Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Procedia Computer Science 00 (2013) 1–10

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

Dynamic Heterogeneous Team Formation for Robotic Urban Search and Rescue

Tyler Gunn[a], John Anderson[a,*]

[a]*Department of Computer Science, E2-445 EITC, University of Manitoba, Winnipeg, MB R3T 2N2*

Abstract

Though much work on coalition formation and maintenance exists from the standpoint of abstract agents, this has not yet translated well to the realm of physically grounded robots. Most multi-robot research has focused on pre-formed teams, with little attention to the formation and maintenance of the team itself. While this is plausible in forgiving domains, it fails rapidly in challenging environments where equipment is lost or broken easily, such as urban search and rescue. This paper describes the team management elements of a framework for coordinating a changing collection of heterogeneous robots operating in complex and dynamic environments such as disaster zones. Our framework helps a team to reshape itself to compensate for lost or failed robots, including adding newly-encountered robots or additions from other teams, and also allows new teams to be formed dynamically starting from an individual robot. We evaluate our framework through an example implementation where robots perform exploration in order to locate victims in a simulated disaster environment.

Keywords: multi-robot systems, team formation, team management, heterogeneity, roles, USAR

1. Introduction

Although much research has been performed on teams and coalition formation in multi-agent systems, most works tend to focus on abstract agents performing high level tasks in domains lacking a physical grounding (e.g. package delivery in abstract space [1, 2]). While important in principle, these do not take into account many of the physical challenges of being grounded in the real world (e.g. communication distance and reliability [3]). Further, work involving teams of heterogeneous robots is usually restricted to relatively controlled laboratory environments and relies on fixed team structures determined in advance (e.g. [4, 5]).

Robots operating in any real-world environment have many challenges to contend with, such as noisy and inaccurate sensor data. Localization is imperfect, and algorithms to intelligently interpret visual data are computationally expensive and inaccurate. Operation in hazardous environments, such as those presented by the exploration of other planets and disaster zones must additionally deal with the fact that robots can be

*Corresponding author

Email address: `andersj@cs.umanitoba.ca` (John Anderson)

damaged or destroyed. In domains such as these, communication between robots is short range, unreliable, and sporadic in nature: in disaster areas, for example, infrastructure can be heavily damaged, and the debris itself can interfere with wireless communication.

A good example of a highly challenging domain in which robots can be of value is in the aftermath of a natural or man-made disaster. This is commonly known as *Urban Search and Rescue* (USAR), and involves exploring damaged structures to locate and assist human casualties. Operation in a USAR environment presents significant mobility and sensory difficulties[6]. Debris and uneven terrain can make navigation difficult and can cause a robot to become stuck. Structural changes to the environment as a result of the disaster can render existing floor-plans and maps useless.

The challenges present in a USAR environment make it likely that robots may become lost or separated from their team. Further, robots can become physically damaged or destroyed, impairing the team's effectiveness. It is also possible for different teams of robots operating in geographically separated areas to encounter one another as the mission progresses, providing an opportunity for teams to exchange members or combine resources. New robots can also be expected to arrive sporadically since not all equipment arrives at once or is sent in at the same time.

While our framework supports task discovery, task allocation, and coordination in USAR using a changing collection of robots, space limitations restrict our focus to team maintenance and formation in this paper. These operations are described in Section 4, followed by an evaluation, while Section 3 provides a high-level overview of concepts key to the operation of our framework. For other elements of this framework not related to team management, see [7].

2. Related Work

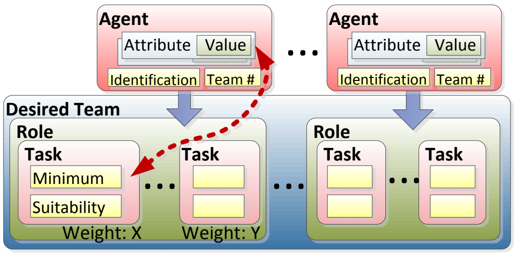
Until recently, there has not been a large focus on how to form and maintain teams of robotic agents. Most previous works assume teams were formed in advance and will not change during the course of operation (e.g. [8–13]).

Extensive prior work (e.g. [1, 2, 14]) has resulted in techniques to enable self-interested agents to form mutually beneficial partnerships in groups of two or more. Although these concepts are generally applicable to robotic domains, these works are demonstrated in domains which are too abstract to show direct applicability for robots in challenging conditions. There is no consideration for issues surrounding the perception of agents in the environment, localization, or the impact of limited range unreliable communication.

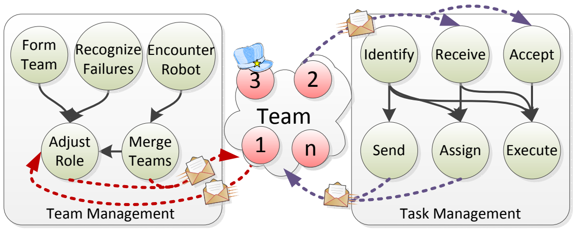
George et al. [15] studied a method to form sub-teams in a larger overall team of unmanned aerial vehicles (UAVs) operating in a region. Their approach is more realistic, as it assumes robots are heterogeneous and must cooperate to achieve a common goal. Communication is assumed to be limited range, and operation takes place in a more realistic domain, but there is little or no consideration of the ability to form new teams as opposed to sub-groups.

Cheng and Dasgupta [16] developed a technique to form teams among robots exploring an area. Although their work assumes a more real-world domain, it aims to form teams for the explicit purpose of maximizing the overall explored area, where our work attempts to maintain teams for carrying out a broader set of overall tasks, the nature of which can change over the course of the mission.

Kiener and von Stryk [4] present a framework for the cooperative completion of tasks by teams of heterogeneous robots. Their framework achieves this by modeling the individual tasks of the overall mission, and storing the degree to which each of the robots can perform these tasks. The capabilities of (only) a single humanoid and single wheeled robot are determined in advance, along with weights identifying the suitability of each to all possible tasks. This information allows a central controller to allocate tasks to each robot. While the tasks involved are significant in that they involve fine motion control and interaction, this is still very primitive in terms of task allocation. The broadly different robot skills and task demands result in a predefined set of tasks with only one logical way to map these tasks to the robots in their system. Our framework instead assumes that there may be potentially a large number of potential mappings. Their approach also requires constant communication with a central controller, where our framework performs task allocation in a distributed manner.



(a) Robots, roles, and desired teams.



(b) Framework operation for a single robot.

Fig. 1

Howard et al. [5] developed a system to automatically deploy a sensor network using heterogeneous robots. Resourceful *leader* robots guide network deployment and provide guidance to sensor nodes to keep them in formation. In contrast to our work, the teams, formation, and deployment positions of the sensor nodes are all pre-computed in advance and rely on reliable communication to a central processing unit. Changes in team structure due to the loss of robots are accounted for by looking up a new pre-computed deployment pattern and adjusting the formation accordingly. No attempt is made to recover lost robots.

Dorigo et al. [17] developed the *swarmanoids* architecture as a means of encouraging research into swarm robotics in real-world domains. In their work, three heterogeneous robot types cooperate to complete the mission of location a book on a shelf and retrieving it. The capabilities of the robot types, however, preclude them from being used in any other combination, and leaves little opportunity to adapt to changes in available robot types.

3. Preliminaries

Before discussing team management in our framework, it is necessary to have some understanding of the way our framework represents knowledge of tasks, roles, and teams in order to facilitate its primary goals of team maintenance and task management. This section describes these concepts, as well as how our framework deals with sharing knowledge among team members.

In our framework, descriptions of possible tasks are created in advance, and describe the units of work required to complete the mission. As illustrated in Figure 1a, a task has both a minimum requirements and suitability expression defined in terms of the attributes of a robot. The minimum requirements determine the set of capabilities a robot must possess in order to carry out the task. For robots that meet these minimum requirements, the suitability expression defines the degree to which a robot is suited to carry out that task. Describing tasks in this manner forms the basis on which robots can reason about the best available team member to carry out a specific task. This knowledge also serves to indicate when the current team structure is less than adequate, and describes the needs of desirable new team members.

To facilitate efficient task allocation and assign a general responsibility of duties, we define roles in terms of the type of tasks a robot filling the role is normally expected to be able to perform. Thus, the tasks expected of a role determine the capabilities required of a robot filling that role. Since task requirements are formulated such that a robot's suitability to carry out a task can be calculated, it follows that a robot's suitability to fill a role is the aggregate of its suitability to complete each task normally expected of the role. Using roles provides a short-cut when assigning tasks to other robots: in the absence of time for further reasoning, a task can be assigned to a team member occupying a role that is normally expected to carry out that task.

The concept of a *desired team* is central to team maintenance in our framework. The desired team identifies the required roles and the quantity of each in order to make an effective team. This description then forms the goal that the framework's team maintenance operations aims to achieve. The desired team composition is highly domain and equipment-dependent and is determined by a human in advance of operation.

Within a team, a *team coordinator* is a special-purpose role responsible for directing the overall operation of the team. This role designates the responsibility of assigning tasks to a single robot and provides a single coordination point where the results of tasks are collected. The team coordinator attempts to ensure that each teammate has a small backlog of tasks to complete, helping robots remain productive in times of communication outages.

4. Methodology

The primary objectives of our framework are *team maintenance* and *task management*. Figure 1b shows how these operations are accomplished from the standpoint of an individual robot. Team maintenance operations, described in Section 4.2, focus on the formation and maintenance of teams, and their reaction to changes in team structure. Task management operations, described in [7], focus on the identification and assignment of tasks to the most suited members of the team. A deficiency in team structure can result in a situation where task assignments are suboptimal; team maintenance operations attempt to correct deficiencies in team structure so that a higher level of suitability can be achieved when assigning tasks.

Thus, using our framework, teams are a fluid aggregation of robots, where robots switch roles within the team and change teams as necessary to make the best use of their abilities. As robots change roles and teams, the overriding goal is to form stable teams that meet the definition of a desired team as closely as possible. Our framework does not differentiate teams of one from larger groups, and so single robots can form larger teams through encounters. A single robot that fills the team coordinator role well will likely retain that role, and one that does not will cede it to a more appropriate teammate.

4.1. Task and Role Suitability

As described in Section 3, the team maintenance and task management operations rely upon tasks defined in terms of a minimum requirements and suitability expression. The minimum requirements expression for a task is a simple boolean expression defining the attributes and corresponding values required for a robot to carry out a task. A victim verification task, for example, could have minimum requirements $HasMap = true \wedge (HasCamera = true \vee HasBreathSensor = true)$.

Suitability expressions are expressed similarly to minimum requirements expressions, except each term is assigned a weight. For each condition in the suitability expression that is satisfied (evaluates to true), a value equivalent to the weight is generated. Conditions evaluating to false generate a value of 0. The evaluated weights for conditions combined with the *and* logical operator are added together. For conditions combined with the *or* logical operator, the result is the maximum weight of the evaluated conditions. The net effect is that conditions combined with the *and* operator increase the suitability for every condition that is met, while the *or* operator acts to increase suitability based on the most valuable condition met. The suitability expression $HasMap[30] = true \wedge (HasCamera[10] = true \vee HasBreathSensor[30] = true)$, for example, favours a robot with a breath sensor over one with a camera, and assigns increased suitability to a robot with a map.

Calculating a robot's suitability to fill a role involves summing up it's suitability to carry out the tasks normally expected of that role. These operations involving tasks, roles and suitability expressions form the basis for the team maintenance operations described in the following section.

4.2. Team Maintenance

Team maintenance aims to ensure robots fill roles on the team which result in a close approximation to the definition of a desired team. Adjusting the roles robots fill on the team can occur when a robot loss or failure is recognized, or when a new robot is encountered and a team merge and redistribution occurs. Team maintenance is also responsible for ensuring the team coordinator role is filled, should that role become open due to a change in team structure.

4.2.1. Recognizing Failures

The most obvious sources of failure in a USAR domain are hardware damage and becoming physically lost. Our framework helps robots detect failures by tracking the last time a robot has heard from the other robots on its team. Robots that have not been heard from in a specified period of time are considered to no longer be a member of the current team.

A team recognizes the failure of robots and responds by adjusting the roles of the remaining robots in a decentralized manner. A robot is responsible for determining if any of its teammates have failed, and adjusting the role it fills on the team in response (a *Role Check*). As shown in Figure 1b, robots gain knowledge of their current team composition through inspection of wireless traffic. In our framework, all wireless messages sent by a robot include the sender's team affiliation and role, which robots in range overhear. This knowledge provides a robot with the necessary information to perform a role check, and potentially change roles in an attempt to ensure the team matches the definition of a desired team a closely as possible. Further, role changes ensure that the team coordinator role is filled by the best suited robot. The recognition of failures is implicit in that a robot adjusts the role it fills on their team in response to deficiencies it detects in its current view of the team complement.

During a role check, a robot calculates a weighted suitability to fill each role on its team, guided by the shared desired team definition. Roles which are currently under-filled according to the desired team definition are given a higher weighting. This encourages robots to fill roles to which they are less suited in the absence of a more suitable team member. If the role check identifies that a role change is necessary, the robot implements it and informs its teammates of the change, which can trigger subsequent role checks to occur for other members of the team. To help prevent two robots from simultaneously determining their role and implementing the same role change, a random interval is added between role checks. Each robot is responsible for changing roles on its own, and can do so without the explicit authorization of any other robot. It is, however, also possible for a robot to be instructed to change roles by another robot in the case where a team merge occurs.

Since we assume communication is unreliable, messages communicating the current team structure are not guaranteed to reach every member. This results in discrepancies between individual views and the actual team structure, the size of which is proportional to communication accuracy. As a result, a robot can potentially initiate a role change that is sub-optimal due to having an incomplete view of its team structure. These deviations are expected in the context of the team of a whole, and are compensated for in part by defining a desired team in terms of a minimum and maximum number of robots filling each role. Restricting the role checks to a periodic basis also helps to prevent the team from continually restructuring itself as its view of the team structure changes. Such oscillations are common in distributed settings, and a similar approach has been shown to be useful for choosing opponents to block in robotic soccer, for example [18].

4.2.2. Encountering Robots

Where two robots on different teams encounter each other, the robots act as representatives for their teams (they may also be individuals and thus represent the entirety of two teams) and calculate a potential merge or redistribution, beginning by exchanging their individual' perspective of team members and roles. The process used in our framework allows individuals to form a larger team, gathering more individuals in future encounters, and also allows the rebalancing of existing teams that could better fit the definition of a desired team by exchanging members.

Our approach selects the most appropriate of the two encountering agents to perform the computational work involved in this process, and abandons the process if neither robot has adequate computational abilities (since better robots may be encountering one another shortly). While it would appear most useful to defer to the two team coordinators perform these negotiations, this is problematic in the types of domains this framework is intended for: team coordinators' knowledge of current team structure is also imperfect, it increases reliance on two specific agents and therefore vulnerability, there are additional levels of indirection involved in contacting team coordinators, and the distance may be much greater between them, leading to less successful communication and greater likelihood of the process failing. Moving the team coordinators also prevent them (and possibly others) from doing useful work at the same time.

The representative robot chosen to consider the merge or redistribution uses information about both teams to form new teams by iteratively finding the best suited robot to fill a role on the new teams and assigning that robot to its best suited role and team (guided by the shared desired team description). Unlike the role check operation, the merge and redistribution attempts to place robots into the best suited roles on the resulting teams. The result will be either a single combined team, two teams with robots swapped between teams to ensure both teams match the definition of a desired team, or no changes (the latter happening if the teams cannot be improved).

After this operation, the other robot is informed of the role and team changes that its teammates must implement as a result. Both encountering robots are then responsible for informing their own teammates of the changes that must be implemented. Inaccurate knowledge of the current team compositions and lost role change instructions can cause the resulting teams to deviate from the desired team definition, and future role checks provide an opportunity to compensate for this.

Where both encountering robots are operating alone, the encounter provides an opportunity for a new team to form out of the individuals. The new team can continue to merge in individuals it encounters, further strengthening the team's capabilities. In this way, it is possible for teams to build up starting with a single robot.

4.2.3. Coping with Inconsistencies

Since our framework assumes communication is unreliable, it is expected that the knowledge any one agent has of its team will be inconsistent with the actual team composition. As a result, a robot could potentially implement a role change that results in the overall team composition deviating away from, rather than closer to, the definition of a desired team. As other robots learn about the role change, there is an opportunity for them to change roles to compensate. Further, as the original robot gains more accurate knowledge of its team, it can change roles again if necessary.

Inaccurate team knowledge can have a similar impact to the team redistribution resulting from encounters (Section 4.2.2). The reformed teams would be determined based on the inaccurate knowledge, causing some robots to be instructed to change to a sub-optimal role. Further, role and team change instructions could fail to be implemented by team members as a result of communication failures, resulting in a deviation from the redistributed team from what the encountering agents intended. As team members perform periodic role check operations, the team has an opportunity to adjust its structure to account for the changes which failed to be implemented.

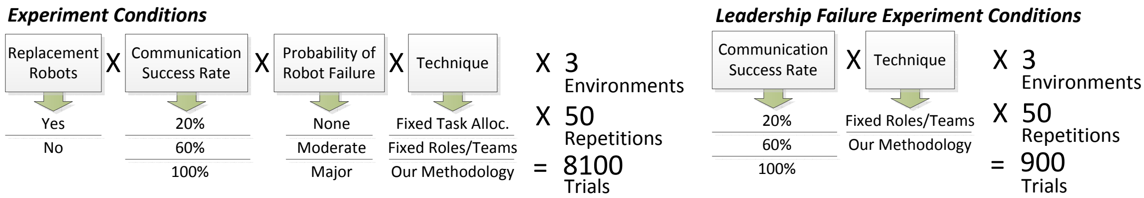
Our framework assumes the role and team change instructions are only used by the robots implementing the changes, and not by other teammates to attempt to learn about the new team composition. This helps ensure only the successfully implemented changes are recognized by the team.

It is also possible for robots to be instructed to change teams when it does not make sense to do so in the context of the other changes which ultimately succeed. The result will be one or both teams operating in a degraded state. Future encounters between the same teams or other teams can provide an opportunity to compensate.

Another potential source of inconsistency is the case where two robots on a team simultaneously encounter other robots (either on the same team or different teams) and negotiate team redistributions at the same time. Such scenarios are considered to be rare (they did not occur often in our experiments), and would likely result in the teams involved deviating from the definition of a desired team. As knowledge of the implemented spreads, team members will change roles in an attempt to compensate. Future encounters with other teams would provide an opportunity to make up for deficiencies in team structure.

5. Experimental Evaluation

We evaluated the effectiveness of our approach using a simulated USAR domain created using the Stage multi-robot simulator [19]. Heterogeneous robots explore damaged structures in order to build a map of the environment and locate casualties. We assume robots can become lost or separated from their team and new robots will be released into the environment as time goes on. Our implementation was coded directly



(a) Experiment conditions.

(b) Leadership failure experiment conditions.

Fig. 2

against the Stage API, allowing for repeatable simulation runs, executed much faster than real-time. A modular design approach ensures the implementation can form the basis of future experiments. We also made modifications to the Stage simulator to provide simulated, unreliable communication between robots.

Using simulation to study our approach is appropriate as the primary focus of our work is to support teamwork and coordination between robots, rather than complete accuracy in USAR in particular. The approach of using a simulated USAR environment for multi-robot research is well established (e.g. [20] used simulated USAR environments).

Our implementation assumes three types of robots are available; *MinBots*, *MidBots* and *MaxBots*. The MinBots are small (10 cm diameter), expendable robots with a wheeled physiology, restricting them to open areas. They are equipped with sonar sensors for navigation and mapping, and sensors capable of detecting the potential presence of victims in the environment. The MinBots do not posses the memory or processing capabilities to coordinate a team. The MaxBots are larger (40 cm diameter), complex robots with the computational capabilities required to coordinate a team and plan an effective exploration of the environment. They are equipped with a tracked drive, allowing them to drive over areas of light debris, giving them access to areas the other robot types cannot access. The MidBots are 20cm in diameter and have computational capabilities that fall in the middle; they are able to coordinate a team, but not as effectively as the MaxBots. The MidBots posses high fidelity victim identification sensors, and are able to confirm potential victim readings reported by the MidBots.

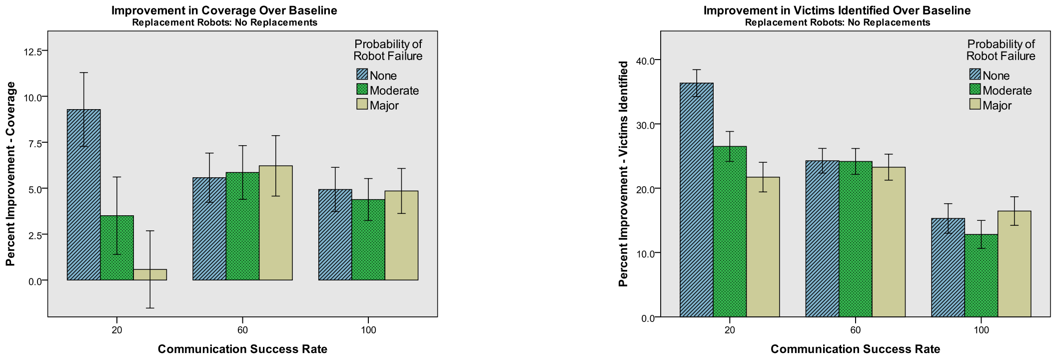
The tasks in our environment are focused on exploration of frontiers identified by more powerful robots and verification of potential victims identified by less powerful robots. These tasks are grouped into roles focused primarily on exploration, and others focused primarily on victim verification.

5.1. Experiment

Environments have two teams which start with one MaxBot, two MidBots, and four MinBot. Teams begin operation in opposite corners of the environment, out of communication range from one another. The environments are 60 mx60 m in size, and include 50 randomly positioned rooms which are 5 -12 m wide, and 5 -12 m long. Access to 60% of the rooms is blocked by debris, restricting access to the MaxBots. The remainder of the environment is filled with randomly placed debris and obstacle elements, 60% of which is passable by the MaxBots, until 13% of the environment is filled with debris, obstacles or rooms. 20 victims are distributed in the environments. An additional 10 debris configurations resembling victims are included, allowing for the potential of misidentification through MinBot sensor errors.

To evaluate our methodology, we performed an experiment to study the efficacy of our methodology when coping with communication and robot failures. As shown in Figure 2a, two of the independent variables in our experiment control communication success rate, and the probability of robot failure. Our methodology also allows a team to adapt to accommodate the arrival of replacement robots. Another independent variable determines whether replacement robots are available or not. Replacements (10 MinBots, 2 MidBots, and 1 MaxBot) begin operation from the edge of the environment at the 5 minute mark.

We compared our methodology against two base cases. In the first, robots are not permitted to change roles and cannot switch teams. This means team structure is fixed: teams cannot gain team members, and team members cannot voluntarily leave the team. Further, robots are not able to change their roles



(a) Improvement in coverage.

(b) Improvement in victims identified.

Fig. 3: Improvement with no replacements.

or team membership to adapt to changes to the team as a result to failures. This provides a means of evaluating our framework's performance from the standpoint of adaptive team management. Because tasks are still allocated in the base case using our task allocation methodology, it shows how much performance improvement is due to improved team management.

The second base case uses fixed roles and team membership, and includes the restriction that there is a 1:1 mapping between tasks and robot types. This means each task type can only be carried out by one type of robot. This essentially provides a worst-case comparison for multi-robot exploration in this domain: we can compare our approach and the base case above to a situation where all robot interaction is completely inflexible.

We used a factorial experiment design (Figure 2b) resulting in 8100 experimental trials. We used 3 different environments to help eliminate potential bias due to features of any one environment. The environments were generated using a tool, ensuring the same environment coverage, number of victims, and equal distance between team start locations. We performed 50 repetitions of each experimental condition, which were run for 30 minutes of simulated time each.

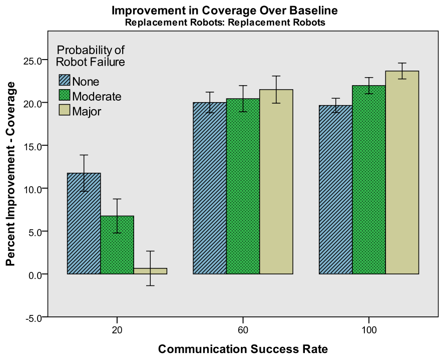
To evaluate the efficacy of our methodology, we recorded two values at fixed times throughout each trial: the percentage of the environment covered, and the percentage of victims successfully identified.

5.2. Failures in Team Leadership

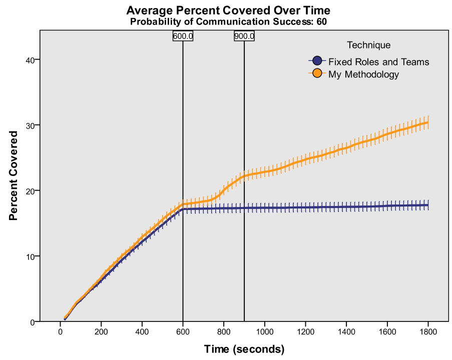
We performed a second experiment with a smaller scope to evaluate our framework's ability to cope with failure of a team's leadership structure (Figure 2b). In this experiment we introduced a failure in the leadership structure of a team at fixed times (10 minutes and 15 minutes) to allow observation of the performance of a single team as it adapts to these failures. This second experiment was limited in scope, and did not consider the availability of replacement equipment. Further, our methodology was compared only against the first base case scenario, where roles are fixed and team membership is fixed.

6. Discussion

Figures 3a and 3b show the improvements realized using our framework in the percent of the environment covered and victims identified, respectively, over the baseline where roles are fixed and team membership is fixed, and no replacement equipment is available. Performance of our framework was hampered at the 20% communication success rate: too few messages were successfully delivered, resulting in tasks failing to be allocated and teams breaking apart. With a communication success rate of 60%, our methodology was able to compensate for the coordination issues associated with robots becoming separated from their team, by allowing them to join another team, or form a new team in response. The smaller number of



(a) Improvement in coverage with replacements.



(b) Leadership failure experiment coverage over time.

Fig. 4

victim identification tasks compared to frontier exploration tasks, coupled with the scarcity of robots with the capability to identify victims resulted in a higher performance improvement in terms of the number of victims identified (Figure 3b) compared to the percent of the environment covered (Figure 3a).

Figure 4a shows the improvement in environment covered over the baseline where roles are fixed and team membership is fixed (the corresponding graph illustrating the improvement in victims shows a considerably higher improvement, but is omit due to space constraints). These results show that the framework is able to allow teams to take advantage of the replacement equipment which becomes available after the start of the mission.

The leadership failure experiment clearly demonstrated our methodology's ability to enable a team to continue operation, despite the failure of a robot filling the critical team coordinator role. Where roles and team membership are fixed, the failure of the team coordinator resulted in the team ceasing to make further progress. Using our methodology, the team adjusted to the failure of the team coordinator, and was able to continue making progress despite the loss of the better suited robot.

Figure 4b shows the percent of the environment covered over the duration of the trials where the communication success rate is 60%. The vertical lines indicate where failures of robots in the team coordinator role were introduced. In the baseline where roles are fixed and teams are fixed, the team ceases to make progress after the first failure. Using our framework, the team is able to adjust itself to compensate for the loss of the robots occupying the team coordinator role, and to continue making useful progress.

7. Future Work

Although our methodology showed significant utility, our example implementation revealed a number of areas where improvements could be made, or other interested research could be performed. An implementation in a physical environment using real robots would be valuable, as it would increase the difficulty of operation considerably.

Our methodology assumes agents can suffer failures and responds to them. It would be useful to incorporate a failure model of the robots and their components. This information could help drive the task assignment process (e.g. [21] found exploration performance could be improved by anticipating failures based on a robot's reliability), ensuring critical tasks are carried out by more reliable agents, for example. Robots could also monitor their performance, and that of others, in order to adjust reliability knowledge based on actual experiences.

In terms of team structure and membership, it would be useful to investigate the use of techniques to enable teams to learn the ideal team structure based on its experiences. It would also be interesting to study the impact of relaxing the restriction that a robot is a member of only one team, providing more opportunity for collaboration between teams and the sharing of rarer capabilities.

8. Conclusion

Our methodology shows strong benefits when helping a team cope with team members getting lost due to unreliable communication and the difficult nature of the environment. Lost team members are able to either form their own team, or join another team they encounter. Where a robot has especially important skills, our methodology helps ensure a lost robot is able to continue providing useful work towards the overall mission. Where critical members of the team suffer a failure, our methodology also shows a significant improvement over the baseline case, allowing a less suited robot to take on the team coordinator role. Where replacement robots are available, our methodology shows a clear benefit in allowing the replacements to form new teams, and be integrated into existing teams.

This research has demonstrated the utility of defining the composition of a team in terms of roles describing the types of work normally expected of its members, and using this as a means of reasoning about the changes which can be implemented to the structure of these teams in response to the loss or failure of team members, or the discovery of new potential team members.

It is our hope that the success of this research will encourage future research into the issues involved with the effective coordination of robots operating in difficult and challenging domains. Robots operating in these domains must cope with difficult conditions, and cannot make assumptions about team structure or composition, making research into team formation and maintenance in these conditions even more important.

References

[1] P. S. Dutta, S. Sen, Forming stable partnerships, Cognitive Systems Research 4 (3) (2003) 211–221.

[2] M. van de Vijsel, J. Anderson, Increasing realism in coalition formation, in: Proc. of CIRAS-2005, 2005.

[3] L. Vig, J. Adams, Issues in multi-robot coalition formation, in: Multi-Robot Systems, Vol. 3, Springer, 2005, pp. 15–26.

[4] J. Kiener, O. von Stryk, Cooperation of heterogeneous, autonomous robots: A case study of humanoid and wheeled robots, in: Proc. of IROS-2007, 2007, pp. 959–964.

[5] A. Howard, L. Parker, G. Sukhatme, Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection, Int. J. Robot. Res. 25 (5–6) (2006) 431–447.

[6] R. Murphy, J. Casper, J. Hyams, M. Micire, B. Minten, Mobility and sensing demands in USAR, in: Proc. of IECON-2000, Vol. 1, 2000, pp. 138–142.

[7] T. Gunn, Dynamic heterogeneous team formation for robotic USAR, Master's thesis, University of Manitoba (2011).

[8] I. Rekleitis, V. Lee-Shue, A. P. New, H. Choset, Limited communication, multi-robot team based coverage, in: Proc. of ICRA-2004, Vol. 4, 2004, pp. 3462–3468.

[9] J. Bruce, M. Bowling, B. Browning, M. Veloso, Multi-robot team response to a multi-robot opponent team, in: Proc. of ICRA-2003, Vol. 2, 2003, pp. 2281–2286.

[10] N. Lau, L. Lopes, G. Corrente, N. Filipe, Multi-robot team coordination through roles, positionings and coordinated procedures, in: Proc. of IROS-2009, 2009, pp. 5841–5848.

[11] N. Boonpinon, A. Sudsang, Constrained coverage for heterogeneous multi-robot team, in: Proc. of ROBIO-2007, 2007, pp. 799–804.

[12] L. Giannetti, P. Valigi, Collaboration among members of a team: a heuristic strategy for multi-robot exploration, in: Proc. of MED-2006, 2006, pp. 1–6.

[13] T. Li, C.-Y. Chen, Y.-C. Yeh, C.-C. Yang, H.-K. Huang, H.-L. Hung, C.-H. Chu, S.-H. Hsu, D.-Y. Huang, B.-R. Tsai, M.-C. Gau, R.-J. Jang, An autonomous surveillance and security robot team, in: Proc. of ARSO-2007, 2007, pp. 1–6.

[14] S. Liemhetcharat, M. Veloso, Modeling and learning synergy for team formation with heterogeneous agents, in: Proc. of AAMAS-2012, 2012, pp. 365–374.

[15] J. George, P. Sujit, J. Sousa, F. Pereira, Coalition formation with communication ranges and moving targets, in: Proc. of ACC-2010, 2010, pp. 1605–1610.

[16] K. Cheng, P. Dasgupta, Multi-agent coalition formation for distributed area coverage: Analysis and evaluation, in: Proc. of IEEE WI-IAT-10, Vol. 3, 2010, pp. 334–337.

[17] M. Dorigo, D. Floreano, L. Gambardella, F. Mondada, et al., Swarmanoid: a novel concept for the study of heterogeneous robotic swarms, Tech. rep., IRIDIA, Universit Libre de Bruxelles (2011).

[18] J. Anderson, J. Baltes, An agent-based approach to introductory robotics using robotic soccer, Int. J. Robot. Autom. 21 (2).

[19] R. Vaughan, Massively multi-robot simulation in stage, Swarm Intelligence 2 (2) (2008) 189–208.

[20] M. Eghbali, M. Sharbafi, Multi agent routing to multi targets via ant colony, in: Proc of ICCAE-2010, Vol. 1, 2010, pp. 587–591.

[21] S. Stancliff, J. Dolan, A. Trebi-Ollennu, Planning to fail – reliability needs to be considered a priori in multirobot task allocation, in: Proc. of IEEE Intl. Conf. on Systems, Man and Cybernetics (SMC), 2009, pp. 2362–2367.