Learning-enhanced Market-based Task Allocation for Disaster Response

E. Gil Jones M.Bernardine Dias Anthony Stentz

CMU-RI-TR-06-48

October 2006

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Learning-enhanced Market-based Task Allocation for Disaster Response	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University ,Robotics Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this work we propose a market-based task allocation system for disaster response domains. We model the disaster response domain as a team of robots cooperating to extinguish a series of fires that arise due to a disaster. Each fire is associated with a time-decreasing reward for successful mitigation, with the value of the initial reward corresponding to task importance, and the speed of decay of the reward determining the urgency of the task. Deadlines are also associated with each fire, and penalties are assessed if fires are not extinguished by their deadlines. The team of robots aims to maximize summed reward over all emergency tasks, resulting in the lowest overall damage from the series of fires. We first implement a baseline market-based approach to task allocation for disaster response. In the baseline approach the allocation respects task importance and urgency, but agents do a poor job of anticipating future emergencies and are assessed a high number of penalties. We then propose a learning-enhanced market-based approach. Our regression-based technique modifies agents? bids resulting in an allocation that avoids many of the penalties assessed when using the baseline approach; by avoiding penalties and better respecting task importance and urgency the robot team achieves substantially higher overall reward. We illustrate the effectiveness of our approach in a simulated disaster response scenario.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**16**	

Abstract

In this work we propose a market-based task allocation system for disaster response domains. We model the disaster response domain as a team of robots cooperating to extinguish a series of fires that arise due to a disaster. Each fire is associated with a time-decreasing reward for successful mitigation, with the value of the initial reward corresponding to task importance, and the speed of decay of the reward determining the urgency of the task. Deadlines are also associated with each fire, and penalties are assessed if fires are not extinguished by their deadlines. The team of robots aims to maximize summed reward over all emergency tasks, resulting in the lowest overall damage from the series of fires. We first implement a baseline market-based approach to task allocation for disaster response. In the baseline approach the allocation respects task importance and urgency, but agents do a poor job of anticipating future emergencies and are assessed a high number of penalties. We then propose a learning-enhanced market-based approach. Our regression-based technique modifies agents' bids resulting in an allocation that avoids many of the penalties assessed when using the baseline approach; by avoiding penalties and better respecting task importance and urgency the robot team achieves substantially higher overall reward. We illustrate the effectiveness of our approach in a simulated disaster response scenario.

Contents

1 Introduction

Disasters have been a constant throughout human history, but the last several years have been especially damaging. According to the 2004 Red Cross World Disasters Report "over the past decade, the number of 'natural' and technological disasters has risen. From 1994 to 1998, reported disasters averaged 428 per year - from 1999 to 2003, this figure shot up by two-thirds to an average 707 disasters each year", and the death toll for 2003 "of nearly 77,000 was triple the total for 2002" [9]. While disaster response requires a variety of resources and skills crossing many disciplines, robotics and agent research communities can play a significant role in enabling efficient coordination of disaster response teams. Disaster response typically involves multiple individuals and teams working together to stem the effects of a variety of emergencies that arise from disasters, often under highly dynamic conditions. Thus, the disaster response domain presents significant challenges for task allocation.

The end goal of this work is a robot team that can be assigned response tasks in the wake of a disaster and will address those emergencies efficiently. A central part of efficient autonomous team operation is allocating tasks within the team. This work centrally concerns itself with creating a system for this task allocation. Disaster response domains present four inter-related challenges to efficient task allocation. The first challenge is that allocation must be highly online and dynamic as the system will be constantly inundated with new emergencies. The second challenge is to respect the relative importance and urgency of emergencies - if a choice must be made, robots should choose a more important task over a less important task, and should address emergencies with highest urgency first. The third challenge is that emergencies will have deadlines for successful completion, possibly creating an oversubscribed environment where it may be impossible for agents to address all emergencies by their deadlines.

A final challenge follows from the previous three. It may occur in the course of operation that an especially important and urgent emergency A occurs. Suppose that attending to the new emergency requires some robots to default on some of their current commitments. Respecting importance and urgency suggests that a robot should forego a less important task B to address the new emergency, but there is a cost associated with this failure. As the robot team is only one part of a larger response effort there are others who could potentially address B, but this can become difficult or expensive when B's deadline is near. Thus if the team cannot do a task it benefits the response effort to know this at the time of issue so that another team has maximal time to respond to the emergency. We assign a penalty to represent the cost to the larger response effort of the team committing to a task and failing to address it by the deadline. The final challenge then becomes anticipating future possibilities when considering the allocation of current tasks; a team that can predict its accomplishments while taking into account possible future tasks will be assigned fewer penalties. Precisely anticipating the future is impossible, especially given the uncertainty in disaster response domains, but a learning algorithm can exploit patterns in the distributions underlying emergency tasks.

Market-based allocation continues to grow in popularity in a variety of multirobot coordination applications [3] [4] [2] but have only recently been applied to the disaster response domain. The primary contribution of this paper is to propose a learning-enhanced market-based approach that includes a mechanism for learning to anticipate future possibilities when allocating tasks. We first propose a baseline market-based approach to task allocation that can efficiently perform online task allocation while respecting relative importance and urgency. The baseline solution has no mechanism however for anticipating future emergencies when performing allocation. This leads to agents committing to perform many low value, low importance tasks, which often either end in penalties or reduce agents' ability to address high importance tasks. The learning-enhanced approach uses a regression-based mechanism to allow individual agents to implicitly anticipate future emergencies in their bids for tasks. Our learning approach leads to allocations that allow the team to achieve substantially higher total reward by being assessed fewer penalties and by retaining the flexibility to address a greater proportion of high importance tasks and to address tasks more quickly.

The following section details related work. We then introduce the particular features of our fire fighting disaster response domain followed by a description of our baseline and learning-enhanced approaches. Experimental results illustrating the performance of both approaches in the fire fighting domain are presented next. We conclude with a summary of contributions and an exploration of future work.

2 Related Work

Market-based approaches have been applied effectively in variety of domains [3]. In some of these domains tasks arrive dynamically throughout execution, for example distributed sensing [2] and box pushing [4]. In none of these domains, however, do tasks have constraints such as deadlines with penalties for failure which cause the system to be oversubscribed. In most existing market-based research agents base their bids on current cost or reward estimates - this is analogous to our baseline approach. These approaches have no mechanism for considering future tasks in bids and will have difficulty with the fourth challenge detailed in Section 1. Two combinatorial auction approaches have been proposed for task allocation in the Robocup Rescue Simulation League, where a number of highly heterogeneous agents must be coordinated to respond to a disaster [7] [12]. While these approaches operate in the Simulation League domain, which includes online task issue and time-varying task rewards, neither approach uses a learning mechanism for improving allocation. In work by Koes et al. a Mixed Integer Linear Program (MILP) formulation is used to allocate emergency tasks with time-discounted rewards to a team of heterogeneous robots [6]. This approach can find optimal allocations given a set of tasks. However an optimal solution for a given set of tasks may become highly suboptimal when new tasks are introduced.

The Trading Agent Competition Supply Chain Management (TAC SCM) scenario has spurred substantial research in adaptive market-based approaches. Of special interest are approaches for adapting and optimizing bidding for customer orders [1] [8]. These approaches seek to predict probabilities of bid acceptance for variously priced bids and to determine optimal bids based on this information to improve one component of TAC SCM agents. While statistical learning techniques are employed to good effect in these approaches, the TAC SCM domain is a competition where each agent seeks to maximize profit at the expense of other agents; our domain involves cooperative agents working together to solve a problem, a very different problem than optimizing agents for TAC SCM.

We are aware of only one previous approach that uses learning to improve bidding over time in a collaborative multirobot market-based approach. Schneider et al. use a notion of opportunity cost to modify the bids of heterogeneous robots in a domain with time-discounted rewards but no deadlines or penalties [10]. This method serves to spread high-reward tasks among robots with different levels of capability, leading to an increase in the overall reward obtained by the team. Schneider et al.'s notion of opportunity cost is of primary benefit in domains with heterogeneous agents and the mechanism is unlikely to limit penalties in our disaster response domain.

3 The Fire Fighting Disaster Response Domain

We evaluate our allocation approach in a fire fighting disaster response domain. In the fire fighting domain teams of robotic fire fighting agents rove around a bounded city area extinguishing fires of various magnitudes that occur in a disaster zone and trying to prevent as much damage to structures as possible. New fires are continuously discovered at various buildings scattered around the city, and the objective score for a given fire depends not only on the value of the affected building but also on the magnitude of the fire. Penalties result when the team agrees to put out a fire but

fails to do so in the allotted time. Good performance in this domain requires coping with the four challenges detailed in Section 1.

In this domain we model the first challenge, continuous issue of new tasks, using a Poisson process, the standard distribution used in queuing theory to represent stochastic arrival times. The parameter λ represents the expected rate of task issuance, as governed by the Poisson probability distribution for given $x = 0, 1, \ldots$:

$$f(x, \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{1}$$

For the second challenge, importance and urgency are associated with building value and fire magnitude respectively. An efficient allocation should respect that a fire at a more valuable building is more important than a fire at a less valuable building, and that a high-magnitude fire is more urgent than a low-magnitude fire. In our experiments we have four building classes with four different Gaussian value distributions, ranging from the least valuable private residences to the most valuable malls. There are more low-value buildings than high-value buildings, so a fire is more likely to occur at a low value building. We also use four different magnitudes of fire, with alarms rated 1 to 4. Larger fires cause damage more quickly than smaller fires, take longer to extinguish, and occur less frequently. Though fires cannot spread in our domain there is still an interest in not letting large fires rage uncontrolled, so the deadlines for larger fires are nearer to issue and the penalties for failure greater. Therefore 16 possible pairings of building type and fire magnitude emerge, providing a rich space of importance and urgency.

4 Market-based Allocation for Disaster Response

In this section we describe our market-based approach to task allocation for the fire fighting disaster response domain.

4.1 Auction Mechanism

Incoming tasks are sent to a team dispatcher, who in our approach acts as an auctioneer. The dispatcher is the only auctioneer - agents do not re-auction tasks amongst themselves. As a new task T is issued, the team dispatcher starts an auction by issuing a call for bids containing all pertinent information about T. The call for bids is sent to all agents in the team. The agents construct a bid for the task (see Section 4.3) and return their bids to the dispatcher. The dispatcher then assigns the task to the highest positive bidder. If no bid is positive the dispatcher refuses the task - presumably some other team will handle the emergency. The dispatcher informs all agents of the outcome of the auction, and the winner adopts the task into its schedule.

4.2 Agent Schedule Optimization

Each agent keeps a schedule of all tasks to which it has been assigned, and each has the ability to optimize their schedules. As the reward function for tasks are monotonically non-increasing, an agent with one or more tasks on its schedule should never be idle - it should always be executing the first task in its schedule. Thus, scheduling entails choosing an ordering of the tasks that yields high summed reward.

Computing the value of a schedule is straightforward, depending only on the ordering of tasks in the schedule, the agent's current location, the current global time, and a method for computing travel time between goal locations. Our algorithm first computes the arrival time at the first scheduled emergency given the starting location and global time, and adds the task duration to get a scheduled task completion time. If that completion time is before the task's deadline then that task's reward given the completion time gets added to a running total and the algorithm computes the completion time of the next emergency task. If the completion time is after the task's deadline then the penalty is subtracted from the running total. As there is no benefit to the agent in moving to the location of the failed task, the algorithm will compute the completion time of the next scheduled emergency using the position and time of completion of the last successfully completed task.

We perform schedule optimization by either generating every possible sequence of tasks for sufficiently small schedules and choosing the highest reward (and thus optimal) schedule or using simulated annealing local search with a set number of iterations for larger schedules. The local search algorithm produces an optimized but possibly non-optimal schedule.

4.3 Agent Bidding

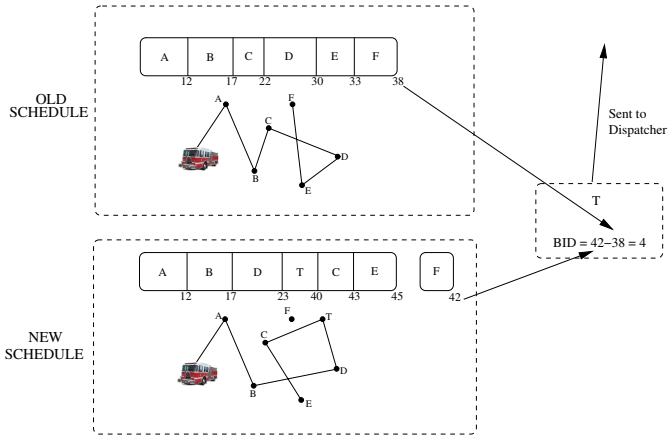

Figure 1: Baseline bidding for a new task T.

When an agent receives a call for bids it creates a new schedule consisting of both its complete old schedule and the new task. It then optimizes the new schedule as described in the previous section and determines the total summed value of the new schedule. The difference between the value of the new schedule and the value of the old schedule is the marginal schedule improvement M associated with the new task. In the baseline version this M value is returned to the auctioneer as the agent's bid. Note that M may be negative if incorporation of the new task into the schedule leads to a marginal decrease in reward. If communication is a concern a negative bid need not be returned to the auctioneer. See Figure 1 for an example of bidding.

4.4 Winning An Auction

When an agent is informed that it has won an auction it can replace its old schedule with the optimized schedule used in bidding. Any time the agent adopts a new schedule it is possible that some tasks assigned to the agent will not be completed by their deadlines. We assume that it is beneficial for the larger disaster response effort to have as much time as possible to cope with this intended failure - thus the dispatcher is informed that the task has failed, and can pass that information back to the proper authorities.

5 Learning-enhanced Market-based Allocation

Our learning approach is inspired by the performance of our baseline approach: agents often do not receive the value for a task that they expect when they are bidding on that task. To illustrate this observation, look at Figure 1. Suppose that task F in the old schedule in the figure was the last task the agent won. We can see that in its position in the old schedule the agent expects a reward of 5 for addressing F. If, however, the bidding agent wins the auction for task T with its bid of 4, then the actual reward received for F will be its penalty, -3. Thus the agent was originally expecting to make 5 when bidding on F, but actually received -3 for the task. If agents can learn to anticipate that some tasks tend to result in lower reward than when scheduled at bid time and modify their bids accordingly then this should lead to an overall increase in performance.

Our approach to learning is to use data accumulated by agents during the course of execution to construct a model mapping scheduled task value at bid time and a host of schedule features to actual value recorded for a task. Once the model is constructed the agents can use it during bidding to map from scheduled reward to predicted reward, and bid based on substituting the predicted value for the scheduled value. We use a support vector regression based approach to perform this mapping.

5.1 Training Data Collection

In all approaches agents collect training data during operation. Each time an agent wins a task from a task auction that agent records a feature vector derived from its bid for the task. The most important entry in the feature vector is the reward for the new task at its scheduled completion time. The rest of the feature vector is populated with salient features to help the regression from scheduled task reward to received task reward. We use the following entries in our feature vector:

1. The new task's scheduled slack - the number of cycles from the scheduled completion time of the task to the task's deadline.
2. The number of previously scheduled tasks in the agent's old schedule.
3. The total time taken for all tasks in the old schedule.
4. The marginal increase in schedule length between the old schedule and the new schedule.
5. The marginal difference in summed slack for all tasks between the old schedule and the new schedule.
6. The scheduled reward for the task.

We chose these features because they correlate with situations where a substantially different reward was received for a task than was scheduled at bid time. For example, if a task is scheduled near its deadline it means that it has a low value for feature one, scheduled slack. This means that any delay in the schedule due to the incorporation of the new task will likely result in failure for the low-slack task. Similarly, if feature four has a high value it means that the agent must add substantially to its current schedule to reach a task. A task that requires an agent to go substantially out of its way is less likely to be successfully completed.

The training target values are collected when the agent receives a reward for either successfully completing the task or when it fails to complete the task and the penalty is assessed. The agent adds the target value to the feature vector for the task and records the vector in a form suitable for the regression model generation program. We assume the data is held in a central repository shared among all agents, though if communication costs were a concern agents could keep individual training data files.

5.2 Learning a Model

Our chosen method of learning a regression model is support vector regression (SVR) [11] with a radial-basis kernel and an ϵ-insensitive loss function. We chose to use SVR as it is naturally well-suited to multivariate regression problems, is quite fast due to kernalization, and has been implemented in several freely available packages. Our choice of available implementations is `libsvm` in C, developed by researchers at National Taiwan University. We train an SVR model by passing the training data file to a `libsvm` training program. This produces a model file which can then be used to produce a predicted target value for a new feature vector.

There are two primary parameters we must set to use SVR - the width of the γ for the radial-basis kernel function we use and the cost parameter C for defining regression error. We used a grid search approach with cross-validation [5] to tune parameters. This cross-validation could occur online and automatically, but we did not find that minute adjustments to parameters resulted in a substantial difference in performance.

5.3 Bidding Using the Model

When a new task T is being auctioned each agent determines a marginal reward M. In the new optimized schedule T will have a scheduled reward S based on T's scheduled time of completion. The agent then computes a feature vector in exactly the same fashion as if generating training data. This feature vector, including the S value, and the model file are passed to `libsvm`, which generates the predicted P value for the task. The agent then substitutes the P value in the bid in place of the S value, giving a final bid of $M - S + P$. This process is illustrated in Figure 2.

5.4 Timing of Model Generation

Our learning approach depends on creating a model file. We have two different approaches to generating the model file. The first approach is off-line learning. In this approach, "Prelearning", we first generate data in several long experiments using the baseline approach. We then create the model outside the standard operation of the system, and then run new experiments using that model without alteration. As this approach is off-line, it is not useful for learning during operation, but provides a good method of testing the soundness of our approach. Our second approach learns in an online fashion. In the "Online Learning" approach, the agents initially use the baseline approach and bid based

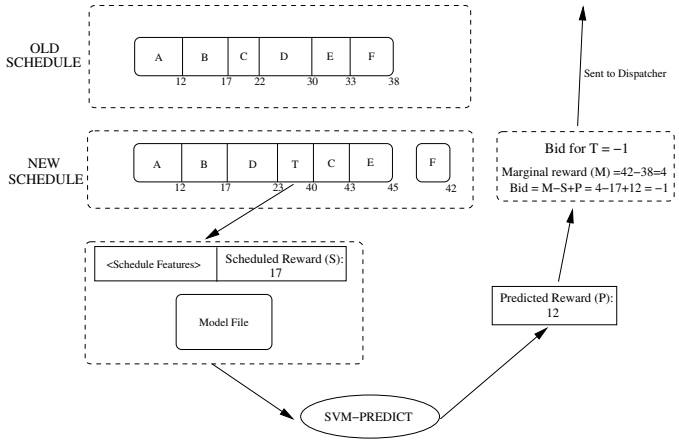

Figure 2: Learning-enhanced bidding for a new task T.

on scheduled task value. Then after a predefined interval the agents create a model file using all the data accumulated thus far in the trial. The agents then begin using that model to bid based on learned predicted task value. The agents continue to log training data after the initial model creation and periodically create a new model based on all data accumulated up to that point in the trial. This approach is fully online and does not depend on outside intervention.

6 Experimental Results

6.1 Simulation Design

For our experiments we use 5 agents modeled as points operating in a bounded world with a number of obstacles. We assume that all agents have full knowledge of the map - the only form of uncertainty occurs in the tasks to be issued. Agents are assigned random start locations in the grid-based world. We use a λ value of $\frac{4}{5}$ in our Poisson process, as shown in Equation 1.

In all trials the environment is exactly the same across approaches - agents start in the same randomly generated start location and are issued the same randomly generated tasks. Thus the differences in the approaches occur only at the allocation level. When comparing approaches we are interested primarily in the accumulated score over the issued tasks - a more effective approach outperforms other approaches in repeated trials.

We ran 15 trials of 10,000 time cycles each to obtain the following results. The Prelearning approach used a model created using the data accumulated from 3 trials of 2000 time cycles of the baseline solution, where all agents were logging training data to a central location. For the Online Learning approach we used a learning time of 750 cycles and centralized logging.

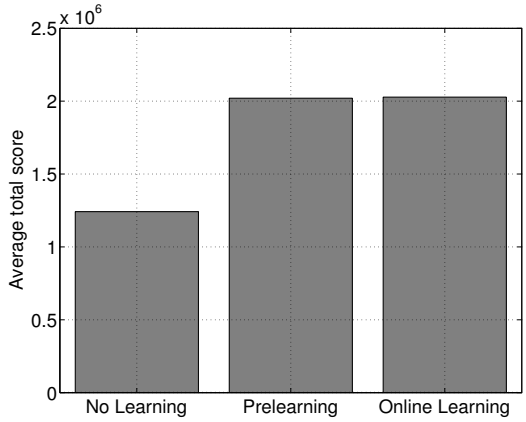

Figure 3: Average total scores (total reward - total penalty) yielded by No Learning, Prelearning, and Online Learning

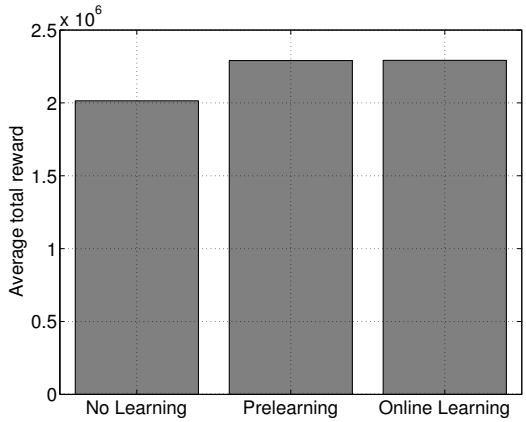

Figure 4: Average total reward for all successfully completed tasks yielded by agents using No Learning, Prelearning, and Online Learning

6.2 Overall Performance

Figure 4 shows total score achieved by the three approaches in our experiments. The learning-enhanced versions significantly outperform the baseline approach, by 62.7% for Prelearning and 63.2% for Online Learning. The improved performance exhibited by our learning approaches is due both to increased reward received for completed tasks and by committing significantly fewer penalties. Figure 4 shows both learning approaches received 14% more reward than the baseline approach, and Figure 5, shows that the learning approaches were assessed around 33% as much in penalties. Both learning approaches prove far better at determining at bid time which tasks are best left to other teams.

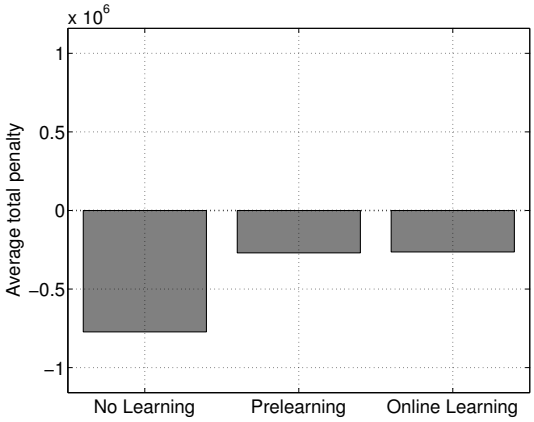

Figure 5: Average total penalties for all failed tasks yielded by agents using No Learning, Prelearning, and Online Learning

The Online Learning approach slightly outperforms the Prelearning approach, despite the fact that for the first 750 cycles of each trial agents use the baseline approach. We believe this is attributable to the iterative improvement of the learning model. While the Prelearning approach is trained exclusively on data obtained from the baseline approach, the Online Learning approach begins with data from 750 cycles of the baseline approach, but then begins using both the baseline data and data obtained from agents bidding using a learned model. As the trial progresses, the Online Learning can iteratively improve the learning model, training on data produced by agents using all of the previous models. That the Online Approach can improve on the Prelearned method makes a strong argument that our learning approach could be effectively employed even in scenarios where data from previous trials was not available.

6.3 Respecting Importance and Urgency

Despite the fact that new tasks are constantly being issued all three of the approaches respect importance and urgency. Figure 6 shows that fires at higher value buildings are addressed at much higher rates than those at lower value buildings across all approaches. The No Learning approach, however, does worse at respecting importance when compared to the learning approaches. It addresses more than twice as many fires at Residences as in the learning approaches, but addresses a smaller percentage of the three classes of more important tasks. By refusing to address low value tasks the agents in the learning approaches have more flexibility to profitably complete higher value tasks.

In Figure 7 we show the Time To Completion (TTC) metric for successfully completed tasks of the four fire magnitudes across all approaches. TTC measures the duration from task issue to task completion. We can see that in all approaches fires of higher magnitudes are extinguished more quickly on average than those of lower magnitudes. The learning approaches have faster TTCs on fires of lower magnitude, while the No Learning approach has slightly faster average TTCs on higher magnitude fires. This is partly due to the fact that agents in the No Learning approach address fewer high urgency tasks. Figure 8 shows TTC averaged over all completed tasks; the learning approaches average almost 16% faster average TTC.

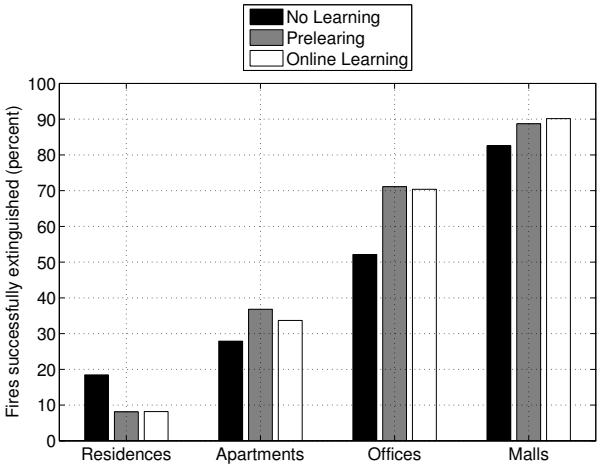

Figure 6: Completed task percentages for the building classes arranged from least average value on the left to greatest average value when agents use No Learning, Prelearning, and Online Learning

.

While the No Learning approach does a reasonable job of respecting importance and urgency, the learning approaches complete more higher value tasks and offer faster service on average. This leads to the increased reward shown in Figure 4.

7 Conclusions and Future Work

While our baseline market-based system is capable of coping with continuous task issue while respecting importance and urgency, its allocations perform poorly when confronted with oversubscription and penalties. We show that by using a learning-enhanced approach our allocation mechanism can learn to make commitments resulting in fewer penalties despite operating in an oversubscribed environment, while improving achieved reward by better respecting importance and urgency. This performance increase is validated on a disaster response domain, and we show that even when there is substantial uncertainty associated with future tasks our learning method can dramatically increase performance. We show that regression-based learning for markets has great promise as an approach to improving bid valuation over time and consequently improving overall team performance.

A central strength of our approach is that regression-based learning can implicitly encapsulate many aspects of task distributions in a manner that is highly relevant to the market without requiring an explicit model of task parameters or rates. Underlying rate and task distributions will become substantially more chaotic as we move to using more real-world data, and modeling parameters explicitly will become increasingly difficult; our approach should yield effective results even when parameters cannot be directly estimated. Also, it is unclear how knowledge of these parameters could be incorporated into agents' bids - our mechanism acts only by modifying bids, remaining true to the market-based paradigm.

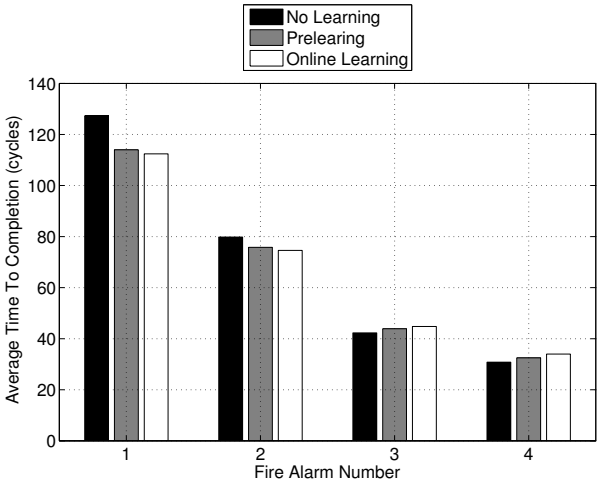

Figure 7: Average time to completions (TTCs) for fires of different magnitudes by agents using No Learning, Prelearning, and Online Learning. Fires are arranged from least urgent on the left to most urgent on the right.

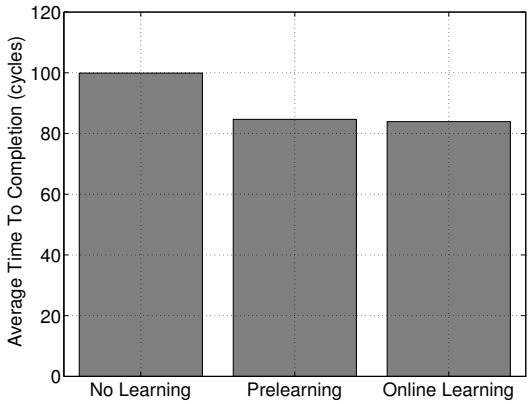

Figure 8: Average time to completions (TTCs) for all successfully addressed fires by agents using No Learning, Prelearning, and Online Learning.

This work takes a few important steps towards effective performance of market-based task allocation for disaster response. However there remain a number of difficult challenges that require additional research. Our future work will explore two main research directions. The first direction involves improving our learning techniques, enabling agents to recognize and avoid even greater sources of inefficiency in allocation. In the near future, we will enable agents to learn about the relative value of schedules instead of tasks. In the second research direction we extend our

learning-enhanced market-based approach to capture more of the domain features and uncertainty to represent the greater complexity of task allocation for disaster response.

We have produced initial results showing substantial improvements from using learning in the fire fighting domain when we incorporate map uncertainty in addition to the uncertainty associated with future emergencies. That regression-based learning can be used to improve bids in domains with multiple sources of uncertainty serves to further validate the approach and attests to its promise as a method with wide applicability.

8 Acknowledgements

This work was sponsored by the U.S. Army Research Laboratory, under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements of the the U.S. Government.

References

[1] Michael Benisch, Amy Greenwald, Ioanna Grypari, Roger Lederman, Victor Naroditskiy, and Michael Carl Tschantz. Botticelli: A supply chain management agent designed to optimize under uncertainty. *SIGecon Exchanges*, 4:29–37, 2004.

[2] M. Bernardine Dias. *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, January 2004.

[3] M. Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE – Special Issue on Multi-Robot Coordination*, 2006. in press.

[4] Brian P. Gerkey and Maja 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] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. *A Practical Guide to Support Vector Classifaction*. Department of Computer Science and Engineering, National Taiwan University. Available http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[6] M. Koes, I. Nourbakhsh, and K. Sycara. Constraint optimization coordination architecture for search and rescue robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3977–3982, 2006.

[7] Ranjit Nair, Takayuki Ito, Milind Tambe, and Stacy Marsella. Task allocation in the robocup rescue simulation domain: A short note. In *RoboCup 2001: Robot Soccer World Cup V*, pages 751–754, London, UK, 2002. Springer-Verlag.

[8] David Pardoe and Peter Stone. Bidding for customer orders in TAC SCM. In *AAMAS 2004 Workshop on Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems*, July 2004.

[9] International Federation of Red cross and Red Crescent Societies World Disasters Report. http://www.ifrc.org/publicat/wdr2004/index.asp, 2004.

[10] Jeff Schneider, David Apfelbaum, Drew Bagnell, and Reid Simmons. Learning opportunity costs in multi-robot market based planners. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[11] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series, October 1998.

[12] S. Suàrez, J. Collins, and B. Lòpez. Improving rescue operations in disasters: Approaches about task allocation and re-scheduling. In *The 24rd Annual Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG)*, pages 66–75, 2005.