The Best Laid Plans of Robots and Men*

Mary Koes, Katia Sycara, and Illah Nourbakhsh
{mberna, katia, illah}@cs.cmu.edu
Robotics Institute
Carnegie Mellon University

Abstract

The best laid plans of robots and men often go awry. In dangerous and uncertain environments initial plans must be revised as robots fail, additional robots join the team, robots discover inconsistencies in their model of the problem, etc. 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 significantly improves team performance.

Introduction

Multirobot teams have many applications including search and rescue (Nourbakhsh *et al.* 2005), space exploration (Schneider *et al.* 2005), gallery monitoring and security sweeps (Kalra, Ferguson, & Stentz 2005). 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* (Dias *et al.* 2005) 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 (Koes, Nourbakhsh, & Sycara 2006).

The resulting domain of *time critical tight coordination team planning problems* are characterized by a set of robots with heterogeneous capabilities seeking to accomplish a set of joint goals with heterogeneous requirements. These goals are distributed throughout some physical environment. Additional system constraints 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. 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 (Koes, Nourbakhsh, & Sycara 2006), (Koes, Nourbakhsh, & Sycara 2005), we describe COCoA, a Constraint Optimization Coordination Architecture, that maps this team planning problem to a mixed integer linear programming problem and then combines domain specific heuristics and standard integer programming algorithms to efficiently generate optimal or near optimal plans.

Unfortunately, few domains are so benign that this original plan can be successfully executed. As robots fail, new robots join the team, goals are added and removed, etc., the plan must be refined. 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. 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.

While the nature of tight coordination problems does not easily allow robots to take advantage of distributed planning (Dias *et al.* 2005), the fractures in the team make centralized planning infeasible forcing distributed replanning. We describe a hybrid approach for coordination in which robots within a fractured subteam replan centrally in order to efficiently generate near optimal plans but each individual robot is capable of independently responding to deviations in the plan. Coordination within a fractured team is challenging because the fractured subteams introduce a new dynamic variable into the system which must be considered while replanning. Since other subteams are unaware of disturbances within the system, care must be taken to minimize disruption

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

to the schedules of these other robots.

This paper has three main contributions. We formally describe fractured subteams, a new model for communication failure in multirobot systems. We present a hybrid coordination architecture capable of generating near optimal plans yet robust to these communication failures. Finally, we show that explicitly reasoning about communication failures through selective disruption minimization may significantly improve team performance.

Related work

Market-based algorithms are a popular approach for multi-robot coordination and have been successfully demonstrated in a variety of domains. A survey of the field (Dias *et al.* 2005) 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. Dias et. al. (Dias *et al.* 2004) describe a market architecture capable of handling partial malfunctions, robot death, and communication failure by maintaining a model of tasks held by teammates and reallocating tasks if the owner loses a heartbeat. In contrast, the fractured subteam model expects certain communication failures, assumes that robots currently out of communication will honor their commitments, and explicitly minimizes changes to the plan that will cause these commitments to fail while maximizing team performance.

Token-based algorithms have been demonstrated in complex domains including search and rescue (Scerri *et al.* 2005) and UAV coordination (Xu *et al.* 2005). These algorithms have been shown empirically to perform well in large scale teams. However, token-based algorithms fail to provide the performance guarantees necessary for time critical domains as they do not consider path planning and separate planning and scheduling, leading to inefficiencies. Existing work has assumed perfect communication.

Pynadath and Tambe provide the COM-MTDP model for reasoning about the cost of communication (Pynadath & Tambe 2002) 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.

Recent work in distributed constraint optimization (DCOP), is promising and has been successfully applied to meeting scheduling (Modi & Veloso 2004) and to the multiagent plan coordination problem (MPCP) (Cox, Durfee, & Bartold 2005). However, MPCPs require loose rather than tight coordination. Existing DCOP algorithms ADOPT (Modi *et al.* 2003) and OptAPO (Mailler & Lesser 2004) lack support for real valued variables and so are insufficient for our problem formulation.

Background

We are interested in the time critical tight coordination team planning problem which is applicable in challenging domains such as search and rescue. The problem and a centralized solution are described in our previous work (Koes, Nourbakhsh, & Sycara 2005), (Koes, Nourbakhsh, & Sycara 2006) but are briefly described here for completeness.

Problem Statement

Before robots set out to accomplish goals, they need to formulate a team plan. The first step in this process is for the robots to agree on a belief of the current state of the world or the problem.

DEFINITION **1.** A **time critical tight coordination team planning problem** is denoted by ρ and consists of a five-tuple, $\langle \mathcal{R}, \mathcal{G}, E, \mathcal{C}, T_{max} \rangle$ where $\mathcal{R} = \{R^1, R^2, ..., R^N\}$ is the set of robots on the team, $\mathcal{G} = \{G^1, G^2, ..., G^M\}$ is the set of goals to be accomplished, E is the robots' representation of the physical environment, $\mathcal{C}$ is the set of additional constraints on the system, and T_{max} is the time limit for achieving the goals. Internal to this representation is the assumption that there exists a set of K capabilities relevant to the completion of the set of goals. Each robot is characterized by its current position and a set of binary capabilities it possesses. Each goal is similarly characterized by its location in the environment, the capabilities required to achieve the goal, the expected time required to achieve the goal, and the reward which we assume decreases over time to 0 at time T_{max}.

The system constraints, $\mathcal{C}$, may include temporal order constraints on the goals, constraints on shared resources, or constraints on the assignments of robots to goals. We describe a first order constraint language for combining goal, robot, and resource constraints in (Koes, Nourbakhsh, & Sycara 2006).

While the problem formulation imposes no constraints on the representation of the environment as long as robots can perform path planning in it, we find it useful to discretize the environment and represent it as a graph with nodes and edges which allows path planning using standard graph algorithms. Another possibility is to use an occupancy grid representation and apply D^* search (Dias *et al.* 2004).

The solution to the problem described above is called the *team plan* though it considers not only the allocation of robots to goals but also the schedule of when each goal is to be accomplished. In order to preserve the semantics of the original problem while maintaining a level of abstraction suitable for a robot with a typical three-tiered architecture, for example (Nourbakhsh *et al.* 2005), we define a team plan as a set of *tightly coupled schedules* containing one schedule for each robot. We preserve the original problem semantics through the goal oriented representation shown in figure 1.

DEFINITION **2.** A **tightly coupled schedule** represents the commitments of an individual robot, R^n to the team and consists of a set of steps, $\mathcal{S}_{R^n} = \{S_1, S_2, ...S_\omega\}$. Each step is defined as $S_i = \langle G^i, P, T_{start}, T_{travel}, T_{wait}, T_{work}, \mathcal{D} \rangle$ where G^i is the goal that the robot works on during step i, P is the path the robot plans to take to get to the goal, T_{start} is the time at which it begins the step, T_{travel} is the time allocated to traverse path P, T_{wait} is the time spent waiting for the other robots working on the joint goal to arrive, T_{work}

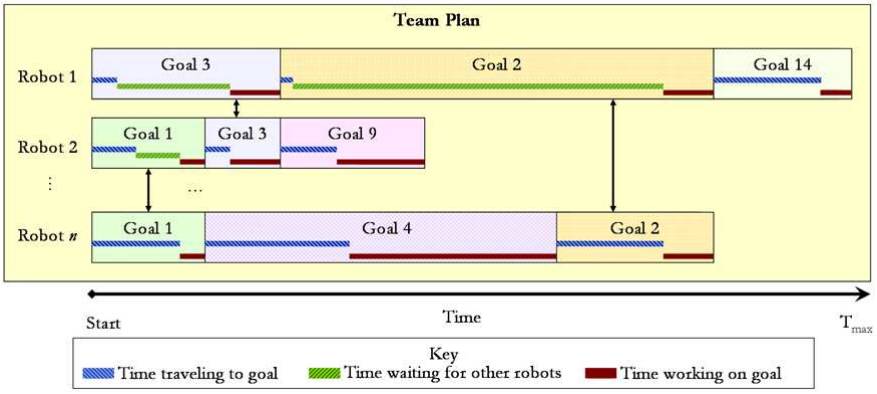


Figure 1: A team plan consists of the tightly coupled schedules from all robots in the team. The tightly coupled schedules for robots 1, 2, and n are illustrated above. All robots have three steps in their schedule so the planning horizon, $\omega = 3$. Each step contains $\langle G^i, P, T_{start}, T_{travel}, T_{wait}, T_{work}, \mathcal{D}\rangle$, where G^i is the goal the robot will work on during this step, P is the path the robot traverses to get to the goal (not shown here), T_{start} is the time at which the step begins, T_{travel} is the time traveling to the goal, T_{wait} is the time waiting for other robots, T_{work} is the time working on the goal, and $\mathcal{D}$ are the coordination commitments for the step. Inter-robot coordination commitments are illustrated with arrows.

is the amount of time allocated to achieving the goal, and $\mathcal{D}$ contains the coordination commitments for this step.

Coordination commitments explicitly represent the requirements of the robot for this goal. An example coordination commitment might be to provide capability 1 in order to achieve goal i. If another robot has the coordination commitment of providing capability 2 for goal i, an inter-robot coordination commitment exists. Coordination commitments signify not only $what$ the robot must do but why it is important.

The number of steps in a robot's schedule is the *planning horizon*, denoted ω. It is possible for different robots to have different planning horizons. It is also possible for robots to have no goal to work on during the later steps in which case the goal may be the *idle* goal.

DEFINITION **3.** A **team plan** consists of the set of tightly coupled schedules for all robots on the team,
$\mathcal{T} = \{\mathcal{S}_{R^1}, \mathcal{S}_{R^2}, ...\mathcal{S}_{R^n}\}$.

A sample team plan is illustrated in figure 1.

COCoA

We have developed COCoA, a Constraint Optimization Coordination Architecture, capable of finding efficient solutions to the time critical team planning problem as described above. The problem is modeled as a constraint optimization problem, specifically a mixed integer linear programming (MILP) problem. Although the problem is $\mathcal{NP}$-hard, modeling it as an MILP allows us to take advantage of state of the art advanced linear programming and mixed integer programming algorithms such as those provided by the commercial solver, CPLEX (ILO 2004).

There are three key enabling ideas behind COCoA. First, since path planning is typically inefficient in ILP solvers, we leverage the goal oriented nature of the team planning problem and preprocess all pairs shortest paths between goals using the Floyd-Warshall algorithm $O(n^3)$ (Cormen, Leiserson, & Rivest 1990). Second, in order to create a problem that is linear, we model goal rewards as decreasing linearly with time. Although there are multiple ways to model time varying rewards, we assume that at the time limit, T_{max}, all rewards are 0. Finally, we use a continuous time model to efficiently represent the scheduling problem.

The details of a basic problem formulation uses a mix of binary and real valued variables. We create scheduling variables for each element, $\langle G^i, P, T_{start}, T_{travel}, T_{wait}, T_{work}, \mathcal{D}\rangle$, for each step in the tightly coupled schedule for each robot.

Additionally, we use goal variables to represent overall team performance. Goals and robots are denoted with superscripts while subscripts denote a variable in the MILP problem formulation that points to the goal or robot. Therefore, G_m is the binary variable used to denote whether or not goal G^m is scheduled and $R_n G_m$ is the binary variable that indicates whether or not robot R^n works on goal G^m.

Finally, our problem formulation includes linking variables to tie the scheduling and goal variables together. The constraints that ensure a legal solution are discussed in (Koes, Nourbakhsh, & Sycara 2005).

The objective function maximized is the sum the reward functions from all tasks accomplished evaluated at the time at which they were accomplished (captured by the variable $G_m\_start^*$ which is equal to T_{max} if the goal is unscheduled). As discussed previously, we model rewards as decreasing linearly with time to 0 at the time limit T_{max} resulting in the following objective function:

$$Maximize: \sum_{m \text{ s.t. } G^m \in \mathcal{G}} \frac{T_{max} - G_m\_start^*}{T_{max}} Q^m$$

The resulting MILP can be solved using well established algorithms (ILO 2004). Incorporating domain knowledge

through heuristics further improves performance. We describe an anytime algorithm for time critical tight coordination team planning problems in detail in previous work (Koes, Nourbakhsh, & Sycara 2006).

Modeling communication failures with fractured subteams

The necessity of graceful degradation of performance in the face of communication failures is acknowledged by the multirobot community (Dias *et al.* 2005). However, many coordination systems assume perfect communication between team members. Those systems that do analyze communication failure assume that an arbitrary message will fail with a uniform probability. This model fails to capture the true nature of communication breakdowns in multirobot teams.

Robots generally depend on peer to peer or ad hoc wireless RF networks for communication. Work in the networking community (Bahl & Padmanabhan 2000), (Rodoplu & Meng 1999) indicates that the 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. Therefore, the ability of two robots to communicate varies depending on their proximity and environmental factors. Since robot communication is frequently based on TCP which is a reliable protocol, dropped packets are relatively unimportant. If robots are sufficiently close, they will be able to communicate reliably and otherwise they will be unable to communicate at all.

The networking community has extensively analyzed communication through message relays in connected and disconnected ad hoc networks (Rodoplu & Meng 1999). However, these models have not yet been applied to communication in the multirobot coordination domain. We introduce the formalism of *fractured subteams* which captures the elements important for multirobot planning while abstracting away the mechanics of the ad hoc network:

DEFINITION **4.** A **fractured subteam** consists of a set of robots $\mathcal{R}_F \subseteq \mathcal{R}$ that

1. Jointly believe in a common problem, $\rho_F = \langle \mathcal{R}, \mathcal{G}, E, \mathcal{C}, T_{max} \rangle$
2. Share a team plan $\mathcal{T}_F = \{\mathcal{S}_{R^1}, \mathcal{S}_{R^2}, ...\mathcal{S}_{R^n}\}$
3. Share a *belief evolution model*, $\mathcal{B}$, defined in definition 4.2
4. Are able to communicate with each other through transitive closure
5. Are unable to communicate with any robot not in $\mathcal{R}_{\mathcal{F}}$

DEFINITION **5.** A **belief evolution model**, denoted $\mathcal{B}$, consists of the set of tuples each containing a time critical tight coordination team planning problem, a team plan, and the set of robots known to have shared this model: $\mathcal{B} = \{\langle \rho_0, \mathcal{T}_0, \mathcal{R} \rangle, \langle \rho_1, \mathcal{T}_2, \mathcal{R}_{F_i} \rangle, ...\langle \rho_n, \mathcal{T}_n, \mathcal{R}_{F_j} \rangle\}$ Since we assume that all robots start with a common problem and team plan, all robots' belief evolution models contain a common element.

It follows from this definition that a robot always belongs to exactly one fractured subteam. A simple example of how a team may be partitioned into fractured subteams based on the topology of the environment is shown in figure 2 where robots 1 and 2 belong to one fractured subteam and robots 3, 4 and 5 belong to a second fractured subteam. If robot 2 crossed into node $[1, 3]$, it would become part of the second fractured subteam and would have to merge its problem and team plan with robots 3, 4 and 5. The beliefs and plan of robot 1 would remain unchanged. Fractured subteams are themselves another variable in the system which must be considered while replanning. At the same time, fractured subteams force distributed replanning.

$\mathcal{R}$: Robot	added removed incapacitated capability added capability removed
$\mathcal{G}$: Goal	added removed requirement added requirement removed reward increased reward decreased duration increased duration decreased
E: Environment path cost	shortened lengthened
$\mathcal{C}$: System constraint	added removed
T_{max}: Time limit	shortened extended

Table 1: Replanning catalysts are divided into categories based on on our definition of a time critical tight coordination team planning problem (Definition 3.1)

Coordination across Fractured Subteams

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. If this occurs, they must refine their model of the problem and replan. There are two main challenges in replanning. First, since robots are operating in time sensitive domains, the replanning must be performed quickly since planning itself delays execution. The second problem arises when failures in communication prevent the refined problem and solution from being disseminated to all team members resulting in inconsistencies. Before addressing our solution to these challenges, we identify the set of replanning catalysts or discoveries that necessitate replanning based as shown in table 1.

The discovery of replanning catalysts is beyond the scope of this paper but is covered in part in other work (Dias *et al.* 2005).

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 re-

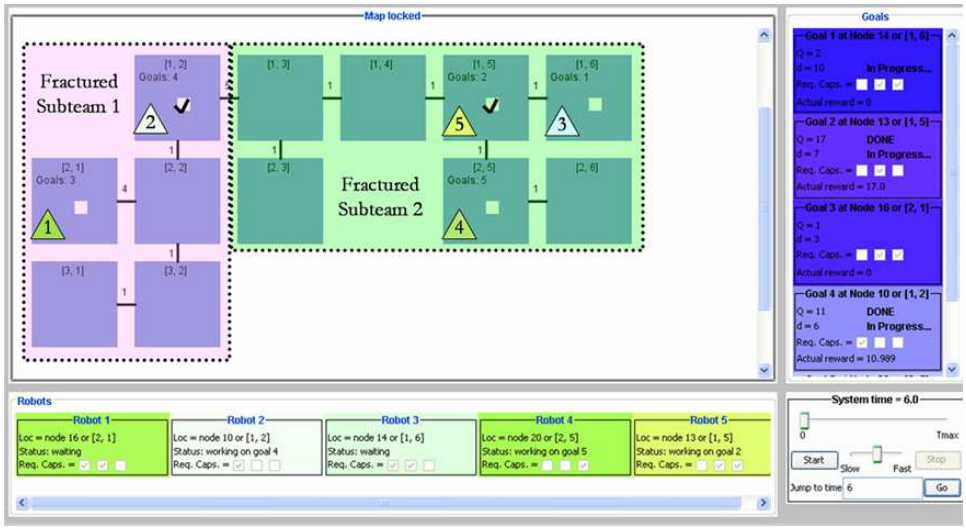
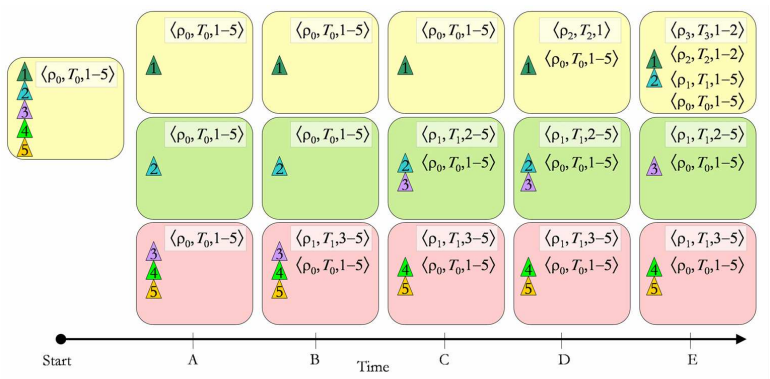


Figure 2: Abstract simulator with small example problem. Robots are illustrated as triangles and fractured subteams are illustrated by bounding boxes.



A: Communication failures result in 3 fractured subteams
B: Robot 4 in fractured subteam 3 is disabled resulting in replanning.
C: Robot 3 moves from fractured subteam 3 to 2 resulting in a merge and the dissemination of robot 4's failure.
D: Fractured subteam 1 is informed of a new goal and replans.
E: Robot 2 moves from fractured subteam 2 to 1; the merge requires replanning.

Figure 3: Example splitting and merging of fractured subteams. Rounded boxes indicate fractured subteams while triangles indicate robots. The belief evolution models for each subteam are shown with the robots for that model indicated as a range (i.e. $\langle \rho_0, \mathcal{T}_0, 1-5 \rangle$ indicates robots 1-5 (R^1, R^2, R^3, R^4, R^5) shared the initial problem ρ_0 and team plan $\mathcal{T}_0$).

planning.

This extension to COCoA has three main components. First, we have extended the architecture itself beyond the master-slave model embedded in the centralized architecture. Next, we provide mechanisms for distributed fractured subteams with independent beliefs and plans to split and merge. The final key to our successful hybrid architecture is that we explicitly reason about communication failure and fractured subteam composition during the opportunistically centralized replanning.

Robustness through redundancy

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 if information is broadcast. 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.

Dynamic fractured subteam formation

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 robots from two fractured subteams are merged into a single fractured subteam, their problems and team plans must likewise be merged. Merging different beliefs is a deep research problem that we intend to address in more detail in future

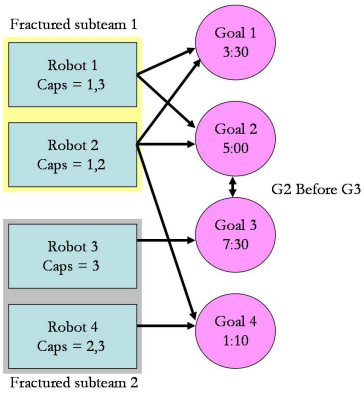


Figure 4: Sample commitment graph. Robot coordination commitments are omitted on the edges for clarity.

work. Presently, we use a simple approach for merging two fractured subteams, F_1 and F_2 described below:

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. The membership of the new fractured subteam is set to the union of the membership of F_1 and F_2.
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 (timestep C in figure 3).
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 merging the last common state in the belief evolution models of F_1 and F_2 with the differences between the common state and the problem of F_1, the differences between the common state and the problem of F_2, and the current status of the members of the merged fractured subteam. Generate a new team plan to the resulting problem. Update the belief evolution model of the merged fractured subteam to include the union of the belief evolution models for F_1 and F_2.

The merge process is illustrated on a sample problem in figure 3. The process of splitting a single fractured subteam into multiple fractured subteams is likewise illustrated in figure 3 and is much simpler than the merge process. Each new fractured subteam maintains its problem, team plan, and belief evolution model.

Replanning with Selective Disruption Minimization

While full replanning works well in perfect communication, it tends to perform poorly with fractured subteams. To illustrate this problem, consider a simple example with three robots. Robots 1 and 2 form one fractured subteam while robot 3 is in a second fractured subteam. In the original

Algorithm 1 Build commitment graph (figure 4)

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 Steps in this robot's schedule, $S_i \in \mathcal{S}_{R^n}$ **do**
    Add edge with robot coordination commitments, $\mathcal{D}$, of S_i from robot node to goal G^i
  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

plan, robot 2 was to perform a certain goal. Before robot 2 was able to complete the goal, it suffers a breakdown. Robot 1 observes this and replans. If robot 3 is closer to the goal than robot 1, the goal will be allocated to robot 3 while robot 1 remains idle. The problem arises because robot 3 is out of communication and unaware of its new commitment. If robot 1 had considered the communication failure while replanning, system performance would have been significantly better even though allocating robot 1 to the goal is suboptimal by our original objective function.

We explicitly reason about communication failure and fractured subteam composition during replanning and prefer solutions that only change the commitments of robots in the fractured subteam. The changes in coordination commitments are known as *disruption* (Bartold & Durfee 2003). Our approach is inspired by Bartold and Durfee's work for disruption minimization for a specific class of multiagent coordination problems based on weighting changes to blocking and synchronization commitments. Since the commitments in the time critical tight coordination team planning problem are significantly different than the multiagent coordination problem in (Bartold & Durfee 2003), we have developed our own metrics for measuring disruption.

Before quantifying change, we must first establish coordination commitments. 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 use *commitment graphs* described in Algorithm 1 and shown in figure 4 to determine where disruption should be minimized. Note that at this time we only consider disruption of temporal goal constraints (based on Allen's 13 temporal relationships (Allen 1983)) and do not model disruption of resource or robot constraints.

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 func-

tions. 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 minimization objective function as follows:

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

The cost function for each goal $Cost(G^m)$ is dependent upon the robots allocated to the goal represented by the commitment graph, the original time at which the goal was scheduled, $oldTime$, the new time at which the goal is to be scheduled (a variable in the MILP), $newTime$, and the system time when replanning, $replanTime$. The cost function for the goal can be constructed based on the following rules:

1. If G^m contains no edges between it and robots not in the subteam and the set of goal nodes reachable from goal node G^m contain no edges between them and robots not in the subteam, the cost is 0 (Goal 1 in figure 4).
2. If G^m is not scheduled and so has no node in the commitment graph, the cost is 0.
3. If G^m contains no edges between it and robots not in the subteam but is reachable from goal nodes with these links, the cost must be formulated to match the temporal goal constraints. Due to space constraints, we are unable to cover this topic in detail. Intuitively, however, the cost must match the nature of the constraint. In the example in figure 4, scheduling goal 2 earlier than 5:00 would have no penalty while scheduling it later would incur a penalty of

$$\alpha \frac{G_2\_start - 5:00}{T_{max}}$$

4. If G^m has edges linking it to nodes of robots not in the fractured subteam (goals 3 and 4 in figure 4) then if the old start time equals the new start time, the cost is 0. If the goal is cancelled, the cost is a constant:

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

Otherwise the cost equals

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

The cost function for each robot $Cost(R^n, G^m)$ can similarly be constructed by following simple rules:

1. If R^n is a member of the fractured subteam the cost is 0.
2. If R^n is not a member of the fractured subteam and is not currently allocated to G^m (independent of whether or not G^m is allocated), the cost is depends on whether the task is scheduled in the near term or long term:

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

3. If R^n is not a member of the fractured subteam but is currently allocated to G^m, the cost is 0 if the new time is the same as the previous time and R^n is still assigned to the goal. If R^n is not assigned to G^m in the new plan, the cost is a constant:

$$\beta_{cancel}(1 - \frac{oldTime - replanTime}{T_{max} - replanTime})$$

Otherwise the cost is determined by:

$$\beta_{cancel}(1 - \frac{oldTime - replanTime}{T_{max} - replanTime}) + \beta_{insert}(1 - \frac{newTime - replanTime}{T_{max} - replanTime})$$

The constants α and β may be tuned depending on the desired emphasis.

The new objective function can be combined with the existing objective function:

$$Maximize: \sum_{m \text{ s.t. } G^m \in \mathcal{G}} \frac{T_{max} - G_m\_start^*}{T_{max}} Q^m - \sum_{m \text{ s.t. } G^m \in \mathcal{G}} (Cost(G^m) + \sum_{n \text{ s.t. } R^n \in \mathcal{R}} Cost(R^n, G^m))$$

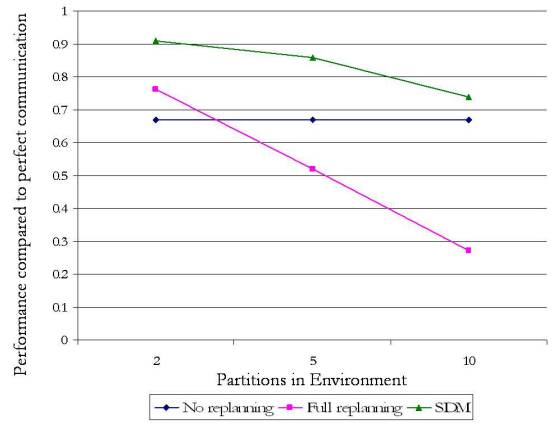


Figure 5: Comparison of no replanning, full replanning, and replanning with selective disruption minimization on a sample environment with 5 robots and 10 goals. One robot was disabled at an early randomly selected time. Strategies for replanning include *no replanning*, *full replanning*, and *selective disruption minimization* (SDM).

Evaluation

We conducted experiments to analyze the effect of fractured subteams on system performance. We used an abstract simulator (figure 2) 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. In the first strategy, *no replanning*, robots attempt to execute their original tightly coupled schedules despite knowledge that the problem is out of date. A second possible strategy is *full replanning* in which robots replan without consideration of communication failure or fractured subteam composition. Inefficiencies may arise since some robots on the team are unaware of the new team plan. The final strategy we consider is *replanning with selective disruption minimization*. The results vary depending on the topology of the environment. The results for a single typical environment are shown in figure 5. As expected, no replanning performs relatively poorly since the tasks assigned to the disabled robot are never accomplished. 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. Finally, by applying selective disruption minimization, we see that the degradation of performance with communication failure is significantly improved.

Conclusions and future work

In this paper, we introduced fractured subteams as a general team coordination model for communication failure in multirobot systems and in particular for tightly coordinated robot teams in time critical domains. We presented a coordination architecture that is distributed with respect to the fractured subteams but centralized with respect to the members of an individual subteam. Our planner explicitly reasons about the communication failure and the makeup of the fractured subteam during replanning and uses this information to selectively minimize disruption. While the results presented in this paper validate our assertion that reasoning about communication failure while replanning can improve performance, further experimentation in necessary to fully characterize this relationship. Our future work includes analyzing the impact of selective disruption minimization on other types of replanning catalysts besides robot failure and to determine the effect of disruption minimization parameters on system performance.

The work here touches on but does not explore several deep problems including failure detection, agent communication, and maintenance of shared beliefs. We also plan to investigate more sophisticated algorithms for merging fractured subteams, incorporation of additional system constraints into distributed replanning, and generating plans that are robust to failures.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26(11):832–843.

Bahl, P., and Padmanabhan, V. N. 2000. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, 775–784.

Bartold, T., and Durfee, E. 2003. Limiting disruption in multiagent replanning. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 49–56. ACM Press.

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. MIT Press/McGraw-Hill.

Cox, J. S.; Durfee, E. H.; and Bartold, T. 2005. A distributed framework for solving the multiagent plan coordination problem. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 821–827. New York, NY, USA: ACM Press.

Dias, M. B.; Zinck, M. B.; Zlot, R. M.; and Stentz, A. T. 2004. Robust multirobot coordination in dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Dias, M. B.; Zlot, R. M.; Kalra, N.; and Stentz, A. T. 2005. Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

ILOG. 2004. *ILOG CPLEX 9.0 User's Manual*.

Kalra, N.; Ferguson, D.; and Stentz, A. T. 2005. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proceedings of the International Conference on Robotics and Automation*.

Koes, M.; Nourbakhsh, I. R.; and Sycara, K. P. 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, 1292–1297. AAAI Press AAAI Press / The MIT Press.

Koes, M.; Nourbakhsh, I.; and Sycara, K. 2006. Constraint optimization coordination architecture for search and rescue robotics. *Submitted to the ICRA 2006*.

Mailler, R., and Lesser, V. 2004. Solving Distributed Constraint Optimization Problems Using Cooperative Mediation. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, 438–445. IEEE Computer Society.

Modi, P. J., and Veloso, M. 2004. Multiagent meeting scheduling with rescheduling. In *Distributed Constraint Reasoning, 2004*.

Modi, P. J.; Shen, W.-M.; Tambe, M.; and Yokoo, M. 2003. An asynchronous complete method for distributed constraint optimization. In *Proc. of the second international joint conference on Autonomous agents and multiagent systems*, 161–168.

Nourbakhsh, I.; Sycara, K.; Koes, M.; Yong, M.; Lewis, M.; and Burion, S. 2005. Human-robot teaming for search and rescue. *IEEE Pervasive Computing: Mobile and Ubiquitous Systems* 72–78.

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

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

Scerri, P.; Farinelli, A.; Okamoto, S.; and Tambe, M. 2005. Allocating tasks in extreme teams. In *Proc. of the fourth international joint conference on Autonomous agents and multiagent systems*.

Schneider, J.; Apfelbaum, D.; Bagnell, D.; and Simmons, R. 2005. Learning Opportunity Costs in Multi-Robot Market Based Planners. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Xu, Y.; Scerri, P.; Yu, B.; Okamoto, S.; Lewis, M.; and Sycara, K. 2005. An integrated token-based algorithm for scalable coordination. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 407–414. New York, NY, USA: ACM Press.