Time-extended Multi-robot Coordination for Domains with Intra-path Constraints

E. Gil Jones   M. Bernardine Dias   Anthony Stentz

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
Email: egjones,mbdias,axs@cs.cmu.edu

Abstract— Many applications require teams of robots to co-operatively execute complex tasks. Among these domains are those where successful coordination solutions must respect constraints that occur on the intra-path level. This work focuses on multi-agent coordination for disaster response with intra-path constraints, a compelling application that is not well addressed by current coordination methods. In this domain a group of fire trucks agents attempt to address a number of fires spread throughout a city in the wake of a large-scale disaster. The disaster has also caused many city roads to be blocked by impassable debris, which can be cleared by bulldozer robots. A high-quality coordination solution must determine not only a task allocation but also what routes the fire trucks should take given the intra-path precedence constraints and which bulldozers should be assigned to clear debris along those routes.

This work presents two methods for generating time-extended coordination solutions – solutions where more than one task is assigned to each agent – for domains with intra-path constraints. While a number of approaches have employed time-extended coordination for domains with independent tasks, few approaches have used time-extended coordination in domains where agents' schedules are interdependent at the path planning level. Our first approach uses tiered auctions and two heuristic techniques, clustering and opportunistic path planning, to perform a bounded search of possible time-extended schedules and allocations. Our second method uses a centralized, non-heuristic, genetic algorithm-based approach that provides higher quality solutions but at substantially greater computational cost. We compare our time-extended approaches with a range of single task allocation approaches in a simulated disaster response domain.

I. INTRODUCTION

Research efforts in robotics have increasingly turned towards using teams of robots to collectively address tasks. A team of robots acting together can often outperform single robot solutions in terms of quality and robustness; however, employing multiple robots requires coordinating robot efforts. Research in multi-robot systems has centered around domains where tasks can be addressed by single robots in isolation – we call these independent task domains. Some approaches to coordination in independent task domains exclusively reason about instantaneous allocation, where each agent is assigned only a single task at a time [4]. Other approaches use time-extended allocation, where agents are assigned a set of tasks to accomplish over a period of time [3] [1]. Reasoning about time-extended allocation can improve performance, as agents can discover synergies and dependencies between tasks. This

increased performance comes at a cost, however, as time-extended coordination introduces substantial challenges for task allocation. For one thing, agents must determine the order in which to perform a number of tasks, a potentially difficult problem that can adversely affect performance if done poorly. Additionally, there are exponentially more potential multi-task assignments than single-task assignments, making it infeasible to consider all possible multi-task allocations and orderings.

The problem of determining high-quality time-extended allocations becomes even more difficult in domains where constraints introduce coupling between agents' schedules; in these domains it can no longer be assumed that agents can independently and accurately determine their fitness for tasks. We are particularly interested in domains where the coupling between agents' schedules occurs at the path planning level. Specifically, we focus on domains where intra-path constraints prevent agents from moving freely within the environment, and where other agents can provide assistance to the constrained agents to permit passage. The presence of intra-path constraints makes it necessary for agents to recruit assistance in order to determine which paths to take through an environment and form accurate fitness estimates for tasks.

One domain where agents' schedules are coupled due to intra-path constraints is precedence-constrained disaster response, a compelling domain that we argue is not well-addressed by existing coordination techniques. In this domain a team of robots capable of extinguishing fires operates in a city that has been ravaged by a disaster of significant proportions. These fire truck agents are tasked with moving to various locations around the city to extinguish fires. In addition to causing fires the disaster has rendered many roads impassable; they are covered with debris and wreckage, creating obstacles around which the bulky fire extinguishing robots cannot navigate. Suppose that there is another group of bulldozer robots in the team that are designed to move freely in rough terrain, and can clear roads of wreckage and debris. We assume that the goal of the team is to maximize a global objective function value that is computed as a sum over time-decreasing rewards offered for extinguishing fires. As in independent task domains, to formulate a high-quality multi-agent plan a coordination solution must allocate fire fighting tasks to fire trucks. Unlike in independent task domains a

coordination solution must also allocate debris-clearing tasks to bulldozers so that fire trucks can quickly and easily move to task locations. Determining time-extended assignments of fires and debris piles requires reasoning about the routes that fire trucks will take and the allocation of intra-path constraint satisfaction duties to bulldozers, adding substantial complexity to the already difficult problem of multi-agent task allocation.

In previous work we made some strides in adapting market-based approaches to domains with intra-path constraints [6]. We used a novel method, tiered auctions, and a number of domain-specific bounding methods to efficiently search the space of task allocations, path plans, and intra-path constraint satisfaction duties. One deficiency of this previous work was that we considered only instantaneous assignment for fire trucks, assigning a single fire to each fire truck at a time. Considering only instantaneous fire truck assignment aids in bounding and helps to drastically decrease the size of the search space; however, it may result in an unacceptable loss in solution quality. Time-extended allocation has been shown to substantially improve performance in domains with independent tasks [1]; we believe that reasoning about time-extended allocation can also offer significant performance increases for domains with intra-path constraints.

Efficiently determining high quality time-extended solutions for domains with intra-path constraints is an open research question that is not well addressed by existing approaches to multi-agent coordination. Some previous work has explicitly considered domains where tasks are interrelated by precedence and simultaneity constraints [9] [8]. Other work has explicitly addressed domains where tasks require the joint efforts of multiple agents [15] [14] [12]. While these approaches support reasoning about precedence constraints and interdependent agent schedules, none of these approaches extensively explore time-extended allocation or are well suited for efficiently searching a space of possible routes and associated task assignment problems. Work in genetic algorithms for multi-agent coordination has been widely studied in vehicle routing [11] and for scheduling in both job-shop [10] and multi-processor [2] contexts, but has not been extensively applied to domains that require multi-robot task allocation, scheduling, and path planning. Shima et al. use a genetic algorithm to do time-extended allocation of tasks that may involve precedence constraints to a set of unmanned air vehicles [13]. Their problem domain does not involve precedence constraint consideration in path planning, however, and they demonstrate their work only in scenarios that involve a maximum of 10 tasks, making the scope much smaller than in our scenarios. Koes et al. use an MILP-based formulation for a search and rescue domain with similar elements to our disaster response domain [7]. While this MILP-based technique could give optimal solutions in our domain the algorithm complexity grows exponentially in the number of tasks and intra-path constraints, making it impractical for any but the smallest problems.

The central contributions of this paper are two novel methods for time-extended coordination in domains with intra-path constraints, each of which strikes a different balance between computation time and solution quality. In the first method, we extend tiered auctions to time-extended allocation using two heuristic techniques, clustering and opportunistic path planning, to efficiently search the immense space of multi-task assignments. This approach improves on the solutions achieved by our previous instantaneous assignment approach but takes somewhat more time to determine the coordination solution. Our second method is a centralized genetic algorithm (GA)-based approach which searches in the full space of coordination solutions, including possible routes for agents, using randomized evolutionary methods. This approach can determine higher quality solutions than time-extended tiered auctions if allowed sufficient running time, but takes orders of magnitude more computation time to do so.

In the next section we describe particular characteristics of the disaster response with intra-path constraints domain. We then detail our methods for time-extended coordination and evaluate their performance in a simulated disaster response domain. We finally conclude and suggest future work.

II. Domains Assumptions

For the purposes of this work we suppose that the goal of the disaster response domain is to maximize the sum of time-decreasing rewards offered for extinguishing fires at affected buildings. When a fire occurs at a building the entire value of the building is put at risk - the building's value becomes the maximum reward offered for addressing the fire. Fires can be of differing magnitudes, where the magnitude governs the linear rate of decay of the building value. The building value is exhausted when the linear decay causes the value to reach zero - no reward can be obtained from extinguishing the fire once the value reaches zero, though no associated penalties are levied. By these assumptions the reward for extinguishing a particular fire is the affected building value less the linear rate of decay multiplied by the number of cycles that have passed since fire onset. In this work the maximum task rewards and linear decay values will be drawn from independent normal distributions. We assume that other performance metrics, such as resource usage, makespan, or number of tasks addressed are not relevant - only maximizing the sum of individual fire rewards. Note that by these assumptions intra-path constraint satisfaction has no intrinsic value to the objective function, and is only useful in allowing access to fire locations.

We assume that each agent has full knowledge of the domain map including debris pile locations, that all fire reward characteristics, locations, and durations are known precisely, and that agents can reliably communicate. While handling greater uncertainty in domain knowledge is certainly a goal for future work, in this work we focus on a low-uncertainty environment; even in domains with low uncertainty the size of the coordination search space offers substantial challenges.

We model the fire trucks as operating on a graph-based representation of our road network - at any given time fire trucks either occupy nodes or are traveling along road edges between nodes. Those nodes can represent intersections, debris piles, or fires. Bulldozers are not restricted to the road network;

they must avoid buildings but are otherwise can move freely. We model bulldozers as operating in an occupancy grid-based representation of the environment.

III. Time-extended Coordination using Tiered Auctions

In this section we first describe our background work in tiered auctions for instantaneous allocation in domains with intra-path constraints. We then discuss how we use clustering and opportunistic path planning to enable tractable and efficient time-extended coordination.

A. Background

Market-based techniques have most frequently been applied to domains with independent tasks. One such approach is our work in a fire-fighting disaster response domain with independent tasks [5]. Our approach used a single tier of auctions; agents determine their task bids in isolation without consulting other agents. To employ a market-based approach in domains with intra-path constraints we needed a market mechanism that would allow agents to assess their fitness for tasks given that schedule quality depends on the schedules of other agents. Efficiency was the primary goal in system design, as the coupling of intra-task constraints and path planning causes a combinatorial explosion in problem complexity.

In order to allow agents to reason about intra-path constraints during the bidding process we added a second tier of auctions to the standard single-tier auction [6]. At the top auction tier fire allocation is determined; in order to bid on fires in those auctions fire trucks search through the space of possible routes, holding second-tier sub-auctions to solicit bulldozer involvement in clearing debris piles to satisfy intra-path constraints. To make the process tractable we structure the auction sequence and route search to permit extensive bounding of the search space.

In our approach during an auction cycle a central Dispatcher/Auctioneer holds auctions periodically to allocate fire fighting tasks. Fire trucks bid on the auctions based on determining the routes to tasks that will allow them to achieve the most reward in terms of the objective function for the task - by our assumptions this is equivalent to searching for the fastest route to the fire. Determining the fastest route to a particular task requires searching the space of possible routes and assigning bulldozers to clear debris piles along those routes. Debris piles are allocated using a series of sequential single item sub-auctions, where bulldozers bid based on adding debris clearing requirements to the end of their schedules. Bulldozers can potentially be assigned clearing duties in many different sub-auctions associated with different fire trucks, different fires, and different routes, though only a single set of assignments corresponding to the fastest route to a single fire will actually be adopted at the end of an auction cycle. Once fire trucks have bid the D/A awards a single fire to the truck that submitted the highest bid for any fire; that truck can then task bulldozers with clearing duties associated with the fastest route. The auction cycle repeats for any idle fire trucks.

Having each fire truck consider each possible route to every unallocated fire during an auction cycle would be prohibitively time consuming. We limit the search by having agents only consider routes that could potentially improve on the best bid submitted thus far. Additionally, we limit the search to routes that need to pass through fewer debris piles than a parameter $NumDebrisCons$; this serves to limit the search to paths which are more likely to yield fast passage and to limit over-exploitation of bulldozer resources.

As a final design consideration, we noted during testing that agents seeking to strictly maximize reward independent of time considerations tended to make somewhat inefficient decisions - going to a fire a long distance away that yielded slightly higher reward, for instance, rather than addressing a much closer fire. While this strategy is effective for maximizing the reward obtained for a single task, it doesn't take into account that the value available for all other tasks is constantly decaying. We determined that agents can obtain more reward over all tasks by trying to concentrate the reward they receive in a given duration and moved to using a model where agents seek to maximize their average reward per time cycle.

B. Methods

While our previous work in tiered auctions for domains with intra-path constraints achieved a substantial performance increase over single-tiered market-based approaches, considering only instantaneous allocations can limit the quality of the produced solutions. At the same time, the space of possible multi-task allocations is exponentially larger than the space of single-task allocations, making maintaining efficiency and tractability difficult. In our approach we attempt to heuristically determine the multi-task assignments that are most likely to yield good performance given the presence of intra-path constraints. We use two central observations in constructing our heuristics for determining likely multi-task assignments. First, we note that addressing a set of intra-path constraints that allow a fire truck to reach one task along a particular route may allow the agent to reach not only the target task but also other tasks near the target task. To exploit this observation we developed a clustering algorithm to group tasks that are located near each other and that are separated by only a few debris; any route from an agent's current location that allows the agent to reach some cluster member will also allow easy access to the other cluster tasks. Second, we note that along many potential routes to a cluster there may be fires that can be addressed with little or no delay to tasks later in the schedule. In fact, it may even result in higher average reward to take a slower path to a task if it allows an agent to address additional fires. We call this heuristic opportunistic path planning, and invoke it during route search. In the rest of this section we describe our two heuristic techniques for time-extended allocation and how they are incorporated into the tiered auction framework.

1) Clustering on Shared Preconditions: Our approach to clustering is to group together tasks that are spatially near each other and that are separated by a small number of debris; these tasks will then be auctioned as a group rather than

being auctioned individually. In our implementation clustering occurs at the auctioneer level. When initiating task auctions the DA runs a clustering algorithm over all unassigned tasks. The algorithm iteratively creates a new cluster centered at an unassigned fire task. It then runs a breadth-first search from the selected fire task node. The BFS considers all unassigned fire tasks connected by edges with a total length less than a distance parameter $ClusterDistance$ and that are separated by fewer debris than a parameter $ClusterDebrisMax$; all such fire nodes that have not been assigned to another cluster are included in the new cluster. A clustering of a number of tasks is shown in Figure 1.
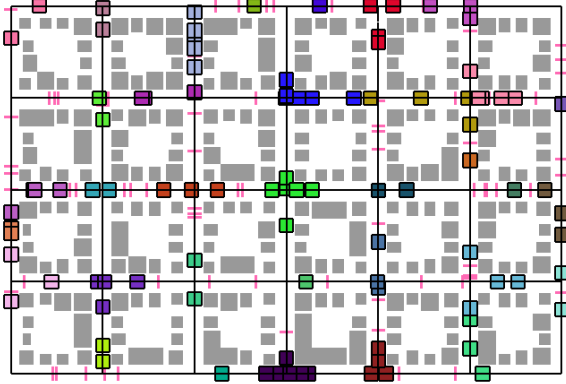

Fig. 1. Output of the clustering algorithm with $ClusterDistance = 2.5$ and $ClusterDebrisMax = 5$ on a disaster response scenario in a five-by-seven road network with 100 fire tasks (squares) and 100 debris (short lines that run perpendicular to the roads). Clusters are grouped by color.

There are a few important differences between the tiered auction for task clusters and the procedure described in Section III-A. First, the DA offers a cluster of tasks in each task auction instead of just a single task. Second, the fire trucks must now assess the reward that can be obtained from addressing a set of tasks rather than a single task. To do this the agent must determine a route to the cluster, the order in which to address cluster tasks, and the routes to take between cluster tasks. Selecting the optimal ordering of tasks within a cluster so as to maximize average reward is a difficult problem; we use a heuristic based on shortest path distance. When bidding on a task cluster a fire truck first sorts the cluster tasks based on each task's shortest path distance from the truck's current location without considering debris. It then selects the task with the shortest distance as its initial target and run the route searching sub-auction process with this task as the goal, producing an efficient fire truck route and corresponding bulldozer assignments. Next the closest task to the initial target task is selected as the next cluster task to address. The route and bulldozer assignments that were previously determined are passed into the route-searching sub-auction process along with the new target fire. Note that debris may lie between cluster tasks, so the new route search may involve assigning additional debris piles to the ends of the bulldozer schedules associated with previous route searches. The new search is explicitly linked to previous searches to enable bulldozers

to form accurate bids given that they may have assignments associated with previous cluster tasks. The process continues until the fire truck has a route that includes all fires in a given cluster. The fire truck then bids for the cluster based on the average reward accumulated for all cluster fires. The DA assigns all relevant cluster tasks to the agent with the highest bid for any cluster, repeating the auction if any agents remain untasked. The winning fire truck adopts the schedule associated with the winning bid and informs the bulldozers they should begin executing their assigned duties.

2) Opportunistic Path Planning: We next focus on our second heuristic approach to time-extended coordination - opportunistic path planning. In all of our tiered auction work described thus far the goal of route planning has been to determine which route will allow the fire truck to reach and extinguish a target fire as quickly as possible. In opportunistic path planning the goal of route search is instead to determine the route to the target fire that will allow the fire truck to accumulate the greatest average reward, where reward can come both from addressing the target fire and from attending to additional fires along or near the route. In many cases taking a somewhat more indirect route to a target fire may result in higher average reward. We use two different techniques to identify unallocated fires along particular routes that can be opportunistically added to the route. The first technique reasons about adding nodes that lie directly along the node sequence that constitutes the route. The second technique reasons about adding tasks that require slight deviations from the route. When considering whether to add a task assignment it must be determined whether or not the added reward achieved by addressing tasks along the route outweighs the reward lost by delaying the completion of tasks later in the route.

a) On-path tasks: The on-path segment of our opportunistic path planning approach reasons only about tasks that lie directly along the route being evaluated. In this algorithm we start the evaluation at the first node in the route. If this node is an unassigned fire we create a new route schedule that is an exact copy of the current route schedule except that the new schedule will include a stop at the unassigned task for the required duration to address the task. The rewards for the schedules are then compared; whichever schedule has the higher per-cycle average reward is adopted. The search continues until the end of the route schedule.

b) Off-path tasks: The off-path segment of our opportunistic path planning approach reasons about adding tasks that do not lie precisely along the given route but require only a slight deviation from the current route. We note that the primary points where a truck can potentially beneficially deviate from the route lie at the beginning of the route and at each intersection. Thus at each of these points we run a distance-limited breadth-first search; the search will continue until the distance limit parameter $OppMaxDistance$ is reached or debris is encountered. When the search finds an unassigned fire a new route schedule is created with the additional required detour to reach the candidate fire and return

to the route. If this new route schedule achieves greater average reward we adopt it; otherwise it is discarded. The process continues until no additions can improve the route.

c) Usage: Opportunistic path planning is used during the route planning portion of the cluster auction. Each route under consideration is passed first to the on-path method and then to the off-path method. The resulting schedule is then passed to the sub-auctioning routine so that a final average reward can be determined. If the route becomes part of a winning bid then the fire truck informs the D/A of all the unassigned fires it is claiming. A plan determined using opportunistic path planning and clustering is shown in Figure 2.
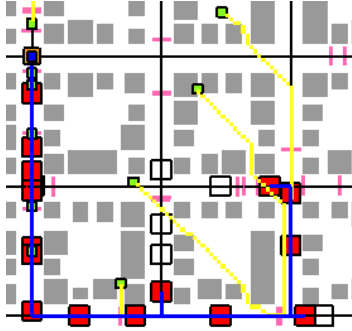

Fig. 2. An example from our simulator of a route determined using clustering and opportunistic path planning. Filled-in squares are fires that are scheduled to be addressed. Small green boxes are bulldozers with their associated paths. The square-within-a-square is the fire truck agent.

3) Bounding for Time-extended Allocation: In Section III-A we noted that a primary bounding technique used in tiered auctions with instantaneous allocation was designed to limit search to routes that could potentially improve on the best task bid. This method no longer is applicable in the time-extended approach, as determining a bound on the potential reward associated with a fire is impossible when opportunistic path planning is being used. Instead we use a method we call g-value bounding. The g-value bound for a particular fire is the time that the fastest route thus far discovered has taken to reach the site of a fire including any extra time taken for addressing additional fires or time spent waiting for debris piles to be cleared. The search for routes to a particular fire will terminate when the remaining potential routes are longer than the current minimum g value. The general intuition is that if a relatively fast path to a fire has been found then it is not necessarily effective to explore increasingly indirect paths, even though some high-value indirect paths go unconsidered. A g-value multiplier parameter can be set that will explore paths that are some factor larger than the real g-value bound, but in practice this does not seem to raise overall system performance and can greatly increase running time.

IV. TIME-EXTENDED COORDINATION USING GENETIC ALGORITHMS

Our time-extended tiered auction approach is designed to quickly converge to a desirable coordination solution; this

solution will in all likelihood be a local maximum in terms of performance. Our genetic algorithm approach, on the other hand, operates in the full space of possible coordinated plans and can converge to a global objective maximum if given sufficient time. Our GA approach is most appropriate for situations where a centralized solution is possible, where computation time is virtually unlimited, and where maximizing solution quality is paramount. We chose to use genetic algorithms as our randomized non-heuristic algorithm for two main reasons. First, they have been successfully used in both the vehicle routing literature [11], which involves elements of path planning and in job-shop scheduling [10], which involves precedence constraints. Second, some approaches, such as simulated annealing, focus on a single solution hypothesis, but genetic algorithms simultaneously maintain many disparate hypotheses, making searching large coordination spaces easier.

Our GA approach consists of the following algorithm, invoked at time zero:

1) Initialize the population of N genomes.
2) For a preset M generations:
   a) Evaluate genomes in terms of fitness.
   b) Select genomes for seeding the next generation.
   c) Crossover selected genomes.
   d) Apply mutation operators to genomes.
3) Take single highest fitness genome from any generation and use it to set agent schedules.

We will next describe our encoding of a problem solution into a genome as well as each of the above genetic algorithm components in more detail.

A. Encoding

The choice of encoding is a very important part of designing a genetic algorithm. The encoding needs to represent a complete solution to a given instance of a problem. In our case we need to specify the full sequence of actions taken by both the fire trucks and the bulldozers. We can represent fire trucks' actions as an ordered sequence of nodes that the fire truck will visit; this sequence can be turned into a full schedule using the methods we will discuss in Section IV-C. In addition to a node sequence each fire truck entry in a genome has a Boolean work vector, with an entry indicating whether the truck will attempt to address a fire as it passes. Note that we explicitly represent the routes that fire trucks but only implicitly represent the assignment of fire trucks to fires - determining which fires will be addressed by particular trucks only occurs during the evaluation phase. As bulldozers are not restricted to the road network the node sequence for each bulldozer needs only consist of a sequence of debris nodes that each will be assigned to clear.

B. Initialization

We use a random method to initialize genomes, both to seed the first generation and to introduce new individuals into the population in subsequent generations to maintain diversity. In this method for each fire truck in the agent population we select a random fire node and determine a shortest path to that

fire node independent of debris considerations. That shortest path node sequence becomes each fire truck's initial entry in the genome. Work vectors are initialized such that trucks will attempt to address all fires on the route. We initialize bulldozers using a simple market-based algorithm that seeks to minimize makespan; the market-based method leads to substantially higher initial population quality than randomly dividing debris clearing responsibilities among agents. This method insures that each debris is assigned to only a single bulldozer and that all debris are assigned to some bulldozer.

C. Fitness Evaluation

The next main component of our GA is the evaluation method - this method must take a genome and determine a fitness value for the genome. In our case the value we want is the total objective function score obtained given that the agents follow the sequences of actions specified by the genome. To determine this value we must simulate the actions of agents in the domain to turn the sequences into schedules, as we cannot determine completion times for tasks without considering how agents' routes or sequences interact. We use an event-driven deterministic simulation method that produces the exact objective score yielded by agents following the sequences indicated in the genome.

D. Selection

We use three methods to select individuals to populate subsequent generations. The first method we use is a binary tournament method. In this method two members of the genome are randomly selected, and the member with the higher fitness is put into the reproduction pool. In this method higher fitness individuals generally are selected but a tournament may involve two low fitness individuals, maintaining diversity. The second method is "Hall-of-fame" (HOF) selection. We maintain a list of a set size of the distinct genomes with the highest fitness that have existed in any generation during the search - this is the "Hall-of-fame." During HOF selection we pick a random individual from the hall and put it in the reproduction pool. The HOF method means that we spend a substantial portion of our time exploring promising hypotheses. Finally, we also can add a randomly generated genome to the pool.

E. Crossover

We do crossover for both fire truck node sequences and bulldozer node sequences. We pair each genome in the selected pool with another node such that each genome has a single partner. Fire truck and bulldozer crossover then occur with some probability. For fire truck crossover we need a method to swap some portion of two agents' node sequences. While work in genetic algorithms for vehicle routing uses somewhat heuristic approaches that attempt to insert sub-routes in beneficial points in agents' schedule [11], we chose to use a simple non-heuristic approach, leaving other approaches for future work. In our method we select a single fire truck in each parent genome for crossover. We select a random crossover point in each sequence and divide the schedules into front and back portions. For each parent we then create a new sequence consisting of its own front portion and the other parent's back portion, with a shortest path segment connecting the two. Corresponding work vectors are crossed over appropriately.

Bulldozer crossover is more complicated, as we want to preserve the genome property that all remaining debris tasks are assigned to one and only one bulldozer. To this end we used the crossover algorithm discussed in Mesghouni et al. in their work on genetic algorithms for job-shop scheduling [10].

F. Mutation

Each of the mutation operators acts on a single genome in isolation. We use four different mutation vectors that operate on the fire truck node sequences, a mutation operator for the fire truck work vectors, and three mutation operators for bulldozer sequences.
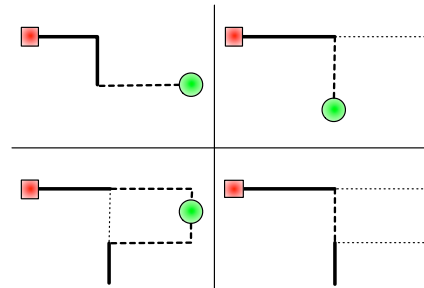

Fig. 3. Examples of the fire truck mutations: Append (upper left), Redirect (upper right), Reduce (lower right), and Lengthen (lower left). The square is the agent's location, the solid line the unaffected part of the original sequence, the smaller dashed line the replaced sub-sequence, the larger dashed line the new portion of the sequence, and the circle a randomly selected target node.

1) Fire Truck Mutation Operators: We could not find examples in the literature of mutator operators applied directly to agent routes. Our primary goal for our mutation operators was to have sufficient richness to take any route and mutate it into any other route in some finite number of steps. Each operator directly alters the path of a single fire truck, either adding to the end of a route (Append), shortening a sub-sequence in the current route (Reduce), expanding a sub-sequence of the current route (Lengthen), or altering the end-point of the route (Redirect). These operations are illustrated in Figure 3. Work vector mutation consists of, with some probability, switching each entry.

2) Bulldozer Mutation Operators: We chose to use three standard mutation operators from the job-shop scheduling literature [10]: swapping task order in a single agent's schedule, swapping two tasks between two different agents' schedules, and reassigning a task from one agent to another. All mutation operations preserve the unique assignment of debris.

V. Experiments and Results

In this section we describe our experimental setup and then present our comparisons of our time-extended approaches and several competing instantaneous allocation approaches.

A. Experimental Setup

Our simulation is designed to directly compare the different approaches. To make the comparison as accurate as possible we have agents in the different approaches operate on exactly the same domain instance: agents begin in the same locations, randomly generated debris are identically located, and the same set of fires must be addressed. This simulation method runs all agents within a single thread of computation.

We run all experiments in the seven-by-five road network shown in Figure 1, with a team consisting of three fire trucks and 12 bulldozers. 100 fires are distributed randomly, with initial reward values drawn from a normal distribution of $N(3000, 250)$ that linearly degrades at a rate drawn from $N(25, 5)$ units/cycle. Fires have zero duration, allowing trucks to extinguish fires immediately upon arrival; debris pile require ten cycles of bulldozer effort to clear. In our experiments we set four debris frequency levels of 0, 10, 200, 300; at the 300 debris level this translates to an average of over five debris per segment. In all cases debris are randomly scattered on the road network. For each debris frequency level we tested on ten randomly generated distinct domain instances.

B. Approach Specifications

We use two instantaneous allocation approaches that do not consider intra-path constraints during allocation to establish a baseline of performance. The first approach we call Independent Tasks. In this approach both fire trucks and bulldozers are assigned to move to the location of the nearest fire and nearest debris, respectively, taking the shortest path. We take care that assignments are unique. In this method there is no explicit coordination between agents. In our second baseline approach we use the "Allocate-then-coordinate" (ATC) approach as described in our previous work [6], where fires are assigned based on shortest path bids; after assignment trucks use the route sub-auction procedure described in Section III-A to find the fastest path to their assigned fire. The final two market-based approaches we compare are tiered auctions with instantaneous fire truck assignment (TA-IA) and tiered auctions with time-extended fire truck assignment (TA-TE). For both tiered auction approaches a value of 8 is set for $NumDebrisCons$. For the TA-TE approach we set $ClusterDistance = 2.5$, $ClusterDebrisMax = 5$, and $OppMaxDistance = 6.0$. In order to make sure that bulldozers are productively occupied even if not tasked by a fire truck we add an Idle behavior to all market-based approaches - if bulldozers are not tasked by fire trucks to clear particular debris they will be assigned to debris piles using the same method taken in the Independent Tasks approach. We test with three different parametrizations of genetic algorithms: 5000 population, 200 optimization generations; 10,000 population, 400 generations; and 20,000 population, 400 generations.

C. Results

We illustrate solution quality in Figures 4 and 5. Figure 4 shows approach reward averaged in proportion to an absolute upper bound on available reward. Performance differences are
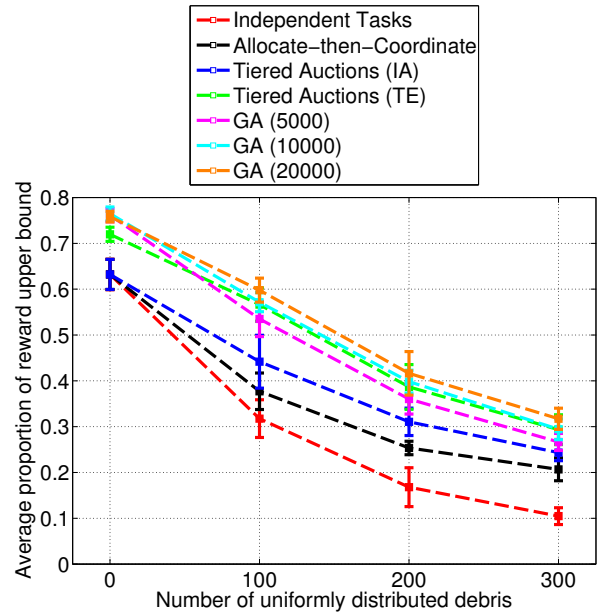

Fig. 4. Approach performance in a simulated emergency response domain. Standard deviations are shown as error bars. Values are shown as a proportion of an upper bound of available reward, where the bound is the summed reward of all issued tasks if they were addressed immediately.

clearer in Figure 5, where we plot how the approaches perform as an averaged proportion of the reward achieved by the best performer, GA-20000. TA-TE outperforms TA-IA by 24% averaged over the trials with debris, exceeding the performance of the baseline instantaneous allocation approaches by an even larger margin. Genetic algorithms with a population of 20,000 outperform TA-TE by 7% over the trials with debris. With a population of 10,000 the GA achieves roughly similar performance to TA-TE, and with a population of 5,000 and fewer generations TA-TE outperforms the GA approach by 7% averaged over the trials with debris.

Total time taken to produce coordination solutions is shown in Figure 6. Time-extended tiered auctions use substantially more computation time than do tiered auctions with instantaneous allocation except at the 300 debris level, where restrictive bounding makes the time-extended approach more efficient. All three settings of genetic algorithms use orders of magnitude more computation than any of the market-based approaches, underscoring that the heuristic techniques offers a good balance between performance and computation.

VI. Conclusion and Future Work

We have shown two different methods for improving performance in a precedence-constrained domain by reasoning about time-extended allocation. Our time-extended tiered auction system uses two heuristics methods to allocate groups of tasks that share preconditions; this approach substantially outperforms a tiered auction approach that uses only instantaneous allocation though may require additional computation time. We also present a genetic algorithm approach to time-extended coordination, the first we know of that uses genetic algorithms
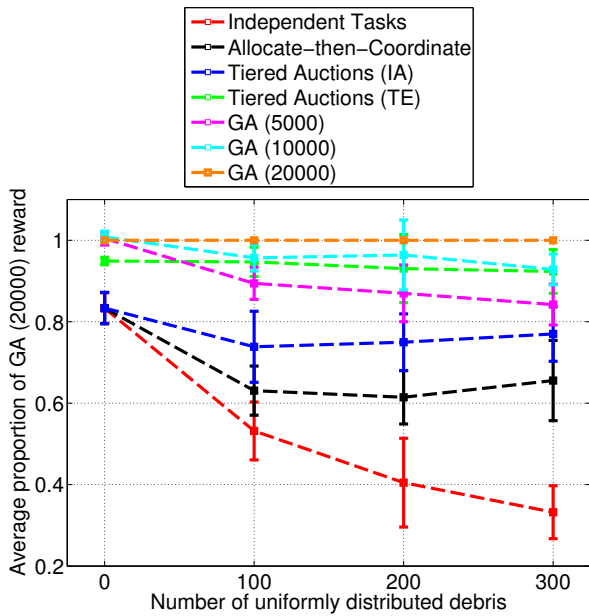
Fig. 5. Average proportion of reward achieved versus the performance of the GA-20000 approach on a per trial basis.
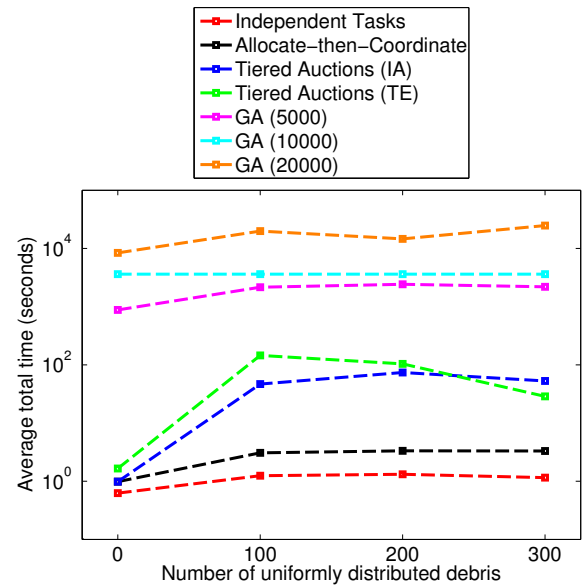

Fig. 6. Average total time taken to produce coordination solutions. Trials were all run in a single thread and allowed 100% of a CPU on a quad-core Intel Xeon 3.8 GHz CPU.

in a domain where allocation and path planning are coupled. The genetic algorithm approach with a sufficiently large population achieves better performance than time-extended tiered auctions at a cost of much greater computation time.

There are many avenues of future work that we would like to consider. First, we would like to extend our work to domains with higher uncertainty. One form of uncertainty we are particularly interested in addressing is dynamic task issue, where tasks at new locations are constantly being discovered. Time-extended coordination in such domains is especially difficult, as a balance must be struck between planning into the future and reacting to new information. We also are interested in extending this work to other forms of intra-path constraints, particularly those that involve simultaneity.

ACKNOWLEDGMENTS

REFERENCES

[1] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
[2] K. Dahal, A. Hossain, B. Varghese, A. Abraham, F. Xhafa, and A. Daradoumis. Scheduling in multi-processor system using genetic algorithms. In *Proceedings of the Genetic Engineering and Evolutionary Computation Conference*, June 2005.
[3] M. B. Dias, R. Zlot, M. Zinck, J. P. Gonzalez, and A. Stentz. A versatile implementation of the *TraderBots* approach to multirobot coordination. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, 2004.
[4] B. P. Gerkey and M. J. Matarić. Sold!: Auction methods for multi-robot control. *IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems*, 18(5), 2002.
[5] E. G. Jones, M. Dias, and A. Stentz. Learning-enhanced market-based task allocation for oversubscribed domains. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
[6] E. G. Jones, M. B. Dias, and A. Stentz. Tiered auctions for multi-agent coordination in domains with precedence constraints. In *Proceedings of the 26th Army Science Conference*, December 2008.
[7] M. Koes, I. Nourbakhsh, and K. Sycara. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2005.
[8] T. Lemaire, R. Alami, and S. Lacroix. A distributed tasks allocation scheme in multi-UAV context. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
[9] D. C. MacKenzie. Collaborative tasking of tightly constrained multi-robot missions. In *Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings of the 2003 International Workshop on Multi-Robot Systems*, volume 2. Kluwer Academic Publishers, 2003.
[10] K. Mesghouni, S. Hammadi, and P. Borne. Evolutionary algorithms for job-shop scheduling. *International Journal of Applied Math and Computer Science*, 14(1):91–103, 2004.
[11] M. A. Russell and G. B. Lamont. A genetic algorithm for unmanned aerial vehicle routing. In *Proceedings of the Genetic Engineering and Evolutionary Computation Conference (GECCO)*, June 2005.
[12] S. Sariel, T. Balch, and N. Erdogan. Incremental multi-robot task selection for resource constrained and interrelated tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
[13] T. Shima, S. Rasmussen, A. Sparks, and K. Passino. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers and Operations Research*, 33:3252–3269, 2006.
[14] F. Tang and L. Parker. A complete methodology for generating multi-robot task solutions using AsyMTRe-D and market-based task allocation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
[15] L. Vig and J. A. Adams. Multi-robot coalition formation. In *IEEE Transactions on Robotics*, volume 22 (4), 2006.