# Auctioning Robotic Tasks With Overlapping Time Windows

# (Extended Abstract)

Ernesto Nunes*
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
enunes@cs.umn.edu

Maitreyi Nanjanath
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
nanjan@cs.umn.edu

Maria Gini*
Dept of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
gini@cs.umn.edu

## ABSTRACT

This work investigates allocation of tasks to multi-robots when tasks are spatially distributed and constrained to be executed within assigned time windows. Our work explores the interaction between scheduling and optimal routing. We propose the Time-Sensitive Sequential Single Item Auction algorithm as a method to allocate tasks with time windows in multi-robot systems. We show, experimentally, that the proposed algorithm outperforms other auction algorithms that we modified to handle time windows.

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.9 [**Artificial Intelligence**]: Robotics

## General Terms

Algorithms, Performance, Experimentation

## Keywords

Auctions, time windows, task allocation, multi-robot systems

## 1. INTRODUCTION

Many real world problems require tasks to be executed within a specified time window. For example, a region may need surveillance at regular intervals or at specific hours, and in search and rescue, much of the exploration has to be done in well defined stages. Time windows make task allocation harder as it is no longer possible to arbitrarily arrange the order of execution of tasks to decrease travel costs.

Auctions are becoming popular for allocating tasks to robots [4]. However, limited attention has been devoted to allocation of tasks that have to be completed within specified time windows (see [3] for an example), and even less for tasks that have overlapping time windows. When time windows are pairwise disjoint, tasks can be strictly ordered and robots can choose any permutation of tasks. With overlapping time windows this is no longer possible.

Time windows are often treated as soft constraints on time of arrival to a task location (see [5] for an example). In this case late arrival to a task is subject to a penalty, which increases the cost but does not affect feasibility. We treat time windows as hard constraints, as a robot can no longer perform a task to which the robot arrives late.

We are interested in approximate algorithms that are computationally efficient and that minimize the sum of the path costs over all the robots, while avoiding time conflicts. The main contributions of this paper are the Time-Sensitive Sequential Single-Item Auction (TS-SSIA) algorithm and experimental results comparing its performance to other single item auction algorithms.

*Work supported in part by NSF IIP-0934327 and the Safety, Security, and Rescue Research Center at the University of Minnesota

**Appears in:** *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.


## 2. AUCTIONS WITH TIME CONSTRAINTS

Formally, $r = \{r_1, r_2, \ldots, r_n\}$ is a set of robots; each robot has a Cartesian location $(x_{r_i}, y_{r_i})$. Let $t = \{1, 2, \ldots, m\}$ be a set of tasks; each task $j$ has a time window defined by its earliest start time $es(j)$, latest finish time $lf(j)$, location $(x_j, y_j)$, and time duration $dur(j)$. The objective is to assign to each robot $r_i$ a subset of tasks $\{j, j+1, ...k\} \subseteq t$ such that that the sum of the path costs is minimized while completing the largest number of tasks.

For each task we compute its latest start time $ls(j) = lf(j) - dur(j)$. Let $RT(r_i, k, j)$ be the time it takes robot $r_i$ to travel between tasks $k$ and $j$. Then, lateness is defined as $l(r_i, k, j) = lf(k) + RT(r_i, k, j) - ls(j)$. If $l(r_i, k, j) > 0$, the robot $r_i$ is not able to reach task $j$ in time to do the task.

In an auction allocation method, the robots bid on tasks based on the amount of effort they need to complete them, which includes the cost of traveling to the task and any additional cost for doing the task itself.

Combinatorial auctions produce optimal solutions, but finding a set of non-conflicting bids that maximizes revenue is NP-complete and impractical for large numbers of tasks, hence it is common to auction each task separately. When all tasks are put up for bids at the same time in a *parallel single-item auction* the solution can be far from optimal, because robots cannot account in their bids for complementarities among tasks. This shortcoming can be reduced by repeating the auctions periodically at fixed time interval [1]. Alternatively, the tasks can be put up for bids one at a time

| Number of Tasks | Parallel single-item auction | | Sequential single-item auction | | TS-SSIA | |
|---|---|---|---|---|---|---|
| (10 robots) | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ |
| 30 | 2062 | (178.04) | 2061 | (184.37) | *1965* | (170.23) |
| 50 | 3420 | (232.17) | 3392 | (228.03) | *3257* | (223.92) |
| 70 | 4824 | (281.54) | 4785 | (278.90) | *4590* | (265.85) |
| 90 | 6396 | (369.55) | 6117 | (367.01) | *5869* | (343.03) |
| (20 robots) | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ |
| 30 | 1894 | (180.96) | 1893 | (183.58) | *1845* | (174.01) |
| 50 | 3217 | (225.67) | 3216 | (223.27) | *3106* | (218.33) |
| 70 | 4496 | (317.13) | 4489 | (317.15) | *4317* | (278.15) |
| 90 | 5766 | (330.75) | 5762 | (320.35) | *5528* | (286.40) |
| (50 robots) | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ | $\mu$ | $(\sigma)$ |
| 1000 | 58520 | (909.12) | 57710 | (948.68) | *56350* | (889.10) |

**Table 1: Solution cost (mean and standard deviation) of auction methods in a $100 \times 100$ grid, averaged over 30 runs. TS-SSIA orders tasks in ascending order of earliest start times. Numbers in italics are the best results.**

in a *sequential single-item auction (SSIA)* [2]. In this case robots account for previous task commitments while bidding on the next task, so they bid the insertion cost in their current path. When minimizing the sum of the path costs the solution is a constant factor away from the optimum.

We extend these auction methods by enabling them to deal with time windows. In *repeated parallel single-item auctions with time windows* the auctioneer chooses each winning bid depending on two criteria: the cost of the winning bid is the minimum amongst all bids, and the winning robot has no time conflict with the task. In *sequential single-item auctions with time windows* the winner of each task is the robot with the minimum insertion cost into its path and no time conflict. The addition of time windows makes the solution to depend on the order in which tasks are put up for auction. This suggests a change in the algorithm to take advantage of both spatial and temporal synergies among the tasks.

In *time-sensitive sequential single-item auctions (TS-SSIA)* the auctioneer orders tasks for auction according to one or more sorting criteria. We experimented with ordering tasks by their earliest or latest start times, either in ascending or descending order. The key difference between TS-SSIA and the sequential single-item auction algorithm is the fact that that the auction process is informed by the time windows of the tasks. By ordering the tasks up for bids according to their time windows, robots can easily take into account the time constraints in their bids instead of considering only their distance to the tasks.

# 3. CASE-STUDY: TASKS ON A 2D GRID

For this set of experiments we created a $100 \times 100$ Cartesian grid and distributed tasks uniformly on the grid. Any robot can reach any task with a cost proportional to the Cartesian distance between the robot and the task. We used 10, 20 and 50 robots and $30, 50, 70, 90$ and 1000 tasks respectively. The cost for each task was uniformly distributed between $0 - 100$. The goal of the experiment is to compare the sum of the path costs produced by TS-SSIA with those produced by the other single-item methods.

In this experimental setup, there was no difference in solution cost between sorting tasks by start times or by deadline, consequently, we present only results for sorting by start times. When tasks are sorted in descending order, on average, the algorithms produced higher costs than when tasks are sorted in ascending order.

In Table 1 we see that TS-SSIA outperforms the other single item auction methods. TS-SSIA reports gains as large as 8.2% against parallel and 4.1% against sequential (90 tasks 10 robots case). The difference in performance between TS-SSIA and the sequential algorithm and between TS-SSIA and the parallel algorithm is statistically significant with p-values 0.0017 and 0.0002 respectively.

# 4. CONCLUSIONS AND FUTURE WORK

We have presented a variant of SSIA, which we call TS-SSIA, that works better for allocation of tasks that have time constraints. We have compared its performance to other single item auction algorithms. In the 2D grid case-study, TS-SSIA produced better solutions than the other single-item auctions in every case. Going forward, we plan to improve the TS-SSIA algorithm to add the ability to adjust the schedule of tasks already allocated when this can result in the allocation of a new task. We also plan to provide a formal theoretical analysis of the algorithm, and extend the experimental work to additional case-studies.

# 5. REFERENCES


[1] M. B. Dias. *TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination*. PhD thesis, Carnegie-Mellon Univ., 2004.

[2] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain. Auction-based multi-robot routing. In *Robotics: Science and Systems*, pages 343–350, Cambridge, USA, June 2005.

[3] J. Melvin, P. Keskinocak, S. Koenig, C. A. Tovey, and B. Y. Ozkaya. Multi-robot routing with rewards and disjoint time windows. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2332–2337, 2007.

[4] M. Nanjanath and M. Gini. Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems*, 58(7):900–909, July 2010.

[5] S. Ponda, J. Redding, H.-L. Choi, J. How, M. Vavrina, and J. Vian. Decentralized planning for complex missions with dynamic communication constraints. In *American Control Conference*, 2010.