Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

Sequential Incremental-Value Auctions*

Xiaoming Zheng and **Sven Koenig**
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781
{xiaominz,skoenig}@usc.edu

Abstract

We study the distributed allocation of tasks to cooperating robots in real time, where each task has to be assigned to exactly one robot so that the sum of the latencies of all tasks is as small as possible. We propose a new auction-like algorithm, called Sequential Incremental-Value (SIV) auction, which assigns tasks to robots in multiple rounds. The idea behind SIV auctions is to assign as many tasks per round to robots as possible as long as their individual costs for performing these tasks are at most a given bound, which increases exponentially from round to round. Our theoretical results show that the team costs of SIV auctions are at most a constant factor larger than minimal.

Introduction

We study the distributed allocation of tasks to cooperating robots in real time, where each task has to be assigned to exactly one robot so that the team cost is as small as possible. We do this in the context of multi-robot routing, where the robots have to visit targets in the plane so that each target is visited by some robot (Dias et al. 2005). The terrain, the locations of all robots and the locations of all targets are known. The team cost is the sum of the latencies of all targets, where the latency of a target is the time when it gets visited. Auction-like algorithms (short: auctions) promise to solve multi-robot routing problems with small communication and computation costs since the robots compress information into a small number of bids, which they compute in parallel and then exchange (Dias et al. 2005; Lagoudakis et al. 2005; Koenig et al. 2008). Thus, auctions promise to be able to control robots in real-time, which is important to prevent robots from being idle each time they allocate targets among themselves. Robotics researchers have recently studied the use of Sequential Single-Item (SSI) auctions for multi-robot routing (Tovey et al. 2005). SSI auctions proceed in multiple rounds, until all targets are assigned to robots. During each round, SSI auctions assign exactly one additional (previously unassigned) target to some robot so that the team cost increases least (= hill-climbing principle). In this paper, we propose a new type of auction, called Sequential Incremental-Value (SIV) auction, that makes use of the hill-climbing principle in a different way to assign targets to robots in multiple rounds. The idea behind SIV auctions is to assign as many tasks per round to robots as possible as long as their individual costs for performing these tasks are at most a given bound, which increases exponentially from round to round. Our theoretical results show that the team costs of SIV auctions are at most a constant factor larger than minimal, which is better than the guarantee on the team costs provided by SSI auctions.

Multi-Robot Routing

We now formalize multi-robot routing problems. A multi-robot routing problem consists of a set of robots $A = \{a_1, \ldots, a_n\}$ and a set of targets $T = \{t_1, \ldots, t_m\}$. The initial locations of all robots can be different. Any tuple $(T_{a_1}, \ldots, T_{a_n})$ of pairwise disjoint bundles of targets $T_{a_i} \subseteq T$ is a partial assignment of the multi-robot routing problem. We define the **path cost** $c_a^{path}(T_a)$ to be the smallest possible travel distance of robot a for visiting all targets T_a from its initial location. We assume that all distances satisfy the triangle inequality. We define the **robot cost** $c_a^{robot}(T_a)$ to be the smallest possible total latency (= sum of the latencies) of all targets T_a on any path that visits all targets T_a from the initial location of robot a, where the **latency** of a target is the time when robot a visits it (measured in the travel distance of robot a, which assumes that the robot moves at unit speed). Finally, we define the **team cost** of a partial assignment $(T_{a_1}, \ldots, T_{a_n})$ to be the total latency $\sum_{a \in A} c_a^{robot}(T_a)$. Any partial assignment with $\cup_{a \in A} T_a = T$ (that is, each target is visited by exactly one robot) is a complete assignment of the multi-robot routing problem. Our objective is to determine a complete assignment of a given multi-robot routing problem and the order in which each robot should visit the targets assigned to it so that the resulting team cost is small. A variety of applications require a small total latency. An example is finding all victims in search and rescue missions. Most existing work is on minimizing the total latency for single robots, called

*This material is based upon research supported by, or in part by, NSF under contract/grant number 0413196, ARL/ARO under contract/grant number W911NF-08-1-0468 and ONR in form of a MURI under contract/grant number N00014-09-1-1031. The presented views and conclusions are those of the authors alone. We thank Craig Tovey for providing his operations research expertise to us and proofreading this paper very carefully.

```
1  function SSI-Auction (\overline{T}, A)
2  inputs: \overline{T}: the set of targets T
3          A: the set of robots A
4  outputs: {T_a}_{a∈A}: a complete assignment
5  for each robot a ∈ A do
6      T_a ← ∅;
7  while (\overline{T} ≠ ∅) do
8      /* Annunciation Stage */
9      the auctioneer announces \overline{T} to each robot a ∈ A;
10     /* Bidding Stage */
11     for each robot a ∈ A do
12         for each target t ∈ \overline{T} do
13             b_a^t ← c_a^{robot}(T_a ∪ {t}) − c_a^{robot}(T_a);
14             robot a submits b_a^t to the auctioneer;
15     /* Winner-Determination Stage */
16     (a, t) ← arg min_{(a∈A, t∈\overline{T})} b_a^t;
17     T_a ← T_a ∪ {t};
18     \overline{T} ← \overline{T} \ {t};
```

Figure 1: Sequential Single-Item Auctions

```
1  function SIV-Auction (\overline{T}, A, b)
2  inputs: \overline{T}: the set of targets T
3          A: the set of robots A
4          b: a constant in (1, 2)
5  outputs: {T_a}_{a∈A}: a complete assignment
6  j ← 0;
7  for each robot a ∈ A do
8      T_a ← ∅;
9  while (\overline{T} ≠ ∅) do
10     j ← j + 1;
11     B ← b^{j+1};
12     A' ← A;
13     while (A' ≠ ∅) do
14         /* Annunciation Stage */
15         the auctioneer announces \overline{T} and B to each robot a ∈ A';
16         /* Bidding Stage */
17         for each robot a ∈ A' do
18             T'_a ← arg max_{T'⊆\overline{T}: c_a^{path}(T')≤B} |T'|;
19             robot a submits T'_a to the auctioneer;
20         /* Winner-Determination Stage */
21         a ← arg max_{a∈A'} |T'_a|;
22         A' ← A' \ {a};
23         T_a ← T_a ∪ T'_a;
24         \overline{T} ← \overline{T} \ T'_a;
```

Figure 2: Sequential Incremental-Value Auctions

the traveling repairman problem (Blum et al. 1994). The first constant factor approximation algorithm (Blum et al. 1994) was later improved (Goemans and Kleinberg 1996). There is much less work on minimizing the total latency for multiple robots, called the k-traveling repairman problem. A constant factor approximation algorithm for the special case where the initial locations of all robots are identical (Fakcharoenphol, Harrelson, and Rao 2003) was later extended to the case where each target has a given repair-time (Jothi and Raghavachari 2007). Our SIV auctions generalize the idea behind the former algorithm to the case where the initial locations of all robots can be different.

Sequential Single-Item (SSI) Auctions

Sequential Single-Item (SSI) auctions (Tovey et al. 2005) assign targets to robots in multiple rounds (which explains the term "sequential") and, during each round, assign exactly one additional (previously unassigned) target to some robot (which explains the term "single-item") so that the team cost increases least, see Figure 1. All targets are initially unassigned (Lines 5-6). The auctioneer starts a new round as long as there are still unassigned targets (Line 7). Each round consists of three stages (Line 8-18): First, the auctioneer announces the unassigned targets to each robot in the annunciation stage (Line 9). Second, each robot bids on each unassigned target in the bidding stage the increase in its robot cost in case it has to visit the target it bids on in addition to all targets already assigned to it in previous rounds (Line 13), which is similar to previous work on marginal-cost bidding in ContractNet (Sandholm 1996). The robot can determine its bids in parallel with the other robots. In the process, it needs to calculate the total latencies of given sets of targets, which is called the traveling repairman problem and is NP-hard (Blum et al. 1994). Thus, the robot needs to approximate these calculations even though it can determine its bids in parallel with the other robots, which is often done with the cheapest-insertion heuristic (Lagoudakis et al. 2005). Third, the auctioneer chooses a bid with the smallest bid cost as the winning bid and assigns the winning target to the winning robot in the winner-determination stage (Line 16-17), which terminates the round. Ties can be broken in an arbitrary way. The following theorem provides the best known bounds on the team costs of SSI auctions.

Theorem 1 ((Lagoudakis et al. 2005)) *The team costs of SSI auctions can be at least a factor of $\Omega(|T|^{1/3})$ larger than minimal, even if each robot calculates its robot costs exactly. They are at most a factor of $O(|T|^2)$ larger than minimal, whether each robot calculates its robot costs exactly or uses the cheapest-insertion heuristic to determine them approximately in polynomial time.*

Sequential Incremental-Value (SIV) Auctions

Sequential Incremental-Value (SIV) auctions assign targets to robots in multiple rounds (which explains the term "sequential") and, during each round, assign the largest number of additional (previously unassigned) targets to each robot with the constraint that the path cost of the set of targets assigned to a robot in this round is at most a given bound, which increases exponentially from round to round (which explains the term "incremental-value"), see Figure 2. All targets are initially unassigned (Lines 7-8). The auctioneer starts a new round as long as there are still unassigned targets (Line 9). In the beginning of each **round**, the auctioneer multiplies the bound with a given constant b (Line 10-11). Different from SSI auctions, each round of SIV auctions consists of $|A|$ iterations (Line 14-24). Each **iteration** consists of three stages and assigns a number of unassigned targets to some robot as follows: First, the auctioneer announces the unassigned targets and the bound to each eligible robot in the annunciation stage. A robot is **eligible** in a round iff it has not been a winning robot in that round. Second, each eligible robot bids as many unassigned targets as possible in the bidding stage with the constraint that the path cost of the set of these targets is at most the given bound (Line 18-19). Note that the bids consist of sets of targets without bid costs, that robots ignore the targets already assigned to them when determining their bids, and that eligible robots can always submit a bid (which could be the empty set). Third, the auctioneer chooses a bid with the largest number of targets as the winning bid and assigns the winning targets to the winning robot in the winner-determination stage (Line 21-23), which makes the robot ineligible in the current round and terminates the iteration. The last robot becoming ineligible terminates the current round. Ties can be broken in an arbitrary way.

Analysis of SIV Auctions

Each eligible robot a bids as many unassigned targets as possible in the bidding stage of an SIV auction with the constraint that the path cost $c_a^{path}(T')$ of the set of these targets T' is at most the given bound B. The problem of calculating its bids is NP-hard (which can be shown by reducing Hamiltonian path problems to it). Thus, the robot needs to approximate the calculations of its bids even though it can determine its bids in parallel with the other robots. We assume that the robot approximates the calculations of its bids by changing Line 18 of SIV auctions to

```
18    T'_a ← Bid(T̄, a, B);
```

The function Bid$(\overline{T}, a, B)$ determines, for $k = 0, \ldots, |\overline{T}|$, a rooted k-MST (whose root is the initial location of robot a) for the unassigned targets $\overline{T}$ and returns the targets T' in the rooted k-MST with the largest k with the constraint that the path cost $c_a^{path}(T')$ is at most the given bound B.[1] Determining rooted k-MSTs and determining the path costs of given sets of targets are both NP-hard (Ravi et al. 1994). We therefore assume that the function Bid$(\overline{T}, a, B)$ uses, for $k = 0, \ldots, |\overline{T}|$, an $(1/\alpha)$**-approximation algorithm** for determining a rooted k-tree (whose root is the initial location of robot a) for the unassigned targets $\overline{T}$ and returns the targets T' in the rooted k-tree with the largest k with the constraint that the travel distance for circumnavigating the rooted k-tree (which is twice the cost of the rooted k-tree) is at most the given bound B. The robot visits the targets assigned to it at the end of the SIV auction by moving with minimal travel distance from each target to the next. It visits the targets it was assigned in earlier rounds before targets it was assigned in later rounds and the targets it was assigned in the same round in the order given by circumnavigating the corresponding tree. We now analyze the team costs of SIV auctions using the following notation:

- $\{T_a^*\}_{a \in A}$: any complete assignment of the multi-robot routing problem and the order in which each robot should visit the targets assigned to it so that the resulting team cost is minimal (short: the optimal assignment) - if there is more than one, choose one arbitrarily;
- $c^* = \sum_{a \in A} c_a^{robot}(T_a^*)$: the team cost of the optimal assignment (short: minimal team cost);
- n_j^*: the number of targets whose latencies are larger than $0.5\,\alpha b^{j+1}$ in the optimal assignment $\{T_a^*\}_{a \in A}$;
- n_j: the number of unassigned targets in the beginning of the jth round of an SIV auction; and
- T_j: the set of unassigned targets in the beginning of the jth round of the SIV auction whose latencies are at most $0.5\,\alpha b^{j+1}$ in the optimal assignment.

[1]The cost of a tree is the sum of the costs of its edges. An unrooted k-tree is a tree that contains exactly k targets, while a rooted k-tree is a tree that contains k targets plus the root, which is the location of the robot. A rooted or unrooted k-Minimum Spanning Tree (k-MST) is a rooted or unrooted (respectively) k-tree of minimal cost.

The constant b influences the runtime and the resulting team cost and can be chosen arbitrarily from the interval $(1, 2)$. We assume that the distance from the robot to any target is larger than b, which implies that n_0^* and n_1 are equal to the number of targets. This relationship can be enforced by eliminating all targets at the locations of the robots and then sufficiently decreasing the units in which distances are measured.

Lemma 1 *Bid$(\overline{T}, a, B)$ returns at least $|T'|$ targets if there exists a set of targets $T' \subseteq \overline{T}$ with $c_a^{path}(T') \leq 0.5\,\alpha B$.*

Proof: Consider $0 \leq k = |T'| \leq |\overline{T}|$. There exists a rooted k-tree whose cost is at most $0.5\,\alpha B$ since $c_a^{path}(T') \leq 0.5\,\alpha B$. Thus, the cost of the rooted k-MST is at most $0.5\,\alpha B$. The $(1/\alpha)$-approximation algorithm returns a rooted k-tree whose cost is at most $0.5\,B$ and the travel distance for circumnavigating the rooted k-tree (which is twice the cost of the rooted k-tree) is at most the given bound B. Thus, Bid(T, a, B) returns at least k targets. ■

The number of targets assigned to robots during the jth round of the SIV auction is $n_j - n_{j+1}$. The following theorem shows that this number is at least half of the number of the unassigned targets in the beginning of the jth round whose latencies are at most $0.5\,\alpha b^{j+1}$ in the optimal assignment.

Theorem 2 *For all $j \geq 1$, $0.5\,|T_j| \leq n_j - n_{j+1}$.*

Proof: Let $T_{a,j}$ be the subset of targets in T_j that are visited by robot a in the optimal assignment. Let $T_{a,j}^{miss}$ be the subset of targets in $T_{a,j}$ that are not assigned to any robot during the jth round of the SIV auction. The set of targets assigned to robot a during the jth round can be partitioned into the following sets: $T_{a,j}^{self}$ is the subset of targets in $T_{a,j}$ that are assigned to robot a during the jth round; $T_{a,j}^{in}$ is the subset of targets in $\cup_{a' \neq a} T_{a',j}$ that are assigned to robot a during the jth round; and $T_{a,j}^{out}$ is the subset of targets not in T_j that are assigned to robot a during the jth round. There are three important relationships among these sets:

1. $\sum_{a \in A}(|T_{a,j}^{self}| + |T_{a,j}^{in}| + |T_{a,j}^{out}|) = n_j - n_{j+1}$ since the sets $T_{a,j}^{self}$, $T_{a,j}^{in}$ and $T_{a,j}^{out}$ for all $a \in A$ partition the set of targets assigned to robots during the jth round.
2. $\sum_{a \in A}(|T_{a,j}^{self}| + |T_{a,j}^{in}| + |T_{a,j}^{miss}|) = |T_j|$ since the sets $T_{a,j}^{self}$, $T_{a,j}^{in}$ and $T_{a,j}^{miss}$ for all $a \in A$ partition the set T_j.
3. $\sum_{a \in A}(|T_{a,j}^{self}| + |T_{a,j}^{in}| + |T_{a,j}^{out}|) \geq \sum_{a \in A}(|T_{a,j}^{self}| + |T_{a,j}^{miss}|)$. The left-hand side of the inequality is the number of targets assigned to robots during the jth round. In the beginning of the iteration of the jth round where robot a is the winning robot, the targets in $T_{a,j}^{self}$ and $T_{a,j}^{miss}$ are unassigned with $c_a^{path}(T_{a,j}^{self} \cup T_{a,j}^{miss}) \leq c_a^{path}(T_{a,j}) \leq$

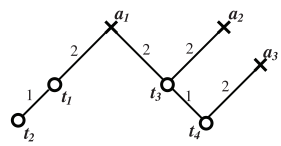

Figure 3: Example 1

$0.5\,\alpha b^{j+1}$. Thus, robot a wins at least $|T^{self}_{a,j}| + |T^{miss}_{a,j}|$ targets by Lemma 1. Summing over all agents A yields the relationship.

Solving Relationships (2) and (3) for $\sum_{a\in A}|T^{in}_{a,j}|$, adding the resulting inequalities and dividing by two yields $\sum_{a\in A}|T^{in}_{a,j}| \geq 0.5\,(|T_j| - \sum_{a\in A}(|T^{self}_{a,j}| + |T^{out}_{a,j}|))$. Substituting this inequality into Relationship (1) yields $n_j - n_{j+1} \geq 0.5\,(|T_j| + \sum_{a\in A}(|T^{self}_{a,j}| + |T^{out}_{a,j}|)) \geq 0.5\,|T_j|$. ■

Corollary 1 *For all $j \geq 1$, $n_{j+1} \leq 0.5\,(n^*_j + n_j)$.*

Proof: Let n'_j be the number of assigned targets in the beginning of the jth round of the SIV auction whose latencies are at most $0.5\,\alpha b^{j+1}$ in the optimal assignment. First, $n'_j \leq |T| - n_j$ since the right-hand side of the inequality is the number of assigned targets in the beginning of the jth round. Second, $|T| = |T_j| + n^*_j + n'_j$ by definition. Third, $0.5\,|T_j| \leq n_j - n_{j+1}$ by Theorem 2. Putting it all together, $0.5\,(n_j - n^*_j) = 0.5\,(|T| - n^*_j - (|T| - n_j)) \leq 0.5\,(|T| - n^*_j - n'_j) = 0.5\,|T_j| \leq n_j - n_{j+1}$. ■

The inequality of Corollary 1 can be tight, as shown in Figure 3. Assume that $\alpha = 1$, $b = 2 - \epsilon$ (where ϵ is a small positive constant so that $b^3 \geq 6$). Then, the optimal assignment is $T^*_{a_1} = \{t_1, t_2\}$, $T^*_{a_2} = \{t_3\}$ and $T^*_{a_3} = \{t_4\}$. Thus, $n^*_0 = n_1 = 4$ (since $b < 2$), $n^*_1 = n_2 = 4$ (since $b^2 < 4$), and $n^*_2 = 0$ (since $c^{path}_{a_1}(\{t_1, t_2\}) = 3 \leq 0.5\,b^3$ and $c^{path}_{a_2}(\{t_3\}) = c^{path}_{a_3}(\{t_4\}) = 2 \leq 0.5\,b^3$). If all three robots bid $\{t_3, t_4\}$ during the second round of the SIV auction and robot a_1 is the winning robot, then no targets are assigned to robots a_2 and a_3 in the second round. Thus, $n_3 = 2$ and $n_3 = 0.5\,(n^*_2 + n_2)$. Corollary 1 allows us to follow (Fakcharoenphol, Harrelson, and Rao 2003) to show that the team costs of SIV auctions are at most a factor of $O(1/\alpha)$ larger than minimal.

Lemma 2 *The minimal team cost c^* satisfies*

$$0.5\,\alpha(b-1)\sum_{j\geq 0} b^j n^*_j \leq c^*$$

Proof: Consider any target t whose latency is in the range $(0.5\,\alpha b^{j+1}, 0.5\,\alpha b^{j+2}]$ in the optimal assignment. First, target t contributes to n^*_i for all $0 \leq i \leq j$ since n^*_i is the number of targets whose latencies are larger than $0.5\,\alpha b^{i+1}$ in the optimal assignment. Second, $0.5\,\alpha(b-1)\sum_{i=0}^{j} b^i = 0.5\,\alpha(b^{j+1} - 1)$ is at most the latency of target t in the optimal assignment. Summing over all targets T yields the lemma. ■

Lemma 3 *The team cost c of an SIV auction satisfies*

$$c \leq 2\sum_{j\geq 0} b^{j+3} n_{j+1}$$

Proof: Consider any target t that is assigned during the $(j+1)$st round of the SIV auction. First, target t contributes to n_{i+1} for all $0 \leq i \leq j$ since n_{i+1} is the number of unassigned targets in the beginning of the $(i+1)st$ round of the SIV auction. Second, the latency of target t in the assignment produced by the SIV auction is at most $2(b^{j+3} - b^2)/(b-1)$ because the path cost of the set of targets assigned a robot during the ith round is at most b^{i+1} and the robot could return to its initial location before visiting the targets assigned to it in future rounds (but actually moves with minimal travel distance from each target to the next), resulting in target t having latency at most $\sum_{i=1}^{j+1} 2b^{i+1} = 2b^2\sum_{i=0}^{j} b^i = 2(b^{j+3} - b^2)/(b-1)$. Third, $2\sum_{i=0}^{j} b^{i+3} = 2(b^{j+4} - b^3)/(b-1)$ is at least $2(b^{j+3} - b^2)/(b-1)$ for all $b > 1$ and $j \geq 0$. Summing over all targets T yields the lemma. ■

Theorem 3 *The team costs of SIV auctions are at most a factor of $O(1/\alpha)$ larger than minimal if each robot calculates its bids with a $(1/\alpha)$-approximation algorithm for determining rooted k-MSTs.*

Proof: Continuing to follow (Fakcharoenphol, Harrelson, and Rao 2003), let $C = 2\sum_{j\geq 0} b^{j+3} n_{j+1}$ be the upper bound on the team cost of an SIV auction from Lemma 3. Then,

$$\begin{aligned}
C &= 2\sum_{j\geq 0} b^{j+3} n_{j+1} \\
&= 2b^3 n_1 + 2\sum_{j\geq 1} b^{j+3} n_{j+1} \\
\text{(Corollary 1)} \quad &\leq 2b^3 n_1 + 2\sum_{j\geq 1} b^{j+3}\, 0.5\,(n^*_j + n_j) \\
&= 2b^3 n^*_0 + \sum_{j\geq 1} b^{j+3} n^*_j + \sum_{j\geq 1} b^{j+3} n_j \\
&= b^3 n^*_0 + b^3\sum_{j\geq 0} b^j n^*_j + b\sum_{j\geq 0} b^{j+3} n_{j+1} \\
(n^*_0 \leq \textstyle\sum_{j\geq 0} b^j n^*_j) \quad &\leq 2b^3\sum_{j\geq 0} b^j n^*_j + b\sum_{j\geq 0} b^{j+3} n_{j+1} \\
\text{(Lemma 2)} \quad &\leq \frac{4b^3}{(b-1)}\frac{1}{\alpha}c^* + 0.5\,bC
\end{aligned}$$

Solving for C yields

$$C \leq \frac{8b^3}{(b-1)(2-b)}\frac{1}{\alpha}\,c^*$$

for the given constant $b \in (1, 2)$. ■

There exist constant-factor approximation algorithms for determining unrooted k-MSTs for given sets of targets $\overline{T}$,

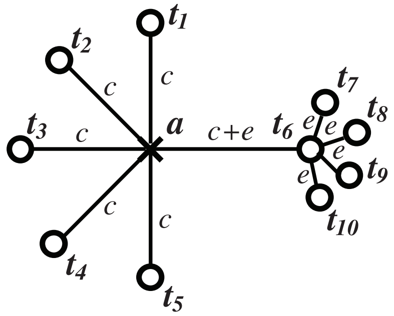

Figure 4: Example 2

such as (Garg 1996; Arora and Karakostas 2000), which can be transformed into constant-factor approximation algorithms for determining rooted k-MSTs for given sets of targets $\overline{T}$ as follows (Awerbuch et al. 1999): For $x = k, \ldots, |\overline{T}|$, determine an unrooted k-tree for the x targets in $\overline{T}$ closest to the root and then connect it to the root. Return the resulting rooted k-tree with the smallest cost. Thus, one can implement SIV auctions so that their team costs are at most a constant factor larger than minimal.

Corollary 2 *The team costs of SIV auctions are at most a factor of $O(1)$ larger than minimal if each robot calculates its bids with a constant factor approximation algorithm for determining rooted k-MSTs.*

However, constant-factor approximation algorithms for determining rooted k-MSTs are slow since they rely on primal-dual algorithms with Lagrangean relaxation (Chudak, Roughgarden, and Williamson 2004). We are not necessarily interested providing the best possible approximations but a good trade-off between runtime and the resulting team cost to prevent robots from being idle each time they allocate targets among themselves. We now show that there exist fast approximation algorithms for determining rooted k-MSTs for given sets of targets $\overline{T}$ that still result in a better guarantee on the team costs than the one currently known for SSI auctions. Consider, for example, a simple **nearest-neighbor algorithm** that determines a minimum spanning tree for the root and the k targets in $\overline{T}$ closest to the root.

Theorem 4 *The nearest-neighbor algorithm produces rooted k-trees whose costs are at most a factor of k larger than minimal.*

Proof: Let T' be the targets in the rooted k-tree produced by the nearest-neighbor algorithm and T'' be the targets in the rooted k-MST. Let D' be the largest distance from the root to any target in T' and D'' be the largest distance from the root to any target in T''. First, $D' \leq D''$ per construction of T'. Second, the cost of the rooted k-MST is at least D'' by the triangle inequality. Finally, the cost of the rooted k-tree produced by the nearest-neighbor algorithm is at most $kD' \leq kD''$ and thus at most a factor of k larger than minimal. ■

The upper bound of Theorem 4 is tight, as shown in Figure 4. The rooted k-tree produced by the nearest-neighbor algorithm contains the targets $t_1, \ldots, t_5$ and has cost $5c$, while the rooted k-MST for $k = 5$ contains the targets $t_6, \ldots, t_{10}$ and has cost $c + 5e$. The cost ratio approaches k as e approaches 0.

Corollary 3 *The team costs of SIV auctions are at most a factor of $O(|T|)$ larger than minimal if each robot calculates its bids with the nearest-neighbor algorithm for determining rooted k-MSTs (since $k \leq |T|$).*

Conclusions

We proposed a new auction-like algorithm, called sequential incremental-value (SIV) auction, which assigns as many tasks per round to robots as possible as long as their individual costs for performing these tasks are at most a given bound, which increases exponentially from round to round. Our theoretical results showed that the resulting sum of the latencies of all tasks is at most a constant factor larger than minimal.

References

Arora, S., and Karakostas, G. 2000. A 2+ϵ approximation algorithm for the k-MST problem. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 754–759.

Awerbuch, B.; Azar, Y.; Blum, A.; and Vempala, S. 1999. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing* 28(1):254–262.

Blum, A.; Chalasani, P.; Coppersmith, D.; Pulleyblank, B.; Raghavan, P.; and Sudan, M. 1994. The minimum latency problem. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 163–171.

Chudak, F.; Roughgarden, T.; and Williamson, D. 2004. Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation. *Mathematical Programming: Series A and B* 100(2):411–421.

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

Fakcharoenphol, J.; Harrelson, C.; and Rao, S. 2003. The k-traveling repairman problem. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 655–664.

Garg, N. 1996. A 3-approximation for the minimum tree spanning k vertices. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, 302–309.

Goemans, M., and Kleinberg, J. 1996. An improved approximation ratio for the minimum latency problem. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 152–158.

Jothi, R., and Raghavachari, B. 2007. Approximating the k-traveling repairman problem with repairtimes. *Journal of Discrete Algorithms* 5(2):293–303.

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

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

Ravi, R.; Sundaram, R.; Marathe, M.; Rosenkrantz, D.; and Ravi, S. 1994. Spanning trees short or small. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 546–555.

Sandholm, T. 1996. *Negotiation among self-interested computationally limited agents*. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst, MA.

Tovey, C.; Lagoudakis, M.; Jain, S.; and Koenig, S. 2005. The generation of bidding rules for auction-based robot coordination. In Parker, L.; Schneider, F.; and Schultz, A., eds., *Multi-Robot Systems: From Swarms to Intelligent Automata*. Springer. 3–14.