Improving Market-Based Task Allocation with Optimal Seed Schedules

G. Ayorkor KORSAH\textsuperscript{a,1}, Balajee KANNAN\textsuperscript{a}, Imran FANASWALA\textsuperscript{b}, and M. Bernardine DIAS\textsuperscript{a,b}

\textsuperscript{a} Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
\textsuperscript{b} Computer Science Program, Carnegie Mellon University, Doha, Qatar

Abstract. Task allocation impacts the performance efficiency of agent teams in significant ways. Due to their efficient and proven performance, Market-based task allocation approaches have grown in popularity for many such multi-agent domains. In addition, market-based approaches are very well suited to dynamic domains such as emergency response, in which the set of the tasks or the environment changes in real time. However, market-based approaches are not guaranteed to produce optimal solutions and researchers have investigated many techniques for improving their performance in different scenarios. Since many application domains have a significant static component coupled with dynamic elements, we explore the option of enhancing team performance in these domains by seeding market-based task allocation with optimal schedules pre-computed for the static tasks. We compare the performance of the TraderBots market-based algorithm with and without the seeded optimal schedules in simulation and on a team of robots. Our results demonstrate that seeding market-based allocation with optimal schedules can improve team performance, particularly when the proportion of static tasks is high.

Keywords. Task Allocation, Market-Based Allocation, Optimal Planning.

Introduction

Complex dynamic domains, such as emergency response and urban search and rescue, require robust and efficient allocation of tasks and resources amongst team members to accomplish a variety of goals. The mission-critical nature of these application domains dictates a need for optimizing efficiency in task allocation while maintaining the flexibility to respond to dynamic conditions. In such domains, the task allocation problem is to compute an optimized assignment of tasks to agents along with a schedule according to which the tasks should be performed, and to modify this task allocation in efficient ways to respond to new situations that arise during execution.

Given the set of tasks a priori, it is possible to express the task allocation and scheduling problem for a team of agents mathematically, and to devise an algorithm to compute an optimal solution. However, computing an optimal allocation and schedule is an NP hard problem, even in the absence of dynamism. As such, in dynamic domains where tasks are constantly entering or leaving the system, it is not computationally feasible to repeatedly compute an optimal allocation and schedule.

\textsuperscript{1} Corresponding Author: G. Ayorkor Korsah (also published as G. Ayorkor Mills-Tettey), Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA; Email: ayorkor@alumni.cmu.edu
Market-based approaches, in which tasks are auctioned among agents in a virtual economy, have proven to be very effective in such domains, enabling the fast computation of efficient solutions.

In reality, the set of tasks to be performed includes both static tasks that can be reasoned about a priori, and dynamic tasks that cannot be anticipated with any certainty. While both types of tasks can be handled seamlessly with a market-based task allocation approach, we observe that an optimal allocation can be computed for the set of static tasks that are known a priori. These schedules can then be used to seed the market-based task allocator that can dynamically adjust the schedules to accommodate unexpected contingencies (such as new tasks and execution failures), by auctioning or re-auctioning tasks as needed. We propose that seeding the market-based planner with initial optimal schedules in this way can improve the overall performance of the team. We investigate this hypothesis by performing experiments and analyzing team performance both in simulation and on physical robots.

As a motivating scenario, we consider a team of agents responding to a series of emergencies (such as a major accident leading to spreading fires) in a small city. The emergency plan for the city will have a pre-determined allocation of tasks to the different emergency response units based on the location and severity of the incidents and the available resources to address them. Pre-computed emergency plans can also include safe locations and routes for evacuating survivors and storing supplies to be used in rescue efforts. Several such likely scenarios can be planned for based on the resources and constraints of the city, and an optimal planner can be used to allocate tasks and resources to address the emergency situations. In the event of actual emergencies, the most relevant pre-computed plan is put into execution and dynamically adjusted to accommodate new information and unforeseen problems.

1. Background

As research efforts in robotics and autonomous systems increasingly move toward using teams of agents to collectively address tasks, coordination and task allocation take on added significance. For example, a well-coordinated team of relatively simple robots can often outperform complex single robot solutions in terms of quality and robustness. Coordinating these teams in an efficient manner is however not a simple task, especially under dynamic and uncertain conditions. Several techniques have been explored to address this problem.

Existing task allocation approaches can be classified as optimal mathematical programming approaches, and heuristic approaches. The class of vehicle routing problems (VRPs) is a well researched general problem class that addresses the distribution of passengers or goods between depots and final users [3] or matching of tasks to agents. VRPs can be expressed as mixed integer programming problems (MIP), defined on a graph in which the nodes correspond to locations of tasks to be performed, and edges correspond to travel segments between these locations. A variety of different mathematical models have been proposed, and these can be broadly categorized as 3-index models, such as that defined by Cordeau [2], and 2-index (or set-partitioning) models, such as that defined by Savelbersgh and Sol [13]. These mathematical models enable the formulation of optimal solution approaches. Another mathematical approach that has been applied to multi-robot coordination [11] discusses allocating multi-robot tasks to a team of robots, where tasks have associated rewards that decay
linearly, and the system can be constrained through task ordering, time-oriented task constraints, capability constraints, etc. Unfortunately, such mathematical programming approaches run in exponential (in the number of constraints) time and requires complete re-computation of the solution when new tasks are added or task features change. The combination of these limitations means that an MILP approach to our domains of interest will not be tractable or effective at generating solutions for the dynamic tasks, but can be used to pre-compute solutions for the static tasks.

An alternate technique that is proven to be highly efficient in dynamic environments is the market-based task allocation strategy [4],[7],[8]. Based on the principles of markets and auctions, robots are designed as self-interested agents that operate in a virtual economy. An auction call for a task is sent, agents place bids on tasks, and the agent with the best bid (according to the specific auction-clearing algorithm), wins the assignment. An advantage of market-based approaches is that they can opportunistically adapt to dynamic conditions to produce fast and efficient, though sometimes sub-optimal, solutions. Market mechanisms can distribute much of the planning and execution over the team and thereby retain the benefits of distributed approaches. Furthermore, most market-based strategies are robust to communication and robot failure. For example, in the case of imperfect communication, an agent can continue to carry out its allotted tasks without necessarily having to participate in auctions. In case an agent is damaged and loses some of its task execution capabilities, it can hold auctions to re-distribute its tasks amongst the remaining agents. TraderBots, developed by Dias [4] is one such task allocation mechanism, inspired by the contract net protocol [14], designed to inherit the flexibility of a market economy, and to exploit these benefits to enable robust and efficient team coordination in dynamic and uncertain environments.

Several quantitative comparative studies have been published for market-based approaches. Dias and Stentz [5] present a comparison of a centralized optimal, a market-based, and a distributed behavioral task allocation and coordination strategy, for a scenario with static tasks. The comparison indicated that the market-based approach compared favorably in terms of cost to the optimal approach, and favorably in terms of computation time to the behavioral approach. Xu et al [15] illustrate that an auction-based (market-based) coordination strategy results in higher rewards than a token-based strategy, but requires more communication. Kalra and Martinoli [10] found that market-based allocation is more efficient than threshold-based allocation when information is accurate, but the latter performs just as well but at a fraction of the expense, when information is not accurate. Thus far, no studies have investigated enhancing market-based approaches by seeding them with optimal initial schedules.

2. Approach

We assume that there is a team of agents \( A \), available to perform a set of tasks \( T \), consisting of a subset of static tasks \( T_S \) which are known a priori, and a set of dynamic tasks \( T_D \) that arrive in real time. An optimal planner is used to allocate and schedule the initial set of static tasks. The resulting seed schedules are given to the agents in the system to execute. As new tasks arrive in real time, or other unexpected changes occur (such as execution failure for a task), a market-based method (TraderBots) is used to adjust the plan, by auctioning new tasks or re-auctioning old tasks as needed.
2.1. Optimal Task Allocation for Static Tasks

For the optimal task allocation strategy, we represent the problem as a mixed integer programming problem using a standard set-partitioning mathematical model, such as the one used by Savelsbergh and Sol [13]. In a set-partitioning model, feasible routes or schedules for agents are represented by columns in a mixed integer linear program. In particular, a binary variable $x^k_r$ indicates whether an agent $k \in A$ performs a route $r$ chosen from among all possible routes $R_k$ that can feasibly be performed by $k$. In our problem, a feasible route is a sequence of tasks to be performed at given locations.

Minimize:

$$\sum_{x \in R_k} d^k_x x^k$$  \hspace{2cm} (1)

Subject to:

$$\sum_{x \in R_k} x^k = 1 \quad \forall k \in A \hspace{2cm} (2)$$

$$\sum_{k \in K} \sum_{x \in R_k} \pi^k_{jr} x^k = 1 \quad \forall j \in T \hspace{2cm} (3)$$

$$x^k \in \{0,1\} \quad \forall k \in A, r \in R_k$$

where $d^k_x$ is the distance along route $r \in R_k$, and $\pi^k_{jr}$ is a binary value that represents whether task $j \in T$ occurs on the route $r \in R_k$.

The objective function, Eq. (1), minimizes the team cost. In this case, the team cost is the total distance traveled. The first set of constraints, Eq. (2), specifies that each agent performs exactly one route (sequence of tasks), which could potentially be an empty sequence. The second set of constraints, Eq. (3), specifies that each task is performed by exactly one agent.

We solve this optimal task allocation problem using a branch-and-price approach [1]. A detailed description of this approach is outside the scope of this paper; however, a short summary follows.

In a branch-and-price approach, instead of trying to enumerate all feasible routes in the mathematical model up front, an initial subset of feasible routes is considered for each vehicle. A relaxed version of the problem above, referred to as the master problem, is then solved using a standard linear programming approach such as the simplex algorithm. Subsequently, the solution to a pricing subproblem, which, in this case, is a constrained route-planning problem, helps identify additional feasible routes to be included. An iteration over these steps is embedded in a branch-and-bound process which is used to find an optimal integer solution to the problem.

When the branch-and-price algorithm is run to termination, the result is an optimal solution to the task allocation and scheduling problem. It is also possible to specify a timeout on the solution to the overall problem, or on each invocation of the route-planning subproblem. However, utilizing these timeouts implies that the algorithm...
may terminate without proving the optimality of the returned solution, which as such may be suboptimal.

2.2. Market-Based Task Allocation for Dynamic Events

As described earlier, the market-based approach has become the task allocation method of choice for many multi-robot systems. In this work, we employ TraderBots [4] as our market-based approach to multi-robot coordination. Tasks are allocated using auctions, where agents submit bids based on a domain-specific cost function. There are two modes in which this approach can be used, centralized and decentralized. The former requires a central/global auctioneer (i.e. the OpTrader) who is responsible for holding auctions, receiving bids, and subsequently allocating the tasks. In the decentralized approach, in addition to the OpTrader, the agents can themselves hold auctions, allowing for greater efficiency. In either case, the task is awarded to the agent with the smallest bid (i.e. the agent for whom the task is least costly). In this work, we use the decentralized version of TraderBots.

2.3. Seeded Market-Based Approach for Static & Dynamic Tasks

In the Seeded TraderBots approach, which we introduce in this work, we start by computing an initial allocation of tasks using the optimal branch-and-price algorithm. This initial schedule is given to the agents for execution. As dynamic events occur, the market-based approach is used to modify the current allocation. The dynamic events may be of two types:

Arrival of new tasks: On the arrival of a new task, an auction is held to assign it to an appropriate agent. In computing the bid value, each agent determines the best point in its current schedule to insert the new task. In addition an agent that has previously been assigned a task may hold an auction to attempt a profitable re-allocation of a subset of its allotted tasks to different agents. This allows an agent’s initial schedule to be efficiently re-organized to accommodate newly arrived tasks.

Execution failure: An agent that for some reason fails to execute a task that has been assigned to it can try to auction that task off to a different agent. Execution failure can be a result of the malfunction of the agent, or due to changes in the environment which make it impossible for the agent to execute the task.

3. Experiments

We have implemented and tested the seeded market-based approach in simulation and on physical robots. In all our tests, we have a set of agents, $A$, and a set of tasks, $T$, (comprising static tasks $T_S$ and dynamic tasks $T_D$). An agent executes a task by visiting the location of that task. We compute the total distance travelled by all agents in the team as a team cost. A plan is considered better than another if it has a lower team cost. In each experiment, we compare three task allocation methods:

- The pure market-based method in which all tasks, both static and dynamic, are allocated using the decentralized TraderBots market-based approach
The seeded market-based method in which the initial set of static tasks is allocated with the optimal mathematical programming approach, and the dynamic tasks are allocated with the TraderBots market based approach.

- The optimal-only method in which a plan for all tasks, both static and dynamic, is computed in hindsight, after the arrival times of all tasks are known, using the mathematical programming approach. Note that this plan is unattainable in practice since the arrival times of dynamic tasks are not known in advance.

As described earlier, if timeouts are used when solving the route-planning subproblem in the branch-and-price algorithm, the algorithm may terminate with a slightly suboptimal solution or without having proved the optimality of the returned solution. As such, in all the results below, the solution reported as returned by the optimal planner is the best solution found by the algorithm, which in some cases has not necessarily been proven optimal.

3.1. Simulation Experiments

The simulation experiments were carried out in a flat 40x40 grid-world. We used 3 problem configurations, comprising (i) 2 agents and 12 tasks, (ii) 2 agents and 16 tasks, and (iii) 5 agents and 20 tasks, respectively. For each configuration, we generated 5 random problem instances, each with different locations of the agents and tasks. For each problem instance, we experimented with 3 degrees of dynamism: (i) 25% of tasks static and 75% dynamic, (ii) 50% static and 50% dynamic, and (iii) 75% static and 25% dynamic. We tested the 3 different allocation strategies: (i) pure market-based (ii) seeded market-based, and (iii) optimal mathematical programming (yielding the best plan in hindsight).

In summary, each simulation experiment was defined by the number of agents, $n_A$; the number of static tasks, $n_S$; the number of dynamic tasks, $n_D$; the random seed, $r$, determining the agent and task locations and task arrival times; and the allocation strategy adopted, $alloc$.

$$\text{Experiment} = \{n_A, n_S, n_D, r, alloc\}$$

In the simulation, the agents move at a speed of 1 grid cell per second. Static tasks are present from $time=0s$ and dynamic tasks arrive at random times between $time=1s$ and $time=100s$. The simulation is initialized by spawning agents at their start locations. For the optimal and seeded market-based methods, the agents’ schedules are seeded with the computed initial plan. For the seeded market-based and pure market-based methods, incoming dynamic tasks are allocated as soon as they arrive, using the TraderBots allocation mechanism.

In the task auctions, each agent computes its bids using the marginal cost of adding a new task in the best insertion point on its current schedule. Costs are computed using the Euclidean distance between the two locations, and the best insertion point is selected as the point of insertion that minimizes the marginal cost. In the pure and seeded market-based experiments, agents are allowed to participate in auctions regardless of their activity (i.e., "idle" or "executing"). Moreover, agents are allowed to re-auction all tasks, except the one that they are currently executing. Agents begin executing their allotted tasks immediately, and upon completion of each task, each agent moves on to the next task in its schedule. In these experiments, all the agents are simulated on a single processor.
While executing their allocated tasks, the agents keep track of the path taken and distance travelled. When the simulation ends, we tally the total distance travelled by each agent thus arriving at the team distance. Note that the team distance in the simulation is the distance travelled in the 8-connected grid and as such is slightly higher than the Euclidean distance used for evaluating the cost of a task. However, Euclidean distance is faster to compute and does a good job of estimating the 8-connected distance so it is appropriate to use it in the bidding process thus enabling faster task allocation.

3.2. Robot Experiments

In addition to the simulation experiments, we implemented and tested the various allocation strategies on a team of two Pioneer 3DX robots, shown in Figure 1, operating in a 10m x 10m world. The Pioneer robots are equipped with SICK LiDar and fiber optic gyros for building maps and maintaining accurate pose during navigation. They communicate with each other over the wireless (802.11g) network. The experiments consisted of 6 static and 5 dynamic tasks; that is, about 50\% of tasks were dynamic. The static tasks were present at time=0s, while the dynamic tasks arrived between time=1s and time=120s. For the pure and seeded market-based strategies, we utilized the decentralized version of TraderBots. This setup represents a much more realistic situation than the simulation results previously described because it is a true distributed system. The agents communicate with each other and with the OpTrader (running on a desktop machine) over the wireless network. This enabled us to capture many of the real-world considerations for such systems, including:

- **Communication**: Auctions and allocations are not instantaneous, but rather take several seconds. Occasional dropped communication packets resulted in some tasks being auctioned multiple times during the experiments.
- **Navigation**: Agents do not turn instantaneously but rather use an arc-based navigation method, affecting the total distance traveled. The speed of the robots is also altered during the course of navigations; the robot slows down as it nears the goal to prevent overshooting.
- **Errors**: The robots occasionally ran into problems due to sensing or other execution errors. These included seeing ghost obstacles where none existed or conversely, running into undetected obstacles. On occasion, a robot would auction all of its tasks off to the other robot, if it ran into a problem from which it could not recover.

![Figure 1: (a) Pioneer 3DX robots in operation (b) Operating Environment (map built from collated SICK LiDar data)](image-url)
4. Results

Figure 2 compares the total team distance incurred, in the simulation experiments, for the pure and seeded market-based approaches for varying numbers of agents and tasks, and for varying degrees of dynamism (represented as the percentage of tasks that are static). The results are averaged over the 5 random problem instances for each configuration, and are expressed as a factor of the best solution found by the optimal mathematical programming approach, a metric we describe as the suboptimality factor.

A couple of observations can be made from the results above. The first is that the market-based approach, although it is a greedy approach, performs fairly well compared to the best solution found by the mathematical programming approach, particularly when there is a higher percentage of static tasks (lower degree of dynamism). That said, for the problems with a higher percentage of static tasks, the solution obtained can often be improved by starting out with an optimal plan for the static tasks; that is, by using the seeded market-based allocation strategy.

The market-based allocation is a greedy approach that, when employed with decentralized auctions, also functions like a local search. As such, having a good initial plan for a subset of the tasks can improve its performance by steering it to a useful part of the state space. The initial plan can however, in some cases, steer the market-based approach to a slightly worse plan than generated by the solely market-based approach. Whether this occurs depends on the particular problem configuration. For example, with information about only the initial set of tasks for a given problem, a subset of tasks might be assigned to a particular agent, whereas if the entire set of tasks for the problem had been known, that subset would have been assigned to a different agent. Although agents can re-auction tasks to other agents, they re-auction only one task at a time. As such, they might not discover that the solution can be improved by transferring a subset of tasks as a group to a different agent. In other words, having an initial solution sometimes traps the market-based approach in local minima. But overall, seeding the market-based approach with an optimized initial plan seems to improve the task allocation performance in most instances.

Table 1 shows the median planning time, on a 2.67 GHz Intel Core i5 processor, for the optimal planner for the various problem configurations tested. Although these
times are for an implementation of the planner that has not been optimized for performance, the combinatorial nature of the optimal approach is evident, emphasizing the suitability of the approach for pre-planning rather than runtime re-computation of optimal solutions.

Table 1. Median planning times for the optimal planner in seconds.

<table>
<thead>
<tr>
<th>Problem configuration</th>
<th>25% static</th>
<th>50% static</th>
<th>75% static</th>
<th>100% static</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 agents, 12 tasks</td>
<td>0.17 s</td>
<td>3.08 s</td>
<td>495.13 s</td>
<td>6417.05 s</td>
</tr>
<tr>
<td>2 agents, 16 tasks</td>
<td>0.30 s</td>
<td>315.39 s</td>
<td>17043.3 s</td>
<td>16906.80 s</td>
</tr>
<tr>
<td>5 agents, 20 tasks</td>
<td>0.49 s</td>
<td>60.88 s</td>
<td>4651.86 s</td>
<td>15400.30 s</td>
</tr>
</tbody>
</table>

(Values in italics indicate problem configurations for which, for all 5 instances, the planner terminated without proving the optimality of the best solution found).

Table 2 summarizes the results from the robot tests. For the pure and seeded market-based approaches, the results are averaged over 5 experimental runs, since the auction outcomes can be different in each run. The table indicates the team distance and the suboptimality factor. In these experiments on real robots (compared to the simulation experiments), seeding the market-based approach with an optimal schedule results in more significant improvements over the pure market-based approach. A contributing factor is that delays caused by real-time distributed auctioning for the market-based approach increases the value of having an optimal plan computed ahead of time for the set of tasks that are known.

Table 2. Comparison of hybrid and market-based allocation method using pioneer robots for a problem with 2 agents and 11 tasks (6 static and 5 dynamic)

<table>
<thead>
<tr>
<th></th>
<th>Hindsight Optimal</th>
<th>Seeded Market-based (averaged over 5 runs)</th>
<th>Solely Market-based (averaged over 5 runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average team distance (m)</td>
<td>25.82</td>
<td>51.78</td>
<td>65.31</td>
</tr>
<tr>
<td>Suboptimality factor (Ratio to optimal)</td>
<td>1.00</td>
<td>2.01</td>
<td>2.53</td>
</tr>
</tbody>
</table>

5. Conclusion and Future Work

We have shown that seeding a market-based task allocation strategy with initial optimal schedules often improves its overall performance, particularly for problems where there is a high proportion of static tasks and where the team is operating in realistic conditions with communication delays.

There are several enhancements to be made in this work. We are in the process of implementing this method in the USARSim simulation framework, to enable us to run a larger number of experiments in a higher fidelity simulated environment. This will enable us to characterize the performance of the approach for a wider range of problem configurations. Another important aspect of future work is to extend the seeded market-based task allocation approach to domains with constraints such as timing, precedence or capacity constraints. We have already proposed an optimal mathematical programming approach to these problems [12] and will work towards enhancing the market-based approach so that it can respect these domain constraints when dealing
with dynamic events. This will greatly extend the range of domains for which this planning approach is useful.

6. Acknowledgement

This work is sponsored in part by the Qatar National Research Fund under contract NPRP 1-7-7-5, and by the Boeing company under contract CMU-BA-GTA-1. The content of this publication does not necessarily reflect the position or policy of the sponsors and no official endorsement should be inferred. We would like to thank Anthony Stentz, Wael Ghazzawi, Ameer Abdulssalam, M. Freddie Dias, Victor Marmol, and Jaime Bourne for their significant contributions to the larger multi-robot coordination projects of which this work is a part.

References


