Active Team Management Strategies for Multi-robot Teams in Dangerous Environments

Geoff Nagy and John Anderson(✉)

Autonomous Agents Lab, University of Manitoba, Winnipeg, MB R3T 2N2, Canada
{geoffn,andersj}@cs.umanitoba.ca

Abstract. Cost-effectiveness, management of risk, and simplicity of design are all arguments in favour of using heterogeneous multi-robot teams in dangerous domains. Robot losses are expected to occur and the loss of useful skills means that replacement robots—either released into the environment or previously lost and rediscovered—must be recruited for useful work. While teams of robots may eventually encounter replacements by chance, more active search strategies can be used to locate them more quickly, either to complete a single task or join a team. These searches, however, must be balanced with existing tasks so that the team can still perform useful work in the domain. This paper describes additions that we have made to an existing framework for managing dynamic teams in dangerous domains in order to support this goal.

1 Introduction

There are many advantages to using autonomous teams of heterogeneous robots. With skills spread out among multiple team members, no single unit needs to possess every capability a mission requires. This reduces the overall cost and complexity of the robots [1]. This is beneficial in dangerous environments such as Urban Search and Rescue (USAR) or minefield clearing. In such settings, risk can be mitigated by deploying fewer elaborate units and more expendable ones.

In any situation where robot damage or loss can occur, team performance will degrade as robots are lost or destroyed. If possible, losses can be compensated for by releasing replacement robots into the environment at periodic intervals during a mission. Existing robot teams should be prepared to accept new members, which may either be replacement robots or rediscovered robots that were previously separated from a team. Robot teams should also recognize when it is necessary to strengthen their team by searching for and integrating other robots as required. Successfully integrated robots should also share information with the new team [14]. Conversely, robot teams may choose not to integrate additional members due to the costs associated with managing a larger team.

Robots should expend effort searching for other robots possessing skills that are valuable to the team or the current tasks at hand. This becomes increasingly necessary as robot teams suffer increased losses, particularly for rare or high-value skills. Aggressive search strategies (such as exploration of a previously unexplored area, or searching a location suspected to contain a useful robot)

M. Mouhoub and P. Langlais (Eds.): Canadian AI 2017, LNAI 10233, pp. 385–396, 2017.
DOI: 10.1007/978-3-319-57351-9_43

are more likely to result in the successful integration of a valuable skill, but at the cost of performing less immediately useful work. Strategies that only rely on random encounters with other robots will allow more work to be completed [7], but are less likely to result in the acquisition of needed skills. These types of recruitment strategies fall on a spectrum that defines how much effort is expended into actively searching for useful robots (Fig. 1). The middle point of this spectrum would contain strategies such as broadcasting wireless requests to nearby robots without actually attempting to locate them physically.

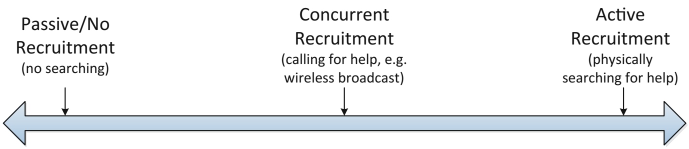

Fig. 1. The recruitment spectrum.

Previous work has resulted in a framework for robotic team management in dangerous domains [7]. We have expanded this framework to include recruitment strategies from across this spectrum. We evaluate our framework in a simulated USAR setting in which robots can be damaged or destroyed. The following sections describe our framework, experimental evaluation, and our results. We begin with a description of related work for comparison.

2 Previous Work

Existing work does not employ many strategies from the recruitment spectrum. It is difficult to form useful comparisons with previous work since many approaches assume an unlimited communications range, or that robots are always close enough to communicate [3,4,10]. Additionally, previous work does not take into account real-world challenges such as communication failures, robot failures, or robot losses in general [5,6,8,12,15]. This section describes previous work that uses active searching or random encounters with other robots to acquire additional team members during the course of a mission.

Krieger and Billeter [9] developed an approach for food-foraging tasks, modelling the behaviour of ants using a large number of homogeneous robots. Although explicit teams were not formed, robots were able to recruit others to known food locations in order to increase the amount of food returned to a central nest. Pitonakova et al. [13] studied active recruitment in more complex foraging missions where resources varied in size and value. This is similar to our work in that certain tasks or robot skills may be more important than others, and the choice of recruitment strategy depends greatly on environmental conditions and the availability of resources. In contrast, we study recruitment in a less abstract domain (USAR) and our work is focused explicitly on long-term team maintenance as well as short-term team goals.

Gunn and Anderson [7] implemented a team-management framework for heterogeneous robot teams in dangerous domains, and considered real-world challenges such as varying communication success rates, and robot failures and losses. This framework was evaluated in a USAR setting, where robots were organized into teams at the onset of each mission and explored the environment to identify as many human casualties as possible within the allotted time. Losses were offset by strategies that allowed teams to acquire new members when they were encountered in the environment by chance, i.e., robots did not explicitly put effort into locating other useful robots. We have made substantial additions to this framework, and have developed recruitment strategies enabling robots to (a) search for and assign tasks to others on an individual basis without the intervention of a team leader, and (b) recruit additional team members more permanently when a crucial skill set is missing from the current team. Preliminary results were published in [11]. This paper describes the additions we have made to the original framework (points (a) and (b) above, as well as extended heterogeneity through a new robot type and behaviours), and presents original results from two large experiments used to evaluate the completed framework.

3 Methodology

3.1 Framework

To adequately describe our additions, a high-level overview of major framework components is required. Readers interested in a thorough description of the complete framework are directed to [7].

Robots. The underlying assumption in our work is that robots are heterogeneous—they differ in how they move, sense the environment, and have varying storage and computational capabilities. Therefore, no single robot is able to perform every task, and certain robots will be able to complete certain tasks better than others, or may not be able to perform some tasks at all. In our framework, robots are aware of their own individual skills and how well-suited those skills make them for a particular task. The distributed nature of our framework means that robots rely on local decision-making processes based on their knowledge of the environment and their teammates. Communication problems, robot failures, or other challenges mean that this knowledge may not always be up-to-date.

Tasks, Roles, and Teams. A *task* in our framework is a single piece of work which can be assigned to and completed by a robot. Although a particular challenge or outstanding job discovered by a robot may ultimately involve more than one robot, such jobs must be broken up into discrete tasks and individually assigned to team members. However, assigned tasks may involve overlapping goals or objectives without explicitly requiring the involvement of more than one robot, e.g., two robots exploring overlapping areas. In the original version of this framework [7], task distribution was solely the responsibility of team leaders. Our improved framework allows any robot to perform task assignment in special circumstances, described in Sect. 3.3. For example, robot A may pass a particular task to another robot B, if A is already too busy or is incapable of performing the task. Every robot maintains an ordered queue of tasks, ensuring that the most important work is completed first.

In our framework, every robot fills a *role* within a *team*. A role is defined by a set of tasks that a robot in that role would be expected to perform, and is used as a simple heuristic for task allocation to quickly find team members capable of performing a particular task. A robot with the appropriate skill set can also perform a task outside of its current role if necessary (e.g., a firefighting robot could still perform exploratory searches for victims). Robots in our framework possess self-knowledge and are capable of determining their *suitability* for a particular role, represented as an integer that indicates how well their skill set matches the tasks that make up the role.

Roles are fluid, rather than permanent: to compensate for robot losses or acquisitions, robots periodically evaluate their role within their team and change to a role that would be more beneficial if required. As robots attempt to address skill deficiencies by adopting new (possibly non-optimal) roles, there is a risk that some robots will fill roles for which they are ill-equipped, taking them away from tasks for which they are better suited. For example, a single robot separated from its team will realize it is alone and adopt a leadership role (i.e. become a team of one), even if it has limited capabilities for such a role. If a more suitable robot is encountered (either by itself or as part of a larger team), it will cede the leadership role to the other robot as they merge teams.

Team Leader and Task Assignment. The *team leader* is a special role that exists for coordinating team actions. The leader also acts as a repository of team knowledge since other team members are expected to communicate significant locations, results of tasks, and other important knowledge to the leader. The team leader is thus expected to have the most complete global knowledge, including knowledge of team structure. It is important to note that this knowledge is unlikely to ever be complete—the leader will not know precisely when a robot is lost, for example, and communication that would indicate this may be long delayed [7]. Therefore, relying on local decision making is important despite the presence of a leader. Due to the memory and processing required, it is expected that leadership should fall to the most computationally-capable robot on the team. Ideally, such a robot would have ample facilities to handle these responsibilities, but as team structure degrades due to hazards, less-capable robots may temporarily fill this role until a better robot is found.

Role Switch Check. Robots are responsible for evaluating the usefulness of their current role and changing to a more valuable role for the team, if necessary.

This is done based on the robot's (potentially inaccurate) knowledge of the current team structure. Robots with empty task queues perform this check every 30 s and switch to a different role if necessary. The intuition behind this process is that robots with no outstanding work might be better suited for a different role. The sections that follow describe the expansions we have made to the original framework in terms of role management and task assignment.

3.2 Role-Level Recruitment

Our extensions provide robots with the ability to perform searches for other robots to join their team. These may be previously lost robots, replacement robots, or even members of another existing team. We refer to this mechanism as *role-level recruitment*. This recruitment process begins with a leader periodically assessing the current structure of the team (using possibly incomplete knowledge), and taking note of the most important role that is not currently filled. This occurs every 6 min in our implementation. The team leader will then attempt to assign a particular robot on the team with the task of acquiring a robot to fill that role. In cases where a team leader is the only robot on a team (and thus might have many missing roles), the team leader will assign itself with the task of finding another robot to fill the most important missing role. Section 3.4 describes the strategies used to find and recruit other robots.

3.3 Task-Level Recruitment

While role-level recruitment provides additional means for acquiring new team members, our framework also supports the acquisition of robot skills in a less-permanent fashion. We refer to this as *task-level recruitment*, and it allows one robot to search for another to complete a task. This is useful when the original robot is already too busy to perform additional work, or if it encounters a task for which it is poorly-equipped.

Task-level recruitment can be initiated in two ways. In our original framework, if a team leader attempts to assign a particular task to a robot, the task will be rejected if the assignee is already too busy with other work. The team leader will have no choice but to continually attempt to reassign the task elsewhere until it is accepted. In our modified framework, rather than rejecting tasks when too busy, robots will offer to locate another capable robot to complete the task. The team leader will agree to this only if no other robot is able to accept the task for execution. When this occurs, it is the responsibility of the assigned robot to recruit another robot who will accept the task for execution. Task-level recruitment can also be initiated when a robot discovers a task on its own that it is unable to complete. In the original framework, a robot which discovers a task beyond its capabilities would attempt to relay the task to a team leader for reallocation. In our improved framework, if a robot discovers a task for which it is poorly-equipped, it will instead attempt to locate another robot who can execute the task. Importantly, this process does not require the intervention of a team leader, and thus, is robust to leadership failures or limitations in communication.

3.4 Recruitment Strategies

A recruitment strategy defines how robots search for others when additional roles should be filled or tasks need to be completed. Active recruitment approaches will involve physical searches. While such approaches are more likely to result in successful recruitment, this is done at the cost of completing less immediately useful work. Less committed strategies may involve no searching at all (as employed in [7]), or may only require minimal searching efforts such as broadcasting a wireless request for assistance. The next two subsections describe the two recruitment mechanisms we have added to the original framework.

Concurrent Recruitment. This strategy relies only on wireless communication to search for other robots. Concurrent recruitment can be used to acquire additional team members (*role-level* recruitment), or to assign tasks to other robots (*task-level* recruitment) when required. This is done concurrently alongside normal work; recruiting robots are able to execute tasks while also sending recruitment broadcasts. The content of these broadcasts and any subsequent communication varies depending on whether a robot is recruiting for a task or a role. A robot will agree to a role-level request only if (a) its task queue is not full, and (b) if the robot has lost contact with its team or is better suited to the new role. The process for accepting task-level recruitment requests is the same that is used for regular task assignments: if a robot is not too busy, it will accept the task, and will offer to recruit someone else for the task otherwise.

Active Recruitment. *Active recruitment* is a more aggressive strategy that relies on a combination of physical search and wireless broadcasting to locate a new teammate. While actively recruiting, a robot is fully committed to exploring the environment in search of other robots, and executes no other tasks. This increases the chances of encountering useful robots, but prevents immediately useful work from being completed. Physical searching also increases a robot's risk of becoming separated from its team, either by exploring too far or by becoming stuck on debris while exploring.

4 Experimental Evaluation

Our framework was evaluated using USAR environments implemented in Stage [16], a well-known and established multi-robot simulator. It provides facilities that allowed us to abstract issues associated with physical sensors and hardware, since our focus was to evaluate our framework's performance with regards to team management. To reduce bias that may exist in any single environment configuration, we used 3 randomly-generated USAR scenarios. Each environment simulated a large area ($60\,\mathrm{m}^2$) of a collapsed building. Environments contained significant debris, partial rooms, and missing walls, as shown in Fig. 2. The presence of debris (dark gray and light gray squares) contributed to a considerable risk of robots becoming stuck, and helped to simulate real USAR conditions. Our environments also contained 20 human casualties that robots were responsible for locating, as well as 10 *false victims* (gray humans in Fig. 2), which appear to be casualties to simpler robots but require the attention of more advanced robots in order to confirm.

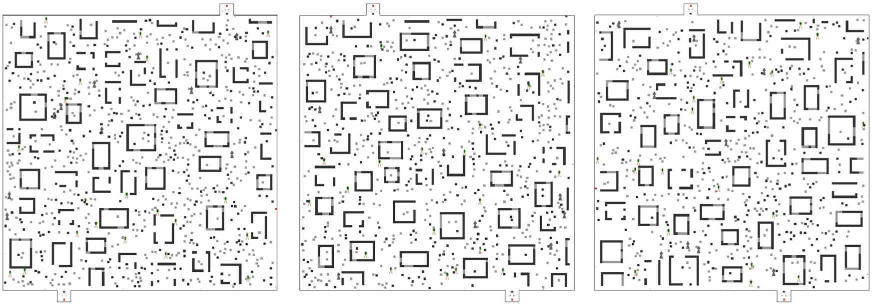

Fig. 2. Overhead views of our 3 randomly-generated USAR environments. Team starting positions are visible at the bottom and top entrances.

We abstracted robot heterogeneity to four types of robots, similar to [2, 7]. The most basic and inexpensive robot was a MinBot, a small robot with very simple victim-sensing capabilities. Its simplicity, low cost, and expendability make it ideal for potential victim discovery and exploration of unknown areas. MinBots are unsuitable as team leaders due to their limited computational and memory facilities, and require more advanced robots to confirm the presence of a human victim. The next robot is a MidBot, which is larger than the MinBot and has more advanced victim-sensing capabilities, enabling it to correctly identify the presence of human casualties at short distances. Increased computational power and memory make MidBots (marginally) suitable as team leaders if no better-equipped robots are available. A MaxBot is the third robot type. Its primary purpose is to function as a team leader due to its large memory capacity and powerful computational capabilities. MaxBots have basic victim sensors only and must rely on MidBots to correctly identify casualties. MaxBots are equipped with a tracked drive enabling them to drive over shorter debris items (light gray boxes in Fig. 2). The fourth robot was a highly-specialized, tracked-drive robot called a DebrisBot whose goal was to find and remove shorter debris items in the environment. Debris removal could be done in an unguided manner (randomly looking for and clearing debris if no other tasks are available), or by request from another robot that has become stuck in the environment. Debris removal results in a piece of debris being completely removed from the environment—this can only be done at ranges less than 1 m, and takes 5 s.

To test our framework against the challenging aspects of robotic USAR, our simulator is configured to allow varying levels of communications reliability, directly affecting the percentage of messages that are delivered successfully. This simulates interference in the environment. Wireless communication was simulated using Stage's indoor ITU radio model, with a 20 m range. Robot failure is also simulated by continually choosing a random value in the range [0,1] and comparing it to set thresholds for temporary or permanent failure in each simulation cycle. Exceeding the permanent failure threshold causes the robot to fail for the remainder of the trial, and exceeding the temporary failure threshold causes a failure for a random length of time (3–4 min). Failure thresholds are specified separately for each robot type, simulating different degrees of reliability.

4.1 Experimental Design

We evaluated our framework in 2 factorial experiments, testing 3 recruitment strategies against 3 levels of communications reliability and 3 levels of robot failure probability. Communications success rates varied between 100%, 60%, and 20%. Because of the random elements in the algorithm for robot failure, rates of robot failure are best expressed as an average percentage of time spent failed for MinBots, MidBots, MaxBots, and DebrisBots, respectively. The values used in this experiment are categorized as none, minimal (15%, 12%, 9%, 15%) and major (25%, 21%, 19%, 25%). Each experimental configuration lasted 30 simulated minutes and was run 50 times. Every trial was repeated in 3 unique environments to help reduce bias.

In our first (*main*) experiment we were interested in evaluating our framework using 2 teams of robots each consisting of 1 MaxBot, 2 MidBots, 4 MinBots, and 1 DebrisBot. Robot teams were initially inserted at the 2 entrances visible in Fig. 2. For this experiment, we varied whether or not replacement robots would be available. In trials where replacements existed, 10 MinBots, 2 MidBots, 1 MaxBot, and 1 DebrisBot would be introduced into the environment, spread evenly along the inner perimeter, after 5 simulated minutes. Each of these units functioned individually as a 1-robot team until forming a new team or joining an existing one. The factorial design for our main experiment yielded 8100 trials.

Our second (*minimal team*) experiment was designed to test the performance of our framework when resources were scarce and replacements were not available. In this experiment, 2 robot teams each consisting of 1 MinBot, 1 MidBot, 1 MaxBot, and 1 DebrisBot were placed in the entrances shown in Fig. 2. The factorial design for our minimal team experiment yielded 4050 trials.

5 Results and Discussion

We evaluated our framework using 2 criteria: the total percentage of true and false victims successfully identified and communicated to a team leader, and the total percentage of the environment covered and communicated to a team leader. We focused only on knowledge conveyed to team leaders, since they are likely to possess the most complete (but still imperfect) set of information. They also represent the best source of knowledge for human extraction teams in real-life scenarios. This makes our results more conservative—other robots may contain additional information that has not been relayed to a leader.

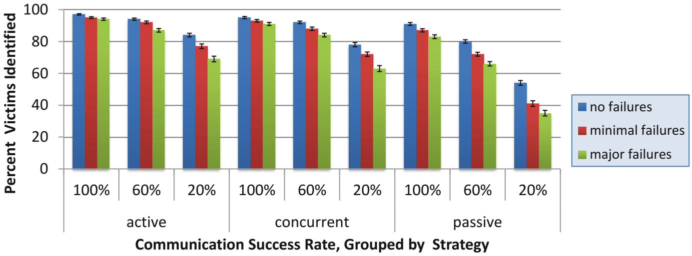

Fig. 3. Main experiment victims identified, with replacement robots available.

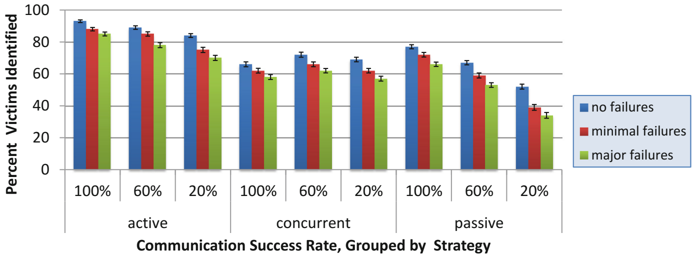

Fig. 4. Main experiment victims identified, with no replacement robots available.

5.1 Main Experiment Results

Victims Identified to Leaders. Figures 3 and 4 show the percentage of victims identified and communicated to a team leader, with and without replacements respectively. In all graphs that follow, error bars show 95% confidence intervals.

A major weakness of [7] was that robot teams could not function effectively when communication success rates were as low as 20%. As communications failed, fewer tasks could be successfully assigned and fewer results could be relayed to leaders. This is reflected in our results: passive recruitment is shown to perform poorly in these conditions. Active and concurrent recruitment strategies, however, provide a significant improvement in terms of victims identified, regardless of the presence of replacement robots, when robot operating conditions or communication reliability are poor.

Interestingly, in ideal conditions where no replacements are available, passive recruitment performs better than concurrent recruitment. This is likely due to a slight duplication of effort that occurs as non-leader robots assign victim-identification tasks that would normally be coordinated by a team leader. Without a team leader's broader perspective on remaining or outstanding work, multiple robots may unnecessarily be tasked with completing the same job. This redundancy can be harmful to team performance in ideal conditions, but is beneficial whenever conditions are not ideal.

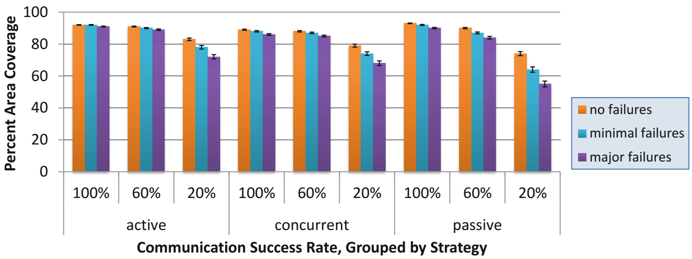

Fig. 5. Main experiment area coverage, with replacement robots available.

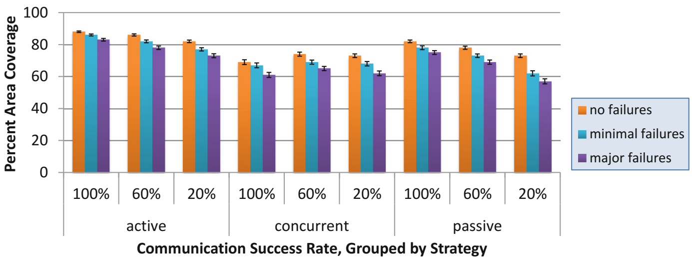

Fig. 6. Main experiment area coverage, with no replacement robots available.

Area Coverage. As shown in Fig. 5, there is not much difference in area coverage between recruitment strategies when replacements are available. It is possible that the area coverage amounts shown here reflect the maximum area that can be covered and conveyed to a team leader in the time allotted (30 min), or that some areas are extremely difficult to reach. As communication success rates fall, and as robot failures increase, active and concurrent recruitment show greater improvements over passive recruitment.

Figure 6 shows that active recruitment is more effective than passive recruitment when no replacements are available. This strategy results in the greatest environment coverage, likely due to the physical searches that take place and the improved likelihood of recruiting robots with useful information about the environment. As with victim identification (Sect. 5.1), concurrent recruitment performed more poorly than passive except where communications were unreliable. As noted previously, this is likely because of a duplication of effort that occurs when tasks are assigned without the coordination of a leader.

5.2 Minimal Team Experiment Results

In this experiment, active recruitment resulted in the greatest number of victims identified, in almost all experimental configurations. This is likely due to the increased exploration that takes place when actively recruiting—exploration provides more opportunities to discover victims and cover the environment. Active recruitment also resulted in the highest levels of area coverage (Fig. 8).

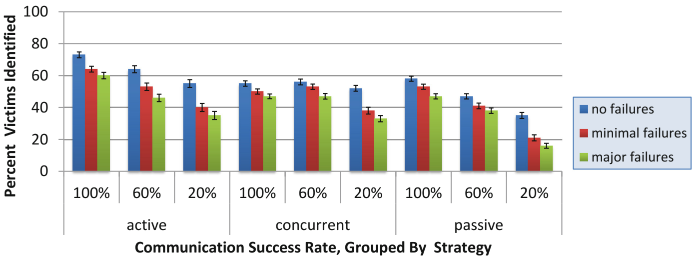

Fig. 7. Minimal team experiment victims identified.

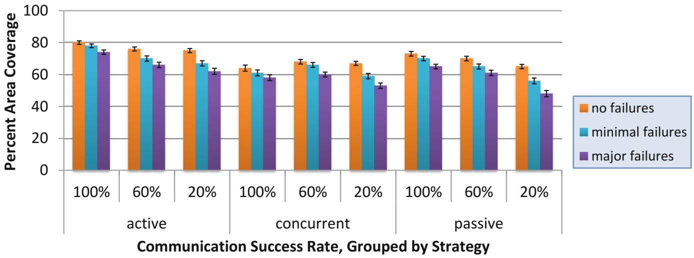

Fig. 8. Minimal team experiment area coverage.

Because of the smaller robot population, robot failures had a greater impact on these results than in our main experiment. However, it is interesting to note that area coverage in this experiment is only slightly less than that of our main experiment where replacements were not available. This suggests that only a small number of robots is necessary to cover most of the environment in 30 min. It is also possible that our experimental environments contain hard-to-reach areas that make it difficult to attain 100% environmental coverage, and that additional robots only improve coverage by a small amount.

6 Future Work and Conclusion

Since increased knowledge transfer between robots has clear benefits, directions for future work include having robots explicitly tasked with spreading information to others. The importance of the information could be used to determine the ideal recruitment strategy for conveying it: more critical information would require a more active approach.

In this paper, we described and evaluated recruitment strategies allowing robot teams in dangerous domains to acquire additional skills temporarily or permanently. Our recruitment strategies provide the greatest benefits when operating conditions are not ideal. Since this is often expected to be the case for many future robotics applications, we believe our framework and results provide substantial insight into how distributed robot teams should be designed to operate in these environments.

References

1. Brooks, R.A.: A robust layered control system for a mobile robot. Robot. Autom. **2**(1), 14–23 (1986)
2. Carnegie, D.: A three-tier hierarchical robotic system for urban search and rescue applications. In: Proceedings of the IEEE International Workshop on SSRR 2007, pp. 1–6 (2007)
3. Costa, E.D., Shiroma, P.M., Campos, M.F.: Cooperative robotic exploration and transport of unknown objects. In: Proceedings of the SBR/LARS 2012, pp. 56–61 (2012)
4. Dos Santos, F., Bazzan, A.L.: Towards efficient multiagent task allocation in the robocup rescue: a biologically-inspired approach. Auton. Agent. Multi-agent Syst. **22**(3), 465–486 (2011)
5. Dutta, P.S., Sen, S.: Forming stable partnerships. Cogn. Syst. Res. **4**(3), 211–221 (2003)
6. Gage, A., Murphy, R.R.: Affective recruitment of distributed heterogeneous agents. In: Proceedings of the AAAI 2004, pp. 14–19 (2004)
7. Gunn, T., Anderson, J.: Dynamic heterogeneous team formation for robotic urban search and rescue. J. Comput. Syst. Sci. **81**(3), 553–567 (2015)
8. Kiener, J., Von Stryk, O.: Cooperation of heterogeneous, autonomous robots: a case study of humanoid and wheeled robots. In: Proceedings of the IROS 2007, pp. 959–964 (2007)
9. Krieger, M.J., Billeter, J.B.: The call of duty: self-organised task allocation in a population of up to twelve mobile robots. Robot. Auton. Syst. **30**(1), 65–84 (2000)
10. Mathews, N., Christensen, A.L., O'Grady, R., Rétornaz, P., Bonani, M., Mondada, F., Dorigo, M.: Enhanced directional self-assembly based on active recruitment and guidance. In: Proceedings of the IROS 2011, pp. 4762–4769 (2011)
11. Nagy, G., Anderson, J.: Active recruitment mechanisms for heterogeneous robot teams in dangerous environments. In: Khoury, R., Drummond, C. (eds.) AI 2016. LNCS (LNAI), vol. 9673, pp. 276–281. Springer, Cham (2016). doi:10.1007/978-3-319-34111-8_34
12. Pinciroli, C., O'Grady, R., Christensen, A.L., Dorigo, M.: Self-organised recruitment in a heteregeneous swarm. In: Proceedings of the ICAR 2009, pp. 1–8 (2009)
13. Pitonakova, L., Crowder, R., Bullock, S.: Understanding the role of recruitment in collective robot foraging. In: Proceedings of the ALIFE 2014 (2014)
14. Roth, M., Vail, D., Veloso, M.: A real-time world model for multi-robot teams with high-latency communication. In: Proceedings of the IROS 2003, pp. 2494–2499 (2003)
15. Van De Vijsel, M., Anderson, J.: Coalition formation in multi-agent systems under real-world conditions. In: Proceedings of the AAAI 2004 Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems (2004)
16. Vaughan, R.: Massively multi-robot simulations in stage. Swarm Intell. **2**(2–4), 189–208 (2008)