Effective Task Allocation for Evolving Multi-Robot Teams in Dangerous Environments

Tyler Gunn and John Anderson
Autonomous Agents Laboratory
Department of Computer Science, University of Manitoba
Winnipeg, Canada
Email: andersj@cs.umanitoba.ca

***Abstract*—This paper describes the task 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 allows a team to discover and distribute tasks among its members, in a distributed fashion, where the structure of the team is under regular change. Robots may become lost or fail at any time, and new equipment may arrive at any time. We evaluate our framework through an example implementation where robots perform exploration and search for victims in a simulated disaster environment.**

***Keywords*-multi-robot systems; task allocation; team management; heterogeneity; roles; USAR**

I. Introduction

Task allocation in a team involves mapping known tasks to the most appropriate agent to perform that task. From a global perspective, the most suitable agent to carry out work (i.e. high likelihood of success; strong skill mapping) may be the same individual for all current tasks. However, due to the geographic spread of the team, it make may more sense to have an agent physically located near the task's location carry out the task. Further, one of the goals of having teams is in part to spread work around among members.

Physical limitations, such as the distance between tasks or geographical limitations of team members, as well as time-limited opportunities mean it is generally impossible to rely on the best suited agent to carry out work. As a result, some tasks must be allocated to agents that are not well suited to carry out those tasks, in order that all tasks have a strong likelihood of being accomplished (taking into account prioritization based on importance as well as fit).

Task allocation has been a subject of multi-agent systems (MAS) research for many years. Much early work was studied using agents operating in abstract domains, where conditions in the real world could be eliminated or relaxed. Real-world task allocation is complicated by many things: tasks are not all available at the outset, but are discovered as work progresses; tasks are not necessarily discovered by those with the capabilities of solving or even judging their importance; and the mapping between agents and tasks may be highly unstable, not only on the basis of agent workload, but by losing and acquiring new agents.

Urban Search and Rescue (USAR) is an example of a domain where this is the case. Operation in such an environment is dangerous and presents many mobility and sensory challenges [1]. Robots can become stuck, lost, or destroyed at any time, which means that no one robot can be considered unexpendable. The cost of losing a robot means that cheaper robots, with sensing and locomotion methodologies spread among a team, are much more cost-effective. The distributed nature of jurisdiction means that agencies with equipment may arrive at different times and have different abilities. Coupled with risk of loss, this means that a team must evolve over time, adapting to the loss of individuals and taking advantage of newly discovered robots. In particular, there is a challenge of never being able to assume a single leader continues to exist, since an agent guiding others may be lost in the same way that any other robot can be. Surrounding all of this are challenges to underlying technology: communication may be sporadic and unreliable, and many solutions commonly used in robotics (e.g. GPS) can be completely disabled by the surroundings (i.e. robots must function in equivalent to an underground environment).

This paper focuses on the task allocation elements of a multi-robot framework for dealing with USAR and other similarly dangerous environments. Because the agent population undergoes continuous change, task allocation must interact with mechanisms for the management of team members and the roles they occupy. While this is an extremely challenging environment, a solution applicable to it should also apply to a range of equally challenging domains, from mining to operating on other planets, as well as domains where only some of these severe challenges are present.

The remainder of this paper describes our framework and its evaluation in a simulated USAR environment. Because task allocation must interact with mechanisms for team management, we begin by reviewing related work in these areas from a multi-robot perspective. For other elements of this framework not related to task management, see [2], [3].

978-1-4799-2902-3/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.114

II. Related Work

Previous multi-agent systems research (e.g. [4], [5], [6]) has produced techniques for agents to form partnerships to accomplish mutual goals. However, these works are generally demonstrated in abstract domains, and do not consider the challenges faced by robots operating in real-world environments. There is little consideration for issues such as the impact of unreliable limited range communication, perception, and localization.

Auction-based approaches (e.g. [7], [8]) have been developed to perform distributed task allocation amongst teams of heterogeneous robots. Similar to our work, these approaches assume robots can fail at any time and that communication may be unreliable. Auction-based approaches assume bidders will bid only on tasks they are capable of carrying out, where our work makes use of roles to guide the task assignment process to robots which are potentially capable of carrying out a task. Further, auction-based approaches typically assume all robots have the necessary capabilities to assign tasks, where our work assumes task allocation is itself a task which is delegated to a more capable robot.

Gage et al. [9] developed an approach to multi-robot task allocation that uses an emotion-based approach to assigning tasks. Similar to our approach, their work assumes unreliable communication and that robots can fail at any time. In their work, robots with tasks to assign continually announce the tasks. Those robots that hear the tasks calculate a *shame* value corresponding to their suitability to carry out the task. The shame value determines whether the robot responds with an offer to carry out the task. Not responding increases the shame value. This results in the best suited robots responding first, and the less-suited robots responding later. Although their approach attempts to reduce communication overhead, it places the burden of task allocation on all robots, even the most primitive. This is unrealistic for dangerous environments, since expendability in robotics will mean that there will be likely many simpler robots that can be sent to areas of greater risk.

Kiener and von Stryk [10] present a framework for the cooperative completion of tasks by teams of heterogeneous robots. Their framework models the tasks that make up the mission, and stores the degree to which each robot can perform these tasks. Their work uses a single humanoid robot and wheeled robot, and pre-computes the task suitabilities in advance. Although their work is an impressive demonstration of heterogeneous robots cooperating in a real world environment, the task allocation involved is very primitive. The capabilities of the robots results in a single mapping between tasks and robot types. We assume there may be any number of robots suitable to carry out a task, with varying degrees of suitability.

Ma et al. [11] developed an approach to performing frontier-based exploration using an auction-based task allocation methodology. Similar to our work, robots respond to task allocation requests with bids describing their suitability to carry out the task. Their approach, however, uses a multi-round auction approach which would be more susceptible to communication failures of the type we assume in our work. Further, our work assumes task allocation is generally centralized to a single robot, where Ma et al. assume it is performed in a completely distributed manner.

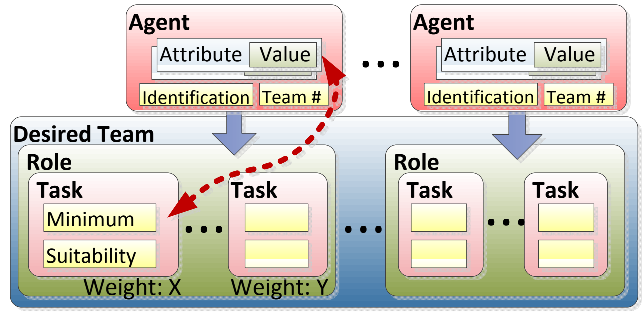

Figure 1. Tasks, roles and desired teams.

III. Preliminaries

Our work assumes heterogeneous robots are operating cooperatively in order to carry out the overall mission (e.g. USAR). We assume the robot population includes few robots with strong computational capabilities and many simpler (i.e. expendable) robots with comparatively limited computational and sensory capabilities.

Tasks form the basis upon which team maintenance decisions are made. We assume tasks are descriptions of the types of work a team normally expects to encounter during a mission. As shown in Figure 1, task type definitions include both a minimum requirements and suitability expression. These expressions are defined in terms of descriptive attributes describing the capabilities of the robots required to carry out these tasks. The minimum requirements expression establishes the bare minimum capabilities a robot requires to carry out a task of that type. Similarly, the suitability expression provides a means of calculating a numerical value corresponding to a robot's suitability to carry out that task. The suitability values are used in the task allocation process to determine the best robot on a team to carry out a task. Team management also makes use of the suitability expression as part of the process of determining the role a robot fills on a team.

We define roles in terms of the types of tasks normally expected of a robot filling that role. In this way, a role provides an abstraction of the underlying capabilities required of a robot to carry out the types of tasks expected of the role. This abstraction simplifies task allocation by avoiding the need to consider each attribute individually, and provides a means of assigning general responsibilities to robots on a team.

Given the challenges inherent with operation in a difficult domain, it is reasonable to expect that the composition of teams will be in a constant state of change. Teams will evolve over the course of the mission, as new members arrive and others fail or leave the team. The team management side of our framework aims to ensure an appropriate mix of robots is available to carry out the tasks the team encounters. The concept of a *desired team* identifies the roles and quantities of each required in order to make an effective team. It serves as a means of characterizing the type of work the team is expected to carry out, and helps team management ensure there is a suitable pool of robots available to assign tasks to. Potential roles and desired team structure are assumed to be defined in advance of the mission.

Because we assume few robots on a team have strong computational capabilities, we assume within a team a special-purpose *team coordinator* role is responsible for directing the overall operation of the team. This role assigns the general responsibility of task allocation to a single robot and provides a single point for collecting the results of carrying out tasks.

IV. Team Management

This section provides a brief overview of how our framework supports the distributed reconfiguration of teams in response to robot failures, the arrival of new equipment, and encounters between teams during the mission. The interested reader is directed to [2], [3] for a detailed discussion of the team management elements of our research. Team management helps ensure a suitable mixture of capabilities exists on a team to provide the task allocation process with the highest chance of succeeding. Although our methodology supports the formation of teams from collections of individual robots, we assume a mission will start with preformed teams.

Robots on a team compensate for lost or damaged team members by periodically performing a *role check* operation. In our framework, all wireless messages include the sender's current role and team information which nearby robots use to discover the composition of their team. Team composition information is timestamped, and robots that have not been heard from in some time are "forgotten". When a robot performs a role check operation, it calculates its suitability to fill the roles on its team. Under-filled roles, as determined by the desired team definition, are given a higher weighting to encourage a less suited robot to fill these roles. If, for example, the team coordinator on a team failed, a less suited robot could fill that role until a better robot is available. If the role check identifies a change in role is necessary, the robot implements it and informs its teammates of the change, potentially triggering role checks by other team members.

Through the role check operations, the members of a team are able to recognize situations where there is a deficiency in capabilities within the team. By changing roles, potentially to a less suited role, the members of a team attempt to fill

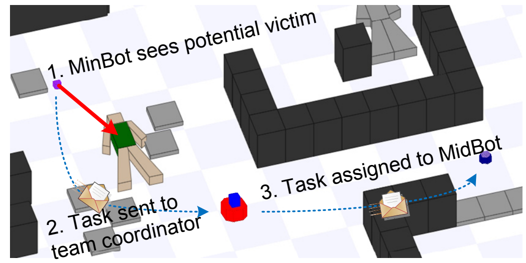

Figure 2. Example of task discovery and assignment.

the void created by the departure of another team member. This helps ensure task allocation can still take place. When robots take on roles they are less suited to the resulting task allocations are expected to be less ideal, where *ideal* means approximating the team responsibilities that would be assigned by a human expert. The team, however, is still able to make useful progress.

Team management also takes advantage of situations where robots on different teams encounter one another in the environment. Close physical proximity provides robots on different teams with the opportunity to visually identify each other and take advantage of short-range communication. During these encounters, the robots act as representatives of their team and exchange mission-related information, and knowledge of each other's teams. The mission-related knowledge exchange provides each team with an opportunity to learn of the work the other team has carried out, and helps reduce duplication of effort during the task allocation process.

After sharing knowledge of each other's teams, the representative robots have an opportunity to merge or re-distribute the robots between the teams. In the case of a lone robot encountering an established team, the robot has the opportunity to join the established team, or to form a new team of its own using some robots from the established team. Where two larger, established teams encounter one another, the merge and re-distribution provides an opportunity for both teams to make up for shortfalls in capabilities (as determined by the desired team definition). This means the task allocations have a better chance of succeeding, and that task allocations will tend to be closer to ideal.

Thus, teams are a fluid aggregation of robots where the robots switch roles within a team and change teams as necessary to make the best use of the available robots. 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, so that the teams can best carry out the tasks encountered in the environment.

V. Task Allocation

Given an evolving team with the potential for ongoing membership changes, our framework supports the identification, allocation, and completion of tasks in challenging environments. Every robot on a team plays a part in the task allocation process. As robots carry out work, they are also responsible for identifying new tasks in the environment to the degree their sensor equipment allows them to do so.

Figure 2 illustrates an example of the task discovery and assignment process. A robot identifies a potential disaster victim as it explores the environment. A task is created to confirm the presence of a victim in the environment. Assuming the robot discovering the task lacks the capabilities to carry it out, the task is sent to the team coordinator for assignment. Through the task assignment process, the task is assigned for completion by a robot with the necessary capabilities.

We assume each robot maintains a prioritized list of tasks. The task list is where each robot tracks new tasks it discovers in the environment, participates in the negotiation of new work to carry out, and assigns tasks it cannot carry out by itself to other agents. Finally, the task list provides a source of tasks for the robot to carry out itself.

A. Task Lists

Robots attempt to carry out the highest priority, oldest task from their task list first. Tasks are normally carried out through completion, with the exception of the case where a higher priority task arrives than the current one. In such a scenario the robot will suspend the current task and begin work on the higher priority task. The suspended task will be resumed once the higher priority task has been completed. In a disaster environment, for example, a robot might stop exploring an area if higher priority task to confirm the presence of a victim arrives.

We assume robots will attempt to maintain a small fixed size backlog of each task type to carry out (we chose a fixed limit of 5 for our implementation, but this of course can vary between robots and the estimated difficulty of tasks). The backlog of tasks helps ensure robots are able to continue carrying out useful work when communication conditions prevent the robot from participating in the task assignment process and acquiring more tasks. The fixed size of the backlog helps encourage the tasks to be spread out among other members of the team and encourages parallel operation. Maintaining a backlog of work also helps distribute tasks among all robots on a team, helping ensure a single damaged robot does not result in the loss of all tasks, which would occur if all tasks were stored by a single robot.

As a robot discovers tasks, it adds them to its task list. The robot uses the minimum requirements expression (Section III) of the task to determine if it is able to carry out the task on its own. If it is able to carry it out on its own, and the robot does not have an excessive backlog of this task type, the robot will retain the task for future completion. If the robot is not able to carry out the task on its own, either due to having a large backlog of that type of task or a lack of appropriate capabilities, the robot will attempt to send the task to the robot it believes fills the team coordinator role on its team.

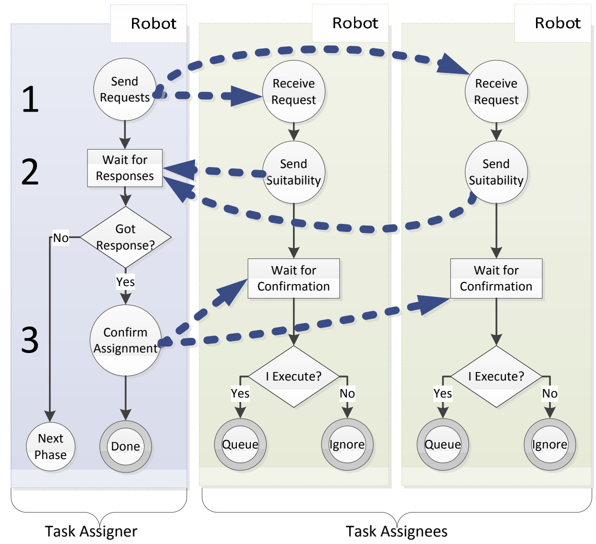

Figure 3. The task assignment process.

Poor communication conditions, or inconsistent knowledge of the robot's team can prevent the robot from sending a task to the team coordinator for assignment. Further, the robot could have become separated from its team coordinator. If there are other members of its team nearby, and the robot has the capabilities required to assign the task on its own, it will attempt to do so. Otherwise, the task will be retained until a point in time when it can be sent to an appropriate robot for assignment (i.e. either when it is within range of its own team, or when it joins a different team).

B. Task Assignment

The robot on a team filling the role of team coordinator is generally responsible for collecting tasks from team members and assigning them to suitable agents. This helps reduce duplication of effort, as the team coordinator has a more complete picture of the work required in the domain and is able to filter out duplicate tasks reported by different team members. It also provides a central point to collect the results of carrying out tasks.

The team coordinator will periodically attempt to assign tasks from its task list to other members of the team. For each task requiring assignment, the robot will first attempt to use a *role-based assignment* process. If no suitable robot is found using this process, an attempt is made to use a less restrictive *exhaustive assignment* process.

1) Role-based task assignment: The role-based assignment of tasks takes advantage of the fact that the definition of a role is in terms of the tasks a robot filling that role is normally expected to carry out. This eliminates the need to match task attributes exhaustively against the attributes of each team member, speeding the task assignment process. Role-based assignment is also advantageous because the agent performing task assignment can send messages addressed to specific recipients rather than broadcasting a request and hoping someone responds. Role-base task assignment occurs in three phases (illustrated in Figure 3): *sending task assignment requests*, *waiting for responses*, and *sending confirmations*.

During the first phase (Figure 3-1), the task assigner iterates through the robots $\{K_1, K_2, \cdots, K_n\}$ on its team it knows fill a role normally expected to be able to complete the task, offering the task to all potential assignees simultaneously. Each potential assignee K_i processes the incoming request and responds based on its current workload and capabilities.

As illustrated in Figure 3-2, the assignee first determines its suitability to carry out the task by evaluating the minimum requirements and suitability expression of the task against its own capabilities. If the assignee meets the minimum requirements, it checks its task list to see if it has reached the backlog limit for tasks of this type. If the assignee does not meet the minimum requirements for a task, or the task would exceed the backlog for tasks of that type, the assignee send a rejection message to the task assigner, indicating it cannot carry out the task. If the assignee meets the minimum requirements and its workload permits, it responds with a message indicating its acceptance of the task, along with its calculated suitability to carry out the task.

After a fixed period of time, the task assigner evaluates the acceptance messages it received (Figure 3-3) to determine the task assignee which is best suited to carry out the task (if any). A mission-specific cost determination is used to break ties between equally suited assignees. A cost could be calculated based on, for example, the distance from each assignee to the task location. This would ensure where multiple assignees are equally suitable to carry out a task, it will be assigned to the one physically closest to the task location.

A confirmation message is sent to the assignees indicating which assignee was chosen to carry out the task. Upon receipt of the confirmation, the chosen assignee marks the task as accepted and sends back an acknowledgment to confirm its acceptance of the task. At this point the chosen assignee is free to carry out the task as workload permits. The other assignees remove the task from their task lists upon receipt of the confirmation message. Where communication conditions are poor, it is possible an assignee may not receive a confirmation. A timeout is used to ensure the assignee only waits a short time for a response; where it does not receive a response in time it removes the task from its task list.

The task assigner waits for the acknowledgment message sent by the task assigner for a fixed period of time. If an acknowledgment is received, the assigner considers the assignment successful and removes the task from its task list. If no acknowledgment is received, task allocation moves on to the next phase.

It is possible poor communication conditions prevents the acknowledgment message from being received by the assigner. In such a scenario, there is a potential for the same task to be assigned to multiple robots. In the worse case there is a potential for duplication of effort due to multiple robots carrying out the same task. In the case of an exploration task in our example implementation, the path both robots travel to arrive at an exploration location will differ, resulting in an opportunity for both robots to discover new work along the way. In other domains, duplication of effort may be undesirable, and an implementation would need to explicitly take measures to minimize the impact of duplicate work.

2) Exhaustive Task Assignment: Due to communication failures, or the task assigner having inconsistent knowledge of its current team composition, it is possible for the role-based task assignment process (Section V-B1) to complete without finding a suitable robot to assign a task to. As shown in Figure 3, in such a scenario the task assigner will move on to the next phase of task allocation, *exhaustive task assignment*, where all robots in range are considered for assignment.

Exhaustive task assignment is not ideal, as it places extra load on the team to process the assignment requests and track responses. It also necessitates a greater reliance on communication between robots.

Exhaustive task assignment uses the same three phases as role-based assignment (see Figure 3). The distinction is that in exhaustive task assignment, a task assignment request is broadcast to any robot in range that can hear it – no specific addressing occurs. This helps broaden the pool of potential recipients, helping to mitigate against particularly poor communication conditions. Robots receiving the task assignment request on the assigner's team send back responses as in role-based assignment.

If after role-based and exhaustive assignment a task is still unassigned, it is re-queued on the assigner's task list so that an attempt can be made to assign it later.

C. Coping with Failures and Inconsistent Knowledge

Due to inconsistent team knowledge, it is possible a task assigner may not assign the task to the best suited robot on the team. The minimum requirements expression of a task ensures only robots that have a chance of carrying out a task will attempt to do so. As the team operates, knowledge of team structure and team composition will change. Thus, at some points in time task assignments will assign tasks to the best suited robots, while at others they will not. Our framework assumes assigning tasks to less suited robots is better than performing no task assignments at all.

During periods of particularly poor communication conditions, in addition to team knowledge being inconsistent between members of a team, the overall task assignments will have a tendency to fail. Poor communication will result in fewer task assignment requests reaching the intended assignee robots. This means the pool of available task assignees will be smaller than would normally be expected. Where none of the potential assignees meet the minimum requirements to carry out a task, a task will tend to remain unassigned. Further, where few robots meet the minimum requirements, there is a lower chance the available robots are well suited to carry out the task, resulting in less than ideal task assignments. The task backlog each robot maintains helps ensure that the robot has a backlog of work to be performed during these periods of poor communication conditions.

Where communication conditions are extremely poor, the result will be few discovered tasks sent to the team coordinator for assignment, and few tasks assigned to the team by the team coordinator. In such scenarios, robots must fall back to their backlog of tasks to ensure useful work is carried out.

When a robot becomes lost or separated from its team, it is possible for the robot to build up a backlog of discovered tasks which it lacks the capabilities to carry out on its own. Further, the robot may lack the necessary capabilities to assign these tasks on its own. In such a scenario the robot will retain the tasks it encounters, as its capabilities permit. As the robot continues operation, it can either re-encounter its current team, or potentially join another team in operation in the environment. At this point, the backlog of tasks would be sent to the team coordinator for assignment. Where the robot joins a different team, the unassigned tasks provide another means of transferring knowledge from one team to the other.

It is also possible for a robot to be destroyed, taking a backlog of tasks with it. Our methodology assumes these tasks become lost, and would be re-discovered by other robots as the mission progresses. In a domain where it is critical to ensure no tasks are lost, the team coordinator could retain knowledge of all tasks and reassign uncompleted tasks assigned to lost team members.

VI. Example Implementation

We implemented and evaluated our framework using a simulated USAR domain created using the Stage multi-robot simulator [12]. Our simulated domain tasks heterogeneous robots with exploring a damaged structure, while building a map of the environment and locating casualties. Robots can become separated from their team or damaged at any time, and new robots are released into the environment as time goes on. We used a modular design approach to ensure our implementation can form the basis of future work. To support our work, we made changes to the Stage simulator to provide simulated, unreliable communication between robots [2].

We chose to use simulation to study our approach, as our work if primarily concerned with supporting effective task allocation where teams evolve over the course of the mission. Using simulation in multi-robot research is a well established practice (e.g. [13] used simulated USAR environments).

Our implementation uses three types of robots chosen to demonstrate different physiologies, and an overlap of capabilities. *MinBots* are small expendable robots with limited sonar sensors for navigation and mapping. The MinBots have a wheeled physiology which restricts their movement to open areas. They can detect the potential presence of victims in the environment, but lack the memory or processing capabilities to coordinate a team. *MaxBots* are larger, complex robots with a tracked drive, enabling them to access areas of the environment the other robots cannot. Superior computational and memory capabilities make them ideal for coordinating a team and planning the exploration process. *MidBots* are wheeled robots sized between the MinBots and MaxBots. They possess specialized sensor equipment capable of confirming potential victim readings reported by the MidBots. Their computational capabilities provide them with the capability of coordinating a team, albeit less effectively than the MaxBots.

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.

VII. Evaluation

Our simulated USAR environments are $60\,\mathrm{m}$x$60\,\mathrm{m}$ in size and include two teams with four MinBots, two MidBots, and one MaxBot. The teams begin operation in opposite corners of the environment. 50 randomly positioned rooms ($5-12\,\mathrm{m}$ wide, and $5-12\,\mathrm{m}$ long) are distributed throughout the environment. 40% of the rooms are accessible by all robots, and 60% are only accessible by the MaxBots. The environment also contains randomly positioned debris (passable by MaxBots), and obstacles (impassable to all), accounting for 13% of the environment. Finally, 20 victims are distributed in the environment, and an additional 10 debris configurations resembling victims are included. The MinBots are unable to distinguish between the victims and debris configurations.

To evaluate our methodology, we performed an experiment to study the effectiveness of task allocation, given the fact team structure can change at any time. Our experiment includes two independent variables to control the communication success rate (20%, 60%, 100% success), and the probability of robot failure (minor, moderate, major).

Another independent variable controls 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, and provide additional resources which can augment the capabilities of existing teams, enabling more discovered tasks to be carried out.

We compared our methodology against two baseline cases. In the first, our task allocation mechanism as outlined in Section VI is employed, but robots cannot change roles, and team membership is fixed (robots cannot leave or join teams). Since there is no ability to improve a team by altering roles or membership, this provides a baseline performance for our task allocation mechanism alone, as opposed to the benefits achieved in tandem with team management.

The second base case also uses fixed roles and team membership, and includes a fixed mapping between tasks and robot types (e.g. similar to [10]). This baseline is a worst-case scenario, where there is no means of adapting to deficiencies in team composition. There is no flexibility to assign tasks to robots other than those robots ideally suited to carry them out.

Our experiment used a factorial design with a total of 8100 experimental trials. To help eliminate bias due to features of any one environment, we used 3 different environments. The environments were generated with a tool to ensure similar coverage, random victim distribution, and equal start distance between teams. Each experimental condition was repeated 50 times, and ran for 30 minutes of simulated time each.

To evaluate our methodology, we recorded two values at fixed points throughout each trial: the percentage of the environment explored not covered by impassable obstacles, and the percentage of victims successfully identified.

Figures 4 and 5 show the improvements realized using our framework in the percent of the environment covered and victims identified, respectively, over the baseline cases. Although our methodology shows an improvement over the base cases at the 20% communication success rate, it is important to note that the performance of all methodologies was actually quite poor. Too few messages were successfully delivered, resulting in a general failure to allocate tasks, and a tendency for teams to break apart over time.

At the 60% communication success rate, our methodology performed considerably better than the base case where tasks are mapped to robots in a fixed manner. The flexibility to assign tasks to less suited team members resulted in the team performing better overall. Where robots were able to change roles and teams, performance was further improved as teams were able to compensate for failures of robots by restructuring the team or forming new teams from remaining robots.

Victim identification showed larger improvements compared to the percent of the environment covered due to the smaller number of victim identification tasks compared to

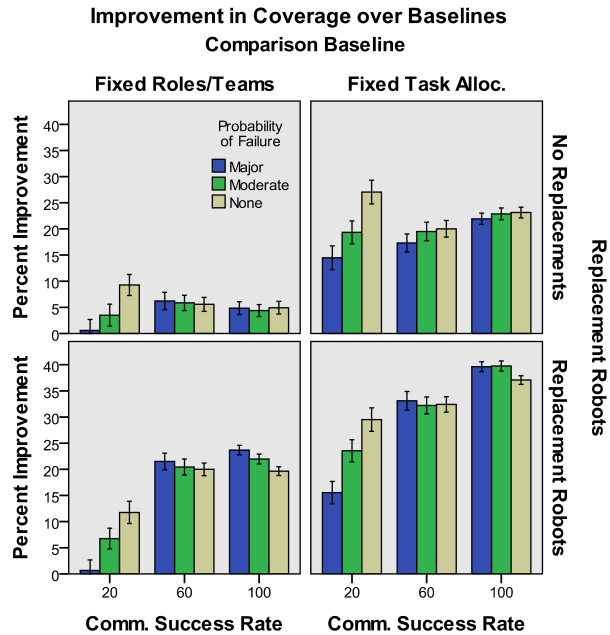

Figure 4. Improvement in coverage over baseline cases.

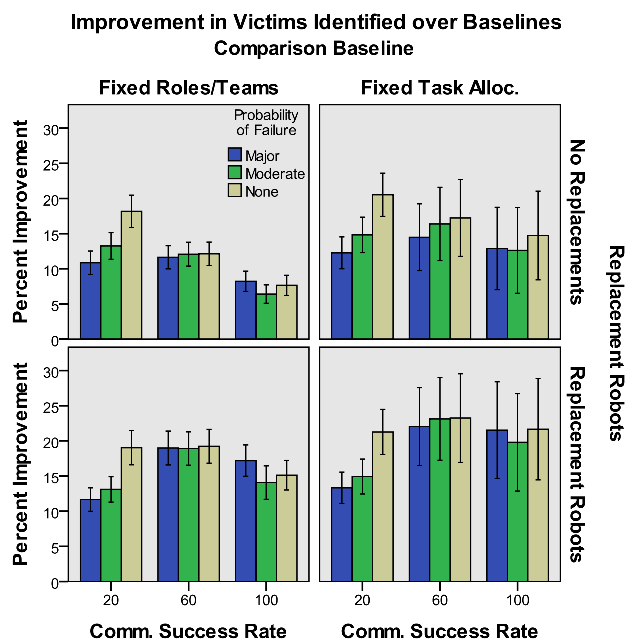

Figure 5. Improvement in victims identified over baseline cases.

exploration tasks. Being able to restructure teams helped ensure the relatively scarce victim identification capability was available more often, resulting in more successful task allocations.

Where replacement robots are available (Figures 4 and 5), our results show that our framework is able to more effectively allocate tasks, taking advantage of the new equipment by augmenting existing teams or forming new ones.

When comparing the improvements in percent of the environment covered against both base cases, it is interesting to note that the performance increases are generally higher when comparing against the baseline where task allocation is fixed. This is due to the fact our task allocation methodology is able to assign exploration tasks to less suited team members. Results for victim identification are similar due to the fact that our implementation has only one reasonable mapping for victim identification tasks.

VIII. Discussion and Future Work

Our methodology demonstrates strong benefits in allocating tasks to appropriate team members in the face of unreliable communication, and a changing team structure. Allocating tasks based on available resources helps ensure the team is able to continue making useful progress, despite a lack of important skills on a team. Allowing the task allocation responsibility to shift to other members of a team provides the ability to continue allocating tasks, despite the failure of a team coordinator. Where replacement robots are available, our task allocation methodology allows the replacement robots to be effectively utilized on the teams they join.

Our research has shown the benefit of using a task allocation methodology where tasks are allocated using role assumptions that also help guide the continuing evolution of the team structure. This ensured task allocation was still effective, despite the loss or failure of team members, and the introduction of replacement equipment.

Our approach and its evaluation have pointed out a number of interesting avenues of future research. First, our methodology assumes robots can suffer failures at any time, but does not actively attempt to re-assign tasks that were assigned to robots the team determines has failed. The rationale for this was that because the environment was assumed to be dangerous, enough robot loss would occur that it would be less resource-intensive to assume robots were lost after a time rather than actively tracking task performance. However, there may be cases where some tracking could avoid unnecessary task reassignment in spite of high robot losses.

It would also be useful to investigate viewing team recruitment as an allocatable task. Currently, team members are obtained by encountering them or balancing membership with another team. However, there may be situations where actively searching for a given type of robot may be more beneficial than allocating tasks in the interim to less-suited robots. The balance between effective task allocation and active management of team membership is a subtle one and deserves greater exploration.

Our current methodology also assumes a task is allocated to a single robot - an obvious extension would be the allowance of shared tasks that would require tight cooperation. Finally, more exploration outside of simulation is warranted.

This research highlights the importance of understanding the issues involved with effective task allocation in difficult and challenging domains. Because of the challenges inherent in these domains, robots cannot make assumptions about team structure or composition, and so must rely on effective task allocation to ensure work is completed. It is our hope that our work will encourage future research in this area.

References

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

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

[3] T. Gunn and J. Anderson, "Dynamic heterogeneous team formation for robotic USAR," in *Proc. of ANT-2013*, July 2013.

[4] P. S. Dutta and S. Sen, "Forming stable partnerships," *Cognitive Systems Research*, vol. 4, no. 3, pp. 211–221, 2003.

[5] M. van de Vijsel and J. Anderson, "Increasing realism in coalition formation," in *Proc. of CIRAS-2005*, Dec 2005.

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

[7] B. Gerkey and M. Mataric, "Sold!: auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct 2002.

[8] S. Sariel-Talay, T. Balch, and N. Erdogan, "A generic framework for distributed multirobot cooperation," *J. Intell. Robot. Syst.*, vol. 63, pp. 323–358, 2011.

[9] A. Gage, R. Murphy, K. Valavanis, and M. Long, "Affective task allocation for distributed multi-robot teams," Center for Robot-Assisted Search and Rescue, 2004.

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

[11] X. Ma, F. Meng, Y. Li, W. Chen, and Y. Xi, "Multi-agent-based auctions for multi-robot exploration," in *Proc. of WCICA-2005*, vol. 2, Jun 2006, pp. 9262–9266.

[12] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, no. 2, pp. 189–208, 12 2008.

[13] M. Eghbali and M. Sharbafi, "Multi agent routing to multi targets via ant colony," in *Proc of ICCAE-2010*, vol. 1, Feb 2010, pp. 587–591.