

Avoid communication outages in decentralized planning

Sameera Ponda^{*}, Olivier Huber[†], Han-Lim Choi[‡], Jonathan P. How[§]
Dept. of Aeronautics and Astronautics MIT, Cambridge, MA, sponda@mit.edu
Département EEA, ENS Cachan, olivier.huber@ens-cachan.fr
Div. of Aerospace Engineering, KAIST, Daejeon, Korea, hanlimc@kaist.ac.kr
Dept. of Aeronautics and Astronautics MIT, Cambridge, MA, jhow@mit.edu

Abstract—This paper proposes an extension to the Consensus-Based Bundle Algorithm (CBBA), a distributed task allocation framework previously developed at the Aerospace Control Laboratory (ACL), MIT, designed to address complex mission planning for a team of heterogeneous agents operating in a dynamic environment. A key assumption of CBBA involves having a connected network, a constraint which is not always feasible in realistic mission scenarios. In this paper, we present a mechanism that aims at preventing and repairing network disconnects. The dynamic network topology is predicted using available information within the planner and relay tasks are created to strengthen the connectivity of the network. By employing underutilized resources, the presented approach improves network connectivity without limiting the scope of the active agents thus improving mission performance.

I. INTRODUCTION

Mission planning consists of coordinating the behavior of a large networked team of agents, with heterogeneous capabilities, in order to efficiently perform a set of tasks. These tasks could include conducting reconnaissance and surveillance operations, target classification, rescue operations, etc [1]. Given the heterogeneous nature of the team, some agents could be better suited to certain tasks than other, leading to different roles and responsibilities within the mission. Ensuring proper coordination between the agents in the team is crucial to efficient mission execution, motivating the development of task allocation methods that improve mission coordination.

The task allocation problem is known to be a significantly complex combinatorial decision problem (NP-hard) with many constraints [2]. The tasks may have different locations and time-windows of validity or could require coordinated execution between several agents [3]–[6]. The agents may have resource limitations or specific agent-task compatibility requirements. Furthermore, the network configuration may be dynamic, with varying communication requirements. The autonomous task allocation process is depicted in Figure 1. The inputs to the task allocation planner include a task list, typically defined and broadcast by a Mission Control Center, the current network configuration, and agent models for planning predictions. The planning process then allocates the tasks among the team, producing schedules for each of the agents. As the agents interact with the “World” to execute the tasks, the network configuration, agent models and task list are updated to reflect changes in the environment. This planning

architecture must be executable in real-time to handle dynamic environments and communication constraints.

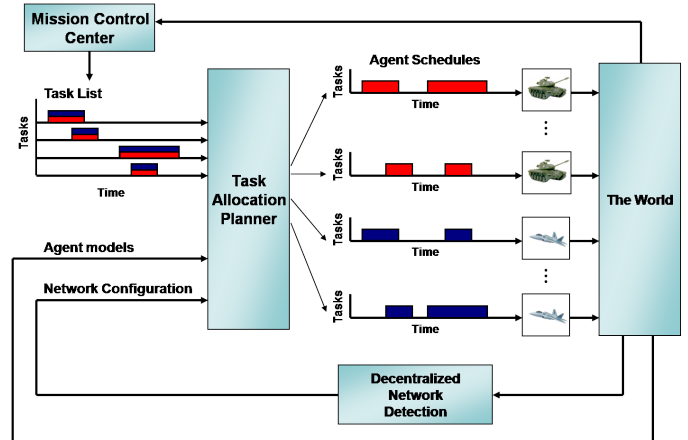


Figure 1. Real-time task allocation architecture for a heterogeneous team

Previous work on multi-agent multi-task allocation has focused on many variants of the Traveling Salesman Problem (TSP) and the Dynamic Vehicle Routing Problem (DVRP, DVRP with time-windows) [7]. Other approaches have focused on multi-agent coordination in dynamic environments with specific applications for teams of UAVs [3,8,9]. Most of these approaches involve solving large, complex combinatorial optimization problems where centralized methods quickly become infeasible and decentralized architectures must be adopted (see [8] and references therein). One class of decentralized combinatorial algorithms involves using auction algorithms augmented with consensus protocols to allocate tasks over a team of agents while resolving conflicting assignments locally among the agents [8]–[11]. Unfortunately, these decentralized algorithms require consensus among the agents, a process reliant on the network structure which deteriorates as the network connectivity decreases, causing degradations in team performance. To address this issue, research on limited communication environments has explored methods of maintaining network connectivity through relay teams [12,13], optimally placing vehicles in order to maintain communication links [14], and dealing with task deconfliction in disconnected networks [6].

This paper presents a real-time decentralized task allocation

framework for a team of heterogeneous agents performing in a dynamic environment. Varying network topologies and dynamic communication constraints are explicitly considered and approaches involving the use of relay tasks are implemented which strengthen the connectivity of the network. In particular, by employing underutilized resources, the presented approach is designed to improve network connectivity without limiting the scope of the active agents, thus achieving similar performance to a fully connected team.

II. PROBLEM FORMULATION

Given a heterogeneous team of N_a agents and a list of N_t tasks, the goal of the task allocation algorithm is to find a conflict-free matching of tasks to agents that maximizes a global reward. An assignment is said to be free of conflicts if each task is assigned to no more than one agent. Our current planning approach uses the Consensus-Based Bundle Algorithm (CBBA) [8], a distributed auction protocol that provides provably good approximate solutions for multi-agent multi-task allocation problems over random network structures. CBBA consists of iterations between two phases: a bundle building phase where each agent greedily generates an ordered bundle of tasks, and a consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents. A few core algorithmic features of CBBA that make it particularly well-suited to the problem at hand are that it is a decentralized decision architecture, a necessity for planning over large teams due to the increase in communication and computation overhead; a polynomial-time algorithm that scales well with the size of the network and number of tasks; and that it can incorporate various nonlinear agent models and scoring structures.

In order to execute the CBBA planning algorithm, the agents must communicate with each other in real-time through established communication links. For the sake of simplicity, we assume that the state of these links depends only on the distance $d_{ij}(t)$ between two agents i and j , where if this distance is greater than some fixed *communication radius*, R , the connection is broken. Specifically, the fleet can be described by a graph, $G = (V, E)$ [15,16], where the agents are represented by nodes, V , and the communications links by edges, E . We assume that the communication radius R is the same for all agents, implying that the graph, G , is undirected and symmetric. The elements of the graph are given by,

$$G_{ij} = \begin{cases} 1 & \text{if } d_{ij} < R \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

As agents move throughout the theater of operation these communication links are dynamically created and destroyed, leading to varying network topologies. As long as the network remains connected, i.e. there exists some communication path between every pair of agents, either directly or through neighboring agents, the CBBA planning algorithm will produce a conflict-free solution. However, if the network is disconnected, agents are not able to communicate their plans to each other

and conflicts may occur. Conflicting assignments negatively impact the overall mission performance, wasting fuel and time by assigning multiple agents to the same tasks [6]. Furthermore, certain tasks rely on maintaining a high bandwidth connection between agents for efficient data transmission. For example, surveillance tasks may require agents to stream video back to a base station for processing, which can only be accomplished if the agents maintain connectivity to the base station while the surveillance task is in progress.

An effective way to adjust the network connectivity in real-time is by maintaining or creating links using other agents as relays. Integrating this notion into the task allocation framework involves: predicting the network structure based on task location information and current agent's plans, and creating relay tasks in positions which will strengthen the network connectivity. In several scenarios, it is common to have agents with empty task lists for certain periods of time. These agents can be used to satisfy the relay requirements of the network. Furthermore, it is possible to have field agents whose specific purpose is to act as relays between more specialized action agents. The next section describes the integration of these relay tasks into the task allocation framework.

III. IMPROVING CONNECTIVITY FOR DYNAMIC NETWORKS

A. Forecasting the network topology

A relay task consists of a trajectory that an agent has to follow in order to be close enough to two (or more) disconnected agents in order to repair the local network connectivity. Our goal is to create relay tasks that prevent network disconnects, as well as to improve the connectivity, since CBBA converges faster when there is a stronger connectivity. These relay tasks are most effective when planned ahead of time, since agents can bid on them and integrate them into their bundles. This can be achieved by predicting the network structure at future times using the task locations and agents' expected path information.

In the implementation of the CBBA algorithm, each agent carries the *winning bid list* and the *winning agent list* [8]. Therefore, every agent knows the other agents' task assignments. Using this information the network structure can be predicted for any future time t . These predictions can be used to determine if an assigned task will cause the network to be disconnected, and appropriate relay tasks can subsequently be created to connect the network. An unfortunate problem is that the introduction of additional relay tasks could cause traveling agents or idle agents to change their paths to accommodate these relay tasks, thereby invalidating the predicted network topology. To mitigate this problem, the network prediction is only performed over the set of agents that are busy doing a task at time t , i.e. agents that are traveling to tasks or are idle at time t are not included in the network prediction. Therefore, if these "free" agents decide to execute relay tasks, the network prediction for the already assigned tasks will still be valid. Using this network prediction, it is possible for every agent to know what the disconnected subnetworks are (with respect to a base station). In other words, if there is more than one

subnetwork, then the subnetwork including the base station is considered the main network and the remaining subnetworks are assumed to be disconnected at that time.

B. Algorithm Description

The CBBA with relays algorithm is described as follows. For each set of tasks, the baseline CBBA allocation algorithm is executed. For each assignment obtained from CBBA, the network structure is predicted at the start time of the task. If execution of this task will cause a network disconnect, then an appropriate number of relays are created to connect the disconnected subnetwork to the main network (the number of relays required is a function of the communication radius R and the closest distance between the subnetworks). These relays are positioned between the 2 closest agents in the corresponding subnetworks and their time of validity is equivalent to the task time. This process is repeated for every assigned task, thus creating a set of all required relay tasks. CBBA is then rerun using the old tasks as well as these new relay tasks as inputs. Agents' bundles are "bootstrapped" to include their previous assignments, implying that they are not allowed to abandon already assigned tasks in order to accommodate relays. During the CBBA task selection process, if agents can accommodate relay tasks into their bundles, they place appropriate bids on these tasks. The value assigned to these relays is equivalent to the sum of the values of all the tasks this relay will help connect. Once CBBA has finished executing, each actual task is checked. If the corresponding relays for that task were all assigned then the task is allowed, if not, the task is marked as not allowed and all dependent relays are deleted. This process iterates until the assignment reaches convergence.

IV. SIMULATION RESULTS

To compare the performance of CBBA with relay tasks to the baseline CBBA algorithm a real-time simulation framework was created in MATLAB. The simulation architecture, depicted in Figure 1, consisted of the task allocation planner, a mission control center, a network detection algorithm, and vehicle managers. The mission control center supplied an initial set of tasks with time windows of validity to the agents, in addition to creating new pop-up tasks every 5 seconds. The task allocation planner received the dynamic list of tasks and allocated them to the agents, producing task lists for each agent. The network detection algorithm used the position of the vehicles to determine the network graph and the set of subnetworks at any given time. And finally, the vehicle manager consisted of a finite state machine which executed the tasks in each agent's task list. Screen shots of the simulation architecture are depicted in Figure 3.

The scenario used to compare CBBA with relays to the baseline CBBA algorithm used 5 agents, task durations of 5 seconds with 10 second time windows for execution, 10 initial tasks, and new pop-up tasks every 5 seconds. The vehicle speed was set to 0.5 m/s and the replan rate was every 5 seconds. In this scenario, agents were only able to execute tasks and receive rewards for them if they were connected

to the base station at the time of task execution. If they were not connected to the base, then they had to abandon the corresponding task and move to the next one in their task list. Figure 2 presents screen shots of CBBA with relays (top) and baseline CBBA (bottom), showing the agent paths and schedules as well as the current network connectivity (black dotted lines). The relay tasks are depicted in blue and the base station is shown as a red diamond. The top figure shows that the network is connected for CBBA with relays, while the bottom figure shows that there are 3 separate subnetworks using baseline CBBA, and 3 of the agents are not in communication with the base station. The score obtained in the CBBA with relays case is shown to be higher than that of the baseline CBBA case. Figure 3 shows plots of the mission score as a function of time as well as the number of disconnected agents as a function of time (with respect to the base station). As seen in the plots, CBBA with relays clearly outperforms the baseline CBBA, achieving a higher score at all times as well as a lower number of disconnected agents.

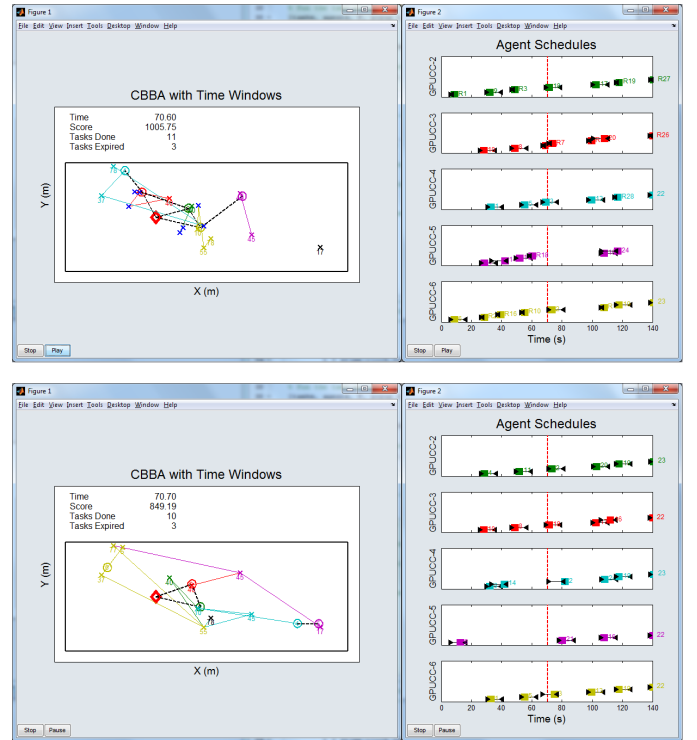


Figure 2. Simulation screen shots comparing baseline CBBA to CBBA with relay tasks

To further analyze the performance of CBBA with relays, a Monte Carlo simulation was implemented, utilizing the same scenario described above. The mission score and the average number of disconnected agents were compared for several different values of the communication radius. Figure 4 depicts the performance of CBBA with relays and baseline CBBA as a function of the normalized communication radius. As seen in the plots, CBBA with relays achieves higher mission

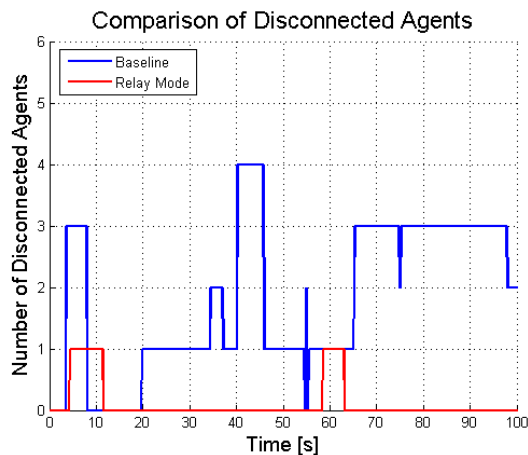
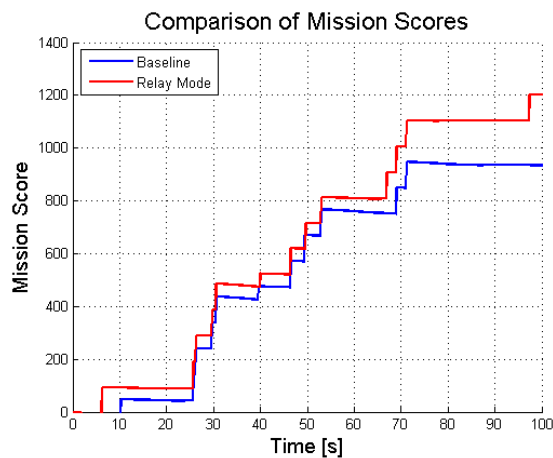


Figure 3. Single simulation run comparing baseline CBBA to CBBA with relay tasks

scores than the baseline CBBA. The score increases as the communication radius increases for both cases. For the CBBA with relays case, this is because the number of disconnects occurring is less and therefore more agents can perform actual tasks instead of acting as relays. For the baseline CBBA case, the increase in performance is due to a lower amount of conflicting assignments as the network connectivity improves. At a normalized communication radius of 0.3, the network remains mostly connected and both algorithms achieve similar performance. The lower plot shows that the average number of disconnected agents decreases as the communication radius increases for both cases, and that CBBA with relays achieves higher network connectivity than the baseline CBBA. These results demonstrate that CBBA with relays successfully addresses the problem of network disconnects, achieving high performance in weak communication environments.

V. CONCLUSIONS

In this paper, we presented an extension to the Consensus-Based Bundle Algorithm (CBBA) that aims at preventing and repairing network disconnects for a team of heterogeneous agents performing in a dynamic environment. Varying

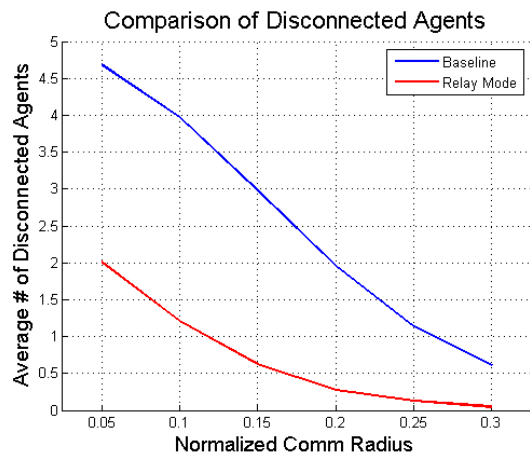
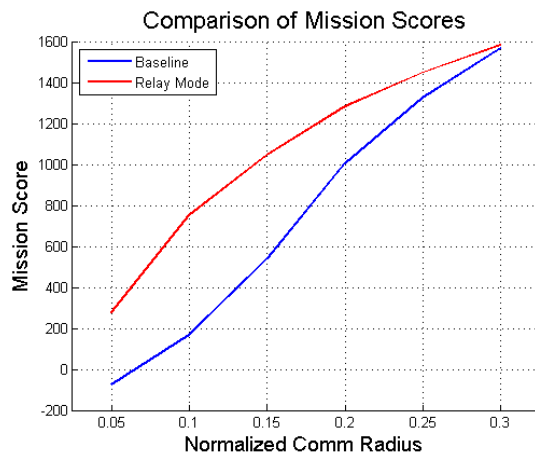


Figure 4. Monte Carlo simulation results comparing baseline CBBA to CBBA with relay tasks

network topologies and dynamic communication constraints are predicted and an approach involving the use of relay tasks is implemented to strengthen the connectivity of the network. In particular, by employing underutilized resources, the presented approach improves network connectivity without limiting the scope of the active agents thus improving mission performance.

ACKNOWLEDGMENTS

This research was supported in part by AFOSR (FA9550-08-1-0086) and MURI (FA9550-08-1-0356).

REFERENCES

- [1] "Unmanned aircraft systems roadmap, 2005-2030," Office of the Secretary of Defense, Tech. Rep., August 2005.
- [2] D. Bertsimas and R. Weismantel, *Optimization over integers*. Dynamic Ideas Belmont, MA, 2005.
- [3] M. Alighanbari, Y. Kuwata, and J. How, "Coordination and control of multiple uavs with timing constraints and loitering," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, June 2003, pp. 5311–5316 vol.6.
- [4] R. Dondo and J. Cerdá, "An MILP framework for dynamic vehicle routing problems with time windows," *Latin American Applied Research*, vol. 36, no. 4, pp. 255–261, 2006.

- [5] H.-L. Choi, A. K. Whitten, and J. P. How, "Decentralized task allocation for heterogeneous teams with cooperation constraints," in *American Controls Conference*, July 2010.
- [6] S. Ponda, J. Redding, H.-L. Choi, J. P. How, M. A. Vavrina, and J. Vian, "Decentralized planning for complex missions with dynamic communication constraints," in *American Controls Conference*, July 2010.
- [7] P. Toth and D. Vigo, *The vehicle routing problem*. Society for Industrial Mathematics, 1987.
- [8] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. on Robotics*, vol. 25 (4), pp. 912 – 926, 2009.
- [9] L. Bertuccelli, H. Choi, P. Cho, and J. How, "Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments."
- [10] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proceedings of the AIAA Workshop on "Integrating Planning Into Scheduling"*, 2005.
- [11] A. Ahmed, A. Patel, T. Brown, M. Ham, M. Jang, and G. Agha, "Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism," in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2005.
- [12] H. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. Spector, "Autonomous communication relays for tactical robots," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2003.
- [13] A. Ibrahim, K. Seddik, and K. Liu, "Improving connectivity via relays deployment in wireless sensor networks," in *Proc. IEEE Global Telecommunications Conference (Globecom 2007)*, 2007, pp. 1159–1163.
- [14] C. Dixon and E. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 281–291, 2009.
- [15] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. New-York, USA: Springer-Verlag, 2001.
- [16] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.