Negotiation with Reaction Functions for Solving Complex Task Allocation Problems

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

***Abstract*— We study task-allocation problems where cooperative robots need to perform tasks simultaneously. We develop a distributed negotiation procedure that allows robots to find all task exchanges that reduce the team cost of a given task allocation, without robots having to know how other robots compute their robot costs. Finally, we demonstrate empirically that our negotiation procedure can substantially reduce the team costs of task allocations resulting from existing task-allocation procedures, including sequential single-item auctions.**

I. Introduction

Task allocation is one of the most important coordination problems for robot teams [4]. We study task allocation where robots collaborate to minimize the team cost rather than their own robot costs. Most research on task allocation considers only simple tasks, which can be performed by single robots [3] [5] [9]. However, one of the main advantages of robot teams is that they can perform tasks that single robots cannot. We therefore consider also complex tasks, which need to be performed by several robots simultaneously [10] [12]. For instance, several robots need to move heavy rocks together, and several fire engines need to extinguish large fires together. Our motivating problem is multi-robot routing, where the tasks are to visit targets in the plane, as shown in Figure 1. The terrain, the locations of all robots and the locations of all targets are known. One needs to determine which targets each robot should visit and when it should visit them so that the team cost (such as the amount of energy or the task-completion time) is as small as possible. Multi-robot routing is a standard task for robot teams, for example, as part of de-mining, search-and-rescue and taking rock probes on the moon. Multi-robot routing with simple tasks is a standard test domain for robot coordination with auctions [2] [9]. Multi-robot routing with complex tasks is more difficult. First, it is difficult to determine which robots should perform a complex task because each complex task has to be assigned to more than one robot. Second, it is difficult to determine when a group of robots should perform a complex task because this requires the robots to solve complex scheduling problems.

We use reaction functions to characterize the robot costs of a given robot for performing a given complex task at any possible time. Reaction functions have been used previously to allow a central planner to determine a task allocation with a small team cost [11]. We, on the other hand, use reaction functions to allow robots to reduce the team cost of a given task allocation by exchanging tasks. Our initial investigation concentrates on disjoint coalitions, where every robot can perform at most one complex task [6] [8]. We proceed as follows: We first review the concepts and properties of reaction functions proposed in the literature. We then develop a distributed negotiation procedure (without a central planner) that allows robots to find all task exchanges that reduce the team cost of a given task allocation. Our negotiation procedure has the advantage that each robot needs to know the reaction functions of the other robots only for the complex targets assigned to them and that no robot needs to know how the reaction functions of the other robots are computed, including how their robot costs are computed. Finally, we demonstrate empirically that our negotiation procedure can substantially reduce the team costs of task allocations resulting from existing task-allocation procedures, including sequential single-item auctions [9].

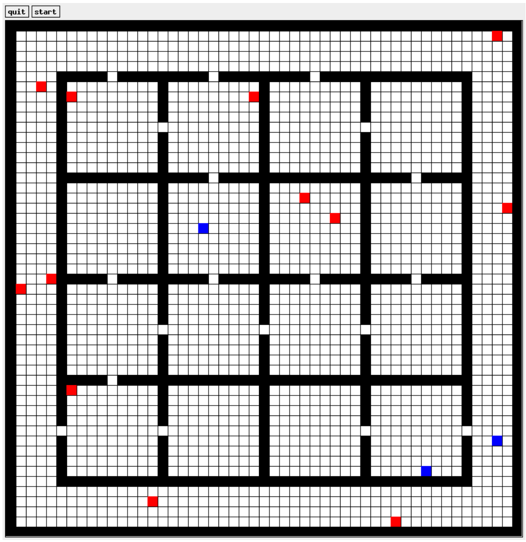

Fig. 1. Multi-Robot Routing Problem

II. Formalization of Multi-Robot Routing

We now formalize multi-robot routing: The finite set of robots is A. The finite set of targets is X. The number of robots that need to visit a target $x \in X$ simultaneously is

$d(x)$. A target $x \in X$ is **simple** if $d(x) = 1$ and **complex** otherwise. We distinguish these two kinds of targets because a robot can freely determine when to visit simple targets but needs to agree with other robots when to visit complex targets. Each robot in the group of $d(x)$ robots that need to visit complex target x at some visit time $0 \leq t < \infty$ has a **commitment**, written as $x \leftarrow t$. An **allocation** of robot r consists of a pair (X_r, C_r), where X_r is the set of simple targets assigned to it and C_r is a set that is either empty or contains the commitment for one complex target. A robot r is **eligible** iff C_r is empty. The **robot cost** $c_r^{robot}(X_r, C_r)$ of robot r is the minimal sum of travel and wait time that it needs to visit all of the targets assigned to it, where it can freely determine when to visit each simple target in X_r subject to the restriction that it has to visit its complex target (if any) at the agreed-on visit time recorded in C_r. (The robot cost is infinity in case the robot cannot satisfy this restriction.) Our objective is to find a solution with a small team cost, where a **solution** requires each target $x \in X$ to be assigned to exactly $d(x)$ robots, each complex target to be assigned a visit time, and each robot to be assigned at most one complex target. In this paper, we consider two ways of defining the team cost. The **team cost** is $\sum_{r \in A} c_r^{robot}(X_r, C_r)$ (roughly proportional to the energy needed by the robots for waiting and moving) for the **MiniSum team objective** and $\max_{r \in A} c_r^{robot}(X_r, C_r)$ (the task-completion time) for the **MiniMax team objective**. We use c^{team} as a special operator for either the sum or max operator, depending on the team objective, and write the team cost as $c_{r \in A}^{team} c_r^{robot}(X_r, C_r)$ to make our notation independent of the team objective.

III. Reaction Functions

To determine the optimal visit time for complex target x (that minimizes the team cost), each eligible robot r computes its **reaction function**

$$\mathcal{F}_r^x(t) := c_r^{robot}(X_r, \{x \leftarrow t\})$$

to characterize its robot costs for visiting complex target x at any possible visit time t in addition to all simple targets in X_r at the optimal visit times. The optimal visit time of complex target x for a given group P_x of eligible robots with given assigned simple targets then is

$$\arg \min_{0 \leq t < \infty} c_{r \in P_x}^{team} \mathcal{F}_r^x(t).$$

A. Approximation

The computation and communication of reaction functions is time-intensive. For example, each robot r has to solve a difficult scheduling problem for each visit time t of complex target x to determine its reaction function $\mathcal{F}_r^x(t)$ because it needs to determine the optimal order in which to visit all targets assigned to it. The computation and communication of reaction functions can be made less time-intensive by approximating them. We discretize them into a constant number of line segments, where each line segment is either linear with slope one (modeling that the robot waits at a complex target for other robots to visit the target simultaneously) or constant at infinity (modeling that the robot cannot visit the complex target at the given visit time), as follows:

- Determine a time interval $(s, e]$ during which robot r can visit complex target x and divide it evenly into k time intervals $(s_i, e_i]$ for a given parameter k.
- Determine the minimal robot cost of robot r for visiting complex target x in time interval $(s_i, e_i]$ without waiting as well as all simple targets assigned to it at the optimal visit times[1] for each $0 \leq i < k$. Assume that robot r visits complex target x at visit time $t_i \in (s_i, e_i]$ for a minimal robot cost of c_i. Then define the following function that calculates the robot cost if all targets are visited in the given order and the robot waits $t - t_i$ time units at the complex target for other robots to visit the target simultaneously:

$$\mathcal{F}_{r,i}^x(t) := \begin{cases} \infty & \text{if } 0 \leq t < t_i \\ c_i + t - t_i & \text{if } t_i \leq t. \end{cases}$$

- Determine the approximate reaction function as the minimum of the functions $\mathcal{F}_{r,i}^x$ for all $0 \leq i < k$ since each function expresses the robot cost if robot r visits its targets in a particular order:

$$\mathcal{F}_r^x(t) := \min_{0 \leq i \leq k} \mathcal{F}_{r,i}^x(t).$$

Let $T(P_x, x)$ be the set of times that correspond to the beginnings of all linear segments with slope one of the approximate reaction functions $\mathcal{F}_r^x(t)$ for all robots $r \in P_x$, where P_x is the group of robots that are assigned complex target x. Then, it holds that

$$\min_{0 \leq t < \infty} c_{r \in P_x}^{team} \mathcal{F}_r^x(t) = \min_{t \in T(P_x, x)} c_{r \in P_x}^{team} \mathcal{F}_r^x(t)$$

[11], which makes it easy to calculate the optimal visit time of complex target x for a given group P_x of eligible robots. In the following, all reaction functions are approximated unless mentioned otherwise.

B. Target Allocation with Reaction Functions

The simple targets need to be allocated before the complex ones because robots can manipulate the order in which they visit their assigned simple targets to accommodate the complex ones. We use two ways of assigning the simple targets to robots.

- **Random Allocation**: Random allocation assigns each simple target randomly to some robot.
- **SSI Auctions**: Sequential single-item auctions [9] assign simple targets to robots in rounds. During each round, one additional simple target is assigned to some

[1] This problem is a special case of the NP-hard traveling salesperson problem with time windows [1] and can be solved approximately with a version of the Or-opt heuristic [7].

robot so that the team cost after assigning that simple target increases the least (hill-climbing).

Afterwards, we assign the complex targets to robots in rounds until all complex targets are assigned to robots. During each round, one additional complex target is assigned to some robot. Let X_r be the set of simple targets assigned to robot r. Each eligible robot r then computes its reaction function $\mathcal{F}_r^x(t)$ for each complex target x and submits

$$\mathcal{V}_r^x(t) := \begin{cases} \mathcal{F}_r^x(t) - c_r^{robot}(X_r, \emptyset) & \text{for MiniSum} \\ \mathcal{F}_r^x(t) & \text{for MiniMax} \end{cases}$$

to a central planner. Let $P(n)$ be the set of all groups of n eligible robots and X_c the set of unassigned complex targets. The central planner determines

$$(P_x, x, t) := \arg \min_{P_x \in P(d(x)), x \in X_c, 0 \le t < \infty} c_{r \in P_x}^{team} \mathcal{V}_r^x(t)$$

and assigns the commitment $x \leftarrow t$ to each robot $r \in P_x$, which terminates the current round [11].

IV. Negotiation with Reaction Functions

Given a solution of a multi-robot routing problem, we exchange targets between two robots so that the team cost of the solution is reduced. We consider two types of target exchanges.

- **Complex target exchanges**: A complex target exchange (r, r', x, x') describes that robot r gives its complex target x to robot r' and robot r' gives its complex target x' to robot r. One of the complex targets can be empty but not both. The number of possible complex target exchanges is bounded by $|A|^2 - |A|$ since each robot is assigned at most one complex target.
- **Simple target exchanges**: A simple target exchange (r, r', X, X') describes that robot r gives its simple targets $X \subseteq X_r$ to robot r' and robot r' gives its simple targets $X' \subseteq X_{r'}$ to robot r. One of the sets of simple targets can be empty but not both. The number of simple target exchanges can be exponential in the number of simple targets. We therefore impose the restriction that $\max(|X|, |X'|) \le K$ for a given constant $K \ge 0$, the **exchange parameter**.

The **gain** $gain(S)$ of a target exchange S is the decrease in team cost that results from performing the target exchange. A target exchange is **profitable** iff its gain is positive.

A. Negotiation Procedure

We now develop a distributed negotiation procedure that allows robots to find all profitable target exchanges. Our negotiation procedure has the advantage that each robot needs to know the reaction functions of the other robots only for the complex targets assigned to them and that no robot needs to know how the reaction functions of the other robots are computed, including how their robot costs are computed. The negotiation procedure consists of three steps.

- **Initialization Step**: Each robot broadcasts the necessary information, including its assigned simple targets, its assigned complex target and its reaction function for its complex target (if any), its robot cost and its index number. The purpose of the index numbers is to order all robots completely.
- **Computation Step**: In the first substep, each robot acts as a **proposer**. It considers each possible target exchange that it can be involved in and, iff the target exchange is potentially profitable, proposes it to the other robot involved in it. In the second substep, each robot acts as a **manager**. It calculates the gain for each target exchange that it receives and stores it iff it is profitable. After the computation step, each profitable target exchange has been stored by at least one robot.
- **Decision Step**: Each robot broadcasts its target exchange with the highest gain. The robots then perform the broadcast target exchange with the highest gain. Ties are broken in favor of the target exchange that involves the robot with the smallest index number. After the decision step, the robots have performed a target exchange with the overall highest gain.

In the following, let $index(r)$ be the index number of each robot r, (X_r, C_r) its current allocation and x_r the complex target assigned to it. The complex target can be empty. Let P_x be the group of robots that are assigned complex target x. Finally, let c_r be the robot cost of robot r and $c := c_{\tilde{r} \in A}^{team} c_{\tilde{r}}$ the team cost of the current solution.

B. Complex Target Exchanges

We first consider complex target exchanges and describe the procedures executed by each robot in the computation step as proposer and manager.

1. Proposer Procedure

If proposer robot r is assigned no complex target, then it does nothing. Otherwise, it executes the following procedure for each robot r'.

Case 1: If robot r' is assigned a complex target $x_{r'}$ that is not assigned to robot r, then robot r considers the complex target exchange $S := (r, r', x_r, x_{r'})$. Let $A' := A \setminus (P_{x_r} \cup P_{x_{r'}})$. Let $P'_{x_{r'}} := P_{x_{r'}} \setminus \{r'\} \cup \{r\}$ be the group of robots that are assigned complex target $x_{r'}$ after the complex target exchange. Robot r calculates its **net loss** $netloss(S, r)$ of the complex target exchange as

$$\begin{cases} \min\limits_{0 \le t < \infty} \sum\limits_{\tilde{r} \in P'_{x_{r'}}} \mathcal{F}_{\tilde{r}}^{x_{r'}}(t) - \sum\limits_{\tilde{r} \in P'_{x_{r'}}} c_{\tilde{r}} & \text{for MiniSum} \\ \max(\max\limits_{\tilde{r} \in A'} c_{\tilde{r}}, \min\limits_{0 \le t < \infty} \max\limits_{\tilde{r} \in P'_{x_{r'}}} \mathcal{F}_{\tilde{r}}^{x_{r'}}(t)) - c & \text{for MiniMax.} \end{cases}$$

Case 2: If robot r' is assigned no complex target, then robot r considers the complex target exchange $S := (r, r', x_r, \emptyset)$. Let $A' := A \setminus (P_{x_r} \cup \{r'\})$. Robot r calculates its **net loss** $netloss(S, r)$ of the complex target exchange as

$$\begin{cases} c_r^{robot}(X_r, \emptyset) - c_r & \text{for MiniSum} \\ \max(\max\limits_{\tilde{r} \in A'} c_{\tilde{r}}, c_r^{robot}(X_r, \emptyset)) - c & \text{for MiniMax.} \end{cases}$$

If its net loss of the complex target exchange is negative, then robot r proposes it to robot r' by sending it the information $\langle S, netloss(S, r)\rangle$.

2. Manager Procedure

If manager robot r' receives a proposal for a complex target exchange $S := (r, r', x_r, x_{r'})$, then let $A' := A \setminus (P_{x_r} \cup P_{x_{r'}})$ if robot r' is assigned a complex target $x_{r'}$ and $A' := A \setminus (P_{x_r} \cup \{r'\})$ if robot r' is assigned no complex target. Let $P'_{x_r} := P_{x_r} \setminus \{r\} \cup \{r'\}$ be the group of robots that are assigned complex target x_r after the complex target exchange. Robot r' calculates its **net loss** $netloss(S, r')$ of the complex target exchange as

$$\begin{cases} \min\limits_{0\le t<\infty} \sum\limits_{\tilde{r}\in P'_{x_r}} \mathcal{F}_{\tilde{r}}^{x_r}(t) - \sum\limits_{\tilde{r}\in P'_{x_r}} c_{\tilde{r}} & \text{for MiniSum} \\ \max(\max\limits_{\tilde{r}\in A'} c_{\tilde{r}}, \min\limits_{0\le t<\infty} \max\limits_{\tilde{r}\in P'_{x_r}} \mathcal{F}_{\tilde{r}}^{x_r}(t)) - c & \text{for MiniMax.} \end{cases}$$

It is easy to show that

$$gain(S) = -c^{team}(netloss(S, r), netloss(S, r')).$$

Proposition 1: Each profitable complex target exchange is stored by at least one robot.

C. Simple Target Exchanges

We now consider simple target exchanges and describe the procedures executed by each robot in the computation step as proposer and manager.

1. Proposer Procedure

Proposer robot r considers the simple target exchange $S := (r, r', X, X')$ for each robot r' with $r \neq r'$, $X \subseteq X_r$, $X' \subseteq X_{r'}$, $X \cap X' = \emptyset$ and $0 < \max(|X|, |X'|) \le K$. Let $X'_r = X_r \setminus X \cup X'$ be the set of simple targets assigned to robot r after the simple target exchange. Robot r then executes the following procedure.

Case 1: If robot r is assigned no complex target, then let $A' := A \setminus (\{r\} \cup P_{x_{r'}})$ if robot r' is assigned a complex target $x_{r'}$ and $A' := A \setminus (\{r, r'\})$ if robot r' is assigned no complex target. Robot r calculates its **net loss** $netloss(S, r)$ of the simple target exchange as

$$\begin{cases} c_r^{robot}(X'_r, \emptyset) - c_r & \text{for MiniSum} \\ \max(\max\limits_{\tilde{r}\in A'} c_{\tilde{r}}, c_r^{robot}(X'_r, \emptyset)) - c & \text{for MiniMax.} \end{cases} \quad (1)$$

If the net loss of the simple target exchange is negative, then robot r proposes it to robot r' by sending it the information $\langle S, netloss(S, r)\rangle$.

Case 2: If robot r is assigned a complex target x_r that is not assigned to robot r', then robot r recomputes its reaction function for its complex target as

$$\mathcal{F'}_r^{x_r}(t) := c_r^{robot}(X'_r, x_r \leftarrow t). \quad (2)$$

This recomputation is necessary since the reaction functions of a robot depend on the simple targets assigned to it and can thus change after simple target exchanges. Define $\mathcal{F'}_{\tilde{r}}^{x_r}(t) := \mathcal{F}_{\tilde{r}}^{x_r}(t)$ for all robots $\tilde{r} \in P_{x_r} \setminus \{r\}$. Let $A' := A \setminus (P_{x_r} \cup P_{x_{r'}})$ if robot r' is assigned a complex target $x_{r'}$ and $A' := A \setminus (P_{x_r} \cup \{r'\})$ if robot r' is assigned no complex target. Robot r calculates its **net loss** $netloss(S, r)$ of the simple target exchange as

$$\begin{cases} \min\limits_{0\le t<\infty} \sum\limits_{\tilde{r}\in P_{x_r}} \mathcal{F'}_{\tilde{r}}^{x_r}(t) - \sum\limits_{\tilde{r}\in P_{x_r}} c_{\tilde{r}} & \text{for MiniSum} \\ \max(\max\limits_{\tilde{r}\in A'} c_{\tilde{r}}, \min\limits_{0\le t<\infty} \max\limits_{\tilde{r}\in P_{x_r}} \mathcal{F'}_{\tilde{r}}^{x_r}(t)) - c & \text{for MiniMax.} \end{cases} \quad (3)$$

If the net loss of the simple target exchange is negative, then robot r proposes it to robot r' by sending it the information $\langle S, netloss(S, r)\rangle$.

Case 3: If robot r is assigned the same complex target as robot r' but has a smaller index number than robot r', then robot r recomputes its reaction function for complex target x_r with Formula (2) and proposes the simple target exchange to robot r' by sending it the information $\langle S, \mathcal{F'}_r^{x_r}(t)\rangle$.

2. Manager Procedure

If manager robot r' receives a proposal for a simple target exchange $S := (r, r', X, X')$, then let $X'_{r'} = X_{r'} \setminus X' \cup X$ be the set of simple targets of robot r' after the simple target exchange. Robot r then executes the following procedure.

Case 1: If robot r' is assigned no complex target, then it calculates its **net loss** $netloss(S, r')$ of the simple target exchange with Formula (1). It easy easy to show that

$$gain(S) = -c^{team}(netloss(S, r), netloss(S, r')).$$

Case 2: If robot r' is assigned a complex target $x_{r'}$ that is not assigned to robot r, then robot r' recomputes its reaction function $\mathcal{F'}_{r'}^{x_{r'}}(t)$ for complex target x_r with Formula (2) and then its **net loss** $netloss(S, r')$ of the simple target exchange with Formula (3). It is easy to show that

$$gain(S) = -c^{team}(netloss(S, r), netloss(S, r')).$$

Case 3: If robot r' is assigned the same complex target as robot r, then robot r' recomputes its reaction function $\mathcal{F'}_{r'}^{x_r}(t)$ for complex target $x_{r'} = x_r$ with Formula (2). Define $\mathcal{F'}_{\tilde{r}}^{x_r}(t) := \mathcal{F}_{\tilde{r}}^{x_r}(t)$ for all robots $\tilde{r} \in P_{x_r} \setminus \{r, r'\}$. Let $A' := A \setminus P_{x_r}$. It is easy to show that $gain(S)$ equals the value calculated with Formula (3).

Proposition 2: Each profitable simple target exchange for exchange parameter K is stored by at least one robot.

V. Experimental Results

We now evaluate the benefits of our negotiation procedure for multi-robot routing problems on known four-neighbor planar grids of size 51×51 with square cells that are either

Robots	Simple Targets	Complex Targets	Initial Cost	K = 0 Cost	K = 0 Cost Reduction	K = 1 Cost	K = 1 Cost Reduction	K = 2 Cost	K = 2 Cost Reduction
MiniSum Team Objective - Initial Solutions Generated with Random Allocation									
4	8	2	566.1	546.0	3.55%	355.7	37.17%	342.2	39.55%
4	18	2	740.6	721.9	2.52%	469.3	36.63%	439.9	40.60%
4	28	2	901.6	882.7	2.10%	552.2	38.75%	511.2	43.30%
6	7	3	618.9	576.3	6.88%	390.8	36.86%	384.5	37.87%
6	17	3	924.5	888.5	3.89%	520.6	43.69%	485.6	47.47%
6	27	3	1150.2	1116.9	2.90%	618.1	46.26%	570.8	50.37%
8	6	4	634.5	585.9	7.66%	428.0	32.55%	423.6	33.24%
8	16	4	1041.2	988.8	5.03%	560.2	46.20%	527.1	49.38%
8	26	4	1352.7	1305.3	3.50%	663.8	50.93%	607.2	55.11%
10	5	5	624.7	579.6	7.22%	443.7	28.97%	439.8	29.60%
10	15	5	1106.8	1044.4	5.64%	590.8	46.62%	563.5	49.09%
10	25	5	1414.6	1345.6	4.88%	695.6	50.83%	654.6	53.73%
MiniSum Team Objective - Initial Solutions Generated with SSI Auctions									
4	8	2	362.4	346.3	4.44%	332.0	8.39%	327.6	9.60%
4	18	2	452.7	437.9	3.27%	418.3	7.60%	412.6	8.86%
4	28	2	519.0	500.3	3.60%	478.2	7.86%	N/A	N/A
6	7	3	399.7	378.9	5.20%	366.8	8.23%	364.8	8.73%
6	17	3	501.5	470.7	6.14%	445.8	11.11%	440.4	12.18%
6	27	3	571.3	532.5	6.79%	504.7	11.66%	498.5	12.74%
8	6	4	435.5	414.8	4.75%	401.6	7.78%	399.5	8.27%
8	16	4	534.4	502.9	5.89%	484.0	9.43%	478.9	10.39%
8	26	4	602.5	563.8	6.42%	537.5	10.79%	529.9	12.05%
10	5	5	459.6	435.6	5.22%	428.4	6.79%	427.5	6.98%
10	15	5	550.1	514.9	6.40%	497.5	9.56%	493.0	10.38%
10	25	5	627.4	586.5	6.52%	562.8	10.30%	554.4	11.64%
MiniMax Team Objective - Initial Solutions Generated with Random Allocation									
4	8	2	199.4	180.1	9.68%	120.3	39.67%	116.0	41.83%
4	18	2	238.6	220.9	7.42%	147.0	38.39%	140.9	40.95%
4	28	2	275.7	259.1	6.02%	174.1	36.85%	159.1	42.29%
6	7	3	203.1	171.8	15.41%	96.1	52.68%	94.8	53.32%
6	17	3	233.8	207.9	11.08%	128.9	44.87%	121.2	48.16%
6	27	3	251.6	226.2	10.10%	151.9	39.63%	142.4	43.40%
8	6	4	170.7	137.8	19.27%	85.9	49.68%	83.9	50.85%
8	16	4	226.5	194.7	14.04%	117.0	48.34%	112.6	50.29%
8	26	4	253.4	221.8	12.47%	139.4	44.99%	128.1	49.45%
10	5	5	152.4	120.6	20.87%	76.5	49.80%	76.6	49.74%
10	15	5	216.1	184.6	14.58%	107.9	50.07%	105.3	51.27%
10	25	5	244.7	210.2	14.10%	132.7	45.77%	126.1	48.47%
MiniMax Team Objective - Initial Solutions Generated with SSI Auctions									
4	8	2	128.0	117.5	8.20%	110.6	13.59%	109.2	14.69%
4	18	2	155.9	142.0	8.92%	130.8	16.10%	127.6	18.15%
4	28	2	173.0	158.7	8.27%	146.0	15.61%	143.4	17.11%
6	7	3	107.8	90.7	15.86%	85.4	20.78%	84.8	21.34%
6	17	3	126.4	108.6	14.08%	100.6	20.41%	99.2	21.52%
6	27	3	141.3	120.6	14.65%	111.5	21.09%	110.0	22.15%
8	6	4	100.9	79.2	21.51%	76.3	24.38%	76.2	24.48%
8	16	4	117.7	95.9	18.52%	90.5	23.11%	89.1	24.30%
8	26	4	128.3	104.0	18.94%	96.0	25.18%	94.6	26.27%
10	5	5	93.2	69.0	25.97%	67.0	28.11%	67.0	28.11%
10	15	5	110.5	85.0	23.08%	80.2	27.42%	79.3	28.24%
10	25	5	118.3	92.0	22.23%	84.9	28.23%	84.0	28.99%

TABLE I

EXPERIMENTAL RESULTS (N/A MEANS THAT THE RUNTIME THRESHOLD WAS EXCEEDED) blocked or unblocked. The grids resemble office environments with randomly closed doors, as shown in Figure 1. All complex targets need to be assigned to groups of two robots. Their number is always half the number of robots, so that every robot visits exactly one complex target. We iteratively apply our negotiation procedure until it no longer reduces the team cost of the current solution. We vary the number of robots from 4, 6, 8, to 10, the number of (simple and complex) targets from 10, 20 to 30, and the exchange parameter K from 0, 1 to 2. For each scenario, we average over 100 samples with randomly chosen cells for the robots and targets. Each robot needs to solve a version of the NP-hard traveling salesperson problem with time windows to calculate its robot cost. We use a version of the Or-opt heuristic [7] in our experiments to approximate this calculation. Table I tabulates the team costs of the initial solutions ("Initial Cost") generated as described in Section "Target Allocation with Reaction Functions" as well as the team costs ("Cost") and the cost reductions over the initial solutions in percent ("Cost Reduction"). The data show that our negotiation procedure can reduce the team costs of the initial solutions significantly. For example, it reduces the team costs of the initial solutions generated with Random Allocation by as much as 55 percent for the MiniSum team objective and 53 percent for the MiniMax team objective. It reduces the team costs of the initial solutions generated with SSI Auctions by as much as 12 percent for the MiniSum team objective and 29 percent for the MiniMax team objective.

VI. Conclusions

We studied task-allocation problems where cooperative robots need to perform tasks simultaneously. We developed a distributed negotiation procedure that allows robots to find all task exchanges that reduce the team cost of a given task allocation, and demonstrated empirically that our negotiation procedure can substantially reduce the team costs of task allocations resulting from existing task-allocation procedures, including sequential single-item auctions. It is future work to extend our results from disjoint coalitions, where every robot can perform at most one complex task, to overlapping coalitions, where some robots can perform more than one task.

VII. Acknowledgments

This research was partly supported by an award from ARO under contract W911NF-08-1-0468 and by awards from NSF under contract 0413196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.

References

[1] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Time constrained routing and scheduling. In *Network Routing. Handbooks in Operations Research and Management Science*, volume 8, chapter 2, pages 35–139. North-Holland, 1995.

[2] M. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2714–2720, 2002.

[3] M. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[4] B. P. Gerkey. *On Multi-Robot Task Allocation*. PhD thesis, Department of Computer Science, University of Southern California, Los Angeles, CA, 2003.

[5] S. Koenig, C. Tovey, X. Zheng, and I. Sungur. Sequential bundle-bid single-sale auction algorithms for decentralized control. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1359–1365, 2007.

[6] E. Manisterski, E. David, S. Kraus, and N. R. Jennings. Forming efficient agent groups for completing complex tasks. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 834–841, 2006.

[7] I. Or. *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. PhD thesis, Northwestern University, Evanston, IL, 1976.

[8] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 655–661, 1995.

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

[10] L. Vig and J. A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.

[11] X. Zheng and S. Koenig. Reaction functions for task allocation to cooperative agents. In *Proceedings of the International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 559–566, 2008.

[12] R. Zlot and A. Stentz. Market-based multirobot coordination for complex tasks. *The International Journal of Robotics Research*, 25(1):73–101, 2006.