

Improving Sequential Single-Item Auctions

Xiaoming Zheng

Computer Science Department
University of Southern California
Los Angeles, California 90089-0781
xiaominz@usc.edu

Sven Koenig

Computer Science Department
University of Southern California
Los Angeles, California 90089-0781
skoenig@usc.edu

Craig Tovey

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205
ctovey@isye.gatech.edu

Abstract—We study how to improve sequential single-item auctions that assign targets to robots for exploration tasks such as environmental clean-up, space-exploration, and search and rescue missions. We exploit the insight that the resulting travel distances are small if the bidding and winner-determination rules are designed to result in hillclimbing, namely to assign an additional target to a robot in each round of the sequential single-item auction so that the team cost increases the least. We study the impact of increasing the lookahead of hillclimbing and using roll-outs to improve the evaluation of partial target assignments. We describe the bidding and winner-determination rules of the resulting sequential single-item auctions and evaluate them experimentally, with surprising results: Larger lookaheads do not improve sequential single-item auctions reliably while only a small number of roll-outs in early rounds already improve them substantially.

I. INTRODUCTION

We study exploration tasks where a team of mobile robots has to visit a number of given targets. Examples include environmental clean-up, space-exploration, and search and rescue missions. How to assign targets to robots is a difficult problem. Centralized control is inefficient in terms of both the required amount of computation and communication since the central controller is the bottleneck of the system. Market-based approaches are decentralized and appear to perform well in many situations. Auctions, in particular, can be efficient in terms of both the required amount of computation and communication since information is compressed into numerical bids that the robots can compute in parallel [1]. Consequently, several research groups are now investigating how to use auctions to coordinate teams of robots [2], [3]. Recent theoretical and experimental results show that sequential single-item auctions (short: SSI auctions) are fast, yet result in small team costs [4]. For example, SSI auctions can provide constant-factor performance guarantees for the sum of the travel distances of the robots even if they use approximations that allow them to run in polynomial time [5]. In contrast, complete combinatorial auctions, where the robots bid on all possible sets of targets in a single round, have prohibitively large computation and communication burden but result in optimal target assignments [6]. In this paper, we study how to improve SSI auctions by increasing their similarity to combinatorial auctions without greatly increasing their communication and computational burden. The two kinds of auctions differ in some salient ways. Combinatorial auctions require each robot to bid on many

overlapping bundles of items, whereas SSI auctions make them bid only on single items and thereby eliminate overlaps. We decrease this difference by increasing the maximum bundle size of SSI auctions to two or three items and permitting overlaps. We prove that this can be done without greatly increasing the communication and computational burden. Surprisingly, our experimental results show that this idea does not improve SSI auctions reliably. We therefore consider another salient difference. Combinatorial auctions evaluate complete target assignments, whereas SSI auctions evaluate partial target assignments. We decrease this difference by making SSI auctions greedily complete the partial target assignments and then evaluate the resulting complete target assignments, a concept that we call rollout. Our experimental results show that this idea improves SSI auctions substantially even if they perform only a small number of rollouts in early rounds. Thus, it appears to be more important to consider complete solutions a few times than to repeatedly pack perfectly a few solution pieces at a time - an important insight for improving auctions that assign targets to robots.

II. SEQUENTIAL SINGLE-ITEM AUCTIONS

Sequential Single-Item Auctions: During each round of a sequential single-item auction (SSI auction), all robots are eligible to bid on all unassigned targets. The robot that places the overall lowest cost bid on any target is assigned that particular target. (Ties can be broken arbitrarily.) A new round of bidding starts, and all robots may bid again on all unassigned targets, and so on until all targets have been assigned to robots. Each robot then calculates the shortest path for visiting all targets assigned to it from its current location and then moves along that path. (A robot does not move if no targets are assigned to it.) To simplify notation, we assume that all targets are initially unassigned, but the auction design can be applied if some targets are pre-assigned. Indeed, the k th round could be thought of as the first round of an auction in which $k - 1$ targets were preassigned.

In each round, it suffices that each robot submits one bid (its lowest cost bid) since only one bid is accepted per round. Therefore, the communication and winner-determination burden of all rounds of an SSI auction combined is much smaller than that of a combinatorial auction, even a severely limited combinatorial auction that restricts bundle sizes. However, the lesser burden apparently entails a loss in ability to consider the

whole rather than the parts, that is, the team performance rather than the individual robot performances. Fortunately, this loss can be offset in large part by incorporating the team objective into the bid calculations.

Team Objectives: We introduce two standard team objectives, which serve as both examples and computational test cases. Denote the set of robots as $R = \{r_1, \dots, r_n\}$ and the set of targets as $T = \{t_1, \dots, t_m\}$. SSI auctions assign a set of targets T_i to robot r_i for all $r_i \in R$, where the set $\{T_1, \dots, T_n\}$ forms a partition of all targets. For any robot r and any set of targets T' , let $D(r, T')$ denote the minimum travel distance that robot r needs to visit all targets in T' from its current location. The MiniSum team objective is to minimize $tc(T_1, \dots, T_n) := \sum_i D(r_i, T_i)$, that is, the sum of the minimum travel distances of the robots (corresponding, for example, roughly to their total energy consumption). The MiniMax team objective is to minimize $tc(T_1, \dots, T_n) := \max_i D(r_i, T_i)$, that is, the largest minimum travel distance of any robot (corresponding roughly to the task-completion time).

Bidding and Winner Determination: The bidding and winner-determination rules depend on the team objective. The winner-determination rule determines which bid should win. The bidding rule determines how much a robot should bid on an unassigned target (we drop the “unassigned” in the following to improve the readability of the text) during any round of the SSI auction. It has been shown that the team cost (= value of the team objective) is small if the SSI auction results in hillclimbing, namely assigns an additional target to a robot in each round of the SSI auction so that the team cost increases the least [4]. Consider any round of the SSI auction, and assume that each robot $r_i \in R$ has already been assigned the set of targets T_i in previous rounds. Then, the team cost is currently $tc(T_1, \dots, T_n)$. If robot r_i is now assigned target t with $t \notin T_1 \cup \dots \cup T_n$, then the team cost becomes $tc(T_1, \dots, T_i \cup \{t\}, \dots, T_n)$. The idea is to assign that target t to that robot r_i so that $tc(T_1, \dots, T_i \cup \{t\}, \dots, T_n) - tc(T_1, \dots, T_n)$ is smallest. [4] showed that this can be achieved by each robot $r_i \in R$ calculating the following bid cost on each target $t \in T$ with $t \notin T_i$, which robot r_i can calculate without having to know where the other robots are or which targets have already been assigned to them:

- $D(r_i, T_i \cup \{t\}) - D(r_i, T_i)$ (= the increase in the minimum travel distance needed by robot r_i to visit all targets assigned to it if target t were assigned to it as well) for the MiniSum team objective, which is similar to previous work on marginal-cost bidding in ContractNet [7], and
- $D(r_i, T_i \cup \{t\})$ (= the minimum travel distance needed by robot r_i to visit all targets assigned to it in case target t were assigned to it as well) for the MiniMax team objective.

Robot r_i needs to calculate $D(r_i, T_i \cup \{t\})$ for both team objectives, which is NP-hard since the robot needs to solve a version of a traveling salesman problem (TSP). The robot can

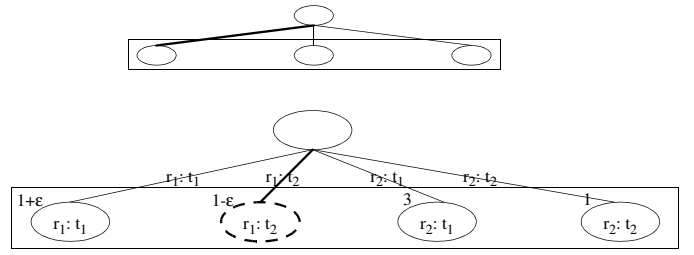


Fig. 1. Standard Hillclimbing (with Lookahead One and without Rollouts)

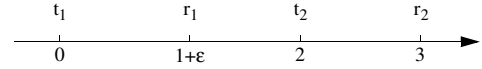


Fig. 2. One-Dimensional Example 1

use polynomial-time TSP heuristics to calculate the minimal travel distance (such as the three-opt or cheapest-insertion heuristics). We use such approximations in the experiments as described in [4] but assume in the theoretical part of this paper that the minimal travel distances are not approximated.

Figure 1 shows the search performed in the first round by the hillclimbing performed by SSI auctions. (We refer to this version of hillclimbing as standard hillclimbing throughout the paper.) The top of the figure shows the search for an abstract example, while the bottom shows the search for the example from Figure 2 in the context of the MiniSum team objective. The robots and targets are located on the real line. (Epsilon is a small positive tie-breaking constant.) The search starts with the current partial target assignment (initially the empty one). All possible target assignments resulting from assigning one additional target to a robot are generated and evaluated according to their team cost. Then, the one with the smallest team cost is chosen and the procedure repeats. Each oval in the figure represents a (partial or complete) target assignment. The resultant team cost is adjacent to the upper left perimeter of each oval. The box indicates which target assignments are compared. Arrows indicate the team cost derivations. A thick line indicates the assignment of an additional target to a robot made by standard hillclimbing. Finally, the dashed oval shows the target assignment from which the search starts in the next round.

A. Experimental Evaluation

Standard hillclimbing was evaluated in [4] for different numbers of robots and targets in eight-neighbor planar grids of size 51×51 that resemble office environments, as shown in Figure 4. The table in Figure 7 reports the team cost for the MiniMax team objective, and the table in Figure 8 reports the team cost for the MiniSum team objective. The team costs for the same number of robots and targets are averaged over the same ten randomly generated initial robot and (unclustered) target locations. Both tables also report the average of the

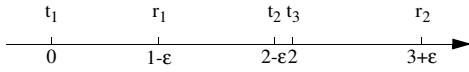


Fig. 3. One-Dimensional Example 2

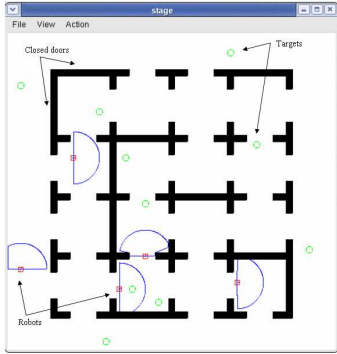


Fig. 4. Screenshot

runtimes over all ten situations, measured in seconds.¹ The case with two robots and ten targets is sufficiently small to be solved optimally with mixed-integer programming. The minimal team cost for the MiniMax team objective is 109.34, and the minimal team cost for the MiniSum team objective is 189.15 [4].

III. IMPROVEMENT: LARGER LOOKAHEAD

A simple idea for improving SSI auctions is to continue to perform hillclimbing but change the lookahead from the assignment of one additional target to a robot to the assignment of $k \geq 2$ additional targets to robots (either the same robot or different robots) so that the team cost increases the least. To be careful, we assign only one of the k targets to its robot, namely the target that increases the team cost the least. In the next round, another target is assigned to a robot, until all targets have been assigned to robots.

Consider again the example from Figure 2 in the context of the MiniSum team objective. Figure 5 shows the search performed in the first round by hillclimbing with lookahead two. All possible target assignments resulting from assigning two additional targets to robots are generated. Each assignment of one additional target to a robot is then evaluated according to the smallest team cost of all assignments of two additional targets to robots that include it as the first step (lookahead-two team cost). Then, the one with the smallest lookahead-two team cost is chosen (using the team cost of the partial target assignment to break ties) and the procedure repeats.

We expect that hillclimbing with larger lookaheads, being less myopic, would result in smaller team costs than standard hillclimbing. Consider, for instance, the example from Figure 2 for both the MiniSum and MiniMax team objectives. Hillclimbing with lookahead one proceeds as follows. Robot r_1 is

¹The runtime of hillclimbing for the same number of targets decreases as the number of robots increases because each robot then tends to visit fewer targets. The bidding subproblems are smaller and can be solved much more quickly.

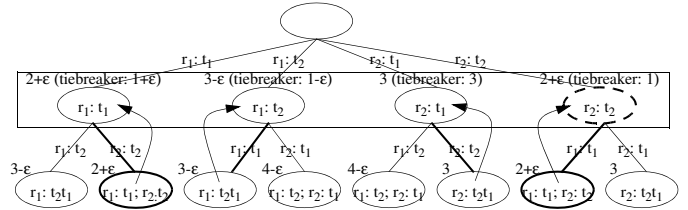
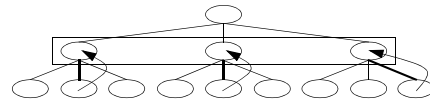


Fig. 5. Hillclimbing with Lookahead Two

assigned target t_2 in the first round (as shown in Figure 1 for the MiniSum team objective) and target t_1 in the second round. Then, robot r_1 minimizes its travel distance by first visiting target t_2 and then target t_1 (we write this as $r_1 \rightarrow t_2 \rightarrow t_1$) and robot r_2 does not move. The resulting team cost of the MiniSum and MiniMax team objectives is $3 - \epsilon$. Hillclimbing with lookahead two, on the other hand, considers all targets right away and thus finds an optimal target assignment. Robot r_2 is assigned target t_2 in the first round (as shown in Figure 5 for the MiniSum team objective) and robot r_1 is assigned target t_1 in the second round. Then, $r_1 \rightarrow t_1$ and $r_2 \rightarrow t_2$. The resulting team cost of the MiniSum team objective is $2 + \epsilon$ and the team cost of the MiniMax team objective is $1 + \epsilon$, in accord with our expectation.

However, hillclimbing with larger lookaheads may result in larger team costs than standard hillclimbing. Consider, for instance, the example from Figure 3. Hillclimbing with lookahead one results in $r_1 \rightarrow t_1$ and $r_2 \rightarrow t_3 \rightarrow t_2$. The resulting team cost of the MiniSum team objective is $2 + 1\epsilon$, and the team cost of the MiniMax team objective is $1 + 2\epsilon$. Hillclimbing with lookahead two, on the other hand, results in $r_1 \rightarrow t_1 \rightarrow t_2 \rightarrow t_3$ for the MiniSum team objective, with a team cost of $3 - \epsilon$, and $r_1 \rightarrow t_1 \rightarrow t_2$ and $r_2 \rightarrow t_3$ for the MiniMax team objective, with a team cost of $3 - 2\epsilon$, supporting our claim. Furthermore, this example can be extended to hillclimbing with even larger lookaheads. If we place m additional targets between targets t_2 and t_3 (for a total of $m + 3$ targets) then hillclimbing with lookahead x finds the optimal target assignment for $x = 1$ and $x = n + 3$ but finds suboptimal target assignments for $1 < x < n + 3$ for both the MiniSum and MiniMax team objectives. An intuitive explanation for this perhaps counter-intuitive result in the context of this example is that target assignments that result in benefits within the lookahead but also costs beyond the lookahead are misjudged to be better than they actually are.

A. Implementation Aspects

Hillclimbing with larger lookaheads can still be implemented with SSI auctions although the bidding and winner-determination rules become more complex. In particular, robots can now bid on sets of targets. The bid costs are

calculated in a way similar to before. The bid cost of a robot for a given set of targets is the increase in its minimum travel distance (for the MiniSum team objective) and the minimum travel distance itself (for the MiniMax team objective) that is needed to visit all targets assigned to it in case the given set of targets were assigned to it as well. We employ a unified notation for the evaluation of a combination B of bids b . Let $v_B := \{v_b : b \in B\}$ denote the set of bid costs. Then $C(v_B)$ denotes the evaluation of the effect on the team objective. For the MiniSum team objective, $C(v_B) = \sum_{b \in B} v_b$. For the MiniMax team objective, $C(v_B) = \max_{b \in B} v_b$. Both the sum and the max functions are obviously monotonic nondecreasing and neutral, that is, independent of the ordering of the elements of B . These two properties, monotonicity and neutrality, will permit a number of bids per robot that is $O(1)$ and a winner-determination rule whose runtime is $O(|R|)$, for each round of an SSI auction that implements hillclimbing with a fixed lookahead k .

It is easy to see that the number of bids submitted per robot only needs to be polynomial in the number of targets for each round. Let T be the set of targets in the current round. Consider the following bids by a robot: For every set $S \subseteq T$ with $|S| \leq k$, the robot bids on a set with the lowest bid cost among all sets $S' \subseteq T$ with $S \cap S' = \emptyset$ and $|S| + |S'| = k$. These bids suffice to implement hillclimbing with lookahead k since hillclimbing with lookahead k assigns all other robots the targets in one of the sets S considered by the robot and by monotonicity it is then optimal to assign the robot the targets in the corresponding set S' that the robot bid on. In fact, the number of bids submitted per robot can be shown to depend only on the lookahead k , regardless of the number of targets. Here we show that the number of bids per robot during any round of the SSI auction is three for hillclimbing with lookahead two and thus does not depend on the number of robots or targets. We also show that the runtime of the winner-determination rule is linear in the number of robots and independent of the number of targets for hillclimbing with lookahead two.

For hillclimbing with lookahead two, robots bid on single targets and pairs of targets. Each robot submits three bids during each round of the SSI auction. It bids on a single target with the lowest bid cost of any single target (Bid a), a single target with the lowest bid cost of any single target except for the target of Bid a (Bid b), and a pair of targets with the lowest bid cost of any pair of targets (Bid c). We use B to denote the set of bids submitted by all robots. We write bids b as $b = (r, T', v)$, meaning that robot r submitted bid cost v on the set of targets T' . We claim that the above three bids submitted per robot suffice to implement hillclimbing with lookahead two. There are two mutually exhaustive cases:

- Case 1: There is an optimal assignment that assigns two targets to the same robot. In this case, there is an optimal assignment that assigns that robot the targets of its Bid c.
- Case 2: There is an optimal assignment that assigns one target each to two robots. Subcase 2.1: If those two robots

differ in the target of their Bid a, then there is an optimal assignment that assigns each of them the target of its Bid a. Subcase 2.2: Otherwise, there is an optimal assignment that assigns one robot the target of its Bid a and the other one the target of its Bid b.

Both cases make use of monotonicity. Case 2 also makes use of neutrality. Our claim easily leads to a winner-determination rule, whose pseudocode is given below. To begin, select the best, that is, the lowest cost, Bid c from among all robots. This is the Case 1 alternative (step 10). Next, select the best two Bid a's from among the robots (steps 1–2). There are two subcases. Subcase 2.1: If the targets of the two Bid a's are distinct, those two bids are the Case 2 alternative. Subcase 2.2: Otherwise (steps 4–9), let the target in question be t and the two robots be r_1 and r_2 . It is easy to see that the winning target assignment assigns target t to either robot r_1 or r_2 . Thus, construct two candidate target assignments as follows. Target assignment 2.2.1: Target t is assigned to robot r_1 and the other target assignment is the best Bid a or Bid b that is for a target other than t , from any robot other than r_1 (step 7). Target assignment 2.2.2: Target t is assigned to robot r_2 , and the other target assignment is the best Bid a or Bid b that is for a target other than t , from any robot other than r_2 (step 9). The Case 2 alternative is the better one of the target assignments 2.2.1 and 2.2.2 (steps 6–9). Finally, select as the winning target assignment the better one of the Case 1 and the Case 2 alternatives (steps 11–14).

- 1) $(r_1, \{t_1\}, v_1) := \arg \min_{(r, \{t\}, v) \in B} v$.
- 2) $(r_2, \{t_2\}, v_2) := \arg \min_{(r, \{t\}, v) \in B: r \neq r_1} v$.
- 3) If $t_1 = t_2$ then
- 4) $(r_3, \{t_3\}, v_3) := \arg \min_{(r, \{t\}, v) \in B: r \neq r_1, t \neq t_1} v$.
- 5) $(r_4, \{t_4\}, v_4) := \arg \min_{(r, \{t\}, v) \in B: r \neq r_2, t \neq t_1} v$.
- 6) if $C(\{v_1, v_3\}) \leq C(\{v_2, v_4\})$ then
- 7) $r_2 := r_3$. $t_2 := t_3$. $v_2 := v_3$.
- 8) else
- 9) $r_1 := r_4$. $t_1 := t_4$. $v_1 := v_4$.
- 10) $(r_5, \{t_5, t_6\}, v_5) := \arg \min_{(r, \{t, t'\}, v) \in B} v$.
- 11) if $C(\{v_5\}) \leq C(\{v_1, v_2\})$ then
- 12) Case 1: consider the assignment of t_5 and t_6 to r_5 .
- 13) else
- 14) Case 2: consider the assignment of t_1 to r_1 and of t_2 to r_2 .

The winner-determination rule assigns only one of the two selected targets to its robot, namely the target that increases the team cost the least. The winner-determination rule can easily determine this target from the bids. For example, in Case 2 in the pseudo code, if $v_1 < v_2$, then the winner-determination rule assigns target t_1 to robot r_1 , otherwise it assigns target t_2 to robot r_2 . For efficiency in selecting one target in Case 1, we adopt the following convention: The target pair is ordered in increasing order of bid costs on individual targets. For example, if (r, t, v) and (r, t', v') are bids with $v < v'$, then the target set that consists of targets t and t' is written as $\{t, t'\}$ rather than $\{t', t\}$.² This convention allows the winner-determination rule to assign target t_5 to robot r_5

²Consider, for instance, the example from Figure 3 for the MiniSum team objective. Then, $B = \{(r_1, t_1, 1 - \epsilon), (r_1, t_2, 1), (r_1, \{t_2, t_3\}, 1 + \epsilon), (r_2, t_3, 1 + \epsilon), (r_2, t_2, 1 + 2\epsilon), (r_2, \{t_3, t_2\}, 1 + 2\epsilon)\}$.

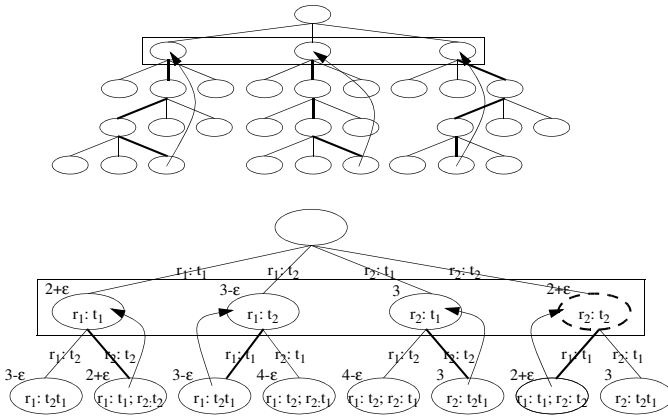


Fig. 6. Hillclimbing with Rollouts

in Case 1 and thus eliminates the need for an additional round of communication with robot r_5 .

To summarize, each robot can determine its three bids by enumerating and evaluating all $O(|T|^2)$ subsets of one or two targets at worst. Thus, the amount of computation of each robot per round is not too large. The number of submitted bids and thus the overall amount of communication as well as the amount of computation to determine the winning robots is even smaller.

B. Experimental Evaluation

The tables in Figures 7 and 8 show that hillclimbing with lookaheads two and three does not reduce the team cost reliably compared to standard hillclimbing. Also, the reductions in team cost that do occur are only marginal even though the runtimes for hillclimbing with lookahead three are substantial, with bid generation responsible for most of the runtime. We therefore investigate a different technique for improving SSI auctions in the following section.

IV. IMPROVEMENT: ROLLOUTS

One problem of standard hillclimbing is that the team costs for partial target assignments do not predict the team costs for the complete target assignments well. A different idea for improving SSI auctions therefore is to continue to perform hillclimbing with lookahead one, as done by standard hillclimbing, but to evaluate the resulting partial target assignments by first using standard hillclimbing to complete them and then using the team costs for the complete target assignments to evaluate the partial ones, rather than using the team costs for the partial target assignments directly. We refer to the completion of the target assignments as rollouts, which is standard terminology in reinforcement learning for evaluating whole trajectories according to their true rewards rather than estimates of their true rewards after the first move [8].

Consider again the example from Figure 2 in the context of the MiniSum team objective. Figure 6 shows the search performed in the first round by hillclimbing with rollouts. Each assignment of one additional target to a robot is evaluated

according to the team cost of the complete target assignment that results when first assigning the target to the robot and then performing standard hillclimbing to complete the target assignment (rollout team cost). Then, the one with the smallest rollout team cost is chosen and the procedure repeats.

We expect that hillclimbing with rollouts, being less myopic, would result in smaller team costs than standard hillclimbing. Consider, for instance, the example from Figure 2 for both the MiniSum and MiniMax team objectives. Since there are only two targets, hillclimbing with rollouts and hillclimbing with lookahead two behave identically, and we have already shown that hillclimbing with lookahead two results in smaller team costs than standard hillclimbing. Similarly, consider again the example from Figure 3. Hillclimbing with rollouts results in $r_1 \rightarrow t_1$ and $r_2 \rightarrow t_3 \rightarrow t_2$. The resulting team costs of the MiniSum team objective is $2 + 1\epsilon$ and the team costs of the MiniMax team objective is $1 + 2\epsilon$, and hillclimbing with rollouts avoids the suboptimality of hillclimbing with lookahead two in this example. In fact, hillclimbing with rollouts cannot result in larger team costs than standard hillclimbing, for the following reason: Each rollout of hillclimbing with rollouts is evaluated according to the team costs for the complete target assignment that it achieves. One of the rollouts of hillclimbing with rollouts is identical to standard hillclimbing, which implies that the first assignment of a target to a robot resulting from hillclimbing with rollouts can be completed to a complete target assignment whose team cost is no larger than the one of the complete target assignment resulting from standard hillclimbing. This argument can now be recursively applied, supporting our claim. This guarantee distinguishes hillclimbing with rollouts from hillclimbing with larger lookaheads, which cannot make such guarantees unless the lookaheads are equal to the number of targets.

A. Implementation Aspects

Hillclimbing with rollouts can still be implemented with SSI auctions. However, the robots now need to run several sets of SSI auctions rather than just one, namely one for each combination of round, robot and target. This can make hillclimbing with rollouts time-consuming. We now discuss two (non-orthogonal) ways of speeding up hillclimbing with rollouts:

- *Hillclimbing with simplified rollouts* speeds up hillclimbing with rollouts by sampling only some of the possible rollouts (including the one that is identical to standard hillclimbing). It runs standard hillclimbing to determine which target t it would assign to which robot r during the current round. Hillclimbing with simplified rollouts then tries the rollouts for all assignments that assign target t to some robot in the current round and for all assignments that assign some target to robot r in the current round. Thus, hillclimbing with rollouts performs $|T||R|$ rollouts in the current round, while hillclimbing with simplified rollouts performs only $|T| + |R| - 1$ rollouts.
- *Hillclimbing with early rollouts* speeds up hillclimbing with rollouts by performing rollouts only during the

Robots	Targets	Standard		Lookahead 2		Lookahead 3		Rollouts		Simplified Rollouts	
		Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)
2	10	125.84	(0.00)	123.73	(0.00)	118.49	(0.01)	109.97	(0.03)	112.13	(0.01)
2	20	163.37	(0.01)	156.24	(0.06)	152.80	(1.07)	142.88	(3.22)	142.76	(1.15)
2	30	185.32	(0.06)	189.93	(0.85)	189.36	(18.57)	173.83	(60.53)	174.10	(19.71)
2	40	221.83	(0.27)	214.66	(5.37)	216.67	(148.67)	191.01	(693.03)	193.42	(183.10)
4	10	73.26	(0.00)	73.15	(0.00)	66.62	(0.01)	59.39	(0.02)	62.27	(0.00)
4	20	94.61	(0.01)	90.32	(0.03)	91.36	(0.43)	81.01	(1.13)	85.12	(0.20)
4	30	105.14	(0.02)	107.94	(0.16)	102.97	(4.00)	88.16	(16.16)	96.04	(2.91)
4	40	138.12	(0.05)	137.06	(0.90)	124.03	(26.46)	102.37	(133.88)	109.09	(24.86)
6	10	55.46	(0.00)	52.33	(0.00)	51.54	(0.01)	47.28	(0.01)	49.77	(0.00)
6	20	74.35	(0.01)	73.68	(0.02)	69.17	(0.29)	60.47	(0.66)	64.41	(0.10)
6	30	88.48	(0.02)	81.53	(0.07)	78.01	(2.23)	62.39	(8.98)	70.16	(1.40)
6	40	86.70	(0.03)	87.21	(0.37)	84.52	(14.05)	71.44	(66.59)	73.18	(9.14)
8	10	44.12	(0.00)	43.80	(0.00)	42.03	(0.01)	38.81	(0.01)	38.49	(0.00)
8	20	58.71	(0.01)	58.56	(0.01)	57.85	(0.24)	47.10	(0.54)	50.17	(0.06)
8	30	60.98	(0.01)	60.99	(0.05)	57.61	(1.66)	50.08	(6.29)	56.98	(0.64)
8	40	74.54	(0.02)	75.12	(0.17)	72.92	(8.04)	58.40	(38.22)	64.28	(3.78)
10	10	36.55	(0.00)	37.09	(0.00)	36.07	(0.02)	34.24	(0.02)	34.71	(0.00)
10	20	48.25	(0.01)	45.38	(0.01)	45.10	(0.24)	36.23	(0.42)	38.29	(0.04)
10	30	55.92	(0.01)	56.22	(0.04)	56.04	(1.42)	44.09	(4.45)	47.29	(0.38)
10	40	60.70	(0.02)	59.79	(0.13)	61.24	(6.19)	48.29	(31.50)	54.60	(2.73)

Fig. 7. MiniMax Team Objective

Robots	Targets	Standard		Lookahead 2		Lookahead 3		Rollouts		Simplified Rollouts	
		Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)	Team Cost	(Runtime)
2	10	193.50	(0.00)	193.88	(0.00)	191.42	(0.02)	189.15	(0.06)	190.32	(0.02)
2	20	264.50	(0.03)	264.51	(0.16)	267.38	(2.76)	262.05	(9.51)	262.05	(2.67)
2	30	324.61	(0.20)	329.66	(2.18)	328.27	(46.81)	315.32	(239.00)	317.15	(62.20)
2	40	367.97	(1.05)	372.31	(13.49)	370.76	(378.95)	358.82	(2529.60)	360.07	(551.77)
4	10	152.63	(0.00)	151.60	(0.00)	151.56	(0.02)	149.83	(0.05)	151.81	(0.02)
4	20	242.52	(0.01)	241.88	(0.06)	240.35	(1.15)	234.13	(5.14)	236.10	(1.16)
4	30	291.48	(0.04)	289.64	(0.48)	290.37	(10.46)	280.54	(92.35)	283.97	(23.38)
4	40	348.47	(0.32)	351.70	(7.55)	352.60	(240.78)	334.99	(695.40)	337.70	(276.36)
6	10	146.83	(0.00)	147.57	(0.00)	146.21	(0.02)	145.41	(0.04)	145.41	(0.01)
6	20	219.14	(0.01)	218.97	(0.06)	217.11	(0.71)	214.87	(4.62)	217.55	(1.39)
6	30	255.46	(0.02)	256.91	(0.31)	256.90	(8.31)	248.07	(35.42)	249.72	(15.30)
6	40	312.67	(0.05)	315.66	(1.33)	316.07	(42.78)	303.29	(346.01)	304.23	(78.75)
8	10	126.16	(0.00)	126.94	(0.00)	126.02	(0.01)	125.88	(0.03)	125.88	(0.00)
8	20	198.65	(0.01)	205.16	(0.03)	200.08	(0.42)	193.38	(2.19)	194.20	(0.49)
8	30	241.08	(0.02)	244.43	(0.13)	241.59	(3.70)	237.35	(32.28)	238.54	(5.97)
8	40	295.68	(0.05)	294.88	(0.82)	294.42	(19.80)	288.01	(227.61)	289.51	(40.81)
10	10	108.03	(0.00)	110.43	(0.00)	107.65	(0.02)	107.59	(0.04)	107.59	(0.01)
10	20	180.12	(0.01)	180.61	(0.02)	179.89	(0.36)	178.65	(1.49)	178.67	(0.32)
10	30	243.19	(0.02)	242.55	(0.09)	241.31	(2.75)	234.80	(20.98)	235.36	(4.15)
10	40	278.16	(0.05)	281.06	(0.72)	279.77	(21.96)	270.70	(217.09)	274.21	(46.54)

Fig. 8. MiniSum Team Objective

Robots	Targets	MiniMax Team Objective						MiniSum Team Objective					
		No Round	Round 1	Round 1-2	Round 1-3	All Rounds	No Round	Round 1	Round 1-2	Round 1-3	All Rounds		
		Team Cost	Team Cost	Team Cost	Team Cost (Runtime)	Team Cost	Team Cost	Team Cost	Team Cost	Team Cost (Runtime)	Team Cost		
2	10	125.84	110.32	109.99	109.97 (0.01)	109.97	193.50	189.15	189.15	189.15 (0.03)	189.15		
2	20	163.37	145.56	143.69	142.88 (0.67)	142.88	264.50	262.05	262.05	262.05 (2.75)	262.05		
2	30	185.32	176.55	174.39	173.83 (9.82)	173.83	324.61	316.32	315.75	315.33 (57.15)	315.32		
2	40	221.83	196.68	193.53	191.94 (117.14)	191.01	367.97	359.07	358.93	358.93 (237.33)	358.82		
4	10	73.26	59.88	59.39	59.39 (0.01)	59.39	152.63	149.83	149.83	149.83 (0.02)	149.83		
4	20	94.61	83.92	81.39	81.22 (0.40)	81.01	242.52	235.13	234.13	234.13 (1.12)	234.13		
4	30	105.14	92.01	89.27	88.64 (3.97)	88.16	291.48	280.76	280.54	280.54 (15.49)	280.54		
4	40	138.12	106.72	103.57	102.78 (26.04)	102.37	348.47	338.10	335.35	334.99 (122.20)	334.99		
6	10	55.46	47.93	47.33	47.28 (0.01)	47.28	146.83	145.41	145.41	145.41 (0.02)	145.41		
6	20	74.35	63.68	60.83	60.63 (0.23)	60.47	219.14	214.87	214.87	214.87 (1.64)	214.87		
6	30	88.48	66.33	63.22	62.75 (1.98)	62.39	255.46	248.28	248.07	248.07 (9.43)	248.07		
6	40	86.70	74.77	72.88	72.19 (12.91)	71.44	312.67	305.65	304.20	304.20 (54.42)	303.29		
8	10	44.12	40.04	38.80	38.81 (0.01)	38.81	126.16	125.88	125.88	125.88 (0.02)	125.88		
8	20	58.71	49.45	47.73	47.47 (0.17)	47.10	198.65	194.18	193.38	193.38 (0.68)	193.38		
8	30	60.98	53.64	51.70	50.94 (1.25)	50.08	241.08	237.87	237.35	237.35 (5.93)	237.35		
8	40	74.54	61.47	58.86	58.66 (6.48)	58.40	295.68	289.94	288.21	288.05 (40.48)	288.01		
10	10	36.55	34.37	34.36	34.24 (0.01)	34.24	108.03	107.59	107.59	107.59 (0.02)	107.59		
10	20	48.25	39.02	37.27	36.65 (0.13)	36.23	180.12	178.76	178.65	178.65 (0.51)	178.65		
10	30	55.92	49.40	45.53	44.98 (0.93)	44.09	243.19	236.40	234.80	234.80 (4.74)	234.80		
10	40	60.70	52.13	49.58	49.12 (5.39)	48.29	278.16	271.45	271.04	271.04 (40.94)	270.70		

Fig. 9. Hillclimbing with Early Rollouts

first few rounds of the SSI auction and using standard hillclimbing in all later rounds. Rollouts can be expected to have larger effects when they are performed in early

rounds rather than later rounds since the team costs of partial target assignments are less predictive of the team costs of the complete target assignments the farther away

the partial target assignments are from being completed.

B. Experimental Evaluation

The tables in Figures 7 and 8 show that hillclimbing with rollouts or simplified rollouts reduces the team costs substantially over standard hillclimbing and hillclimbing with lookaheads two and three. Hillclimbing with rollouts even reaches the minimal team costs for two robots and ten targets (almost) but its runtimes are larger than the runtimes of hillclimbing with lookahead three. The table in Figure 9 contains experimental results for hillclimbing with early rollouts. The column “No Round” is identical to standard hillclimbing, and the column “All Rounds” is identical to hillclimbing with rollouts. Hillclimbing with a smaller number of early rollouts cannot result in smaller team costs than hillclimbing with a larger number of early rollouts for the same reason why standard hillclimbing cannot result in smaller team costs than hillclimbing with rollouts. Hillclimbing with rollouts in only the first round already reduces the team costs substantially over standard hillclimbing. Hillclimbing with rollouts in only the first three rounds achieves team costs that are almost identical to the ones of hillclimbing with rollouts in all rounds, for both team objectives. The runtimes of hillclimbing with lookahead three, hillclimbing with simplified rollouts and hillclimbing with rollouts in only the first three rounds are comparable but the team costs of hillclimbing with rollouts in only the first three rounds are smaller than the ones of the other two versions. For 2 robots and 10 targets, hillclimbing with rollouts in only the first three rounds achieves the minimal team costs within 0.6 percent for both team objectives. For 10 robots and 40 targets, it improves the team cost of standard hillclimbing by about 19 percent for the MiniMax team objective and by about 2 percent for the MiniSum team objective, despite the margins for improvement being rather small. It is NP-hard to minimize the team cost for both team objectives [5], and the team cost of standard hillclimbing has been reported to be roughly within 10 percent of minimal for the MiniSum team objective and within 50 percent of minimal for the MiniMax team objective [4]. Overall, it was surprising to us that rollouts improved standard hillclimbing much more than larger lookaheads. While we expected rollouts to have larger effects when they are performed in early rounds, it was also surprising to us that one needs to perform rollouts only for the first few rounds because additional rollouts in later rounds improve hillclimbing with rollouts only marginally.

V. CONCLUSIONS

Sequential single-item auctions (SSI auctions), which sequentially allocate targets to robots, require less computing resources but yield poorer target assignments than combinatorial auctions. In this paper, we have investigated techniques for improving SSI auctions, in the spirit of [9], although our techniques do this by improving the evaluation of partial target assignments. We developed a method to implement lookahead efficiently in SSI auctions, so that the computational and communication burden still compares favorably with combinatorial

auctions. Specifically, the overall amount of computation by each robot in SSI auctions that implement hillclimbing with lookahead k is similar, in the worst case, to the amount of computation by each robot in case of combinatorial auctions where each robot bids only on sets of at most k targets. In practice, SSI auctions should require substantially less computation because branch-and-bound usually prunes much of an enumeration tree. Moreover, SSI auctions require both fewer submitted bids and thus less overall communication and much less computation to determine the winning robots. We also developed roll-outs for SSI auctions to evaluate partial assignments more accurately. We described the bidding and winner-determination rules of the resulting SSI auctions and evaluated them experimentally, with surprising results: Larger lookaheads do not improve SSI auctions reliably while only a small number of roll-outs in early rounds already improve them substantially. All robots can formulate their bids and run the winner-determination rule in parallel, but it remains future work to truly distribute the determination of the winning robots, which also includes synchronizing the auctions and making them robust in the face of communication errors and malfunctioning robots.

ACKNOWLEDGMENT

We thank Brad Clement for helpful discussions. This research was partially supported by seed funding from NASA’s Jet Propulsion Laboratory as well as NSF awards under contracts IIS-0413196 and IIS-0412912.

REFERENCES

- [1] M. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” Robotics Institute, Carnegie Mellon University, Pittsburgh (Pennsylvania), Tech. Rep. CMU-RI-TR-05-13, 2005.
- [2] B. Gerkey and M. Mataric, “Sold!: Auction methods for multi-robot coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [3] S. Sarel and T. Balch, “Efficient bids on task allocation for multi robot exploration,” in *Proceedings of the International FLAIRS Conference*, 2006, p. (to appear).
- [4] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig, “The generation of bidding rules for auction-based robot coordination,” in *Multi-Robot Systems: From Swarms to Intelligent Automata*, L. Parker, F. Schneider, and A. Schultz, Eds. Springer, 2005, pp. 3–14.
- [5] 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.
- [6] 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.
- [7] T. Sandholm, “Negotiation among self-interested computationally limited agents,” Ph.D. dissertation, Department of Computer Science, University of Massachusetts, Amherst (Massachusetts), 1996.
- [8] R. Sutton, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [9] M. B. Dias and A. Stentz, “Opportunistic optimization for market-based multirobot control,” in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2002.