Minimising Undesired Task Costs in Multi-robot Task Allocation Problems with In-Schedule Dependencies

Bradford Heap and **Maurice Pagnucco**
School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW, 2052, Australia
{bradfordh,morri}@cse.unsw.edu.au

Abstract

In multi-robot task allocation problems with in-schedule dependencies, tasks with high costs have a large influence on the total time required for a team of robots to complete all tasks. We reduce this influence by calculating a novel task cost dispersion value that measures robots' collective preference for each task. By modifying the winner determination phase of sequential single-item auctions, our approach inspects the bids for every task to identify tasks which robots collectively consider to be high cost and ensures these tasks are allocated prior to other tasks. Our empirical results show this method provides a significant reduction in the total time required to complete all tasks.

Introduction

To coordinate efficiently, distributed robots often use explicit communication techniques (e.g., auctions, coalitions) to allocate tasks and ensure a global team objective is completed. This problem is known as *multi-robot task allocation* (MRTA) and many variants are known to be NP-Hard (Korsah, Stentz, and Dias 2013). Minimising the total task completion time is a common objective in these problems. For example, consider a team of autonomous robots delivering packages in an office-like environment. Each task represents a package to be delivered. The cost to complete each task is the time required for a robot to travel to the pickup point and then the delivery point for a package. To complete all tasks, robots may be required to deliver multiple packages and may be constrained in the number of packages they can carry at any time. When robots carry multiple packages, they may take advantage of positive *inter-task synergies*, that is, when the cost to complete two or more tasks in parallel is lower than the cost to complete each task in isolation.

To minimise the total team cost to complete all tasks, it is vital that robots are allocated tasks that minimise each robot's local costs. A common approach for minimising local costs is greedy-based auctions which let robots bid for tasks relative to the costs of completing their existing commitments. Generally, this approach works well, however, it is poor at identifying tasks that are considered undesirable by all robots, e.g., tasks that have large completion times and/or poor inter-task synergies. These undesired tasks are often the last to be allocated and can cause large increases in the total team cost, resulting in task allocations where the majority of tasks are completed quickly but the total task completion time is high.

In this paper we propose the novel calculation of a *Task Cost Dispersion (TCD) Value* which measures the robots' collective preference for each task. Our method inspects all robots' bids to identify tasks for which robots have no strong collective preferences. We then ensure that these tasks are allocated before other tasks. Our empirical results show this method reduces the total time taken to complete all tasks.

Problem Formalisation

MRTA problems vary widely and their scope includes problems where costs, tasks and robots may not be independent. Tasks may be complex requiring other tasks to be completed *a priori* and completion of some tasks may require multiple robots to coordinate; robots may be constrained to a maximum number of tasks they can execute at any one time or complete overall; and, task costs may change according to a robot's other task commitments. A recently published taxonomy classifies MRTA problems according to these differences (Korsah, Stentz, and Dias 2013). At the top level it divides MRTA problems according to inter-task relatedness:

No Dependencies tasks are completed by individual robots without inter-task synergies and dependencies.

In-schedule Dependencies tasks are completed by individual robots with intra-schedule dependencies. That is, task costs are relative to a robot's other task commitments.

Cross-schedule Dependencies tasks may need to be completed in unison with other robots and the costs are relative to robots negotiating the task completion order.

Complex Dependencies tasks may need to be decomposed into subtasks with cross-schedule dependencies, each robot's costs is dependent upon the task decomposition and tight cooperation with other robots.

In this paper we study problems with in-schedule dependencies, for which multi-robot routing is considered the standard testbed of this classification of MRTA problems (Dias et al. 2006). We explore two different types of task structures: *elemental tasks* (e-tasks) which require robots to visit a single point location $t = l_v$ and *simple tasks* (s-tasks) which consist of multiple e-tasks and have constraints on the task execution ordering. In our scenarios, s-tasks are tasks with pickup l_p and delivery l_d locations, the structure of these tasks is a tuple $t = \langle l_p, l_d \rangle$.

For problems with e-tasks, robots are classed as *single-task* (ST), that is a robot can only execute one task at a time and capacity constraints may apply to the total number of tasks a robot can execute. This is a variant of the multiple travelling salesman problem (Bektas 2006). For problems with s-tasks, robots may also be of type *multi-task* (MT). This class of robot is able to execute multiple tasks in parallel. A robot is considered to be executing a task when it begins travelling to a task's pickup location until it reaches its delivery location. Robots may also be constrained in the number of tasks that they are able to execute in parallel. These constraints are representative of real robots which may have a fixed maximum number of items that they can carry. In the operations research domain this problem is referred to as the *Vehicle Routing Problem with Pickup and Delivery* (Desaulniers et al. 2002).

We follow (Koenig et al. 2007) to define a solution to this type of MRTA problem. Given a set of robots $R = \{r_1, \ldots, r_m\}$ and a set of tasks $T = \{t_1, \ldots, t_n\}$. A partial solution to the MRTA problem is given by any tuple $\langle T_{r_1}, \ldots, T_{r_m} \rangle$ of pairwise disjoint task subsets:

$$T_{r_i} \subseteq T \text{ with } T_{r_i} \cap T_{r_{i'}} = \emptyset, i \neq i', \forall i = 1, \ldots, m$$

Each task subset T_{r_i} is then assigned to a single robot $r_i \in R$. To determine a complete solution we must find a partial solution with all tasks assigned to task subsets:

$$\langle T_{r_1} \ldots T_{r_m} \rangle \text{ with } \cup_{r_i \in R} T_{r_i} = T$$

Each robot always has private knowledge of its current location and can calculate the cost λ to travel between locations. The cost to travel between any two locations is equal across all robots. The robot cost $\lambda_{r_i}(T_{r_i})$ is the minimum cost for an individual robot r_i to visit all locations T_{r_i} assigned to it. Synergies between tasks assigned to a robot may be less than (positive synergy), or greater than (negative synergy) the sum of the individual costs for each task:

$$\lambda_{r_i}(\{t\}) + \lambda_{r_i}(\{t'\}) \neq \lambda_{r_i}(\{t\} \cup \{t'\})$$

Synergies allow robots to calculate costs for additional tasks relative to their current commitments.

Team objectives are used to provide additional guidance in the search for solutions to the task allocation that meet certain criteria. Lagoudakis *et al.* discuss team objectives in detail and their application to MRTA (Lagoudakis et al. 2005). In this work we use the MiniMax team objective: $\min \max_{r_i \in R} \lambda_{r_i}(T_{r_i})$, as this team objective reflects our desire to minimise the total task completion time.

Related Work

Solutions to MRTA problems can be found using centralised methods, such as, mixed integer programming (Koenig et al. 2007) or graph partitioning (Liu and Shell 2011). However, in all but the simplest problems, centralised methods are not efficient for MRTA (Dias and Stentz 2000). In addition, the information required by the centralised controller in decision making introduces large, and generally impractical, communication overheads into the system.

A number of common approaches for solving MRTA problems are based on distributed market-based auction algorithms (Dias et al. 2006). A standard auction is composed of three phases: the *initial phase* in which an auctioneer informs the robots of the tasks for auction; a *bidding phase* in which each robot evaluates the tasks for auction and responds with bids representing the costs to complete each task; and, a *winner determination* phase in which the auctioneer determines the winner for each task. In a MRTA auction every robot bids on the set of tasks available and is awarded tasks according to their bids. Bids are calculated using *bidding rules* which enable individual robots to measure the costs of tasks relative to the team objective. Additionally, some auction algorithms allow inter-task synergies to be considered during bid calculations. Furthermore, auctions can be run without any centralised auctioneer if all robots send all bids to each other and in parallel perform the same winner determination routine (Lagoudakis et al. 2005).

The *Contract Net Protocol* (CNP) is the foundation of many market-based systems (Smith 1980). Originally used in distributed computing, this protocol considers a system with no central control and describes a framework that covers the three auction phases. The CNP has been applied to the MRTA domain through the M+ scheme. Robots are assigned tasks one at a time. They are able to indicate tasks they wish to complete in the future and other robots are able to make counter offers (Botelho and Alami 1999). However, this approach gives no guarantees on the solution bounds.

In contrast, optimal solutions to the MRTA problem can be found using a *single-round combinatorial auction* where each robot is allocated at most one disjoint subset of tasks. To generate optimal solutions, robots calculate bids for every possible subset of tasks with inter-task synergies considered. This method is NP-complete and the computation tends to be intractable. It is therefore, not feasible for anything but the smallest scenarios (Berhault et al. 2003).

Sequential single-item (SSI) auctions offer a middle ground and solve MRTA problems with bounded solution costs over multiple bidding rounds. In each auction round, each robot calculates bids for all unallocated tasks and submits bids for every task on offer. The robot that bid the lowest for any task is then awarded that task, such that the overall team cost increases the least according to the team objective. Despite this, SSI auctions are not guaranteed to generate optimal solutions even if the robot costs are calculated optimally (Tovey et al. 2005). However, solutions generated relative to a variety of team objectives are bounded (Lagoudakis et al. 2005), they run in polynomial time (Koenig et al. 2006), and have been shown experimentally to perform very well (Tovey et al. 2005).

In this work we modify the winner determination phase of SSI auctions to reduce the influence of undesired tasks. A distributed algorithm for standard SSI auctions is given in Figure 1. This algorithm assumes a set of robots are supplied with a map of the environment, have perfect locali-

function SSI-Auction ($\bar{T}$,T_{r_i}, r_i, R)
Input: $\bar{T}$: the set of tasks to be assigned
T_{r_i}: the set of tasks presently assigned to robot r_i
r_i: the robot
R: the set of robots
Output: T_{r_i}: the set of tasks assigned to the robot r_i

1: **while** ($\bar{T} \neq \emptyset$)
2: *\/* Bidding Phase *\/*
3: **for each** task $t \in \bar{T}$
4: $\beta^t_{r_i}$ ←CalcBid(T_{r_i},t);
5: Send(β, R) | $B \leftarrow \bigcup_i$Receive(β_i, R);
6: */* Winner Determination Phase */*
7: **for each** task $t \in \bar{T}$
8: $M^t_{r'} \leftarrow$ arg min$_{(r' \in R)}$ B_t;
9: (r', t) ← arg min$_{(r' \in R, t \in \bar{T})}$ M;
10: **if** $r_i = r'$ **then**
11: $T_{r_i} \leftarrow T_{r_i} \cup \{t\}$;
12: $\bar{T} \leftarrow \bar{T} \backslash \{t\}$;

Figure 1: Algorithm for Sequential Single-Item Auctions.

sation, error free communication, and do not break down. These constraints are applied to enable us to focus on the auction process alone. A SSI auction begins and continues while there are unassigned tasks (Line 1). During the bidding phase (Lines 3-5) each robot calculates bids for every unassigned task and sends these bids to all other robots. The function CalcBid takes robot r_i's set of previously assigned tasks T_{r_i} and the task t to be bid on and uses a bidding rule to calculate a bid cost (Line 4). Each bid is a triple $\beta = \langle r_i, t, b_\lambda \rangle$ of a robot $r_i \in R$, a task $t \in T$ and a bid cost b_λ. For the MiniMax team objective, each robot's bid cost is the cost to complete all previously allocated tasks plus the cost for the additional task $b_\lambda = \lambda_{r_i}(\{T_{r_i} \cup t\})$. Robots send their bids and receive bids from all other robots in parallel (Line 5). The winner determination phase (Lines 6-12) consists of each robot finding the task with the lowest bid across all submitted bids. First the lowest bid for each task $M^t_{r'}$ is determined (Lines 7-8), from this the lowest overall bid is determined (Line 9) and the task awarded to the bidding robot (Lines 10-11). Ties are broken in an arbitrary manner. All robots then remove the awarded task from the set of unassigned tasks and the next round begins (Line 12).

A key strength of SSI auctions is their ability to build upon inter-task synergies during each bidding round. However, when robots have few tasks allocated, each robot's bids have a greedy bias towards tasks that are close to their initial locations. While this makes intuitive sense — as these nearby tasks have low costs relative to a robot's initial location — these initial task allocations have a large influence over the bid costs and preferences for other tasks and, without any ability to reallocate tasks, may result in large overall team costs. To address this, a variety of further improvements and extensions to SSI auctions have been studied which modify the bidding and winner determination phases of the auction process, trading off allocation time against overall team

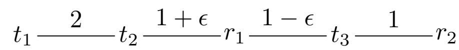

Figure 2: MRTA problem with three e-tasks and two robots.

	$T_{r_1} = \emptyset$	$T_{r_1} = \{t_1\}$	$T_{r_1} = \{t_2\}$	$T_{r_1} = \{t_3\}$
t_1	$3+\epsilon$	-	$3+\epsilon$	$5-\epsilon$
t_2	$1+\epsilon$	$3+\epsilon$	-	$3-\epsilon$
t_3	$1-\epsilon$	$5-\epsilon$	$3-\epsilon$	-

Table 1: Task bid costs for robot r_1 with prior commitments.

costs (Koenig, Keskinocak, and Tovey 2010).

One particular extension which seeks to overcome this problem, without modifying the bidding process, is *regret clearing*. In this extension, the winner determination phase of each auction round is modified such that the difference between the lowest and second lowest bids for any task which increase the overall team cost are maximised. The objective of this modification is, in each auction round, to allocate the task for which any robot has the strongest preference over all other robots. Despite the bounds for standard SSI auctions not applying to regret clearing, empirical evaluation indicates this approach works well for allocating tasks according to the MiniMax team objective with and without capacity constraints (Koenig et al. 2008).

Identifying Undesired Tasks

SSI and other sequential and greedy-based auction schemes, by design, prefer tasks with low costs. While these approaches are good at locally minimising the majority of each robot's task costs, certain tasks may have high costs and the allocation of these relative to other tasks can have a large impact on the overall team cost. Additionally, many of these approaches fail to consider alternative task allocation permutations in scenarios with tasks that are strongly preferred by multiple robots or tasks for which no robot has a strong preference.

Consider a simple MRTA problem with three e-tasks and two robots in a line (Figure 2). In this example, it is easy to see that, for both robots, with no existing task commitments, task t_1 has the highest cost and task t_3 the lowest cost. The optimal solution to this problem for the MiniMax team objective is $T_{r_1} = \{t_1, t_2\}$, $T_{r_2} = \{t_3\}$ which yields a cost of $3 + \epsilon$. For this problem, when no other tasks are allocated, any greedy-based auction algorithm will allocate task t_3 to robot r_1 which (a) immediately deviates from the optimal solution and (b) causes all other task costs to rise for robot r_1. This shows us that, despite task t_3 having a low cost and a high preference among the robots, it has poor inter-task synergies among other tasks (Table 1). In contrast, task t_1 has high costs but high inter-task synergies. For instance, if robot r_1 was allocated task t_1 before any other tasks, r_1's cost for task t_2 rises to $3+\epsilon$ but the relative increase in overall cost is 0 (as robot r_1's overall cost is already $3 + \epsilon$).

Unfortunately, avoiding team costs being heavily influenced by tasks with poor inter-task synergies, is not as simple as determining the tasks with highest costs and allocating

$$t_1 \overset{1}{\text{———}} t_2 \overset{1+\epsilon}{\text{———}} r_1 \overset{3-\epsilon}{\text{———}} t_3 \overset{3}{\text{———}} r_2$$

Figure 3: MRTA problem with modified task costs.

them first — although this idea contributes to the basis of our approach — as this can also easily result in poor solutions. For instance, in Figure 3 we modify the previous example's task costs. In this new scenario, if we allocate tasks by taking the maximum of the lowest bids for each task, task t_3 would still be allocated to robot r_1 in the first bidding round. This allocation would again deviate from the optimal solution and cause an increase in robot r_1's costs to complete other tasks. Furthermore, this approach ignores any robot's strong preferences (low bids) for certain tasks, e.g., in this example, robot r_1 has the strongest preference for task t_2.

A possible approach for overcoming these problems, which trades off individual task preferences with inter-task synergies, is for each robot to calculate bids for each task according to the influence this task allocation would have on the subsequent costs of all remaining unallocated tasks. Returning to the first example MRTA problem, allocating task t_2 to robot r_1 would have the smallest maximum increase in costs for other tasks (Table 1). However, if this approach was to be used in an auction algorithm, the number of calculations required per bid is increased by a factor of $|T|$, which in many cases would cause a substantial increase in the computation time for each bid. As an alternative, we seek to use each robot's preference for each task, which is supplied to us in the bidding phase of SSI auctions, to determine tasks that robots collectively find undesirable. In the following section, we explore a number of approaches that inspect the bids from all robots for all tasks and use this to measure the collective preferences for each task.

Winner Determination with Collective Preferences

A winner determination algorithm that is well informed and considers a large number of task allocation permutations is vital to minimising the overall team cost. Standard winner determination in SSI auctions is simplistic and selects the lowest overall bid for any task without consideration for other robots who may also have a high preference for this task. While regret clearing seeks to improve on this by selecting the robot and task with the largest difference in preference over all the second highest bidding robot, this approach still only focuses on preferences of two robots and not the collective desires of the whole team.

In contrast, approaches that attempt to allocate multiple tasks to multiple robots at the same time suffer from additional complexities in bid formation and winner determination. For instance, in combinatorial auctions, to determine the optimal allocation, after each robot has calculated bids for every subset of tasks, an exhaustive search of every tuple of pairwise disjoint combinations that are valid problem solutions is required. As this is generally impractical, robotics researchers have developed strategies to bid on limited subsets of tasks, such as, bidding on bundles of n or fewer targets (Berhault et al. 2003; Sandholm 2002), or forming greedy bids on bundles with low path costs (Berhault et al. 2003). However, even if the number of bids is reduced and a near optimal solution is accepted, winner determination still remains NP-complete (Sandholm 2002).

In SSI auctions, during the bidding process, every robot's individual preferences for each unallocated task is calculated. These task preferences include the costs to complete each task relative to each robot's previously allocated tasks. Using these task preferences, we can modify the winner determination phase of SSI auctions to calculate a TCD value for each task and then assign the task with the highest TCD value to the robot that bid the lowest for this task.

Each task's TCD value is a measure of the robots' collective preference for this task. A low TCD value indicates a task for which many robots have high preference, and a high value indicates a task which few robots desire. By allocating tasks according to the highest TCD value, we ensure that tasks which robots collectively find undesirable are allocated before tasks for which many robots have strong preferences. Formally, lines 7 - 9 of the SSI algorithm presented in Figure 1 are removed and replaced with:

for each task $t \in \bar{T}$
  $M^t_{r'} \leftarrow \text{calcDispersion}(R, t, B_t)$;
$(r', t) \leftarrow \arg\max_{(r' \in R, t \in \bar{T})} M$;

The new function CalcDispersion takes all the bids for a task B_t and returns a single replacement bid $M^t_{r'}$. This replacement bid continues to be a tuple $M^t_{r'} = \langle r_i, t, b_\lambda \rangle$ of a robot $r_i \in R$, the task t and a bid cost b_λ. In our replacement bid, the bidding robot is the robot that originally bid the lowest for this task, $r_i \leftarrow \arg\min_{(r' \in R)} B_t$. The replacement bid cost b_λ is the TCD value for the task.

Any function that measures the robots' collective preferences for a task (more formally the distribution of bids) can be used as a metric for calculating TCD values. In our empirical evaluations we explore a number of different standard statistical measures which seek to balance robot's individual preferences and the collective team preferences:

Minimum Bid (TCD-Min) By using the minimum bid for each task $b_\lambda \leftarrow \arg\min(B_t)$ as a task's TCD value, the next task assigned is the task for which every robot has the collective weakest preference. This method is good at identifying undesirable tasks, however, as analysed in the previous section, is poor at identifying tasks for which robots have strong preferences.

Average Bid (TCD-Avg) Calculating the average bid for each task $b_\lambda \leftarrow \text{average}(B_t)$ enables all robots' preferences to contribute to the TCD value. In this approach, the TCD value can be heavily influenced by robots with strong or weak preferences and, when these two extremes in preferences are both present, they may cancel each other out, thereby making it difficult to distinguish between tasks for which all robots have neither strong or weak preferences and those for which the preferences are highly divided.

Median Bid (TCD-Mid) By taking the median of each task's bids $b_\lambda \leftarrow \text{median}(B_t)$ we are able to remove the influence that extremely strong or weak preferences have on the collective preferences of the robots. This approach, however, only analyses the bids of at most two robots. The robot who will be assigned the task — through initially bidding the lowest for it — if this bid is deemed the winner, has no influence over the TCD value. Therefore, while the TCD-Mid value may be representative of the 'general' view of all robots, the robot assigned the task may not have a strong preference for the task.

Bid Range (TCD-Rng) Using the range between the highest and lowest bids for a task $b_\lambda \leftarrow \arg\max(B_t) - \arg\min(B_t)$ allows us to determine how varied the preferences for a task are. A low range suggests a task for which all robots have strong preference, whereas, a high range suggests a task that at least one robot has a strongly weak preference. While this approach enables the robot that will be awarded the task to contribute to the replacement bid cost, the value can be highly influenced by a single robot's weak preference and does not allow us to determine if other robots have high preferences for this task.

Maximum Bid Delta (TCD-Dlt) If we take the difference between the lowest and second lowest bids $b_\lambda \leftarrow \arg\min(B_t \backslash t) - \arg\min(B_t)$ we can determine tasks for which individual robots have a strong preference. This approach is similar to regret clearing with the difference that, for the MiniMax team objective, regret clearing uses *'increased bids'* which also consider the overall team cost for the current allocation to select the bid delta that increases the team cost the least (Koenig et al. 2008).

Experiments

We evaluate each TCD value calculation method on a variety of MRTA problems with in-schedule dependencies. We compare the overall team costs for each method to the costs obtained using standard SSI auctions and regret clearing.

Our simulated test world resembles an office-like environment with 16 rooms. Each room contains four doors that are independently opened or closed to allow or restrict travel between rooms (Figure 4). Robots can only travel through open doors and they cannot open or close doors. In each experiment configuration, it is guaranteed that there is at least one path between each room and every other room. This environment has become the standard testbed in recent literature (Koenig et al. 2007; 2008).

We repeat each experiment on 25 randomly generated configurations of opened and closed doors, with 10 robots and 60 tasks. In each configuration, each robot starts in a different random location and every robot is supplied a map of the environment at a resolution of 510x510 grid units. A grid unit covers a 5cm by 5cm area of space and gives an overall simulated space of 25.5m by 25.5m.

We test six different MRTA problem formulations with in-schedule dependencies. The first two of these problems use single-task (ST) robots and e-tasks with and without maximum task capacity constraints. These two configurations have been used previously to test SSI-based auction algorithms (Koenig et al. 2007; 2008). The third problem uses ST robots with s-tasks without capacity constraints. The final three problems use multi-task (MT) robots with s-tasks

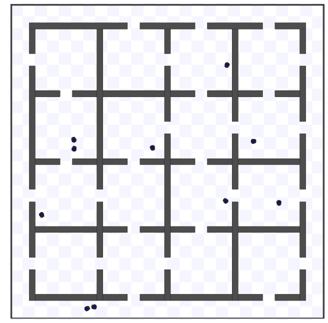

Figure 4: Simulation of robots in an office-like environment.

and parallel task execution constraints of three, five and unlimited tasks. These later four problem configurations are extensions of the previous tests to represent problems with pickup and delivery.

The mean team costs for each of these six problems are presented in Table 2. Across the ST robot results we make the following observations: allocations using TCD-Min values do not decrease the team cost relative to standard SSI auctions; for problems with e-tasks, allocations using TCD-Avg, TCD-Rng, and TCD-Dlt values result in lower final team costs than both standard SSI auctions and regret clearing; for s-tasks no TCD value calculation method produces lower average costs than SSI auctions; and across all three ST robot problems, allocations using TCD-Rng values produce costs that are nearest to the costs generated using standard SSI auctions.

The results for MT robot problems show all TCD value calculation approaches produce lower team costs than standard SSI auctions. However, only allocations using TCD-Avg or TCD-Mid values produce costs that are lower than SSI auctions with regret clearing. Overall, across each tested MRTA problem types (excluding ST robots with s-tasks), task allocations using either TCD-Avg or TCD-Mid values consistently outperform all other approaches tested.

To measure the strength of these results, we form the hypothesis that *modifying the winner determination phase of SSI auctions to select tasks with the maximum TCD value calculated using either the average or median of all submitted bids produces lower overall team costs than standard SSI auctions or regret clearing for the MiniMax team objective*. To validate this hypothesis, we perform one-sided Wilcoxon signed-rank tests for the results of each of the six MRTA problem types to measure the statistical significance of our results. For each test, the null hypothesis is defined as:

$$H_0 : \mu\lambda_{\text{Average/Median}} \geq \mu\lambda_{\text{SSI/Regret}}$$

and alternative hypothesis as:

$$H_1 : \mu\lambda_{\text{Average/Median}} < \mu\lambda_{\text{SSI/Regret}}$$

For the first problem with ST robots, e-tasks and no capacity constraints, allocations using TCD-Avg values have statistical significance of 0.94, and using TCD-Mid values, statistical significance of 0.98 over standard SSI auctions. However, compared to allocations with regret clearing neither approach produces statistically significant lower costs.

Single-Task Robots	SSI	Regret	TCD-Min	TCD-Avg	TCD-Mid	TCD-Rng	TCD-Dlt
E-Tasks No Constraints	707	671	816	**668**	**662**	737	**637**
E-Tasks Max Capacity 6	1258	1270	1431	**1096**	**1202**	**1250**	**1214**
S-Tasks No Constraints	4005	4260	4310	4191	4167	4084	4229

Multi-Task Robots	SSI	Regret	TCD-Min	TCD-Avg	TCD-Mid	TCD-Rng	TCD-Dlt
S-Tasks 3 Parallel Task Limit	2591	2344	2422	**2294**	**2216**	2422	2353
S-Tasks 6 Parallel Task Limit	2249	1945	2013	**1869**	**1862**	2047	1976
S-Tasks No Constraints	2080	1885	1971	**1826**	**1825**	1945	1887

Table 2: Mean Team Costs—bolded costs are lower than corresponding SSI and Regret Clearing costs.

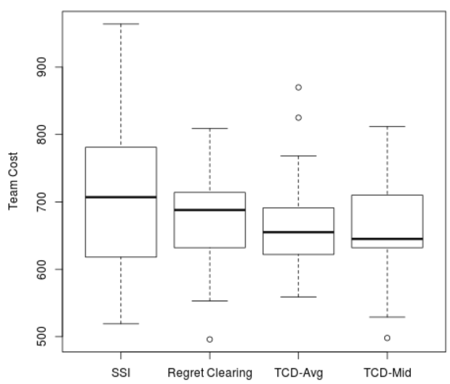

Figure 5: Boxplot of results distribution for e-tasks.

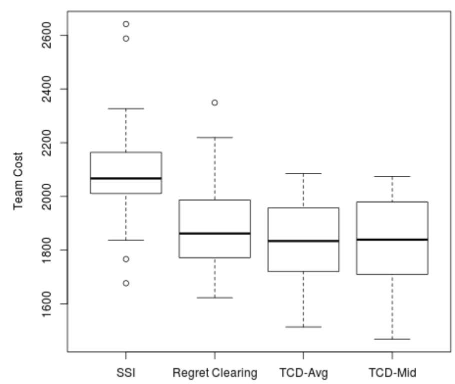

Figure 6: Boxplot of results distribution for s-tasks.

For the second problem (ST robots, e-tasks with maximum task capacity constraints) allocations using TCD-Avg values are statistically significant at 0.999 for both SSI and regret clearing, however, allocations using TCD-Mid values are not significant at the 0.95 threshold. This suggests that, overall, for problems requiring ST robots and e-tasks, allocations using TCD-Avg values produce the lowest costs.

For problems with MT robots and s-tasks even stronger statistical validity was measured. For all problems, statistical significance results greater than 0.999 were recorded for TCD values calculated with either average or median task costs when compared to standard SSI auctions. And when compared to regret clearing, the two approaches measured statistical significance in a range from 0.92 - 0.999.

As noted in the previous section, TCD values calculated with average costs are prone to heavy influence by weak preferences and values calculated using median bid values ignore strong preferences. While strong statistical significance results describe the average behaviour of each method, we are further interested in the relative distribution of the result set. In Figure 5 we plot the result distribution for ST robots with e-tasks and no capacity constraints, and in Figure 6 for MT robots with s-tasks and no capacity constraints.

The first of these plots shows that allocations generated with TCD-Mid values are heavily skewed towards lower costs compared to the other task allocation approaches. However, outlying high cost results cause the average cost for allocations using TCD-Mid values to be very close to those using TCD-Avg values (662 vs 668). This plot also shows that, while statistical significance testing showed no difference between regret clearing and TCD-Mid values, the distribution skew of each respective results differ substantially. The second distribution plot shows the clear benefit of using either TCD-Avg or TCD-Mid values for task allocation problems with s-tasks and MT robots. Additionally, it shows standard SSI auctions perform poorly on this type of problem, which is a type of problem that SSI auctions have not previously explored.

Conclusion and Future Work

In this paper we have shown the benefits of using the collective information contained across all bids in SSI auctions. By calculating a task cost dispersion value for each task we have ensured that tasks that are collectively undesired by robots are allocated before tasks which are more strongly preferred. Our empirical results have shown that this approach lowers the team cost for the MiniMax team objective when compared to SSI auctions.

In future work, focus should be given to constructing TCD value calculation functions for other team objectives. Additionally, further study of auctions for problems with pickup and delivery is essential as this is currently underrepresented in the literature.

References

Bektas, T. 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34(3):209–219.

Berhault, M.; Huang, H.; Keskinocak, P.; Koenig, S.; Elmaghraby, W.; Griffin, P. M.; and Kleywegt, A. J. 2003. Robot exploration with combinatorial auctions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1957–1962.

Botelho, S. C., and Alami, R. 1999. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1234–1239.

Desaulniers, G.; Desrosiers, J.; Erdmann, A.; Solomon, M. M.; and Soumis, F. 2002. VRP with pickup and delivery. *The vehicle routing problem* 9:225–242.

Dias, M., and Stentz, A. 2000. A free market architecture for distributed control of a multirobot system. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, 115–122.

Dias, M. B.; Zlot, R.; Kalra, N.; and Stentz, A. 2006. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE* 94(7):1257–1270.

Koenig, S.; Tovey, C. A.; Lagoudakis, M. G.; Markakis, E.; Kempe, D.; Keskinocak, P.; Kleywegt, A. J.; Meyerson, A.; and Jain, S. 2006. The power of sequential single-item auctions for agent coordination. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 1625–1629.

Koenig, S.; Tovey, C. A.; Zheng, X.; and Sungur, I. 2007. Sequential bundle-bid single-sale auction algorithms for decentralized control. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, 1359–1365.

Koenig, S.; Zheng, X.; Tovey, C. A.; Borie, R. B.; Kilby, P.; Markakis, V.; and Keskinocak, P. 2008. Agent coordination with regret clearing. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 101–107.

Koenig, S.; Keskinocak, P.; and Tovey, C. A. 2010. Progress on agent coordination with cooperative auctions. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)*, 1713–1717.

Korsah, G. A.; Stentz, A.; and Dias, M. B. 2013. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32(12):1495–1512.

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

Liu, L., and Shell, D. A. 2011. Multi-level partitioning and distribution of the assignment problem for large-scale multi-robot task allocation. In *Robotics: Science and Systems*.

Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artifical Intelligence* 135(1-2):1–54.

Smith, R. G. 1980. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* 29(12):1104–1113.

Tovey, C.; Lagoudakis, M.; Jain, S.; and Koenig, S. 2005. The generation of bidding rules for auction-based robot coordination. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, volume III. Springer. 3–14.