Sequential Bundle-Bid Single-Sale Auction Algorithms for Decentralized Control

Sven Koenig
USC
Computer Science
skoenig@usc.edu

Craig Tovey
Georgia Institute of Technology
Industrial and Systems Engineering
ctovey@isye.gatech.edu

Xiaoming Zheng
USC
Computer Science
xiaominz@usc.edu

Ilgaz Sungur
USC
Industrial and Systems Engineering
sungur@usc.edu

Abstract

We study auction-like algorithms for the distributed allocation of tasks to cooperating agents. To reduce the team cost of sequential single-item auction algorithms, we generalize them to assign more than one additional task during each round, which increases their similarity to combinatorial auction algorithms. We show that, for a given number of additional tasks to be assigned during each round, every agent needs to submit only a constant number of bids per round and the runtime of winner determination is linear in the number of agents. The communication and winner determination costs do not depend on the number of tasks and thus scale to a large number of tasks for small bundle sizes. We then demonstrate empirically that the team cost of sequential bundle-bid single-sale (= single-item) auction algorithms can be substantially smaller than that without bundles for multi-agent routing problems with capacity constraints.

1 Introduction

We study the distributed allocation of tasks to cooperating agents, where each task has to be assigned to exactly one agent so that the team cost is small (= team performance is high). Auction algorithms promise to solve these combinatorial task-assignment problems with small communication and computation costs since the agents compress information into a small number of bids, which they compute in parallel and then exchange [Dias *et al.*, 2005]. Ideal combinatorial auctions consist of a single round, after which all tasks have been assigned to agents. All agents bid on all bundles (= sets) of tasks and the auctioneer then assigns all tasks to agents, which allows the agents to take synergies among tasks into account in their bids and results in a minimal team cost but incurs prohibitively large communication and winner determination costs. The communication and winner determination costs remain large even if the agents bid on selected bundles only [Berhault *et al.*, 2003]. Researchers have therefore recently advocated sequential single-item auctions (SSI auctions) instead [Lagoudakis *et al.*, 2005]. SSI auctions consist of several rounds, until all tasks have been assigned to agents [Boutilier *et al.*, 1999; Fatima, 2006]. During each round, all agents bid on all unallocated tasks and the auctioneer then assigns one additional task to some agent, which incurs small communication and winner determination costs. However, the team cost of SSI auctions tends to be larger than that of combinatorial auctions since the agents cannot take as many synergies among tasks into account in their bids. To reduce the team cost of SSI auctions, we extend them to assign $k > 1$ additional tasks among the agents during each round. These sequential bundle-bid single-sale (= single-item) auction algorithms (short: SSI auctions with bundles) still consist of several rounds, until all tasks have been assigned to agents. During each round, all agents can now bid on bundles of at most k tasks and the auctioneer then assigns k additional tasks to agents, making SSI auctions with bundles similar to combinatorial auctions. We expect the team cost of SSI auctions with bundles to be smaller than the one of standard SSI auctions since the agents can take more synergies among tasks into account in their bids. In this paper, we develop a general theory for such SSI auctions with bundles, proving that they can be implemented for small bundle sizes k without greatly increasing the communication and winner determination costs of standard SSI auctions.

2 Task-Allocation Problem

We now formalize the task-allocation problems. A task-allocation problem consists of a set of agents $A = \{a_1 \dots a_m\}$ and a set of tasks $T = \{t_1 \dots t_n\}$. Any tuple $(T_{a_1} \dots T_{a_m})$ of pairwise disjoint bundles $T_{a_i} \subseteq T$, for all $i = 1 \dots m$, (= no task is assigned to more than one agent) is a partial solution of the task-allocation problem, with the meaning that agent a_i performs the tasks T_{a_i}. Let $c_a^{agent}(T')$ be the cost needed by agent $a \in A$ to perform the tasks $T' \subseteq T$, called **agent cost**. There can be synergies among tasks, that is, $c_a^{agent}(T') + c_a^{agent}(T'')$ does not necessarily equal $c_a^{agent}(T' \cup T'')$ even if $T' \cap T'' = \emptyset$. The cost of the partial solution, called **team cost**, depends on the team objective. In this paper, we consider two different team objectives. The team cost of the partial solution is $\sum_{a \in A} c_a^{agent}(T_a)$ for the team objective MiniSum and $\max_{a \in A} c_a^{agent}(T_a)$ for the team objective MiniMax. We use c^{team} as a special operator (shorthand) for either the sum and max operator, depending on the team objective, and write $c_{a \in A}^{team} c_a^{agent}(T_a)$ to make our notation independent of the team objective. Any partial solution $(T_{a_1} \dots T_{a_m})$ with $\cup_{a \in A} T_a = T$ (= each task is assigned to exactly one agent) is a complete solution of the task-allocation problem. We want to find a complete solution of the task-allocation problem with a small team cost.

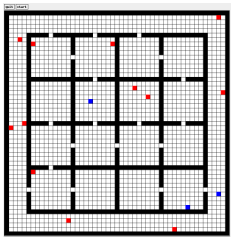

Figure 1: Multi-Agent Routing Problem

3 Multi-Agent Routing

We study multi-agent routing problems as examples of task-allocation problems, as shown in Figure 1. Multi-agent routing problems are task-allocation problems where the tasks are to visit given targets with exactly one agent each. The terrain, the locations of all agents and the locations of all targets are known.[1] The agent cost of an agent to visit a set of given targets corresponds to the smallest travel distance needed to visit the targets from its current location. There can be synergies among tasks, for example, the smallest travel distance needed to visit two close-by targets is typically smaller than the sum of the travel distances needed to visit each target individually. Multi-agent routing is a standard task for robot teams that needs to be solved, for example, as part of de-mining, search-and-rescue and taking rock probes on the moon. In multi-agent routing without capacity constraints, every agent can perform an arbitrary number of tasks. Multi-agent routing problems without capacity constraints are standard test domains for agent coordination with auctions [Dias *et al.*, 2005]. In multi-agent routing with capacity constraints, every agent can perform at most a given number of tasks (= its capacity), for example, can take only a given number of rock probes before its drill bit becomes useless due to wear and tear. Multi-agent routing problems with capacity constraints are novel applications for agent coordination with auctions.

4 SSI Auctions with Bundles

We now develop a blend of combinatorial and sequential single-item auctions (SSI auctions), which we call sequential bundle-bid single-sale (= single-item) auction algorithms (short: SSI auctions with bundles) to solve the task-allocation problems: Initially, all tasks are unassigned. SSI auctions with bundles consist of several rounds, until all tasks have been assigned to agents, which then execute the tasks assigned to them. During each round, all agents bid on all non-empty bundles of at most k unassigned tasks (bidding phase) and the auctioneer then assigns exactly k additional tasks to agents (or all tasks in case the number of tasks is smaller than k), either to the same agent or different agents (winner determination phase). We call k the bundle size.

[1] One can solve multi-agent routing problems in unknown terrain by making assumptions about the unknown terrain, such as the assumption that it is traversable, making it in effect "known" and thus solvable with auctions. One then runs another auction to re-allocate all unvisited targets to agents whenever this assumption turned out to be wrong and needed to get revised.

We now consider any round of the SSI auction with bundles. Assume that agent $a \in A$ has already been assigned the tasks $T_a \subseteq T$ in previous rounds for all $a \in A$. Thus, $U = T \setminus \cup_{a \in A} T_a$ is the set of unassigned tasks. We leave out the "unassigned" in the following for readability since only unassigned tasks are bid on and subsequently assigned. A bid b is a triple of an agent, a bundle and a bid cost (= numerical value of the bid). If b is a bid, then we use b_a to denote the agent, b_t to denote the bundle it bids on and b_c to denote the bid cost. We now explain exactly which bids the agents submit (including how much they bid) and which of these bids win.

Bidding Phase: The set of submitted bids B satisfies: 1) for all $b \in B$, it holds that $b_a \in A$, $b_t \subseteq U$, and $0 < |b_t| \leq k$; and 2) for all $a \in A$ and $T' \subseteq U$ with $0 < |T'| \leq k$, there exists exactly one bid $b \in B$ with $b_a = a$ and $b_t = T'$ (= each agent bids on all non-empty bundles of at most k tasks). For the MiniSum team objective, $b_c = c^{agent}_{b_a}(T_{b_a} \cup b_t) - c^{agent}_{b_a}(T_{b_a})$. In other words, the agent bids the increase in its agent cost for all tasks assigned to it if it is additionally assigned the tasks that it bids on, which is similar to previous work on marginal-cost bidding in ContractNet [Sandholm, 1996]. For the MiniMax team objective, $b_c = c^{agent}_{b_a}(T_{b_a} \cup b_t)$. In other words, the agent bids its agent cost for all tasks assigned to it if it is additionally assigned the tasks that it bids on.

Winner Determination Phase: Each collection $B' \subseteq B$ of bids is potentially winning iff 1) $b_a \neq b'_a$ and $b_t \cap b'_t = \emptyset$ for all $b, b' \in B'$ with $b \neq b'$; and 2) $|\cup_{b \in B'} b_t| = \min(k, |U|)$ (= a potentially winning collection of bids must have been made by different agents, since several bids by the same agent do not express synergies, and cover exactly k tasks or all tasks in case the number of tasks is smaller than k). The term $|U|$ covers the case where the number of tasks is smaller than the bundle size k. We denote the set of all potentially winning collections of bids as $P(B)$. The auctioneer evaluates a potentially winning collection $B' \subseteq P(B)$ of bids according to the value $c^{team}_{b \in B'} b_c$, called **evaluation cost**, for both the MiniSum and MiniMax team objectives. Thus, the winning collection $B_w \in P(B)$ of bids satisfies, for all $B' \in P(B)$, $c^{team}_{b \in B_w} b_c \leq c^{team}_{b \in B'} b_c$ (= the winning collection of bids has the smallest evaluation cost among all potentially winning collections of bids). Any such collection of bids can be chosen. The auctioneer then assigns the additional tasks b_t to agent b_a for all $b \in B_w$. At this point in time, agent $a \in A$ has been assigned the tasks $T_a(B_w) = T_a \cup b_t$ if there exists $b \in B_w$ with $b_a = a$, and $T_a(B_w) = T_a$ otherwise.

We now explain why we expect these bidding and winner determination rules to result in a small team cost. (However, their team cost can be worse than that of standard SSI auctions.)

Theorem 4.1 *For all $B' \in P(B)$, it holds that $c^{team}_{a \in A} c^{agent}_a(T_a(B_w)) \leq c^{team}_{a \in A} c^{agent}_a(T_a(B'))$ for both the MiniSum and MiniMax team objectives.*

Proof: Let X and Y be any two potentially winning collections of bids. We show that $c^{team}_{b \in X} b_c \leq c^{team}_{b \in Y} b_c$ (a) implies $c^{team}_{a \in A} c^{agent}_a(T_a(X)) \leq c^{team}_{a \in A} c^{agent}_a(T_a(Y))$ (b). Consider

the MiniSum team objective. Adding $c^{team}_{a\in A} c^{agent}_a(T_a)$ to both sides of Inequality (a) yields Inequality (b) since by definition $b_c = c^{agent}_a(T_a \cup X_a) - c^{agent}_a(T_a)$ where X_a is the (possibly empty) bundle of tasks bid on by agent a in the collection X of bids. Now consider the MiniMax team objective. By definition, $c^{team}_{a\in A} c^{agent}_a(T_a(X)) = \max_{a\in A} c^{agent}_a(T_a(X)) = \max\{\max_{a\in A:b_a\notin X} c^{agent}_a(T_a), \max_{a\in A:b_a\in X} c^{agent}_a(T_a \cup X_a)\} = \max\{\max_{a\in A} c^{agent}_a(T_a), \max_{a\in A:b_a\in X} c^{agent}_a(T_a \cup X_a)\}$, where the second argument is the evaluation cost of the collection X of bids from Inequality (a). Similarly, $c^{team}_{a\in A} c^{agent}_a(T_a(Y)) = \max\{\max_{a\in A} c^{agent}_a(T_a), \max_{a\in A:b_a\in Y} c^{agent}_a(T_a \cup Y_a)\}$, where the first argument is identical to the first one above and the second one is the evaluation cost of the collection Y of bids from Inequality (a). Thus, Inequality (a) implies $c^{team}_{a\in A} c^{agent}_a(T_a(X)) \leq c^{team}_{a\in A} c^{agent}_a(T_a(Y))$ and thus Inequality (b). ■

In other words, each round of SSI auctions with bundles assigns $\min(k, |U|)$ additional tasks to agents so that the team cost after the assignment is as small as possible. Thus, SSI auctions with bundles perform hill-climbing and the resulting team cost can thus be expected to be small. Unfortunately, the communication and winner determination costs of SSI auctions with bundles (as described so far) are large since the number of bids is a kth-order polynomial in $|U|$, the number of tasks. This is a problem because the communication and winner determination costs are bottlenecks for SSI auctions with bundles, especially since the auctioneer is often centralized. Thus, SSI auctions do not scale to large numbers of tasks even for small bundle sizes. We now show how to modify SSI auctions with bundles so that they continue to assign the same additional tasks to the same agents during each round (modulo tie breaking) but so that, for a given bundle size, every agent needs to submit only a constant number of bids per round and the runtime of winner determination is linear in the number of agents and independent of the number of tasks. This is the main contribution of our paper and perhaps surprising since SSI auctions with bundles are similar to ideal combinatorial auctions, for which winner determination is known to be NP-hard and thus needs to be approximated [Hoos and Boutilier, 2000]. The result is due to the fact that agents bid only on bundles of at most k tasks for SSI auctions with bundles but on all bundles for ideal combinatorial auctions. The idea behind our modification is that the agents do not need to submit bids on all bundles of at most k tasks since some bundles have no chance of winning.

5 Bidding Phase

We now explain exactly which bundles the agents need to bid on. To determine which bids to submit in a round of an SSI auction with bundle size k, an agent constructs k bid trees, one for each $1 \leq k' \leq k$. Each node of the k'th bid tree is labeled with a bundle of k' tasks. Each edge is labeled with a task. The k'th bid tree is recursively constructed, starting with its root. Consider any node in the bid tree. It is labeled with the bundle that has the smallest bid cost among all bundles of k' tasks that do not contain any of the tasks that label the edges from the root to the node in question. Each node at depth $\min(k - k', |U| - k')$ is a leaf. The bid tree is empty if this depth is negative. (The term $|U| - k'$ covers the case where the number of tasks is smaller than the bundle size k.) Otherwise, each edge from the node in question to its k' children is labeled with one of the tasks contained in the bundle. This completes the construction of the bid trees. The agent then submits bids on all bundles that label nodes in its bid trees, calculating the bid costs as before. (The same bundle can label several nodes but, of course, results in only one bid.) Thus, the number of bids per agent is constant for SSI auctions with a given bundle size.

Theorem 5.1 *The winner determination phase assigns the same* $\min(k, |U|)$ *additional tasks (modulo tie breaking) to agents no matter whether the agents bid on all non-empty bundles with at most* k *tasks or submit bids on the bundles from their bid trees only.*

Proof Sketch: Assume that the agents bid on all non-empty bundles with at most k tasks and that the winning collection of bids is W. We show that we can achieve the same team cost if the agents bid on the bundles from their bid trees only. Obviously, the team cost cannot be smaller if the agents bid on the bundles from their bid trees only, a subset of all non-empty bundles with at most k tasks. We thus need to show that it cannot be larger either. To this end, consider an arbitrary winning bid $b \in W$ whose bundle b_t is not in any bid tree of agent b_a. We now show that we can replace this bid b with a bid b' so that the bundle of bid b' is in a bid tree, the collection of bids remains potentially winning, and the bid cost of bid b' is no larger than the bid cost of bid b, which implies that the substitution cannot increase the team cost since the c^{team} operator is monotonically increasing in its arguments. Similar substitutions can then be made for all bids in W, if necessary, until all bids are on bundles from the bid trees of the agents, which proves the theorem. To prove the property, consider the tasks $F = \cup_{b''\in W\setminus\{b\}} b''_t$ that are part of the winning bids except for bid b. The following properties will hold: b' is in a bid tree, $b'_a = b_a$ (= both bids are from the same agent), $|b'_t| = |b_t|$ (= both bids are on the same number of tasks), $b'_t \cap F = \emptyset$ (= no task is assigned to more than one agent) and $b'_c \leq b_c$ (= the bid cost of the new bid is no larger than the bid cost of the previous bid). We achieve these properties by choosing the bid b' with the smallest bid cost among all bids from the $|b_t|$th bid tree of agent b_a whose bundles do not contain tasks in F, as follows: We start at the root of the $|b_t|$th bid tree of agent b_a as current node. If the bundle of the current node does not contain any tasks in F, then we pick the bid on this bundle. Otherwise, choose a task contained in both this bundle and F (ties can be broken arbitrarily) and follow the edge labeled with this task from the current node to one of its children, and repeat the procedure. We are guaranteed to reach a node whose bundle does not contain any tasks in F because F contains $\min(k - |b_t|, |U| - |b_t|)$ tasks and the $|b_t|$th tree has depth $\min(k - |b_t|, |U| - |b_t|)$. The bundle of this node is the bundle with the smallest bid cost that does not contain any tasks in F and has the stated properties, per construction of the bid trees. ■

As example, consider the multi-agent routing problem without capacity constraints shown in Figure 2 for the MiniMax team objective. The agents and targets are located on the real line. (Epsilon is a small positive tie-breaking constant.)

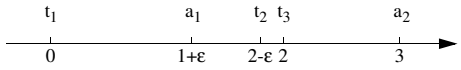

Figure 2: Multi-Agent Routing Problem without Capacities

	{t1}	{t2}	{t3}	{t1,t2}	{t1,t3}	{t2,t3}	{t1,t2,t3}
a_1	$1+\epsilon$	$1-2\epsilon$	$1-\epsilon$	$3-3\epsilon$	$3-\epsilon$	$1-\epsilon$	$3-\epsilon$
a_2	3	$1+\epsilon$	1	3	3	$1+\epsilon$	3

Table 1: Bid Costs

Table 1 shows the bid costs of both agents on all bundles. Figure 3 shows the bid trees of agent a_1 during the first round of an SSI auction with bundle size three. For example, the bundle of two targets with the lowest bid for agent a_1 and thus the root of its bid tree two is $\{t2, t3\}$. The bundle with the lowest bid for agent a_1 among all bundles with two targets different from target t_2 and thus a child of the root of its bid tree two is $\{t_1, t_3\}$. Similarly, the bundle with the lowest bid for agent a_1 among all bundles with two targets different from target t_3 and thus the second child of the root of its bid tree two is $\{t_1, t_2\}$. Overall, the agent bids on all non-empty bundles of at most three targets. Figure 4 shows the bid trees of agent a_1 during the first round of an SSI auction with bundle size two. Consequently, the agent bids on the bundles $\{t_2, t_3\}$, $\{t_2\}$, $\{t_3\}$, which are only half of all non-empty bundles of at most two targets. For example, the agent does not bid on all single tasks but only on the ones with the lowest and second-lowest bid cost. Table 2 demonstrates the substantial reduction in the number of bids of one agent for different bundle sizes in case there are 20 tasks.

6 Winner Determination Phase

From now on, we use B to refer to the bids of the agents on the bundles from their bid trees. Although the number of bids is small, it is not clear whether the auctioneer can determine the winning bids with a small runtime. In the following, we construct a winner-determination rule that is linear in the number of bids and thus linear in the number of agents and independent of the number of tasks for SSI auctions with a fixed bundle size. It is curious that the construction and justification of the winner-determination rule are rather more complicated than those of the bundle construction.

Theorem 6.1 *The auctioneer can determine the winning bids with a runtime that is linear in the number of bids.*

Proof Sketch: The auctioneer needs to assign $\min(k, |U|)$ additional tasks to agents. It first finds all different non-decreasing sequences of at most $|A|$ positive integers that sum to $\min(k, |U|)$. For example, there are three such sequences for $\min(k, |U|) = 3$ and $|A| \geq 3$, namely the sequences (1,1,1), (1,2) and (3), but only two such sequences (the latter two) for $|A| = 2$. Each sequence represents a possible set of winning bid sizes. The number of such sequences must be $O(1)$ because $k = O(1)$. A portfolio for such a sequence $(s(1) \dots s(l))$ is a collection of bids $(b(1) \dots b(l))$ with $b(i) \in B$ and $|b(i)_t| = s(i)$ for all $1 \leq i \leq l$. The portfolio is consistent iff $b(i)_a \neq b(j)_a$ and $b(i)_t \cap b(j)_t = \emptyset$ for all $1 \leq i < j \leq l$, that is, no agent gets more than one bid, and the bids form a partition

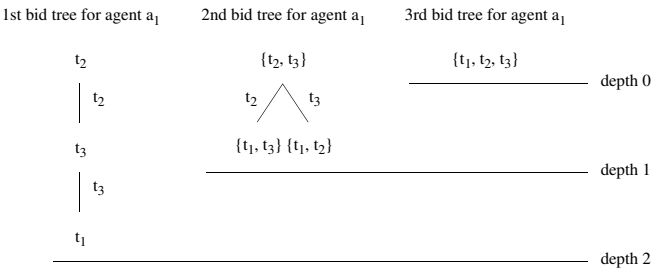

Figure 3: Bid Trees for SSI Auctions with Bundle Size Three

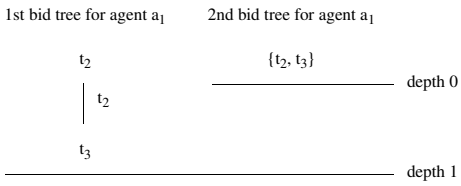

Figure 4: Bid Trees for SSI Auctions with Bundle Size Two

Number of Targets $\lvert U\rvert$	Bundle Size k	Previous Number of Bids	New Number of Bids	
20	20	1048575	1048575	(100.00%)
20	19	1048574	1047074	(99.86%)
20	18	1048554	1030781	(98.30%)
20	17	1048364	944747	(90.12%)
20	16	1047224	809850	(77.33%)
20	15	1042379	561204	(53.84%)
20	14	1026875	343120	(33.41%)
20	13	988115	184060	(18.63%)
20	12	910595	85684	(9.41%)
20	11	784625	34510	(4.40%)
20	10	616665	12130	(1.97%)
20	9	431909	3764	(0.87%)
20	8	263949	1048	(0.40%)
20	7	137979	315	(0.23%)
20	6	60459	105	(0.17%)
20	5	21699	39	(0.18%)
20	4	6195	16	(0.26%)
20	3	1350	7	(0.52%)
20	2	210	3	(1.43%)
20	1	20	1	(5.00%)

Table 2: Reduction in the Number of Bids

of the items. Note that a consistent portfolio is a potentially winning collection of bids. To find the winning collection of bids, the auctioneer constructs a search tree for each sequence. Each node of the search tree for a given sequence $(s(1) \dots s(l))$ is labeled with a portfolio for the sequence. Each edge is labeled with a constraint, which can be of two kinds: 1) "$t \in b(j)_t$" for a given $t \in U$ and a given $1 \leq j \leq l$ and 2) "$b(j)_a = b_a$ and $b(j)_t = b_t$" for a given $b \in B$ with $|b_t| = s(j)$ and a given $1 \leq j \leq l$. The search tree for a given sequence $(s(1) \dots s(l))$ is recursively constructed, starting with its root. Consider any node in the search tree. It is labeled with a portfolio $(b(1) \dots b(l))$ with the smallest evaluation cost among all portfolios that satisfy the constraints that label the edges from the root to the node in question, as follows: If the constraint is "$t \in b(j)_t$" then the portfolio needs to satisfy $t \in b(i)_t$ for $i = j$ and, for all $1 \leq i \leq l$ with $i \neq j$, $t \notin b(i)_t$. If the constraint is "$b(j)_a = b_a$ and $b(j)_t = b_t$," then the portfolio needs to satisfy $b(i)_a = b_a$ and $b(i)_t = b_t$ for $i = j$ and, for all $1 \leq i \leq l$ with $i \neq j$, $b(i)_a \neq b_a$

and $b(i)_t \cap b_t = \emptyset$. The node is deleted from the search tree if no portfolio satisfies the constraints. Each node with a consistent portfolio is a leaf. Otherwise, the auctioneer chooses any $1 \leq i \leq l$ with $b(i)_a = b(j)_a$ or $b(i)_t \cap b(j)_t \neq \emptyset$ for some $1 \leq j \leq l$ with $i \neq j$ to generate the constraints that label the edges from the node in question to its children: 1) "$t \in b(j)_t$," one constraint for each $1 \leq j \leq l$ with $j \neq i$ and each $t \in b(i)_t$; and 2) "$b(j)_a = b_a$ and $b(j)_t = b_t$," one constraint for each $1 \leq j \leq l$ and each $b \in B$ with $b(i)_a = b_a$ and $|b_t| = s(j)$. The justification for these constraints is that at least one of them must be satisfied: Either some bid by agent $b(i)_a$ is part of the portfolio or there is a bid part of the portfolio whose bundle includes some task $t \in b(i)_t$. If this were not the case, then one could substitute bid $b(i)$ for the bid at the ith position, resulting in a potentially winning bid without increasing the evaluation cost. This completes the construction of the search trees. The winning collection of bids then is any portfolio $(b(1) \dots b(l))$ with the smallest evaluation cost among all consistent portfolios that label nodes in the search trees, per construction of the search trees. There are $O(1)$ sequences and thus $O(1)$ search trees. The depth of the search tree for a given sequence $(s(1) \dots s(l))$ is at most $k + l$ and thus $O(1)$ since each constraint imposes an additional restriction on a portfolio and no portfolio can satisfy more than $k + l$ constraints. The number of children of every node in each search tree is $O(1)$. Thus, each search tree has $O(1)^{O(1)} = O(1)$ nodes. The portfolio of each node can be determined in time $O(|B|)$ by finding the bid with the smallest bid cost that satisfies the constraints independently for each position in the portfolio. Consequently, the runtime of winner determination is $O(1) \times O(1) \times O(|B|) = O(|B|)$. ■

Consider again the multi-agent routing problem without capacity constraints shown in Figure 2. Figure 5 shows the corresponding search trees for the first round of an SSI auction with bundle size three. The underlined bids were used to generate the constraints. Consistent portfolios are annotated with their evaluation cost.

7 Refinement

The auctioneer can be cautious and assign only one additional task of the $\min(k, |U|)$ tasks per round, namely one with the smallest agent cost. In this case, the bidding phase changes as follows: Every agent also needs to bid on all tasks that are part of larger bundles that it bids on since it bids its agent cost on single tasks. This increases its number of bids by a constant that is usually small. As example, consider again the multi-agent routing problem without capacity constraints shown in Figure 2. Figures 3 and 4 show that agent a_1 already bids on all tasks that are part of larger bundles that it bids on during the first round of SSI auctions with bundle sizes two and three and thus does not need to submit any additional bids during the first round. The winner determination phase changes as follows: The auctioneer first determines the winning collection of bids W, as before. Then, it determines the bid $b \in W$ and task $t \in b_t$ with $c^{agent}_{b_a}(t) \leq c^{agent}_{b'_a}(t')$ for all $b' \in W$ and $t' \in b'_t$ (= the task with the smallest agent cost or, equivalently, bid cost among all winning tasks). It does this by comparing the bids of agent b'_a on task t' for all $b' \in W$ and $t' \in b'_t$ and choosing the one with the smallest bid cost (ties can be broken arbitrarily). It then assigns only this additional task t to the bidding agent b_a.

As example, consider again the multi-agent routing problem without capacity constraints shown in Figure 2 for the MiniMax team objective.[2] For this particular example, SSI auctions with larger bundle sizes indeed result in a smaller team cost, since they are less myopic, although this is not guaranteed in general. 1) For SSI auctions with bundle size one (where the auctioneer automatically assigns only one additional target), targets are assigned to agents in the following order. Round 1: agent a_1 is assigned target t_2; Round 2: agent a_1 is assigned target t_3; and Round 3: agent a_1 is assigned target t_1. We write this as: $a_1 \leftarrow t_2; a_1 \leftarrow t_3; a_1 \leftarrow t_1$. The largest travel distance of any agent (= makespan) is $3 - \epsilon$. 2) For SSI auctions with bundle size two where the auctioneer is cautious (that is, assigns only one additional target) targets are assigned to agents in the following order, where parentheses enclose targets that are not assigned: $a_1 \leftarrow t_2(t_3); (a_1 \leftarrow t_1), a_2 \leftarrow t_3; a_1 \leftarrow t_1$. The makespan is $3 - 3\epsilon$, which is slightly smaller than the makespan of SSI auctions with bundle size one. The makespan of SSI auctions with bundle size two where the auctioneer assigns two additional targets (if possible) is $3 - \epsilon$, which is slightly larger than the makespan of SSI auctions with bundle size two where the auctioneer assigns only one additional target and the reason why the auctioneer assigns only one additional target in our experiments. 3) For SSI auctions with bundle size three where the auctioneer assigns only one additional target, targets are assigned to agents in the following order: $(a_1 \leftarrow t_1), a_2 \leftarrow (t_2)t_3; a_1 \leftarrow t_1, (a_2 \leftarrow t_2); a_2 \leftarrow t_2$. The makespan is $1 + \epsilon$, which is the minimal team cost since the bundle size equals the number of targets and smaller than the makespan of SSI auctions with bundle size two.

8 Experiments

We now evaluate the benefit of SSI auctions with bundles for multi-agent routing problems on known eight-neighbor planar grids of size 51×51 with square cells that are either blocked or unblocked. The grids resembled office environments, as shown in Figure 1. We averaged all results over 25 instances of the same office environment with randomly closed doors. We solved multi-agent routing problems with capacity constraints using SSI auctions with bundles where the agents stopped bidding once the number of tasks assigned to them reached their capacity. We set the capacities of all agents to the ratio of the number of targets and agents. An agent needs to determine the bid costs of many bids to determine which bids to submit. To determine each bid cost for multi-agent routing problems, it needs to solve a version of a TSP problem (where it does not need to return to its initial location) when calculating the smallest travel distance needed to visit the targets from its current location, an NP-hard problem. These calculations therefore need to be approximated to run fast. For this purpose, we used a combination of the two-opt and cheapest-insertion heuristics in our experiments. Table 3 tabulates the team cost for SSI auctions with bundle sizes one, two and three as well as approximations of

[2] Similar results holds for the MiniSum team objective. Only the results for SSI auctions with bundle size two where the auctioneer assigns only one additional target is slightly different.

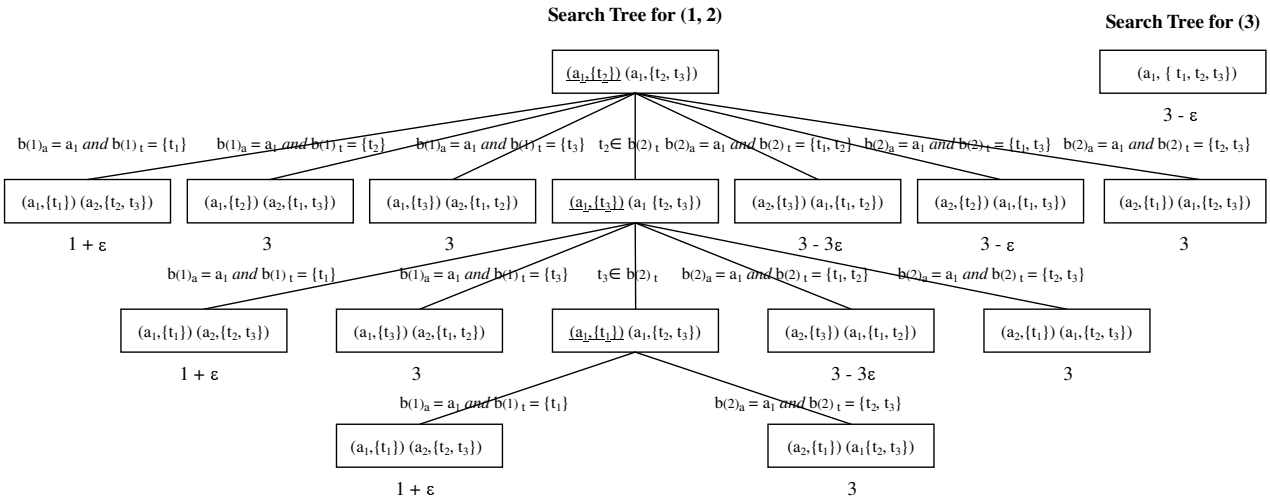

Figure 5: Search Trees

			MiniSum Team Objective				MiniMax Team Objective			
Capacities	Agents	Targets	Minimal	Bundle Size 1	Bundle Size 2	Bundle Size 3	Minimal	Bundle Size 1	Bundle Size 2	Bundle Size 3
2	2	4	120.8	125.7	121.7 [122.5]	120.8 [121.6]	76.3	83.9	81.1 [81.6]	77.3 [81.4]
2	4	8	176.6	197.1	189.4 [189.7]	187.5 [189.3]	60.3	93.1	87.6 [89.6]	78.2 [94.7]
2	6	12	212.9	243.6	235.8 [237.8]	228.2 [230.1]	54.2	88.7	79.6 [80.9]	74.6 [75.3]
2	8	16	251.7	287.5	283.2 [290.0]	274.9 [281.5]	(51.2)	87.5	86.3 [85.4]	77.2 [85.8]
2	10	20	285.7	344.8	331.4 [331.7]	328.5 [329.2]	(52.6)	97.1	90.3 [94.7]	83.9 [94.3]
3	2	6	166.2	176.1	172.5 [174.6]	169.3 [172.0]	96.2	113.2	106.3 [110.7]	99.0 [99.1]
3	4	12	229.1	265.2	258.6 [261.8]	250.2 [255.0]	(72.2)	107.7	105.4 [106.9]	102.9 [105.1]
3	6	18	(265.9)	323.0	310.1 [314.4]	297.8 [310.8]	(65.9)	110.8	104.3 [109.6]	101.7 [109.4]
3	8	24	(297.7)	371.3	362.7 [360.4]	356.7 [359.9]	(69.4)	107.5	108.2 [105.8]	99.7 [102.1]
3	10	30	(340.4)	421.2	411.9 [415.3]	402.9 [407.3]	(82.9)	113.5	113.3 [110.9]	97.7 [105.1]
4	2	8	187.4	201.6	207.6 [204.9]	203.0 [202.0]	106.0	132.6	127.5 [128.5]	118.7 [131.9]
4	4	16	(264.4)	303.5	302.9 [305.9]	297.1 [306.5]	(89.2)	132.9	125.5 [123.5]	116.5 [126.7]
4	6	24	(296.9)	376.4	362.8 [364.4]	359.3 [356.1]	(100.0)	124.5	125.9 [122.0]	119.6 [118.3]
4	8	32	(357.7)	437.2	425.9 [431.7]	415.6 [421.3]	(114.9)	125.7	122.6 [123.4]	113.2 [130.8]
4	10	40	(413.3)	488.4	504.9 [512.0]	479.2 [499.9]	(147.3)	129.5	124.2 [126.3]	116.1 [126.2]

Table 3: Comparison of Bundle Sizes: Experimental Results

the minimal team cost calculated with a mixed integer program (MIP) for both the MiniSum and MiniMax team objectives. (Team costs for a non-cautious auctioneer are reported in square brackets and tend to be larger than the team costs for a cautious auctioneer, who assigns only one additional target.) The MIPs were arc-based Vehicle Routing Problem formulations with routing constraints and Miller-Tucker-Zemlin (MTZ) sub-tour elimination constraints, which are further augmented by introducing additional first and second order lifted versions of MTZ constraints and arrival-time constraints. They were solved with CPLEX 9.0, a commercial MIP solver, at default settings on a Dell Precision 670 computer with a 3.2 GHz Intel Xeon Processor and 2 GB RAM running Red Hat Linux 9.0. For MIPs that could not be solved within one hour of runtime, a binary search was carried out with CPLEX as subroutine, which sometimes allowed us to determine additional minimal team costs within one additional hour of runtime. A "minimal team cost" is enclosed in parentheses in the table if the average contains at least one non-minimal team cost and it is thus only an upper bound on the minimal team cost. The runtime to calculate this gold standard quickly increased with the problem size. For example, we were not able to determine the minimal team cost for any of the 25 multi-agent routing problems with 10 agents and 40 targets for the MiniMax team objective within the runtime limit. On the other hand, all SSI auctions terminated within two seconds and their runtime increased only very slowly with the problem size. For the MiniSum team objective, SSI auctions with bundle size three reduced the team cost by 4-5 percent on average compared to standard SSI auctions (with bundle size one). For the MiniMax team objective, SSI auctions with bundle size three reduced the team cost by 10-11 percent on average compared to standard SSI auctions. Nonparametric statistical comparisons between SSI auctions with different bundle sizes support the hypothesis that larger bundle sizes reduce the average team cost across cases with different numbers of agents and targets with confidence 0.9963 (1 versus 2, MiniMax); 0.9995 (1 versus 2, MiniSum; 1 versus 3, MiniSum); and 0.99997 (1 versus 3, MiniMax; 2 versus 3, both team objectives). These results justify our generalization of standard SSI auctions to SSI auctions with bundles.

We also tested SSI auctions with bundles against a different heuristic, as suggested by the reviewers. The experimental setup was identical to that of the first experiment, except that the conference schedule limited us to 12 instead of

Capacities	Agents	Targets	MiniSum Team Objective			MiniMax Team Objective		
			CPLEX Heuristics	Bundle Size 2	Bundle Size 3	CPLEX Heuristics	Bundle Size 2	Bundle Size 3
2	2	4	120.2	122.0	120.2	74.7	76.7	74.7
2	4	8	185.5	202.5	196.5	66.0	94.8	80.1
2	6	12	205.0	221.0	231.5	52.3	79.9	75.9
2	8	16	245.3	283.0	271.9	(50.8)	84.4	78.4
2	10	20	295.1	342.7	337.3	(60.6)	94.7	89.6
3	2	6	184.1	192.9	189.9	103.5	115.1	108.5
3	4	12	238.6	296.6	262.6	(76.3)	109.2	112.9
3	6	18	(260.7)	300.1	295.7	(65.0)	104.6	105.6
3	8	24	(293.7)	354.1	345.0	(73.7)	106.9	96.1
3	10	30	(368.8)	436.9	431.5	(89.4)	128.7	109.1
4	2	8	185.4	202.9	197.8	106.5	123.9	119.7
4	4	16	(273.9)	313.6	304.7	(91.7)	119.2	121.8
4	6	24	(304.8)	373.7	377.6	(95.5)	130.9	127.3
4	8	32	(352.5)	415.9	411.9	(123.7)	114.3	111.0
4	10	40	(423.4)	492.4	469.0	(186.8)	123.9	120.8

Table 4: Comparison of Heuristics: Experimental Results

25 instances of the office environment. We set the CPLEX parameters to strongly emphasize searching and improving feasible solutions, and further tuned the parameters to improve the performance of the CPLEX local search, RINS, and other heuristics. The time limit was 5 minutes for cases with fewer than 16 targets and 10 minutes otherwise. The results are given in Table 4, which follows the same format as Table 3. The CPLEX heuristics result in significantly smaller team costs than SSI auctions with bundles, except for cases with larger numbers of agents and targets in conjunction with the MiniMax team objective, where SSI auctions with bundles are superior. On the other hand, the CPLEX heuristics ran much more slowly than SSI auctions with bundles. The CPLEX heuristics exhausted the allotted time for more than one third of the instances, while the SSI auctions with bundles never ran for more than 0.1 seconds in case of bundle size two and 1.9 seconds in case of bundle size three. It is future work to compare SSI auctions with bundles against specialized vehicle routing problem heuristics.

9 Conclusions

In this paper, we studied sequential single-item auctions for the distributed allocation of tasks to cooperating agents. Rollouts [Zheng *et al.*, 2006] reduce the team cost of sequential single-item auctions but increase their runtime substantially. We therefore studied an alternative. We generalized sequential single-item auctions to assign more than one additional task per round and developed a general theory of such sequential bundle-bid single-sale (= single-item) auctions (short: sequential single-item auctions with bundles). Our research generalizes earlier (somewhat ad-hoc) work that is equivalent to SSI auctions with bundle sizes one and two [Lagoudakis *et al.*, 2005; Zheng *et al.*, 2006]. We showed, for sequential single-item auctions with a given bundle size, that every agent needs to submit only a constant number of bids per round and the runtime of winner determination is linear in the number of agents. The communication and winner determination costs do not depend on the number of tasks and thus scale to a large number of tasks for small bundle sizes.

Acknowledgments

This research was partly supported by NSF awards under contracts ITR/AP0113881, IIS-0098807 and IIS-0350584 as well as a seed funding from NASA's Jet Propulsion Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.

References

[Berhault *et al.*, 2003] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2003.

[Boutilier *et al.*, 1999] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 527–523, 1999.

[Dias *et al.*, 2005] M. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Robotics Institute, Carnegie Mellon University, Pittsburgh (Pennsylvania), 2005.

[Fatima, 2006] S. Fatima. Sequential versus simultaneous auctions: A case study. In *Proceedings of the International Conference on Electronic Commerce*, 2006.

[Hoos and Boutilier, 2000] H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the National Conference on Artificial Intelligence*, pages 22–29, 2000.

[Lagoudakis *et al.*, 2005] M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, S. Koenig, A. Kleywegt, C. Tovey, A. Meyerson, and S. Jain. Auction-based multi-robot routing. In *Proceedings of the International Conference on Robotics: Science and Systems*, 2005.

[Sandholm, 1996] T. Sandholm. *Negotiation among Self-Interested Computationally Limited Agents*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst (Massachusetts), 1996.

[Zheng *et al.*, 2006] X. Zheng, S. Koenig, and C. Tovey. Improving sequential single-item auctions. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2006.