Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Learning-enhanced Market-based Task Allocation for Oversubscribed Domains

E. Gil Jones M. Bernardine Dias Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
{egjones,mbdias,axs}@cs.cmu.edu

Abstract— **This paper presents a learning-enhanced market-based task allocation approach for oversubscribed domains. In oversubscribed domains all tasks cannot be completed within the required deadlines due to a lack of resources. We focus specifically on domains where tasks can be generated throughout the mission, tasks can have different levels of importance and urgency, and penalties are assessed for failed commitments. Therefore, agents must reason about potential future events before making task commitments. Within these constraints, existing market-based approaches to task allocation can handle task importance and urgency, but do a poor job of anticipating future tasks, and are hence assessed a high number of penalties. In this work, we enhance a baseline market-based task allocation approach using regression-based learning to reduce overall incurred penalties. We illustrate the effectiveness of our approach in a simulated disaster response scenario by comparing performance with a baseline market-approach.**

I. Introduction

Several application domains that require teamwork, such as disaster response and grid computing scheduling, present scenarios where agents cannot complete all tasks even if they act optimally. These domains are categorized as oversubscribed, where the team resources are insufficient to complete all tasks within the required deadlines. Oversubscribed domains present several challenges to task-allocation. The primary challenge is to determine which tasks should be completed, based on estimates of future constraints, and thus minimize penalties. Within this problem space, we focus on domains where the selection of tasks for execution significantly impacts the quality of the allocation solution. Specifically, we are interested in domains that exhibit the following characteristics:

- The set of tasks are not known prior to execution and new tasks are issued throughout the mission.
- All tasks are not equal in importance and urgency.
- Failed or broken commitments incur a cost in proportion to the importance and urgency of the tasks.

Existing market-based approaches to task allocation contain no mechanism for reasoning about future tasks and thus perform poorly in domains that demonstrate the above characteristics. Of course, precisely anticipating the future is impossible, especially given the uncertainty in many oversubscribed domains. However, learning techniques can be used to identify and exploit patterns in the characteristics and rate of emergence of tasks. Thus, the primary contribution of this work is the design, implementation, and evaluation of a learning-enhanced market-based allocation approach for oversubscribed domains.

The following section explores related work. We then introduce an oversubscribed fire fighting disaster response domain that has the requisite characteristics followed by a description of our approach and implementation. Experimental results and discussion are then presented followed by conclusions and an exploration of future work.

II. Related Work

Market-based approaches have been applied effectively to coordinate teams in a variety of domains [4]. The widest application in multi-robot domains has been in scenarios where the set of tasks to be allocated is known ahead of time, and the goal of the allocation is to assign tasks to robots to minimize a cost function such as total path length across all robots. For these domains sequential single-item auctions have been shown to give solutions within a constant factor of the optimal allocation [8] and to provide solutions that compare well against parallel and combinatorial auctions [7] while being inexpensive in terms of computation and communication. Some work has addressed improving single-item auctions to gain even better performance. Zheng et al. evaluate using lookaheads and rollouts to get better allocations [13], and several approaches focus on using inter-agent re-auctions to improve performance [3] [14]. However, none of this work considers tasks that have constraints such as deadlines with penalties for failure. Thus, existing methods of improving single-item auctions are ineffective when a team must reason about the urgency and importance of different tasks. For example, a re-auction can improve the allocation solution by changing the assignment for a particular task, but in the oversubscribed domains explored in this paper, some robot must still perform the task or the team will be assessed a penalty for not completing the task. The re-auction can address small inefficiencies in task allocation, but cannot affect the possibly larger inefficiencies associated with the acceptance of a relatively unimportant task in a domain oversubscribed with important tasks.

The Trading Agent Competition Supply Chain Management (TAC SCM) scenario has spurred substantial research

1-4244-0912-8/07/$25.00 ©2007 IEEE.

in adaptive market-based approaches. Of special interest are approaches for adapting and optimizing bidding for customer orders [1] [9]. 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; the oversubscribed domains addressed in this paper instead focus on cooperative agents working together to solve a problem.

Substantial work exists in reinforcement learning for improving task allocation [12] [2]. We are aware of only one previous approach, however, that uses learning to improve bidding over time in a collaborative multi-robot 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] - the omission of deadlines and penalties makes the decision about whether or not to allocate certain tasks have little effect on overall efficiency. This method serves to spread high-reward tasks among robots with different capabilities, 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. However, their mechanism is unlikely to limit penalties in the class of oversubscribed domains addressed by our work as it does not help agents to reason about the effects that current allocations will have on the future possibilities.

III. The Oversubscribed Fire Fighting Disaster Response Domain

We evaluate our allocation approach in an oversubscribed fire fighting disaster response domain. In the fire fighting domain teams of robotic fire fighters rove around a bounded disaster zone extinguishing fires of various magnitudes. New fires are continuously discovered at various buildings scattered around the city, and an objective score is assigned to each fire based not only on the value of the affected building but also on the magnitude of the fire. This means objective score for a particular fire is dependent on the time at which it is extinguished, the initial value of the building, and the fire's magnitude. 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 reasoning about importance and urgency and making good decisions about what tasks the team should and should not accept.

In this domain we model the continuous issue of new tasks using a Poisson process, the standard distribution used in queuing theory to represent stochastic arrival times of independent tasks [5]. The Poisson process is governed by a parameter λ which represents the expected rate of task issuance, as governed by the Poisson probability distribution.

Relative importance and urgency are associated with the value of the affected building and the magnitude of the fire respectively. An efficient allocation should consider a fire at a more valuable building to be more important than a fire at a less valuable building, and should consider a high-magnitude fire to be more urgent than a low-magnitude fire. In our experiments we include four building classes with four different Gaussian value distributions, ranging from the least valuable private residences to the most valuable malls. We also use four different magnitudes of fire, with alarms rated 1 to 4. There are more low-value than high-value buildings, and more low-magnitude than high-magnitude fires, so a particular fire issued from the Poisson distribution is more likely to occur at a low-value building and to be a small fire. Larger fires cause damage more quickly than smaller fires and take longer to extinguish. Though fires cannot spread in this example 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.

IV. Market-based Allocation for Oversubscribed Domains

The basic idea behind a market-based task-allocation mechanism is to assign tasks via an auction, where agents bid a value in a shared currency based on their perceived fitness for a task [4]. Tasks are awarded to the lowest bidder if the goal is minimizing cost, or to the highest bidder if the goal is to maximize reward.

A. Auction Mechanism

In our implementation, incoming tasks are sent to a team dispatcher, who acts as an auctioneer. The dispatcher is the only auctioneer in this implementation and 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 bids for the task (see Section IV-C) 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, allowing the disaster response coordinators to recruit additional agents to handle refused tasks. The dispatcher then informs the winning agent that it has won, and that agent adopts the task into its schedule.

B. 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 schedule value 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 task 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 is 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 in moving to the location of the failed task, the algorithm will compute the completion time of the next scheduled task 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 by using a 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.

C. Agent Bidding

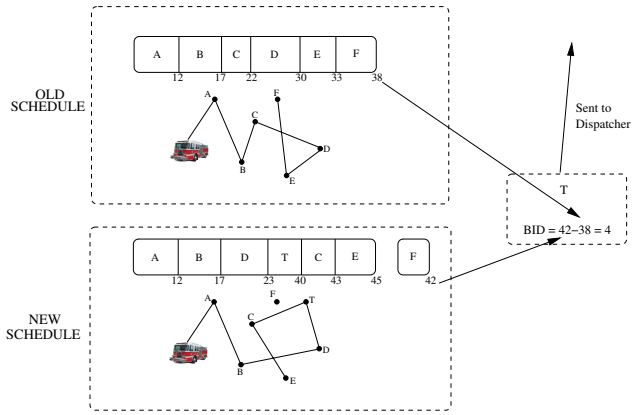

Fig. 1. Baseline bidding for a new task T as illustrated in the oversubscribed fire fighting domain.

When an agent receives a call for bids it creates a new schedule consisting of both its old schedule and the new task. It then optimizes the new schedule as described in Section IV-B and determines the total 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 market implementation 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 bid calculation.

D. Winning An Auction

As stated previously, the auction will be awarded to the agent placing the highest postive bid for the task. 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 other assets 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.

V. 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, consider Figure 1. In the old schedule before bidding for T the task F has an expected reward of 5. If the agent wins the auction for task T with a bid of 4 based on its new schedule, then the actual reward received for F will be its penalty, -3. Thus the agent was originally expecting to make 5 for 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, we can expect an overall increase in solution quality.

Our approach to learning is to use data accumulated by agents during the course of execution to construct a model that maps 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 [11] to perform this mapping.

A. Training Data Collection

In order to study the performance during the different runs, all 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 schedule to reach the location of 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. However, if communication costs were a concern agents could keep individual training data files.

B. 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; we use the `libsvm` package [6] . We train an SVR model by passing the training data file to a `libsvm` training program, which 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 and the cost parameter C for defining regression error. We used a grid search approach with cross-validation [6] to tune parameters. This cross-validation could occur online, but we found that small adjustments to parameters did not result in a substantial difference in performance.

C. Bidding Using the Model

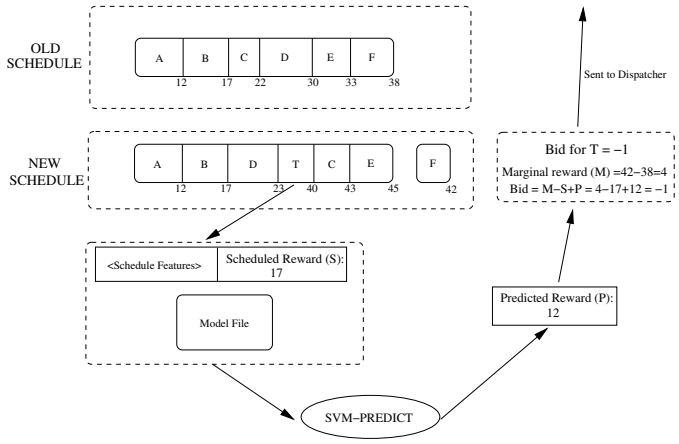

Fig. 2. Learning-enhanced bidding for a new task T as illustrated in the oversubscribed fire fighting domain.

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.

D. 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 "Prelearning" approach 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 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 automated.

VI. Experimental Results

This section describes experimental results in which we evaluate our learning-based approaches versus our baseline approach in the oversubscribed disaster response domain.

A. Simulation Design

For our experiments we use 5 agents, modeled as points and assigned random start locations, operating in a bounded world with a number of known obstacles. In each trial the same set of fire-fighting tasks is randomly generated – using a λ value of $\frac{4}{5}$ in our Poisson process – and then issued at the indicated times to agents operating in three parallel worlds associated with the three approaches. This ensures that performance differences between the three worlds should occur exclusively at the allocation level.

We ran 15 trials of 10,000 time cycles each to obtain the reported 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.

B. Overall Performance

Our simulation results show that the learning-enhanced versions significantly outperform the baseline approach, by 62.7% for Prelearning and 63.2% for Online Learning. Figure 3 shows the total score achieved by the three approaches in our experiments.

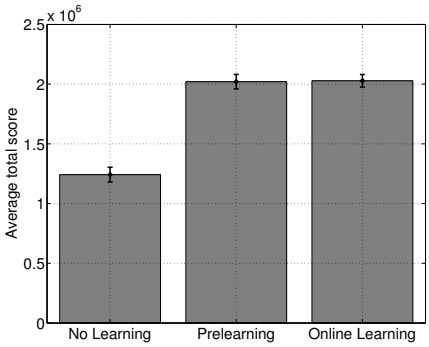

Fig. 3. Average total scores (total reward - total penalty) and standard deviations yielded by agents using the baseline and learning approaches.

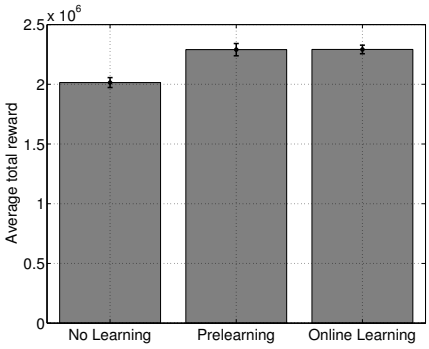

Fig. 4. Average total reward and standard deviations for all successfully completed tasks yielded by agents using the baseline and learning approaches.

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 approximately 33% of the penalties incurred in the baseline approach. Thus, both learning approaches are significantly better at determining (at bid time) which tasks are best suited for execution.

The Online Learning and Prelearning approaches perform equivalently within the standard deviations. That the Online Learning method can equal the performance of the Prelearning approach makes a strong argument that our learning approach could be effectively employed even in scenarios where the data from previous trials is not available.

C. Respecting Importance and Urgency

Despite the fact that new tasks are constantly being issued all three approaches demonstrate the ability to consider importance and urgency during task allocation. 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, addressing more low importance tasks

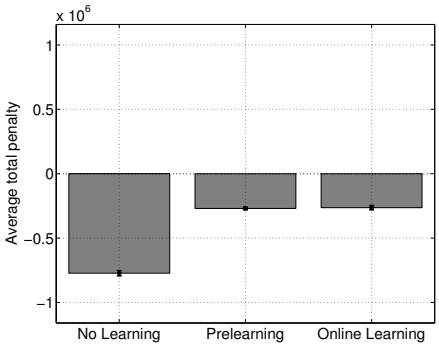

Fig. 5. Average total penalties and standard deviations for all failed tasks yielded by agents using the baseline and learning approaches.

and fewer high importance tasks. By refusing to address low importance tasks the agents in the learning approaches have more flexibility to profitably complete higher value tasks.

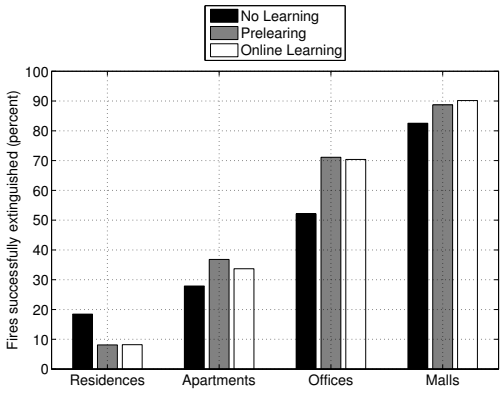

Fig. 6. Completed task percentages for the building classes arranged from least average value on the left to greatest average value when agents use the baseline and learning approaches.

In Figure 7 we show the Time To Completion (TTC) metric for successfully completed tasks. TTC measures the duration from task issue to completion. We can see that in all three approaches fires of higher magnitude 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 TTC.

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

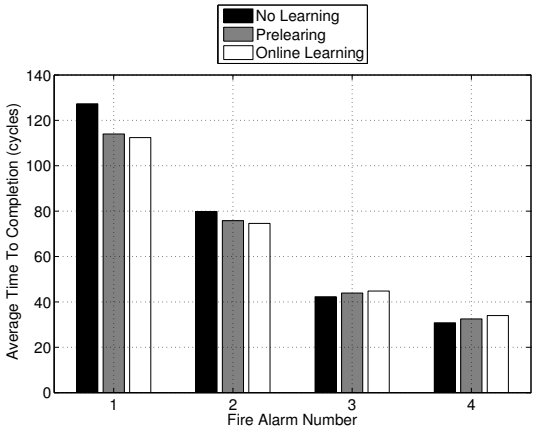

Fig. 7. Average time to completions (TTCs) for fires of different magnitudes by agents using the baseline and learning approaches. Fires are arranged from least urgent on the left to most urgent on the right.

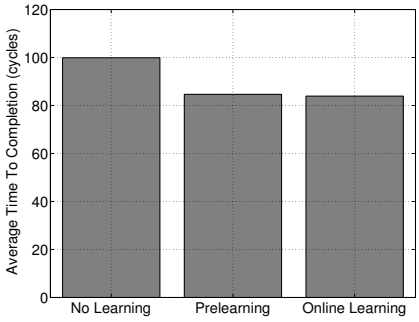

Fig. 8. Average time to completions (TTCs) for all successfully addressed fires by agents using the baseline and learning approaches.

VII. Conclusions and Future Work

In this paper we demonstrate that a learning-enhanced market-based approach can perform allocations that incur few penalties despite operating in an oversubscribed environment while respecting the relative importance and urgency of different tasks. This approach outperforms a baseline market-based allocation as demonstrated in a simulated disaster response domain. We show that even when there is substantial uncertainty associated with future tasks our learning method can dramatically increase 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. The underlying rate and task distributions will become substantially more chaotic in real-world data sets, and modeling parameters explicitly will become increasingly difficult. Our approach should yield effective results even when parameters cannot be directly estimated.

This work takes a few important steps towards effective performance of market-based task allocation for oversubscribed domains. However there remain a number of 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 additional sources of environmental uncertainty beyond the uncertainty associated with future tasks.

VIII. 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] M. Benisch, A. Greenwald, I. Grypari, R. Lederman, V. Naroditskiy, and M. C. Tschantz. Botticelli: A supply chain management agent designed to optimize under uncertainty. *SIGecon Exchanges*, 4:29–37, 2004.

[2] T. S. Dahl, M. J. Matarić, and G. S. Sukhatme. A machine learning method for improving task allocation in distributed multi-robot transportation. In D. Braha, A. Minai, and Y. Bar-Yam, editors, *Complex Engineering Systems*. Perseus Books, 2004.

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

[4] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE – Special Issue on Multi-Robot Coordination*, 2006.

[5] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley, New York, 3rd edition, 1998.

[6] C.-W. Hsu, C.-C. Chang, and C.-J. 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/.

[7] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain. The power of sequential single-item auctions for agent coordination. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.

[8] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt. Simple auctions with performance guarantees for multi-robot task allocation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[9] D. Pardoe and P. 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.

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

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

[12] M. J. A. Strens and N. Windelinckx. Combining planning with reinforcement learning for multi-robot task allocation. In D. Kudenko, D. Kazakov, and E. Alonso, editors, *Adaptive Agents and Multi-Agent Systems II*, volume 3394, pages 260–274. Springer-Verlag, 2005.

[13] X. Zheng, S. Koenig, and C. Tovey. Improving single-item auctions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[14] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.