A Constraint Optimization Framework for Fractured Robot Teams *

Mary Koes†
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
mberna@cs.cmu.edu

Katia Sycara
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
katia@cs.cmu.edu

Illah Nourbakhsh
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
illah@cs.cmu.edu

ABSTRACT

In dangerous and uncertain environments initial plans must be revised. Communication failures hamper this replanning. We introduce *fractured subteams* as a novel formalism for modeling breakdowns in communication. We present a hybrid approach that employs distributed coordination mechanisms to provide robustness to these communication breakdowns and exploits opportunistic centralization. By modeling the problem as a mixed integer linear programming problem, we are able to apply constraint optimization techniques to efficiently find optimal or near optimal solutions to the difficult class of time critical tight coordination team planning problems. We then demonstrate that explicitly reasoning about communication failures through the incorporation of selective disruption minimization can improve team performance.

Keywords

Autonomous robots and robot teams, cooperation and coordination among agents, multi-agent planning

1. BACKGROUND

Teams of robots are necessary for a number of applications in which a single robot is unable or inefficient at accomplishing the goals. For example, time critical domains such as search and rescue, where lives are at stake, require careful consideration of plan efficiency. While some domains consist entirely of tasks that may be independently completed by individual robots permitting *loose coordination*, we are interested in the more challenging problem of *tight coordination* [2] where robots must work together to accomplish a *joint goal*. This requires robots to simultaneously solve the dual $\mathcal{NP}$-hard scheduling and task allocation problems with additional consideration for path planning [3].

The resulting domain of *time critical tight coordination team planning problems* are characterized by a set of robots, $\mathcal{R}$, with heterogeneous capabilities seeking to accomplish a set of joint goals, $\mathcal{G}$, with heterogeneous requirements. These goals are distributed throughout some physical environment, E. Additional system constraints, $\mathcal{C}$, may apply to robots, goals, or resources in the problem. The objective is to maximize team utility by accomplishing goals while goal rewards decrease over time to 0 at some time limit, T_{max}.

For teams of robots operating in physical environments, coordination inefficiencies may by magnified by the spatial constraints inherent to the problem. The time needed for robots to traverse the environment often dominates team performance. Consequently, it is beneficial to generate an optimal or near optimal plan at the start. In previous work [3], we describe COCoA, a Constraint Optimization Coordination Architecture, that maps this team planning problem to a mixed integer linear programming (MILP) problem and then combines domain specific heuristics and standard integer programming algorithms to efficiently generate optimal or near optimal plans. The resulting *team plan* generated by COCoA includes a schedule for each robot on the team broken down into the three components necessary to accomplish each of the robot's goals: time traveling to the goal, time waiting for teammates, and time working on the goal.

Unfortunately, few domains are so benign that this original plan can be successfully executed. Robots must interleave planning and execution as robots fail or are incapacitated, new robots join the team or receive new capabilities, goals are added and removed, goal requirements, rewards, or durations are changed, paths through the environment are found to be longer or shorter than believed, system constraints are added or removed, or the time limit is changed. Events that necessitate replanning are known as *replanning catalysts*. The discovery of replanning catalysts is beyond the scope of this paper but is discussed in other work [2].

Due to the physical nature of the environment, robots may not be in full communication at any given time. Although communication failure is a common phenomenon in multirobot coordination, models for these communication breakdowns are relatively primitive. Multirobot coordination architectures that address the problem assume any given message is lost with some uniform probability. However, since robots generally depend on peer to peer wireless

*This work is supported by NSF Award IIS-0205526. The authors would also like to thank the ILOG corporation for their generous support.

†Student Author

RF networks for communication, signal strength between two robots varies according to $\frac{1}{d^n}$ where d is the distance between the robots and n is the path loss exponent that is dependent on the environment [6]. Since robot communication is frequently based on TCP which is a reliable protocol, dropped packets are relatively unimportant. A more accurate model for robots which frequently have some limited range of communication is to model subsets of the team with good local communication but little or no communication between subsets. We introduce the terminology *fractured subteam* to refer to subsets of robots in the original team that are able to communicate with each other, directly or indirectly, but not with the rest of the team.

2. FRACTURED SUBTEAMS

A fractured subteam consists of a set of robots that jointly believe in a time critical tight coordination team planning problem, share a team plan, share a belief evolution model, are able to communicate with each other through transitive closure, and are unable to communicate with any robot not in the fractured subteam. A *belief evolution model* is a data structure that represents this fractured subteam's knowledge of beliefs of robots in other fractured subteams. Since all robots start with a common problem and team plan, all robots' belief evolution models contain a common element.

As robots seek to execute their team plan in dangerous and uncertain environments, they may discover that the problem and environment which they encounter does not match the problem for which they planned requiring them to refine their model of the problem and replan. This replanning must be performed quickly due to the time sensitive nature of the domain; COCoA has algorithms that can find good solutions in real time [3]. The second problem arises when failures in communication prevent the refined problem and solution from being disseminated to all team members resulting in inconsistencies. Fractured subteams force distributed replanning.

We extend COCoA from a purely centralized solver to a hybrid architecture. This hybrid architecture employs opportunistic centralization but is fundamentally distributed in that each fractured subteam is unable to communicate with other fractured subteams requiring distributed replanning. Each fractured subteam maintains its own knowledge from all previous subteams encountered which it uses for this replanning. Fractured subteams are themselves another variable in the system which must be considered while replanning.

Since COCoA was designed to be a centralized planner, only a single robot was required to possess the planning capabilities and complete problem and team plan. Other than this centralized planning robot, robots required only their individual tightly coupled schedule. In the distributed version of COCoA, each robot possesses full planning capabilities and a copy of the complete problem and team plan. The increase in communication cost is marginal. When a replanning catalyst is discovered, the architecture leverages the nature of a fractured subteam for opportunistic centralization. A robot from that fractured subteam is elected to replan and broadcast the results to the subteam.

Although the replanning is locally centralized, COCoA is distributed with respect to the individual fractured subteams. Each fractured subteam maintains its own problem and team plan. Each robot carries with it the complete knowledge of the fractured subteam to which it belongs. When two fractured subteams, F_1 and F_2, are merged into a single fractured subteam, their problems, team plans, and belief evolution models must likewise be merged. We have developed a simple algorithm for this merge, though we intend to investigate more sophisticated methods in the future since merging different beliefs is a deep research problem.

1. If the currently held problem and team plan of F_1 match the currently held problem and team plan of F_2, no changes need to be made.
2. If the currently held problem and team plan of F_1 match a state in the belief evolution model of F_2 and the team plan of F_2 is valid given the current status of the members of F_1 and F_2, all members of the merged fractured subteam adopt the problem, team plan, and belief evolution model of F_2.
3. By symmetry, the previous rule applies if the currently held problem and team plan of F_2 match a state in the belief evolution model of F_1.
4. If none of the above hold, create a new problem by combining the team planning problems of F_1 and F_2 and generate a new team plan to the resulting problem.

While full replanning works well in perfect communication, it tends to perform poorly with fractured subteams due to the inconsistencies in team plans. We can improve performance by explicitly reasoning about communication failure and fractured subteam composition during replanning and preferring solutions that only change the commitments of robots in the fractured subteam. The changes in coordination commitments are known as *disruption*.

One of the advantages of the MILP problem formulation in COCoA is the relative ease in combining multiple objective functions. We selectively minimize disruption by adding a cost function. There are several possible cost functions. We penalize near term changes more than long term changes since there is a higher probability that changes in the plan will be discovered in time for the robots to respond if the change is sufficiently far in the future. We then define our disruption objective function as follows. Minimize:

$$\sum_{m \text{ s.t. } G^m \in \mathcal{G}} [Cost(G^m) + \sum_{n \text{ s.t. } R^n \in \mathcal{R}} Cost(R^n, G^m)]$$

In general, every goal assigned to a robot represents a commitment to the team. However, not all changes to commitments are undesirable. If a robot is in the fractured subteam and aware of the replanning catalyst, some changes in the robot's schedule are advantageous as they increase system performance. Since the problem is tightly coupled, distinguishing between commitments that should be preserved and those that may be modified is challenging. We have developed *commitment graphs* to represent dependencies between robots in different fractured subteams. Commitment graphs can be easily generated from the team plan as described in Algorithm 1.

The cost function for each goal, $Cost(G^m)$, can be computed from the commitment graph and the current membership of the fractured subteam. The original time at which the goal was scheduled is denoted by $oldTime$. The new time at which the goal is to be scheduled (a variable in the MILP) is $newTime$, and the system time when replanning, is $replanTime$. If G^m has edges linking it to nodes of robots not in the fractured subteam and G^m was originally sched-

Algorithm 1 Build commitment graph

for all Goals $G^m \in \mathcal{G}$ such that $isScheduled(G^m)$ **do**
  Add goal node to graph
end for
for all Robots $R^n \in \mathcal{R}$ **do**
  Add robot node to graph
  $\mathcal{S}_{R^n}$ = last known schedule for R^n in belief evolution model
  for all Goals in this robot's schedule **do**
    Add edge with robot coordination commitments from robot node to goal node
  end for
end for
for all Temporal goal constraints $C_j \in \mathcal{C}$ **do**
  Add edge C_j between constrained goal nodes
end for

uled, canceling the goal results in a penalty of

$$\alpha_{cancel}\left(1 - \frac{oldTime - replanTime}{T_{max} - replanTime}\right)$$

and changing the start time results in a penalty of

$$\alpha_{insert}\left(1 - \frac{newTime - replanTime}{T_{max} - replanTime}\right)$$

The cost function for each robot not a member of the fractured subteam, $Cost(R^n, G^m)$, can also be calculated. If R^n is not currently allocated to G^m, the cost adding the goal into robot n's schedule is determined by equation (1).

$$\beta_{insert}\left(1 - \frac{newTime - replanTime}{T_{max} - replanTime}\right) \qquad (1)$$

Similarly, the cost of removing a goal from robot n's schedule is determined by equation (2).

$$\beta_{cancel}\left(1 - \frac{oldTime - replanTime}{T_{max} - replanTime}\right) \qquad (2)$$

We model the cost of changing the time of a goal in robot n's schedule as canceling and then adding a goal or the sum of equations (1) and (2).

The constants α and β may be tuned depending on the desired emphasis but should match the level of communication disruption in the environment. The new MILP combines the reward maximizing objective function [3] and the disruption minimization objective function.

3. EVALUATION

We conducted preliminary experiments to analyze the effect of fractured subteams on system performance. We used an abstract simulator with a randomly generated environment, 5 robots, and 10 goals. The environments are randomly partitioned into various blackout zones which impose the communication limitations, resulting in a set of fractured subteams. The number of fractured subteams is dependent on the topology of the environment and the composition of these subteams changes as robots traverse the environment. As an starting point for understanding the effect of communication breakdown on performance, we conducted a simple set of experiments in which one robot in the team was randomly disabled at some random time early in the plan.

We compared three different replanning strategies: *no replanning* where robots attempt to execute their original schedules, *full replanning* in which robots replan without consideration of communication failure, and *replanning with selective disruption minimization*. The results vary depending on the topology of the environment but we found that, in general, if the communication failures are minimal, full replanning outperforms the no replanning strategy, but as violations of the implicit assumption of full communication in the full replanning strategy increase, full replanning actually decreases team performance. By applying selective disruption minimization, as the number of partitions in the environment (communication failures) increases, the performance eventually converges to the *no replanning* strategy. Choosing α and β to match the amount of communication failure in the environment is an important challenge that we will address in future work.

4. RELATED WORK AND CONCLUSIONS

Market-based algorithms are a popular approach for multirobot coordination and have been successfully demonstrated in a variety of domains. A survey of the field [2] cites the lack of performance guarantees as one of the biggest drawbacks to current market-based approaches and suggests opportunistic centralization in future challenges. Our coordination approach addresses both of these issues.

In the MAS literature, token-based algorithms have been shown empirically to perform well in large scale teams [7]. However, token-based algorithms fail to provide the performance guarantees necessary for time critical domains. Existing work has assumed perfect communication. Pynadath and Tambe provide the COM-MTDP model for reasoning about the cost of communication [5] but do not explicitly model communication failure. The COM-MTDP model does not apply to time critical team planning problems since it uses a Markov assumption which does not immediately accommodate the time varying rewards or temporal constraints in our formulation of the problem. Distributed constraint optimization (DCOP) has been successfully applied to meeting scheduling [4] and to the multiagent plan coordination problem (MPCP) [1]. However, MPCPs require loose rather than tight coordination.

This preliminary work in developing an architecture interleaving planning and execution with communication failures for tightly coupled robot teams is promising and warrants additional investigation.

5. REFERENCES

[1] J. S. Cox, E. H. Durfee, and T. Bartold. A distributed framework for solving the multiagent plan coordination problem. In *AAMAS '05*, pages 821–827, New York, NY, USA, 2005. ACM Press.

[2] M. B. Dias, R. M. Zlot, N. Kalra, and A. T. Stentz. Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Pittsburgh, PA, April 2005.

[3] M. Koes, I. Nourbakhsh, and K. Sycara. Constraint optimization coordination architecture for search and rescue robotics. In *Submitted to the International Conference on Robotics and Automation*, 2006.

[4] P. J. Modi and M. Veloso. Multiagent meeting scheduling with rescheduling. In *Distributed Constraint Reasoning, (DCR) 2004*, 2004.

[5] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[6] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE J. Select. Areas Commun.*, 17(8):1333 – 1344, Aug. 1999.

[7] Y. Xu, P. Scerri, B. Yu, S. Okamoto, M. Lewis, and K. Sycara. An integrated token-based algorithm for scalable coordination. In *AAMAS '05*, pages 407–414, New York, NY, USA, 2005. ACM Press.