB appealing, especially since RDA showed difficulty at the start when too many robots were fighting for space to move, and also ended up doing redundant coverage in the initial area. We created the second modification, RDA-MS (multi-start), in which each starting location holds a separate team, which then runs the regular RDA algorithm. Though we can’t always be certain that there will be multiple locations from which to start the robots, it is common to have multiple points of entry [24]. BoB allows the robots to communicate with any other robot, no matter their entry point, and any robot may cross another robot’s path to reach a new frontier. In contrast, RDA-MS robots only consider the robots from their starting point as part of their team, and if they encounter robots or beacons from other teams, these are treated as closing loops, and the meeting point is blocked off so that the robots from each team cannot pass in the other team’s territory. This maintains RDA-MS’s coverage guarantees, while not compromising the communication restrictions or requiring RDA-MS to make and share maps.

4 Simulation Experiments

We used the ROS/Stage [13] simulator using the Pioneer robot model with 16 sonar sensors for the mobile robots, and a modified Pioneer robot for the beacons. We used three different environments—the random obstacles of the Cave in Fig. 1, the structured NHH environment with few intersections in Fig. 2, and the structured Hospital Section with many intersections and loops in Fig. 3. We used 1 and 4 robots in the NHH environment (more robots were detrimental to performance), and 1, 4, and 8 robots in each environment in the Cave and Hospital Section, with 15 runs per algorithm/robot combination, and using 4 different starting configurations for BoB and RDA-MS.

Table 1 shows the performance metrics for each algorithm in each environment. Both BoB and MR-DFS discretize the environment such that the robots move a uniform distance per time step. The distance is a count of the graph vertices reached (15 for Cave, 21 for NHH, and 130 for Hospital Section), and each edge traversal takes one time step, or 20 s (robots move at roughly 0.5 m/s). Graphs showing the rate of coverage for each algorithm are shown in Figs. 4, 5, and 6.

With a single robot, the DFS algorithms perform the same and are thus consolidated to a single line. In the Cave and Hospital Section environment, using a single robot, BoB performed slightly better on all metrics, but in the NHH environment, with long corridors and no room for BoB to cut corners, all algorithms performed the same with a single robot. In comparing the performance of the multi-robot runs, we expect BoB and RDA-MS (multi-start) to perform the best, because starting in separate locations reduces interference and allows the robots to cover more of the environment before interacting with each other (if at all). This greatly reduces the time to provide full coverage.

In the Cave and NHH environments, BoB and RDA-MS take the same amount of time to achieve coverage, but RDA-MS has a slightly longer return time and distance traveled because the robots must return along the same path, while the BoB robots can plan the shortest path back to their starting locations. However, in the Hospital Section environment, though BoB is best in all categories on 4 robots, RDA-MS is similar in distance and return time. This is because the BoB robots will cross areas already covered on the way towards a frontier, only to get partway there and then be informed that the area has been covered. With 8 robots in the Hospital section, RDA-MS still has a longer total distance and return time, but averages a faster time to complete coverage. This was due to the robots operating individually and not entering areas covered by other robots. Additionally, the differences between BoB and RDA-MS are not statistically significant (using a threshold of $p < 0.05$), which means the differences are within random variation. Thus, the fact that RDA-MS can match and in some cases outperform BoB, which requires global communication and perfect mapping/localization, makes RDA-MS more robust and appealing.

Fig. 1 Cave environment, randomly sized and placed obstacles (15 vertices)

Fig. 2 NHH environment, accessible area in green (21 vertices)

Fig. 3 Hospital Section environment (130 vertices)

As expected, RDA does not do well against the comparison algorithms, but it does result in lower distance traveled than MR-DFS with the same number of robots in all but the NHH environment. Comparing MR-DFS and RDA-EC, however, shows that RDA-EC achieves a total distance traveled of 80% of MR-DFS, even though it takes longer to achieve coverage and return in most cases. In the Hospital Section with 4 robots, RDA-EC returns all the robots to the starting location faster than MR-DFS, because the robots explore and return along their paths earlier in the exploration

Table 1 Travel distance in number of edges traversed, and time to coverage and return for each algorithm and number of robots in each environment. Values in bold denote the best for that metric in that section

Environment		Cave		
# of robots	Metric\Alg.	Travel distance (edges)	Time to coverage (s)	Time to return (s)
1	BoB	**20**	**340**	**380**
	DFS	31	320	600
4	BoB	**24**	**80**	**120**
	MR-DFS	51	120	280
	RDA	65	880	1000
	RDA-MS	37	**80**	180
	RDA-EC	33	120	240
8	BoB	**28**	**40**	**80**
	MR-DFS	67	80	200
	RDA	63	360	480
	RDA-MS	**28**	**40**	**80**
	RDA-EC	49	140	240
Environment		NHH		
# of robots	Metric\Alg.	Travel distance (edges)	Time to coverage (s)	Time to return (s)
1	BoB	**41**	**560**	**820**
	DFS	**41**	**560**	**820**
4	BoB	**46**	**120**	**220**
	MR-DFS	60	180	300
	RDA	64	700	860
	RDA-MS	47	**120**	**220**
	RDA-EC	48	200	340
Environment		Hospital Section		
# of robots	Metric\Alg.	Travel distance (edges)	Time to coverage (s)	Time to return (s)
1	BoB	**205**	**4080**	**4100**
	DFS	259	4860	5180
4	BoB	**238**	**960**	**1260**
	MR-DFS	396	1420	2080
	RDA	381	2100	2660
	RDA-MS	254	1100	1340
	RDA-EC	324	1500	2020
8	BoB	**225**	580	**640**
	MR-DFS	685	1200	1800
	RDA	486	2160	2580
	RDA-MS	240	**540**	760
	RDA-EC	528	1100	1620

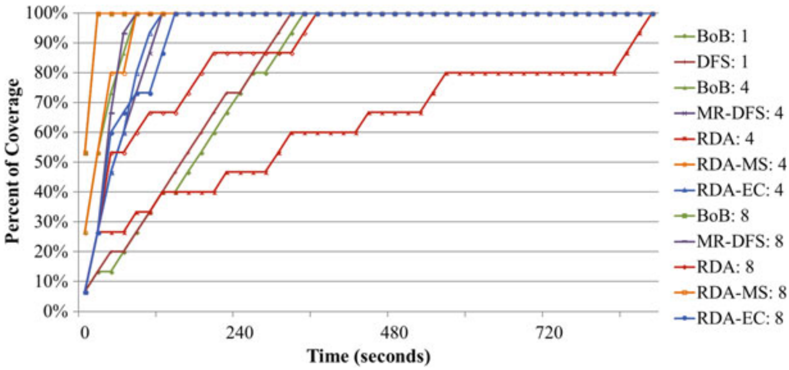

Fig. 4 Rate of coverage for the Cave environment, with 1, 4, and 8 robots. BoB and RDA-MS are identical. RDA and RDA-EC are identical up to 300 seconds

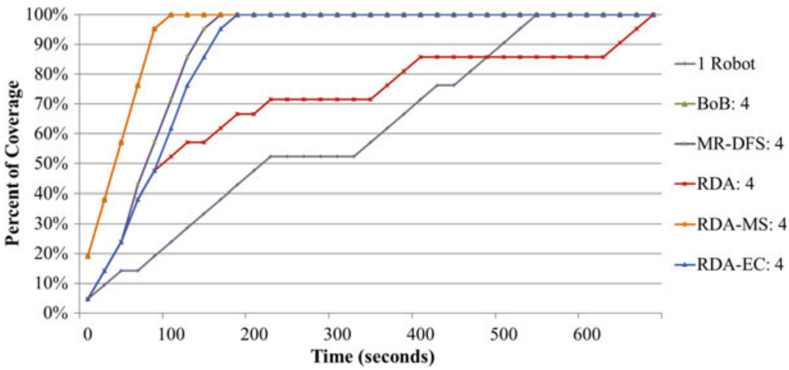

Fig. 5 Rate of coverage for the NHH environment, using 1 and 4 robots. BoB and RDA-MS again have identical rates of coverage

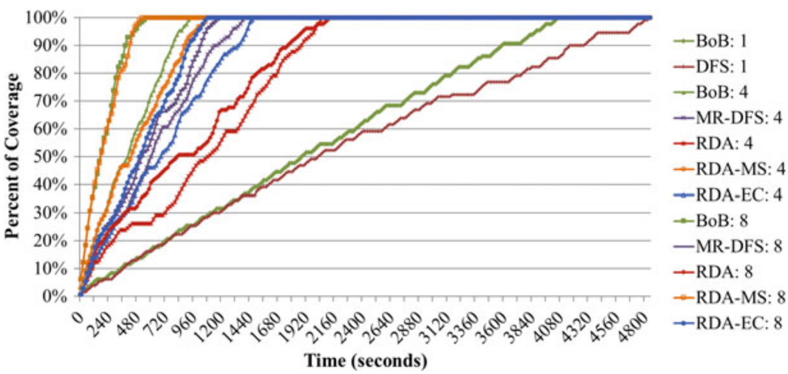

Fig. 6 Rate of coverage for the Hospital Section environment, using 1, 4, and 8 robots

phase, while MR-DFS ends up sending robots to return along side paths to ensure every edge is traversed in each direction, which adds to the return time. In addition, when using 8 robots in the Hospital Section environment, RDA-EC performs better than MR-DFS on all measures, because it avoids repeated coverage when at all possible. While RDA-EC performs worse than BoB and RDA-MS, it performs better than MR-DFS with $p < 0.05$. When intersections are close together, as in the Cave and Hospital Section environments, RDA-EC does not scale as well as RDA, with the best speed-up at 4 robots.

5 Physical Robot Experiments

We also completed physical robot experiments using RDA, RDA-MS, and RDA-EC in the NHH environment (see Fig. 2) to confirm the trends seen in simulation would hold under real-world conditions. For these experiments, we used the LEGO EV3 robots [19], running the ev3dev OS [9]. The robot has a bump sensor, ultrasonic sensor, and color sensor to interact with the beacons (Wi-Fi enabled motes were not available). A robot and three beacons are shown in Fig. 7. A robot shares the beacon's state with neighboring robots when the beacon is deposited or updated, and that information is passed on to the rest of the team.

Our experiments consisted of 10 runs each using 1, 2, and 3 robots for each of the algorithms (5 runs each for 2 starting configurations with 2 and 3 robots for RDA-MS). As expected, based on the simulation results, RDA-MS performed the best and RDA the worst on all metrics. The rate of coverage trends also match the simulations (see Fig. 8). In RDA [15], as the number of robots increases, the number and length of plateaus in the rate of coverage also increases, due to the need to fully return along a path before moving down a different branch. We do not see this as much in RDA-EC, due to the longer communication range, which allows the robots to move further apart. RDA-MS shows none of this because the robots are completely independent. With larger environments and more robots, we expect to see more plateaus.

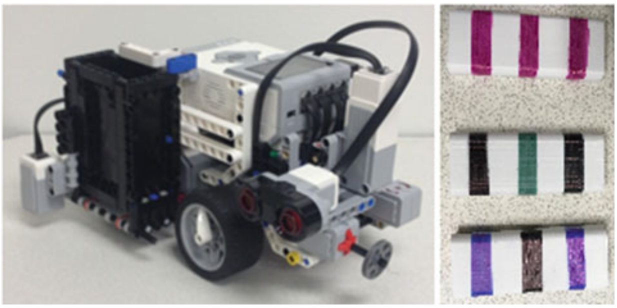

Fig. 7 The LEGO EV3 robot and three of the beacons used in experiments

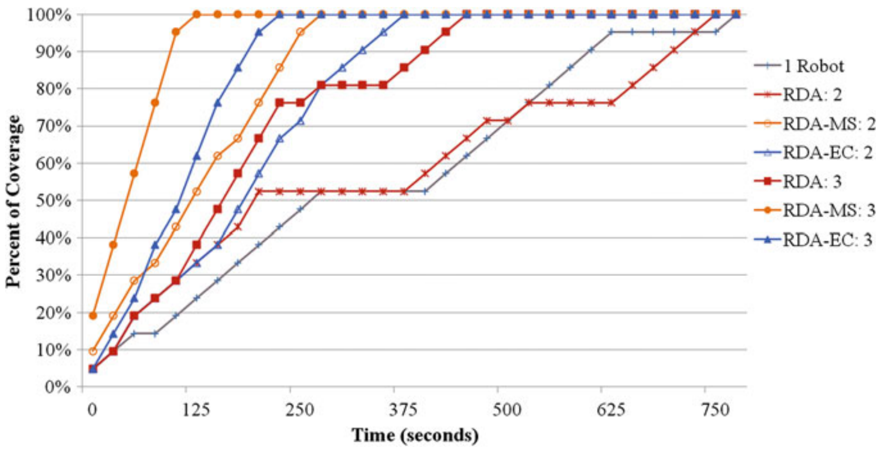

Fig. 8 Results of running 1, 2, and 3 Lego EV3 robots using RDA, RDA-EC, and RDA-MS in the NHH environment. Results are averaged over 10 runs

6 Conclusions

We have discussed how communication impacts online multi-robot coverage algorithms' viability for real-world scenarios, and have shown how communication can affect the algorithms' performance. We also developed two modifications to RDA that can maintain the communication restrictions while matching the performance of the respective comparison algorithms. The experimental results obtained in simulation and with physical robots showed that both modifications compete in performance with their respective comparison algorithms, and are significant improvements over the RDA algorithm. In the future we will investigate the impact that the environmental structure has on performance, and on the scalability of each algorithm. We will also consider ways of combining RDA-EC and RDA-MS.

References

1. Ahmadi, M., Stone, P.: A multi-robot system for continuous area sweeping tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1724–1729 (2006)
2. Amigoni, F., Banfi, J., Basilico, N.: Multirobot exploration of communication-restricted environments: a survey. IEEE Intell. Syst. **32**(6), 48–57 (2017)
3. Batalin, M.A., Sukhatme, G.S.: The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. IEEE Trans. Robot. **23**(4), 661–675 (2007)

4. Brass, P., Cabrera-Mora, F., Gasparri, A., Xiao, J.: Multirobot tree and graph exploration. IEEE Trans. Robot. **27**(4), 707–717 (2011)
5. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Trans. Robot. **21**(3), 376–386 (2005)
6. Chand, P., Carnegie, D.A.: Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots. Robot. Auton. Syst. **61**(6), 565–579 (2013)
7. Choset, H.: Coverage for robotics—a survey of recent results. Ann. Math. Artif. Intell. **31**(1), 113–126 (2001)
8. Doriya, R., Mishra, S., Gupta, S.: A brief survey and analysis of multi-robot communication and coordination. In: Proceedings of the International Conference on Computing, Communication and Automation (ICCCA), pp. 1014–1021 (2015)
9. ev3dev.org: ev3dev documentation (2017). http://www.ev3dev.org/docs/
10. Fazli, P., Davoodi, A., Mackworth, A.K.: Multi-robot repeated area coverage. Auton. Robots **34**(4), 251–276 (2013)
11. Gage, D.W.: Command control for many-robot systems. In: Proceedings of the 19th Annual AUVS Technical Symposium (1992)
12. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**(12), 1258–1276 (2013)
13. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics (ICAR), pp. 317–323 (2003)
14. Hazon, N., Mieli, F., Kaminka, G.A.: Towards robust on-line multi-robot coverage. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1710–1715 (2006)
15. Jensen, E.A., Gini, M.: Rolling dispersion for robot teams. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 2473–2479 (2013)
16. Jensen, E., Lowmanstone, L., Gini, M.: Communication-restricted exploration for search teams. In: Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems (DARS), pp. 17–30 (2016)
17. Kantaros, Y., Zavlanos, M.M.: Distributed communication-aware coverage control by mobile sensor networks. Automatica **63**, 209–220 (2016)
18. Latimer IV, D., Srinivasa, S., Lee-Shue, V., Sonne, S., Choset, H., Hurst, A.: Towards sensor based coverage with robot teams. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol. 1, pp. 961–967 (2002)
19. LEGO: About ev3. https://www.lego.com/en-us/mindstorms/about-ev3. Accessed 26 Sep 2017
20. Ludwig, L., Gini, M.: Robotic swarm dispersion using wireless intensity signals. In: Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS), pp. 135–144 (2006)
21. Mamei, M., Zambonelli, F.: Pervasive pheromone-based interaction with RFID tags. ACM Trans. Auton. Adapt. Syst. (TAAS) **2**(2) (2007)
22. Mitchell, D., Corah, M., Chakraborty, N., Sycara, K., Michael, N.: Multi-robot long-term persistent coverage with fuel constrained robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1093–1099 (2015)
23. Mizuguchi, T., Sugawara, K.: Proportion regulation in task allocation systems. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **89**(10), 2745–2751 (2006)
24. Murphy, R.R.: Disaster Robotics. The MIT Press (2014)
25. Nieto-Granda, C., Rogers III, J.G., Christensen, H.I.: Coordination strategies for multi-robot exploration and mapping. Int. J. Robot. Res. **33**(4), 519–533 (2014)
26. Rekleitis, I., New, A.P., Choset, H.: Distributed coverage of unknown/unstructured environments by mobile sensor networks. In: Multi-Robot Systems. From Swarms to Intelligent Automata Volume III: Proceedings from the 2005 International Workshop on Multi-Robot Systems, pp. 145–155 (2005)

27. Robin, C., Lacroix, S.: Multi-robot target detection and tracking: taxonomy and survey. Auton. Robots **40**(4), 729–760 (2016)
28. Rogge, J.A., Aeyels, D.: A strategy for exploration with a multi-robot system. Inf. Control Autom. Robotics **24**, 195–206 (2009)
29. Rooker, M.N., Birk, A.: Multi-robot exploration under the constraints of wireless networking. Control Eng. Pract. **15**(4), 435–445 (2007)
30. Saeedi, S., Paull, L., Trentini, M., Seto, M., Li, H.: Map merging for multiple robots using Hough peak matching. Robot. Auton. Syst. **62**(10), 1408–1424 (2014)
31. Stump, E., Jadbabaie, A., Kumar, V.: Connectivity management in mobile robot teams. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1525–1530 (2008)
32. Sweatt, M., Ayoade, A., Han, Q., Steele, J., Al-Wahedi, K., Karki, H.: WiFi based communication and localization of an autonomous mobile robot for refinery inspection. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4490–4495 (2015)
33. Viet, H.H., Dang, V.H., Choi, S., Chung, T.C.: BoB: an online coverage approach for multi-robot systems. Appl. Intell. **42**(2), 157–173 (2015)
34. Wurm, K.M., Stachniss, C., Burgard, W.: Coordinated multi-robot exploration using a segmentation of the environment. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1160–1165 (2008)
35. Yan, Z., Fabresse, L., Laval, J., Bouraqadi, N.: Metrics for performance benchmarking of multi-robot exploration. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3407–3414 (2015)