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 \(c^{\text{path}}_{a}(T_a)\) 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 \(c^{\text{robot}}_{a}(T_a)\) 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 \(\sum_{a \in A} c^{\text{robot}}_{a}(T_a)\). Any partial assignment with \(\bigcup_{a \in A} T_a = T\) (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...
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 (Blum et al. 1994). 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 (Lines 8-18): First, the auctioneer announces the unassigned targets to each robot in the announcement stage (Line 9). Second, each robot bids on each unassigned target in the bidding stage (Line 10). 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 11), 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 $O(|T|^{1/3})$ 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 some robot (which explains the term "single-item") so 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 additional targets to each robot as follows: First, the auctioneer announces the unassigned targets and the bound to each eligible robot in the announcement stage. A robot is eligible in a round if 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 targets assigned to a robot in this round is at most a 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 be-
coming 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 pos-
sible in the bidding stage of an SIV auction with the con-
straint that the path cost \( c_a^{\text{path}}(T') \) of the set of these targets
\( T' \) is at most the given bound \( B \). The problem of calcul-
ing 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 as-
sume that the robot approximates the calculations of its bids
by changing Line 18 of SIV auctions to

\[
T'_a \leftarrow \text{Bid}(T, a, B);
\]

The function \( \text{Bid}(T, a, B) \) determines, for \( k = 0, \ldots, |T| \), a rooted \( k \)-MST (whose root is the initial location of robot
\( a \)) for the unassigned targets \( T \) and returns the targets \( T' \) in
the rooted \( k \)-MST with the largest \( k \) with the constraint that
the path cost \( c_a^{\text{path}}(T') \) is at most the given bound \( B \).\(^1\) De-
termining 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 \( \text{Bid}(T, a, B) \) uses,
for \( k = 0, \ldots, |T| \), an \((1/\alpha)\)-approximation algorithm
for determining a rooted \( k \)-tree (whose root is the initial loca-
tion of robot \( a \)) for the unassigned targets \( T \) 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;
- \( c^* = \sum_{a \in A} c_{\text{robot}}(T'_a) \): 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 \( j \)th round of an SIV auction; and
- \( T_j \): the set of unassigned targets in the beginning of
  the \( j \)th round of the SIV auction whose latencies are at most
  \( 0.5 \alpha b^{j+1} \) in the optimal assignment.

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 \( n_j \) 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(\( T, a, B \)) returns at least \( \{|T'| \} \) targets if there
exists a set of targets \( T' \) with \( c_a^{\text{path}}(T') \leq 0.5 \alpha B \).

**Proof:** Consider \( 0 \leq k = |T'| \leq |T| \). There exists
a rooted \( k \)-tree whose cost is at most \( 0.5 \alpha B \) since
\( c_a^{\text{path}}(T') \leq 0.5 \alpha B \). 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 \( j \)th
round of the SIV auction is \( n_j = n_{j+1} \). The following the-
orem shows that this number is at least half of the number
of the unassigned targets in the beginning of the \( j \)th round
whose latencies are at most \( 0.5 \alpha b^{j+1} \) in the optimal assign-
ment.

**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 \( T_{a,j}^{\text{miss}} \)
be the subset of targets in \( T_{a,j} \) that are not assigned to any robot
during the \( j \)th round of the SIV auction. The set of targets
assigned to robot \( a \) during the \( j \)th round can be partitioned
into the following sets: \( T_{a,j}^{\text{self}} \) is the subset of targets in \( T_{a,j} \)
that are assigned to robot \( a \) during the \( j \)th round; \( T_{a,j}^{\text{in}} \) is
the subset of targets in \( \bigcup_{a' \neq a} T_{a',j} \) that are assigned to robot
\( a \) during the \( j \)th round; and \( T_{a,j}^{\text{out}} \) is the subset of targets not in
\( T_{a,j} \) that are assigned to robot \( a \) during the \( j \)th round. There
are three important relationships among these sets:

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

\(^1\)The cost of a tree is the sum of the costs of its edges. An un-
rooted \( k \)-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 root. A rooted or unrooted \( k \)-Minimum Spanning
Tree (\( MK \)) is a rooted or unrooted (respectively) \( k \)-tree of minimal
cost.
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 \( 2(\alpha b^{j+3} - b^2)/(b-1) \) because the path cost of the set of targets assigned a robot during the \( i \)th 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

\[
\sum_{i=0}^{j+1} 2 b^{i+1} = 2 b^2 \sum_{i=0}^{j} b^i = 2(\alpha b^{j+3} - b^2)/(b-1).
\]

Third, \( 2 \sum_{i=0}^{j} b^{i+3} = 2(\alpha b^{j+4} - b^3)/(b-1) \) is at least \( 2(\alpha b^{j+4} - b^3)/(b-1) \) for all \( b > 1 \) and \( j \geq 0 \). Summing over all targets \( T \) yields the lemma. \( \blacksquare \)

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,

\[
C = 2 \sum_{j \geq 0} b^{j+3} n_{j+1} = 2 b^3 n_1 + 2 \sum_{j \geq 1} b^{j+3} n_{j+1}.
\]

(Corollary 1)

\[
\leq 2 b^3 n_1 + 2 \sum_{j \geq 1} b^{j+3} 0.5 (n_j^* + n_j) = 2 b^3 n_0 + \sum_{j \geq 1} b^{j+3} n_j^* + \sum_{j \geq 1} b^{j+3} n_j.
\]

\[
= b^3 n_0^* + b^3 \sum_{j \geq 0} b^j n_j^* + b \sum_{j \geq 0} b^{j+3} n_{j+1}.
\]

\[
\sum_{j \geq 0} b^{j+3} n_{j+1} \leq 2 b^3 \sum_{j \geq 0} b^{j+3} n_{j+1} \leq \frac{4b^3}{(b-1) \alpha} c^* + 0.5 b C.
\]

Solving for \( C \) yields

\[
C \leq \frac{8b^3}{(b-1)(2-b) \alpha} c^* + 0.5 b C.
\]

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

There exist constant-factor approximation algorithms for determining unrooted \( k \)-MSTs for given sets of targets \( T \).
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 $T$ as follows (Awerbuch et al. 1999): For $x = k, \ldots, \lvert T \rvert$, determine an unrooted $k$-tree for the $x$ targets in $T$ 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 latencies of all tasks is at most a constant factor larger than minimal.

**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 $kD'$ 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 $O(\lvert T \rvert)$ larger than minimal if each robot calculates its bids with the nearest-neighbor algorithm for determining rooted $k$-MSTs (since $k \leq \lvert T \rvert$).

**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**


