Sequential Single-Cluster Auctions for Robot Task Allocation

Bradford Heap and Maurice Pagnucco

ARC Centre of Excellence in Autonomous Systems
School of Computer Science and Engineering, UNSW
Sydney, NSW, 2052, Australia.
E-mail: {bradfordh|morri}@cse.unsw.edu.au

Abstract. Multi-robot task allocation research has focused on *sequential single-item auctions* and various extensions as quick methods for allocating tasks to robots with small overall team costs. In this paper we outline the benefits of grouping tasks with positive synergies together and auctioning clusters of tasks rather than individual tasks. We show that with task-clustering the winner determination costs remain the same as sequential single-item auctions and that auctioning task-clusters can result in overall smaller team costs.

1 Introduction

Consider a team of autonomous mobile robots operating in an office-like environment. These robots may be required to deliver documents between departments, clean up spillages, or act as tour guides to visitors. In many situations there will be a set of tasks to be completed and we wish for the robots to distribute these tasks amongst themselves in a manner that satisfies a global team objective. Recently, multi-robot cooperative auctions have become a popular approach for solving task-allocation problems [3].

We can achieve an optimal allocation of a set of tasks to robots using a *single-round combinatorial auction*. However, in most situations where there are many tasks, combinatorial auctions fail to perform efficiently due to high communication and winner determination costs [1]. As an alternative, much of the research focus has been on the use of *sequential single-item auctions* (SSI auctions) for task allocation over multi-round auctions [6]. Although SSI auctions produce a team cost that is at least as large as combinatorial auctions, they have much lower communication and winner determination costs which results in a much quicker allocation of tasks. To lower the team cost in SSI auctions researchers have looked at improvements and extensions to the bidding phases of SSI auctions through the use of techniques like *rollouts*, *regret clearing* and *bundle-bids* (the interested reader is referred to [5, 7]).

SSI auctions with bundles are an interesting hybrid of standard SSI auctions and combinatorial auctions in which each robot can bid on dynamic combinations of up to k tasks and, during the winner determination phase, a robot can be allocated between $0 - k$ tasks. In general, this approach results in lower team costs as each bundle bid takes into account more synergies between tasks, however, because of the additional calculations involved in the bidding and winner determination phases it performs a lot slower than standard SSI auctions.

In this paper we extend the idea of bidding on a collection of tasks and allow robots to bid on fixed clusters of tasks where a robot will either win all items in the cluster or none. We show empirically that this method results in lower team costs than standard SSI auctions and performs much faster than SSI auctions with bundle bids. More specifically, we demonstrate that SSC auctions result in lower MiniMax distances than SSI auctions when the number of robots is greater than 2. Moreover, for the MiniSum team objective, SSC auctions perform well when the capacity constraint is small.

2 Multi-Robot Task-Allocation

We formalise the definition of the task-allocation problem in the same manner as Koenig *et al.* [7]. Given a set of robots $R = \{r_1, \ldots, r_m\}$ and a set of tasks $T = \{t_1, \ldots, t_n\}$, any tuple $\langle T_{r_1}, \ldots, T_{r_m} \rangle$ of pairwise disjoint bundles $T_{r_i} \subseteq T$ and $T_{r_i} \neq T_{r_j}$ for $i \neq j$, for all $i = 1, \ldots, m$, is a partial solution of the task-allocation problem. This means that robot r_i performs the tasks T_{r_i}, and no task is assigned to more than one robot. To determine a complete solution to the task-allocation problem we need to find a partial solution $\langle T_{r_1} \ldots T_{r_m} \rangle$ with $\cup_{r_i \in R} T_{r_i} = T$, that is, where every task is assigned to exactly one robot.

The standard testbed of the task-allocation problem is multi-robot routing. The tasks represent locations to visit. Robots know their locations and can calculate the costs between locations. We assume costs are symmetric, $\lambda(i, j) = \lambda(j, i)$ and are the same for 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. There can be synergies between tasks, such that, $\lambda_{r_i}(T_{r'}) + \lambda_{r_i}(T_{r''})$ may not equal $\lambda_{r_i}(T_{r'} \cup T_{r''})$. A positive synergy is when $\lambda_{r_i}(T_{r'} \cup T_{r''}) < \lambda_{r_i}(T_{r'}) + \lambda_{r_i}(T_{r''})$. Robots can also have capacity constraints where they can have at most a fixed number of tasks. We wish to find a solution to the task-allocation problem that achieves a team objective. In this paper we study two common team objectives:

MiniMax $\max_{r_i \in R} \lambda_{r_i}(T_{r_i})$, that is to minimise the maximum distance each individual robot travels.

MiniSum $\sum_{r_i \in R} \lambda_{r_i}(T_{r_i})$, that is to minimise the sum of the paths of all robots in visiting all their assigned locations.

These two team objective result in different allocations of tasks due to how each robot calculates their bids incorporating synergies between tasks. Lagoudakis *et al.* [8] explores these differences in more detail.

3 Sequential Auctions with Clusters

Auction-based methods for task allocation have become increasingly popular in the recent literature. An auction is composed of three separate phases: the *initial phase* in which an auctioneer sends a request to all robots indicating the tasks up for auction; a *bidding phase* in which each robot evaluates the tasks up for auction and responds with a bid for those in which it is interested; and, a *winner determination* phase in which the auctioneer determines the winner for each task. Common auction types include *combinatorial auctions*, *parallel auctions* and *sequential auctions*. In combinatorial auctions each robot bids on all subsets of the tasks on offer. This yields optimal results but the computation tends to be intractable and is certainly not feasible for any but the smallest scenarios. In parallel auctions the robots develop a bid for each task and the auctioneer then allocates the tasks all at once. The computational complexity is minimal but solutions are likely to be sub-optimal. Sequential auctions represent a compromise between these two extremes. They progress over several rounds in which a subset of tasks is auctioned in each round. In the case of SSI auctions, one item (i.e., task) is auctioned in each round.

We now develop an extension to SSI auctions in which individual tasks are organised into clusters taking into account positive synergies between tasks. Robots bid on these clusters to solve the task-allocation problem. We call this *sequential single-cluster auctions* (SSC auctions). An SSC auction consists of three phases: clustering phase, bidding phase, and winner determination phase. Initially, all tasks are unassigned. Before the auction, a clustering algorithm is used to allocate all individual tasks into a cluster with the goal of maximising the positive synergy between tasks in each cluster (clustering phase). Each task can be assigned to one, and only one cluster. Clusters can be of varying sizes. During each round, all robots bid on all unassigned task clusters (bidding phase), the auctioneer then determines the winner and assigns the winning cluster to the winning robot (winner determination phase). The winning robot must then complete all tasks in that cluster.

Clustering Phase: Expanding upon our definition of the task-allocation problem given in Section 2 we introduce the set of clusters $C = \{c_1, \ldots, c_o\}$. We now need to allocate all tasks to one and only one cluster. This is achieved by taking any tuple $\langle T_{c_1}, \ldots, T_{c_o} \rangle$ of pairwise disjoint bundles $T_{c_j} \subseteq T$ for all $j = 1, \ldots, o$ that satisfies $\cup_{c_j \in C} T_{c_j} = T$. For multi-robot routing the synergy between tasks is represented by the distance between them. Tasks with a large distance separating them have a low synergy, whereas, tasks with a small distance have a high positive synergy. In this paper, we use the standard k-means algorithm [4] for clustering tasks during the empirical experimentation. However, our proposal does not depend upon k-means and other clustering methods that satisfy these properties may produce better results.

Once we have organised all tasks into clusters we must ensure that all clusters are allocated to one and only one robot. We do this by taking any tuple $\langle C_{r_1}, \ldots, C_{r_m} \rangle$ of pairwise disjoint bundles $C_{r_i} \subseteq C$ for all $i = 1, \ldots, m$ that satisfy $\cup_{r_i \in R} C_{r_i} = C$. As a result of this we have now allocated all tasks into clusters, and assigned all clusters to robots and therefore it holds that we still have a valid solution to the task-allocation problem of all tasks being allocated such that each task is allocated to one and only one robot.

Now we consider a single round of a SSC auction. We assume that robot $r_i \in R$ has already been assigned the set of task clusters $C_{r_i} \subseteq C$ in previous rounds for all $r_i \in R$. Therefore $U = C \setminus \cup_{r_i \in R} C_{r_i}$ is the set of unassigned task clusters. Let a bid b be a triple of a robot b_r, a task cluster b_c and a bid cost b_λ, such that, $b = \langle b_r, b_c, b_\lambda \rangle$.

Bidding Phase: The set of submitted bids $B = \{b_1, \ldots, b_m\}$ satisfies: 1) for all $b \in B$, it holds that $b_r \in R$ and $b_c \in U$; and 2) for all $r_i \in R$ and $c' \in U$ there exists exactly one bid $b \in B$ with $b_r = r_i$ and $b_c = c'$. That is each robot submits one bid on each task cluster. For the MiniMax team objective, $b_\lambda = \lambda_{b_r}(C_{b_r} \cup \{b_c\})$. That is the robot bids the costs to do all tasks assigned to it plus the tasks in the cluster it is bidding on. For the MiniSum team objective, $b_\lambda = \lambda_{b_r}(C_{b_r} \cup \{b_c\}) - \lambda_{b_r}(C_{b_r})$. That is the robot bids the increase in its costs for doing all of its currently allocated tasks plus the tasks in the cluster it is bidding on.

Winner Determination Phase: Once all bids have been received, the auctioneer evaluates a potentially winning bid $b' \in B$ according to the value b'_λ. The winning bid for both the MiniMax and MiniSum team objective is the bid b' with the smallest b'_λ. The auctioneer then assigns all tasks in the cluster b'_c to the robot b'_r.

4 Properties

We now describe the unique behavioural properties of SSC auctions. These properties allow SSC auctions to operate in an efficent manner and generally result in a small team cost.

1. The number of rounds in a SSC auction is no more than the number of rounds in a SSI auction.
   Proof: We define an SSI auction as the tuple $A_{ssi} = \langle R, T \rangle$ where R represents the set of available robots and T the set of tasks. The number of rounds in A_{ssi} is equal to the number of tasks, $N_{ssi} = |T|$, as only one task is allocated per round. We define an SSC auction is the tuple $A_{ssc} = \langle R, T, C \rangle$. The number of rounds in A_{ssc} is equal to the number of clusters, $N_{ssc} = |C|$, as one cluster is allocated per round. Each cluster can have one or more tasks, therefore, $|C| \leq |T|$, and as a result of this $N_{ssc} \leq N_{ssi}$.
2. Winner determination time in a SSC auction is equal to winner determination time in a SSI auction.
   Proof: In an SSI auction each bid b_s consists of a robot b_r, a task b_t, and a cost b_λ. In an SSC auction the structure of a bid remains the same, with the exception that b_t is replaced by b_c (as defined in Section 3). For winner determination, we have a set of bids B and the value of each b_λ is compared in the same manner in both auction frameworks and $|B|$ does not change. Therefore the winner determination time does not change.
   N.B. SSC winner determination time is much faster than SSI with bundles. This is because in SSI with bundles each bid must include b_λ for each combination of the k tasks that is being bid on. To determine the winner in SSI

with bundles each b_λ for each combination needs to be compared to all other bids and combinations to determine the winner.

3. When clusters employ positive synergies between tasks the resultant team cost in a SSC auction is less than in a SSI auction.

   Take for example, the same task-allocation problem as *Exploration Task 4* in Koenig *et al.* [6] (Figure 1). In this problem an SSI auction fails to consider enough synergies between tasks and results in a less than optimal solution. For the MiniSum team objective the overall distance sum is 20 and the resultant paths for each robot to traverse are $r_1 \rightarrow t_2 \rightarrow t_1$ and $r_2 \rightarrow t_4 \rightarrow t_3$. For an SSC auction we define our clusters $c_1 = \{t_1, t_3\}$ and $c_2 = \{t_2, t_4\}$. Auctioning with the MiniSum team objective results in an allocation of c_1 to r_2 and c_2 to r_1 with the resultant paths $r_1 \rightarrow t_2 \rightarrow t_4$ and $r_2 \rightarrow t_3 \rightarrow t_1$. The overall distance sum is 15. However, it should be noted that if a cluster fails to employ synergies correctly SSC auctions may result in team costs that are worse than SSI auctions.

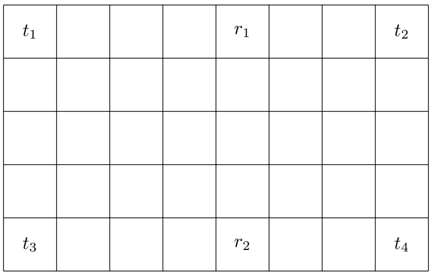

Fig. 1: Exploration Task 4 (Koenig *et al.* [6])

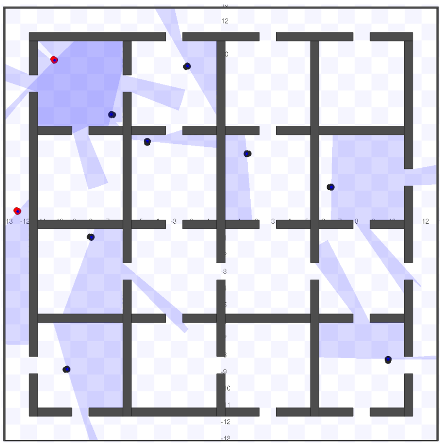

Fig. 2: Simulation of an office-like environment (cf. Koenig *et al.* [6])

5 Experiment Setup

To test SSC auctions we simulate an office-like environment (Figure 2) as in Koenig *et al.* [6]. For each experiment, doors between different rooms and the hallway are either opened or closed. We tested on 25 different randomly generated configurations of opened and closed doors with each robot in each configuration starting in a different random location which is standard in the literature and therefore provides a common setting for comparison. Robots can only travel between rooms through open doors and cannot open or close doors. In each experiment robots are set a fixed task-capacity constraint of the ratio of the number of tasks to the number of robots. Robots stop being allocated additional tasks once these capacities are met. For each configuration we test with $|R| \in \{2, 4, 6, 8, 10\}$ and $|T| \in \{6, 7, \ldots, 60\}$.

We use standard k-means clustering to quickly create clusters of geographically close tasks to be auctioned. It is important to note that k-means clustering does not take into account walls and closed doors. This means that it is possible for tasks to be clustered together that may have a large navigational distance between them (low synergy). However, this approach best represents a real world situation where it would be extremely complex to always create an optimal grouping of tasks. For our experiments we test two different total numbers of task-clusters. Our first experiment uses a cluster count of half the number of tasks, and the second uses a cluster count of two-thirds the number of tasks.

For each auction round robots bid on the cluster that will result in the lowest increase to the team objective. To determine their bid cost each robot needs to solve a version of the travelling salesperson problem (TSP) where it needs to travel to all tasks allocated to it but does not return to its initial location. Solving the TSP is an NP-Hard problem so we need to approximate the true cost. We do this by using the cheapest-insertion heuristic to add new tasks into our path and then use the two-opt heuristic [2] to improve our solution.

To compare the effectiveness of SSC auctions we also run parallel, SSI, and SSI with bundles auctions on the same 25 configurations. For SSI with bundles we test $k = 2$ and $k = 3$ with a *non-cautious auctioneer*, that is, all k tasks are allocated in each round. Furthermore, we test *hard* and *soft* capacity constraints for SSI with bundles. Hard capacity constraints ensure that all robots are allocated exactly their capacity of tasks. Soft capacity constraints allow robots to go slightly over their capacity, provided they are under their capacity before the round winner determination and allocation. This comparison of capacity constraints is necessary because SSC auctions may result in allocations where robots are slightly over their capacities because of the requirement that all tasks in a cluster are allocated to the same robot.

6 Results

We begin our analysis with the MiniMax Team Objective with the mean experimental results shown in Table 1. We observe that in all Robot/Task combinations tested that SSC auctions result in a lower mean MiniMax result than SSI auctions. Overall there is an average MiniMax distance reduction of 20% where the number of clusters $|C| = \frac{1}{2}|T|$ and a reduction of 25% where the number of clusters $|C| = \frac{2}{3}|T|$. However, $|C| = \frac{2}{3}|T|$ does not result in lower mean MiniMax distances than $|C| = \frac{1}{2}|T|$ in all Robot/Task combinations. We also note that SSI auctions with bundles also result in lower mean MiniMax distances than both standard SSI auctions and SSC auctions. Interestingly SSI auctions with bundles where $k = 3$ do not always result in lower results than SSI auctions with bundles where $k = 2$ for all Robot/Task combinations. This result, however,

			Standard		SSC		SSI bundles $k = 2$		SSI bundles $k = 3$	
Capacity	Robots	Tasks	Parallel	SSI	$\|C\| = \frac{1}{2}\|T\|$	$\|C\| = \frac{2}{3}\|T\|$	Hard-Cap	Soft-Cap	Hard-Cap	Soft-Cap
3	2	6	1039	1130	1085	944	823	811	607	613
3	4	12	1094	1138	880	946	828	808	762	755
3	6	18	1060	1156	899	833	743	704	730	675
3	8	24	1199	1112	853	760	668	680	763	706
3	10	30	1092	1159	802	733	656	651	670	636
4	2	8	1318	1284	1242	1108	965	950	1060	1194
4	4	16	1430	1239	1034	1042	880	851	1038	969
4	6	24	1301	1352	1030	868	779	781	762	679
4	8	32	1310	1299	857	856	747	767	789	999
4	10	40	1438	1249	889	758	704	687	821	855
5	2	10	1464	1364	1260	1257	1132	1101	1326	1248
5	4	20	1545	1297	1138	1142	928	905	1001	1119
5	6	30	1485	1289	1087	1003	850	835	915	853
5	8	40	1506	1341	989	952	819	797	974	891
5	10	50	1574	1347	933	872	773	732	850	1051
6	2	12	1699	1690	1421	1459	1231	1197	1092	1117
6	4	24	1711	1457	1274	1142	1039	1010	972	923
6	6	36	1782	1409	1129	1051	840	884	1076	1061
6	8	48	1713	1463	1132	1012	907	812	894	964
6	10	60	1736	1492	957	909	836	813	928	856
Overall Mean:			1425	1313	1045	983	857	839	901	908

Table 1: Mean MiniMax Experimental Results

is consistent with Koenig's prior results for SSI auctions with bundles where a non-cautious auctioneer has been used [7]. Despite, SSI with bundles producing lower results than SSC the computational overhead is significantly higher and the consequences of this are discussed further below.

To confirm the validity of our results we perform *two-sample independent one-tailed t tests* comparing the SSC auction results to the SSI auction results for each Robot/Task combination. We define our null hypothesis as $H_0 : \mu A_{ssc} \geq \mu A_{ssi}$ and our alternative hypothesis as $H_a : \mu A_{ssc} < \mu A_{ssi}$, that is, we wish to prove that the mean result for SSC auctions are lower than SSI auctions. We declare any result a significant difference if the result of the t test P is less than 0.05, that is, the probability of the decrease between the mean results of SSC auctions compared to SSI auctions being a result of random variation is less than 5%.

The significance tests show that in all but three Robot/Task combinations we have a statistically significant reduction in the MiniMax distance, that is, we accept the alternative hypothesis. The non-significant results occur, in both $|C|$ sizes, when there are only 2 robots with total tasks $\{6, 8, 10\}$. However, in these scenarios we can expect that clustering will not perform well due to the low numbers of robots and tasks.

Finally we perform *two-sample independent two-tailed t tests* for the difference between the cluster sizes for all Robots/Tasks combinations ($H_0 : \mu A_{|C|=\frac{1}{2}|T|} = \mu A_{|C|=\frac{2}{3}|T|}$, $H_a : \mu A_{|C|=\frac{1}{2}|T|} \neq \mu A_{|C|=\frac{2}{3}|T|}$). Only two combinations, $\langle |R| = 6, |T| = 24\rangle$ and $\langle |R| = 10, |T| = 40\rangle$, result in a significant difference between the two cluster sizes, in which $|C| = \frac{2}{3}|T|$ produces the smallest distances. Overall we can conclude that SSC auctions result in lower MiniMax distances than SSI auctions when the number of robots is greater than 2.

			Standard		SSC		SSI bundles $k = 2$		SSI bundles $k = 3$	
Capacity	Robots	Tasks	Parallel	SSI	$\|C\| = \frac{1}{2}\|T\|$	$\|C\| = \frac{2}{3}\|T\|$	Hard-Cap	Soft-Cap	Hard-Cap	Soft-Cap
3	2	6	1653	1819	1589	1615	1661	1617	1398	1398
3	4	12	2757	2867	2331	2411	2243	2378	1997	1984
3	6	18	3580	3542	2864	2982	2643	2628	2284	2285
3	8	24	4723	4395	3366	3596	3191	3281	2489	2462
3	10	30	5057	4928	3764	3869	3394	3408	2751	2663
4	2	8	2085	1941	1889	1971	1857	1796	1850	1844
4	4	16	3564	3180	2783	2892	2641	2637	2514	2497
4	6	24	4417	4033	3448	3718	3268	3301	2612	2652
4	8	32	5428	4780	3749	4144	3607	3703	3125	3391
4	10	40	6370	5391	4371	4605	3998	4181	3183	3442
5	2	10	2378	2149	2202	2198	2154	2127	2444	2145
5	4	20	4026	3029	3170	3372	2981	3038	3019	2933
5	6	30	5129	4086	4044	4078	3677	3825	3025	2842
5	8	40	6334	4741	4549	4637	4221	4320	3529	3403
5	10	50	7087	5353	4745	5078	4526	4848	3850	3947
6	2	12	2834	2628	2397	2417	2482	2464	2402	2478
6	4	24	4435	3537	3512	3498	3360	3407	3014	2859
6	6	36	5941	4475	4302	4139	4207	4081	3377	3593
6	8	48	7234	5268	5028	5022	4753	4961	3791	3824
6	10	60	8059	5805	5523	5731	5289	5093	4007	4170
Overall Mean:			4654	3897	3481	3599	3308	3355	2833	2841

Table 2: Mean MiniSum Experimental Results

The mean results of the MiniSum Team Objective is shown in Table 2. Overall there is a mean MiniSum distance reduction of 12% where the number of clusters $|C| = \frac{1}{2}|T|$ and a reduction of 8% where the number of clusters $|C| = \frac{2}{3}|T|$ when compared to SSI auctions. However, in contrast to the MiniMax results, there is not a mean distance reduction in every Robot/Task combination. In particular, the combination $\langle |R| = 4, |T| = 20 \rangle$ shows a substantial increase in the MiniSum distances in both cluster sizes. The results for our experiments using SSI with bundles show that they result in lower distances than SSI and SSC auctions. We observe that in experiments with bundle size $k = 3$ the mean distance is consistently lower than experiments with bundle size $k = 2$. This is in line with, and validates, Koenig's previous work on SSI with bundles.

We perform *two-sample independent one-tailed t tests* comparing the SSC auction results to the SSI auction results for those Robot/Task combinations where the SSC result is less than the SSI result ($H_0 : \mu A_{ssc} \geq \mu A_{ssi}$, $H_a : \mu A_{ssc} < \mu A_{ssi}$), that is, we test for a statistically significant decrease in the mean results of SSC compared to SSI. When the SSC result is greater than the SSI result we perform *two-tailed t tests* for a difference between the two samples ($H_0 : \mu A_{ssc} \neq \mu A_{ssi}$, $H_a : \mu A_{ssc} = \mu A_{ssi}$), that is, we test for no statistically significant difference between the mean results.

The results of these tests give an interesting partition of the data. In experiments where the robot capacity is 3 or 4 we confirm a significant result in the reduction of the mean MiniSum distances for all combinations except those where $|R| = 2$. However, in all cases where the robot capacity is 5 or 6 we get no significant difference between the SSI and SSC auctions, except, in the previously mentioned combination $\langle |R| = 4, |T| = 20 \rangle$ with $|C| = \frac{2}{3}|T|$ which, in the two-tailed t tests, confirmed a significant increase in distance.

The MiniSum results are not in line with our predictions. The raw data appears to show SSC auctions mostly performing better than SSI auctions. However, our statistical testing does not confirm this. We can conclude that when the capacity constraint is small SSC auctions perform well. However, more experiments are needed to examine situations where robots are allocated many tasks. For instance using a different clustering algorithm, such as potential fields or graph partitioning, may produce a significant reduction in the distance.

			Standard		SSC		SSI bundles $k = 2$		SSI bundles $k = 3$	
Capacity	Robots	Tasks	Parallel	SSI	$\|C\| = \frac{1}{2}\|T\|$	$\|C\| = \frac{2}{3}\|T\|$	Hard-Cap	Soft-Cap	Hard-Cap	Soft-Cap
3	2	6	1.9	2.4	2.6	2.5	3.3	2.4	2.9	2.9
3	4	12	4.0	7.8	8.2	8.1	14.4	14.9	14.9	15.2
3	6	18	7.3	16.3	16.4	16.1	46.9	47.2	47.4	47.5
3	8	24	12.7	29.6	32.1	29.5	117.8	119.0	121.3	120.8
3	10	30	20.0	47.2	49.3	46.4	245.3	244.8	244.9	242.3
4	2	8	2.3	3.4	3.5	3.4	4.1	4.0	4.4	4.4
4	4	16	5.1	11.7	11.7	11.5	23.4	23.9	24.9	24.9
4	6	24	9.4	25.4	26.2	24.8	80.2	80.3	82.9	81.0
4	8	32	16.1	45.2	46.8	45.8	201.0	200.4	207.5	207.7
4	10	40	29.8	72.9	73.4	72.5	415.0	422.8	419.9	431.4
5	2	10	2.7	4.4	4.6	4.6	5.5	5.5	5.5	5.4
5	4	20	6.1	16.1	16.2	16.3	34.2	34.2	35.3	35.6
5	6	30	11.4	35.9	36.5	35.6	120.8	123.1	124.7	125.4
5	8	40	19.9	65.3	66.1	65.5	312.5	326.3	320.5	315.2
5	10	50	31.4	104.3	105.0	102.4	649.5	649.4	659.9	661.9
6	2	12	3.2	5.7	6.9	5.7	7.1	7.1	7.2	8.1
6	4	24	7.2	21.4	22.1	25.5	48.3	48.4	50.5	50.6
6	6	36	14.1	47.1	50.5	49.0	171.7	170.5	180.4	178.7
6	8	48	24.3	87.1	87.8	87.0	445.7	432.6	445.4	458.4
6	10	60	37.9	138.4	146.4	140.0	908.4	890.0	929.5	1082.3
Overall Mean:			8.9	26.3	27.1	26.4	128.5	128.2	131.0	136.7

Table 3: Mean Total Task-Allocation Determination Time *(seconds)*

Table 3 shows the mean time to run auctions and allocate all tasks for each Robot/Task combination. For all auctions except SSC we begin timing when the robots are informed of the tasks to bid on and stop timing when all tasks have been allocated. For the SSC auctions we begin timing when the clustering algorithm begins and stop when all tasks have been allocated.

Parallel auctions are always the quickest auction to finish, however, they produce the most sub-optimal distance results. Standard SSI are on average around three times slower than Parallel auctions. SSC auctions run in a comparable time to SSI auctions. This is an important point because SSC auctions need to generate the task clusters before auctions can begin which can take considerable time. However, once the auctioning phases begin they are quicker than SSI auctions because they have fewer auction rounds. This result validates our properties from Section 4 and analysing both the mean distance results and the timing results empirically demonstrates that SSC auctions can result in a lower team objective distance in a similar time to SSI auctions.

Finally, SSI auctions with bundles perform around five times slower than SSI and SSC auctions and 13 times slower than Parallel auctions. Although SSI auctions with bundles produce the lowest team objective distances the performance trade-off cost is very high.

7 Conclusions and Further Work

In this paper we have shown the benefits of SSC auctions as an alternative to SSI auctions for the allocation of tasks to robots. We developed the theoretical foundations of SSC auctions and outlined their unique behavioural properties. Using the standard multi-robot routing test-bed we demonstrated empirically that SSC auctions can produce smaller team objective results than SSI auctions. We also compared these results to another extension of SSI auctions which involves grouping tasks, *SSI auctions with bundles*, and showed that SSC auctions perform much quicker.

This paper provides scope for further investigation of SSC auctions. For instance, a comparison of the effectiveness of different clustering algorithms could provide an insight into the trade-off between run-time speed and the optimality of the final allocation. Applying SSC auctions to dynamic task allocation and reallocation in a manner similar to [9] can also be considered. Finally, clustering non-homogeneous tasks could be advantageous in the quick allocation of complex task sets.

References

1. Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., Kleywegt, A.: Robot exploration with combinatorial auctions. In: Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. vol. 2, pp. 1957–1962. IEEE (2003)
2. Croes, G.: A method for solving traveling-salesman problems. Operations Research 6(6), 791–812 (1958)
3. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (2006)
4. Hartigan, J., Wong, M.: A k-means clustering algorithm. Journal of the Royal Statistical Society C 28(1), 100–108 (1979)
5. Koenig, S., Keskinocak, P., Tovey, C.: Progress on agent coordination with cooperative auctions. In: Proceedings of the AAAI Conference on Artificial Intelligence (2010)
6. Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., Jain, S.: The power of sequential single-item auctions for agent coordination. Proc. AAAI-06 (2006)
7. Koenig, S., Tovey, C., Zheng, X., Sungur, I.: Sequential bundle-bid single-sale auction algorithms for decentralized control. Proc. IJCAI-07 pp. 1359–1365 (2007)
8. Lagoudakis, M., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., Jain, S.: Auction-based multi-robot routing. Proc. Int. Conf. on Robotics: Science and Systems pp. 343–350 (2005)
9. Schoenig, A., Pagnucco, M.: Evaluating sequential single-item auctions for dynamic task allocation. AI 2010: Advances in Artificial Intelligence pp. 506–515 (2011)