An Efficient Stochastic Clustering Auction for Heterogeneous Robot Teams

Kai Zhang and Emmanuel G. Collins, Jr. and Adrian Barbu

***Abstract*— Stochastic Clustering Auctions (SCAs) constitute a class of cooperative auction methods that enable improvement of the global cost of the task allocations obtained with fast greedy algorithms. Prior research had developed Contracts Sequencing Algorithms (CSAs) that are deterministic and enable transfers, swaps, and other types of contracts between team members. In contrast to CSAs, SCAs use *stochastic* transfers or swaps between the task clusters assigned to each team member and have algorithm parameters that can enable tradeoffs between optimality and computational and communication requirements. The first SCA was based on a "Gibbs Sampler" and constrained the stochastic cluster reallocations to simple single transfers or swaps; it is applicable to heterogeneous teams. Subsequently, a more efficient SCA was developed, based on the generalized Swendsen-Wang method; it achieves the increased efficiency by connecting tasks that appear to be synergistic and then stochastically reassigning these connected tasks, hence enabling more complex and efficient movements between clusters than the first SCA. However, its application was limited to homogeneous teams. *The contribution of this work is to present an efficient SCA for heterogeneous teams; it is based on a modified Swendsen-Wang method.* For centralized auctioning and homogeneous teams, extensive numerical experiments were used to provide a comparison in terms of costs and computational and communication requirements of the three SCAs and a baseline CSA. It was seen that the new SCA maintains the efficiency of the second SCA and can yield similar performance to the baseline CSA in far fewer iterations. The same metrics were used to evaluate the performance of the new SCA for heterogeneous teams. A distributed version of the new SCA was also evaluated in numerical experiments. The results show that, as expected, the distributed SCA continually improves the global performance with each iteration, but converges to a lower cost solution than the centralized SCA.**

I. Introduction

Algorithms for effective coordination of heterogeneous robotic systems have numerous applications. For example, they can be used for efficient task allocation for teams of robots in many application domains [1], [2] such as collaborative manipulation, transportation, assembly and maintenance of large structures, collective surface exploration and mapping, disaster response, and planetary survey and habitat construction. A critical issue is how to assign tasks to heterogeneous robots in order to optimally or near-optimally complete a given mission. It is well known that this task allocation must not always rely on complete communication between the allocator and each of the robots. Hence, a practically meaningful solution must allow distributed task allocation.

Kai Zhang and Emmanuel G. Collins, Jr. are with Center for Intelligent Systems, Control and Robotics (CISCOR), Department of Mechanical Engineering, FAMU-FSU College of Engineering, Florida A&M University-Florida State University, Tallahassee, FL 32310, USA {zhangka, ecollins}@eng.fsu.edu

Adrian Barbu is with Department of Statistics, Florida State University, Tallahassee, FL 32306, USA abarbu@stat.fsu.edu

A. Stochastic Clustering Auctions

Auction methods are effective approaches for resource allocation. Stochastic Clustering Auctions (SCAs) [3],[4],[5] constitute a class of cooperative auction methods that enable improvement of the global cost of the task allocations obtained with fast greedy algorithms. An SCA is in the class of Markov Chain Monte Carlo methods [6] and optimizes using simulated annealing [7]. The first SCA [3], [4] is called here the Gibbs Sampler SCA (GSSCA) since the underlying optimization algorithm is a Gibbs Sampler, which has been successfully used for shape clustering and segmentation in computer vision [8]; GSSCA is applicable to heterogeneous robots. The second SCA is called the generalized Swendsen Wang SCA (SW^1SCA) [5] and is applicable only to homogeneous robots; however, for this class of robots it is far more efficient than GSSCA. The contribution of this work is to introduce a third SCA, called the modified Swendsen Wang SCA (SW^2SCA), that is applicable to heterogeneous robots and maintains the efficiency of SW^1SCA. Each SCA algorithm rearranges the clusters using transfer moves (with 50% probability) or swap moves (with 50% probability) and allows not only downhill movements, but also uphill movements, which provides it with some ability to avoid local minima. By tuning the annealing suite and turning the uphill movements on and off, the team performance obtained after algorithm convergence can slide in the region between the global optimal performance and the performance of a random allocation.

In SW^1SCA the agent serving as the auctioneer connects tasks that appear to be synergistic to form "connected components" and then trades these connected components instead of the single tasks traded by GSSCA. Since a set of tasks is connected based on costs computed by the auctioneer, the synergy of a connected set is only valid for each of the robots with which the auctioneer is negotiating if each robot is identical to the auctioneer. Hence, SW^1SCA is only applicable to homogeneous robot teams. In contrast, in SW^2SCA each set of connected tasks is computed by each robot, which enables SW^2SCA to be applied to heterogeneous robotic teams.

B. Deterministic Contract Algorithms

OCSM (Original, Cluster, Swaps, and Multiagent) contracts [9], [10] were developed to provide a general framework for combinatorial auctioning. This work designs deterministic algorithms, called here *Contracts Sequencing Algorithms* (*CSAs*), that are closely related to SCAs. In particular OCSM involves four types of contracts:

1) The *original* (*O*) *contract* enables one task to move from one robot to another robot. In SCA terminology this contract is called a *transfer* and is allowed by all SCAs.
2) The *cluster* (*C*) *contract* enables two or more tasks to move from one robot to another. C contracts were sequenced by trying all combinations of two tasks followed by all combinations of three tasks, and so on. SW^1SCA and SW^2SCA allow transfer of an arbitrary number of "connected" tasks (see Section II-B) and is hence a cluster contract.
3) The *swap* (*S*) *contract* enables two robots to exchange tasks such that each robot receives one task from the other. All SCAs allow this movement and it is called a *swap*.
4) The *multiagent* (*M*) *contracts* enable exactly three tasks to be transferred between exactly three robots. Current SCAs do not allow this type of movement.

A CSA has the structure shown in Algorithm 1.

Algorithm 1 Principal Mechanisms for a Contracts Sequencing Algorithm (CSA)

```
repeat
    Propose a reclustering based on the contract types in
    a deterministic sequence.
    Decide whether to accept the proposed cluster.
    if the solution is better then
        Accept.
    end if
until the deterministic sequence is finished
```

SW^1SCA and SW^2SCA enable the swap of multiple tasks, which is not allowed in the current OCSM framework. However, here using contractual language [9], [10], it will be called a *swap*$^+$ (S^+) *contract*. From the above discussion it follows that GSSCA is a stochastic algorithm built on OS contracts, while SW^1SCA and SW^2SCA are stochastic algorithms built on OCSS^+ contracts.

It should be noted that it has been proved [9] that using OCSM contracts an algorithm can be devised that leads to the optimum allocation in a finite number of steps, a powerful result that has not been proved for a SCA. This algorithm requires that a very large number of possible contracts be considered for each algorithm iteration. These contracts are combinations of O, C, S, and M contracts and are the largest number possible. If O, C, S, and M contracts are used at each iteration, then local optima can be encountered [9]. Unfortunately, the problem with the huge contract neighborhood of the provably convergent algorithm is the exponentially large cost of a single iteration. As a result, for larger problems reaching the global optimum can take an impractically long time. Hence, to develop auction algorithms that have more reasonable computational times, negotiations that consider only certain types of contracts were explored in [9]. The best of these is based on O and C contracts and will be called here OCCSA. It is used as a baseline in this paper.

C. Paper Organization

The remainder of this paper is organized as follows. Section II formulates the basic optimization problem for task allocation and provides a description of SW^2SCA. Section III first considers centralized auctioning for homogeneous teams and presents simulation results from random scenarios with an initial focus on comparing OCCSA, GSSCA, SW^1SCA, and SW^2SCA. Then, it considers heterogeneous teams with an emphasis on evaluating the performance of SW^2SCA. Section IV considers distributed auctioning and presents simulation results from random scenarios using communication links motivated by a generic topology called a "scale free network"; the focus is on demonstrating the performance achieved by SW^2SCA. Finally, Section V summarizes the results.

Table I summarizes the acronyms used in this paper.

TABLE I
ACRONYMS

CSA	Contracts Sequencing Algorithm
OCSM	Original, Cluster, Swaps, and Multiagent
OCCSA	Original and Cluster based CSA
SCA	Stochastic Clustering Auction
GSSCA	Gibbs Sampler SCA
SW^1SCA	Generalized Swendsen-Wang SCA
SW^2SCA	Modified Swendsen-Wang SCA
S^+	Swap contracts based on connected components
PA	Parallel Auction
LBPA	Look-Back Parallel Auction
SA	Sequential Auction
LBSA	Look-Back Sequential Auction
AC	Auction Cycle
MCI	Mean Cost Improvement
RTV	Robot Type Vector
TPI	Tournament Participation Index
SFN	Scale Free Network

II. Stochastic Clustering Auctions

This section first presents the basic problem statement. Then it presents the algorithm for SW^2SCA.

A. Task Allocation Problem Statement

Let $\mathcal{H}$ denote a set of k heterogeneous robots, and $\mathcal{T}$ denote a set of n tasks, i.e. $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$ and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$. The tasks are specifically "goto points," which can be sequenced in any order and can be performed by any robot. Also, heterogeneity does not necessarily imply that the robots perform different tasks. It implies that they do not perform each task with the same efficiency. For example, in the simulation results of Section III-B the robots move at different speeds, so they cannot reach a destination (i.e., perform a task) with equal efficiency, even when starting from the same spatial location.

Let $\mathcal{A}$ denote the allocation, $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, The cost associated with $\mathcal{A}$ is given by either

$$C(\mathcal{A}) = \sum_{s=1}^{k} c_s(a_s), \tag{1}$$

or

$$C(\mathcal{A}) = \max_{s} c_s(a_s), \tag{2}$$

where $c_s(a_s)$ is the minimum cost for robot h_s to complete the set of tasks a_s. The individual cost function $c_s(\cdot)$ is based on characteristics of each robot, e.g., the dynamic model of the robot, the state of the market, current task commitments and/or a human-inspired measure. The problem is to solve the optimization $\min\limits_{\mathcal{A}} C(\mathcal{A})$. In practice the cost function in Equation (1) might be used to represent the total distance traveled or the total energy expended by the robots while the cost function in Equation (2) might be used to represent the maximum time taken to accomplish the tasks.

B. The Modified Swendsen-Wang Stochastic Clustering Auction

As previously discussed, there are currently three variants of a SCA: the Gibbs Sampler SCA (GSSCA), the generalized Swendsen-Wang SCA (SW^{1}SCA), and the modified Swendsen-Wang SCA (SW^{2}SCA). Below, the SCAs are put in the framework of the auctioning algorithms (or contract approaches) developed for the CSAs [9], [10] and the general features of a SCA are described before detailing SW^{2}SCA.

Algorithm 2 Principal Mechanisms for the Non-greedy and Greedy Versions of a Stochastic Clustering Auction

```
1:  Initialize the annealing suite.
2:  repeat
3:     Randomly select 2 agents for reclustering.
4:     Stochastically propose a reclustering for the selected
       robots.
5:     Decide whether to accept the proposed cluster.
6:     if the solution is better (for greedy SCAs only) then
7:        Accept.
8:     else
9:        Accept with an acceptance probability.
10:    end if
11:    Update the annealing suite.
12: until termination is reached
```

As previously mentioned, a SCA is similar to the auction methods of the CSAs [9], [10], described by Algorithm 1. Algorithm 2 describes the generic structure of the non-greedy and greedy versions of a Stochastic Clustering Auction (SCA). A SCA is always guaranteed to result in an allocation that has a cost less than or equal to the cost of the initial allocation. The GSSCA, the SW^{1}SCA and the SW^{2}SCA each follow Algorithm 2. These algorithms primarily differ in line 4, where they propose reclustering. The proposals of GSSCA are based on treating tasks individually and hence involve simple transfer and swaps of individual tasks. In contrast, SW^{1}SCA and SW^{2}SCA are based on transfers and swaps of interconnected tasks. All SCA algorithms can be made greedy by not allowing the uphill movements associated with line 9, which enable the algorithm to escape local minima. The ability to initialize and update the annealing suite in lines 1 and 11 and turn the uphill movements on and off in lines 6 and 7 provides SCA with the ability to trade off the converged algorithm cost with computational and communication efficiency. This is a novel feature of a SCA.

Comparing Algorithms 1 and 2 reveals that CSA and SCA have similar features. However, unlike CSA, SCA makes choices stochastically. In particular, referring to Algorithm 2, SCA randomly chooses two robots for reclustering in line 3, it stochastically propose the reclustering in line 4, and probabilistically accepts a proposed reclustering in line 9. It follows that SCA is an auction method where the participants are chosen randomly, there is a certain randomness in the choice of tasks to be bid, and the auctioneer may choose to make choices stochastically, not always greedily.

It was shown in a prior research [5] that for homogeneous teams SW^{1}SCA can achieve much greater performance than GSSCA in fewer iterations. Unfortunately, as discussed in Section I, SW^{1}SCA is not suited for heterogeneous teams since an auctioneer connects the tasks and has to be homogeneous with the negotiating robots. The novelty of this paper is the presentation of a modified Swendsen-Wang SCA for heterogeneous robotic teams in which each robot decomposes the adjacency graph for connected components in Step 6 of SW^{2}SCA, given below.

Modified Swendsen-Wang Auction[2] (SW^{2}SCA)

1) The auctioneer partitions $\mathcal{T}$ into k clusters to form an initial allocation $\mathcal{A}^{(0)} = \{a_1^{(0)}, a_2^{(0)}, \ldots, a_k^{(0)}\}$, where each cluster $a_s^{(0)}$ is an unordered subset of $\mathcal{T}$. Let $\mathcal{A} = \mathcal{A}^{(0)}$ and $\mathcal{A}^* = \mathcal{A}^{(0)}$. ($\mathcal{A}$ is the current algorithm allocation, while $\mathcal{A}^*$ is the allocation that has the lowest cost.)
2) Each robot $h_p \in \mathcal{H}$ $(p = 1, 2, \ldots, k)$ uses a "constrained Prim's Algorithm"[1] (a greedy algorithm) to efficiently approximate the cost $c_p(a_p)$ and submits its cost to the auctioneer. In this bid valuation stage, each cluster a_p becomes an ordered subset of $\mathcal{T}$. The auctioneer computes the global cost $C(\mathcal{A})$ using Equation (1) or Equation (2) and sets the temperature T to a high initial value T_0.
3) The auctioneer randomly selects two robots h_s and h_t for either transfers or swaps and lets $\mathcal{H} = \{h_s, h_t\}$.
4) Each robot $h_p \in \mathcal{H}$ constructs an adjacency graph $G(a_p) = \langle \mathcal{T}(a_p), \mathcal{E}(a_p) \rangle$ for each cluster a_p, where $\mathcal{E}(a_p)$ is the edge set of $\mathcal{T}(a_p)$.
5) Each robot $h_p \in \mathcal{H}$ submits $l_{\min}(a_p)$ to the auctioneer, where $l_{\min}(a_p) = \min\limits_{e \in \mathcal{E}(a_p)} l(e)$, and $l(e)$ denotes the Euclidean length of the edge e.

[1]This algorithm fixes the initial vertex with a single edge in Prim's Algorithm [11], [12] as building a minimum spanning tree, and hence, unlike Prim's algorithm, is not guaranteed to be optimal. It is well known to be 2-approximate (i.e., the cost of the resulting allocation is at most twice the total cost of an optimal allocation).

6) For each $e \in \mathcal{E}(a_p)$ ($p = s, t$), the robot turns the edge e off (such that the tasks at the end of the edges are no longer considered connected) with a probability $1 - p_e$, e.g., $p_e = \frac{l_{\min}(a_p)}{l(e)}$. For $p = s, t$ this results in the decomposition of $a_p \in \mathcal{A}$ into n_p connected components a_{pi} such that $a_p = \bigcup_{i=1}^{n_p} a_{pi}$.
7) The auctioneer collects all the connected components in the set $\mathcal{CP}$.
8) With a 50%-50% probability the auctioneer selects either a transfer a) or swap b).
   a) **Connected Component Transfer**: Select a connected component $a_{si} \in \mathcal{CP}$ from robot h_s with a probability in $P(a_{si}|\mathcal{CP})$, e.g., $P(a_{si}|\mathcal{CP}) = \frac{1}{||\mathcal{CP}||}$ in a uniform distribution. Assume that a_{si} is reassigned to robot h_t with a probability in $P(A(a_{si}) = t|a_{si}, \mathcal{A})$, e.g., $P(A(a_{si}) = t|a_{si}, \mathcal{A}) = \frac{1}{k}$ in a uniform distribution, resulting in the new allocation $\mathcal{A}_i^{(s,t)}$ that has two modified clusters[2] $a_s^{(-i)}$ and $a_t^{(+i)}$. Assume that robot h_s computes[3] $c_s(a_s^{(-i)})$ and for $t = 1, 2, \ldots, k$ ($t \neq s$) robot h_t computes[3] $c_t(a_t^{(+i)})$, which the auctioneer uses to compute the corresponding cost $C(\mathcal{A}_i^{(s,t)})$ (based on (1) or (2)). The probability $\alpha_S(\mathcal{A} \to \mathcal{A}_i^{(s,t)})$ of the acceptance of the transfer of the connected component a_{si} from robot h_s to robot h_t is computed. (The acceptance Theory is omitted for simplicity.) If $C(\mathcal{A}_i^{(s,t)})$ is less than $C(\mathcal{A}^*)$, then the new $\mathcal{A}^*$ is updated to $\mathcal{A}_i^{(s,t)}$.
   b) **Connected Component Swap**: Select two connected components in a_s and a_t, one connected component a_{si} from robot h_s with a probability in $P(a_{si}|\mathcal{CP})$ and the other connected component a_{tj} from robot h_t with a probability in $P(a_{tj}|\mathcal{CP})$, and swap them, resulting in the new allocation $\mathcal{A}_{i,j}^{(s,t)}$ that has two modified clusters[2] $a_s^{(-i,+j)}$ and $a_t^{(+i,-j)}$. Assume that robot h_s computes[3] $c_s(a_s^{(-i,+j)})$ and robot h_t computes[3] $c_t(a_t^{(+i,-j)})$, which the auctioneer uses to compute the corresponding cost $C(\mathcal{A}_{i,j}^{(s,t)})$ (based on (1) or (2)). Then, the probability $\alpha_D(\mathcal{A} \to \mathcal{A}_{i,j}^{(s,t)})$ of swapping the two connected components is computed. (The acceptance Theory is omitted for simplicity.) If $C(\mathcal{A}_{i,j}^{(s,t)})$ is less than $C(\mathcal{A}^*)$, then the new $\mathcal{A}^*$ is updated to $\mathcal{A}_{i,j}^{(s,t)}$.
9) For the non-greedy SCA the auctioneer accepts the proposal with probability $\alpha_S(\mathcal{A} \to \mathcal{A}_i^{(s,t)})$ or $\alpha_D(\mathcal{A} \to \mathcal{A}_{i,j}^{(s,t)})$, while for the greedy SCA the auctioneer accepts the proposal if $C(\mathcal{A}_i^{(s,t)})$ or $C(\mathcal{A}_{i,j}^{(s,t)})$ is less than $C(\mathcal{A}^*)$. Then $\mathcal{A}$ is updated and the cost $C(\mathcal{A})$ is updated and stored. Otherwise, the auctioneer declines the proposal and the auctioneer keeps the current configuration.
10) If the auction evolution termination criterion is satisfied, i.e., $T < T_{cut}$, where T_{cut} is some threshold temperature, then the auction is terminated and the final allocation is $\mathcal{A}^*$ with final cost $C(\mathcal{A}^*) \le C(\mathcal{A}^{(0)})$. If the criterion is not satisfied, reduce T, using $T \leftarrow T/\beta$ where $\beta > 1$ and go to Step 3).

In order to search for the global optimum, a simulated annealing method has been adopted. It is well known that the convergence rate of simulated annealing depends on the "depth" of the instance [13] (i.e., the highest you have to go upward to eventually reach a lower (better) point than where you are, where the maximum is taken over all starting points) and that changing the neighborhood structure can drastically change the depth. Simulation results (omitted for brevity) showed that adding swaps has decreased the depth, which accounts for the faster convergence.

III. EXPERIMENTAL RESULTS FOR CENTRALIZED APPLICATION OF OCCSA, GSSCA, SW^{1}SCA AND SW^{2}SCA

This section provides simulation results for centralized application of OCCSA, GSSCA, SW^{1}SCA and SW^{2}SCA to homogeneous and heterogeneous teams using the multi-robot routing problem, which is a standard test domain for robot coordination using auctions [1], [14], [15], [9], [10], [16]. The task allocation is time-extended assignment such that all tasks are assigned to robots before the assignments are carried out [17]. It is free of conflicts since each task is assigned to no more than one robot. The tasks in the multi-robot routing problem considered here are to visit targets and complete an assignment. The costs resulting from OCCSA, GSSCA, SW^{1}SCA and SW^{2}SCA are compared to the best cost obtained using the Sequential (single-item) Auction (SA) and the Parallel Auction (PA), which are standard auction methods in the existing literature [1], [2], [18], [19], and their variants, the Look-Back Sequential (single-item) Auction (LBSA) and the Look-Back Parallel Auction (LBPA), which were described in prior research [3], [4].

For each simulation the stochastic random scenario appears in a $10000m \times 10000m$ area involving 1000 random scenarios for a given number of tasks and robots. The initial robot positions were evenly distributed along one edge of the area. (When the initial robot positions are distributed differently, e.g., randomly, the qualitative results given below remain the same.) Also, for each simulation the following SCA parameters were used: initial temperature, $T = 1000$; and termination temperature, $T_{cut} = 20$. (T and T_{cut} are a priori knowledge regarding the scenario scale so they are scenario-dependent. But they are always the same for different robots, different tasks and different missions in a fixed scenario scale.) In terms of the experimental scenario, a particular case was selected where all robots begin evenly distributed along a line. This assumption has been justified in prior distributed implementation of SCA [4] using the four benchmark auction communication scenarios, derived

[2]complexity analysis of SW^{2}SCA is omitted due to space limitation but will be included in the future paper.

from fundamental network topologies. These and other experiments performed by the authors make it clear that the insights gained from the initial experiments are not an artifact of that particular setting but are indeed a general phenomena which is expected to see for a wide variety of robot and task instantiations.

In order to evaluate the performance of a SCA the concept of *Mean Cost Improvement* (*MCI*) is introduced in Definition 1. The communication complexity of a SCA is measured by the number of *Auction Cycles* (*AC*), also as given in Definition 2. In addition, the concept of *Robot Type Vector* is introduced in Definition 3 to measure the different types of robots that constitute a given heterogeneous team in the evaluations of Section III-B.

Definition 1 *For m stochastic scenarios let $\{C^{SCA}(r) : r = 1, \cdots, m\}$ denote the set of m costs resulting from SCA and let $\{C^{BestGreedy}(r) : r = 1, \cdots, m\}$ denote the set of minimum costs achievable with the greedy algorithms: SA, LBSA, PA and LBPA. The* ***Mean Cost Improvement (MCI)*** *is the geometric mean of the normalized improvement of the SCA cost over the best of the greedy algorithms, such that*

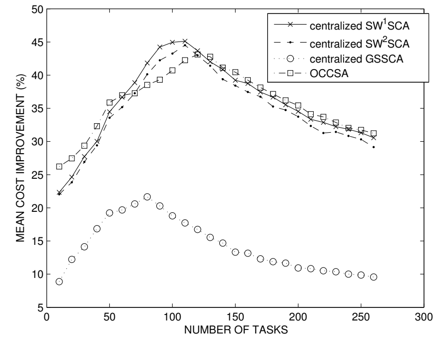
$$MCI \triangleq \left(\prod_{r}^{m} \left(\frac{C^{BestGreedy}(r) - C^{SCA}(r)}{C^{BestGreedy}(r)} \right) \right)^{\frac{1}{m}}. \quad (3)$$

Definition 2 *An* ***Auction Cycle (AC)*** *is the number of iterations to algorithm convergence, where for a CSA an iteration corresponds to Steps 2 through 6 of Algorithm 1, while for a SCA an iteration corresponds to Steps 3 through 10 of Algorithm 2.*

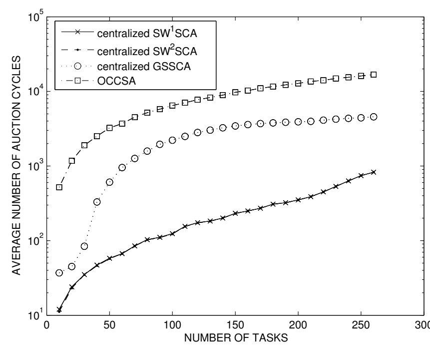
Definition 3 *Let $at(p)$ denote a type p robot for $p \in \{1, 2, ..., v\}$, where $at(p) \neq at(q)$ for $p \neq q$. Assume that a team consist of k robots, where $k \geq v$ and that $m(p)$ denotes the number of robots of type $at(p)$. The* ***Robot Type Vector (RTV)*** *for this k-robot team is denoted by $M(k)$ and defined by $M(k) = [\ m(1) \quad m(2) \quad \ldots \quad m(i) \quad \ldots \quad m(v)\ ]$, where $\sum_{p=1}^{v} m(p) = k$.*

A. Comparison of OCCSA with GSSCA, SW^{1}SCA and SW^{2}SCA Using Homogeneous Teams

As originally mentioned in Section I-B, it has been shown [10] that within the CSA framework, OSCSA, an algorithm based on O and C contracts, yields the best performance among the algorithms based on various sub-combinations of the OCSM contracts. Hence, for centralized auctioning it is compared here with GSSCA, SW^{1}SCA and SW^{2}SCA. Since SW^{1}SCA cannot be directly applied to heterogeneous teams, the comparison uses homogeneous robots. The comparison metrics are MCIs and ACs and the cost function is the MINSUM cost function of Equation (1), corresponding in this study to the total distance traveled by the robots. Each of the robots was assumed to have the constant speed (i.e., the robots are homogeneous), chosen randomly from the interval $(0m/s, 20m/s]$ assuming a uniform distribution. The



(a) Mean cost improvement vs. Number of tasks



(b) Average auction cycles vs. Number of tasks

Fig. 1. Mean cost improvement (MCI) and average number of auction cycles (ACs) of OCCSA, GSSCA, SW^{1}SCA and SW^{2}SCA for 5 robots with the number of tasks ranging from 10 to 260 in increments of 10

comparison was based on 5 robots with the number of tasks ranging from 10 to 260 in increments of 10. The cooling schedule ratio is chosen as $\beta = 1.001$, which yields relatively slow annealing. Further comparison of GSSCA and SWSCA based on various cooling schedule ratios is given in prior research [5].

Figure 1 illustrates the main results. The MCIs of SW^{2}SCA are in the interval [21.04%,44.52%] and are very similar to those for SW^{1}SCA, which are in [22.34%,45.96%] and those for OCCSA, which are in [26.23%,43.12%]. The MCIs for GSSCA are substantially smaller, lying in [8.87%,21.65%]. In contrast the ACs of SW^{2}SCA lie in [11,826] similar to SW^{1}SCA lie in [12,827], while the ACs of OCCSA lie in [519,16792], and for each specified number of tasks the ACs of SW^{1}SCA and SW^{2}SCA is at least a factor of 20 times smaller than the corresponding AC for OCCSA. The ACs of GSSCA lie in [37,4561] and for each specified number of tasks the AC of GSSCA lies between the corresponding ACs for SW^{2}SCA and OCCSA.

The results show that for homogeneous teams SW^{2}SCA can be far more computationally efficient than the OCCSA at obtaining very similar performance, has similar performance to SW^{1}SCA, and is superior to GSSCA. The performance of the OCCSA is fixed. In contrast, as illustrated in prior research [5], the SCAs can sacrifice performance (i.e., MCI) for computations (i.e., ACs). For example, at the expense of greater ACs it is possible for SW^{2}SCA to increase its MCIs by further slowing the annealing. The better outcomes reported in this paper and in the prior paper [5] are due in large part to the stochasticity of the algorithm, which in effect permits a swap to occur as a sequence of O or C moves, and in other cases due to the existence of swaps themselves in the neighborhood structure.

B. Evaluation of SW^{2}SCA for Heterogeneous Teams

TABLE II
ASSUMED ROBOT TYPE VECTOR (RTV) AS THE NUMBER OF ROBOTS VARIES FROM 2 TO 10

no. of robots (k)	RTV
2	$M(2) = [\ 2\ \ 0\ \ 0\ ]$
3	$M(3) = [\ 2\ \ 1\ \ 0\ ]$
4	$M(4) = [\ 2\ \ 1\ \ 1\ ]$
5	$M(5) = [\ 2\ \ 2\ \ 1\ ]$
6	$M(6) = [\ 2\ \ 2\ \ 2\ ]$
7	$M(7) = [\ 3\ \ 2\ \ 2\ ]$
8	$M(8) = [\ 3\ \ 3\ \ 2\ ]$
9	$M(9) = [\ 3\ \ 3\ \ 3\ ]$
10	$M(10) = [\ 4\ \ 3\ \ 3\ ]$

This section evaluates SW^{2}SCA for heterogeneous teams. The cost function is a MINMAX cost function in Equation (2) corresponding to the maximum time taken to accomplish the tasks. The speeds for each of the robots were assumed to be constant and were chosen randomly from the speed set $\{20m/s, 10m/s, 1m/s\}$. In the subsequent simulations, $v \leq 3$ (i.e., the number of robot types is at most 3). The robots were assumed to differ in terms of their speed of travel. In particular, $at(1)$ traveled at $20m/s$, $at(2)$ at $10m/s$ and $at(3)$ at $1m/s$. Table II shows the assumed RTVs when the number of robots k varies from 2 to 10. The auctioneer was assumed to be of type $at(1)$. As before, 1000 random scenarios were studied for a given number of robots and tasks with the number of robots now ranging from 2 to 10 and the number of tasks ranging from 10 to 260 in increments of 10.

Figure 2 displays the costs (in this case for SW^{2}SCA with $\beta = 1.01$) as a function of the number of robots and tasks. It shows that as the number of tasks increases, a substantial performance improvement (i.e., time savings) can be achieved by adding a small number of robots. For example in Figure 2(b), which shows the costs for 260 tasks, the cost corresponding to 2 robots is 250 $hours$, while the costs with 4 robots improves to 82 $hours$. In general for a fixed number of tasks, the corresponding "slice" of a 3-D curve such as Figure 2(b) may be used to trade off performance vs. the number of robots and hence provides a guideline for choosing the desired number of robots for the expected mission. It is important to note that Figure 2 shows that the cost decreases more than linearly with the addition of robots. It is because 4 robots not only split the work among each other, but also start from different locations so that every job is reasonably close to some robot, compared with what happens with two robots. Using the given metric, a solution would only improve linearly if robots are added but the tasks are *not* reallocated. Overall, the curves of Figures 2 provides cost information to aid in choosing the number of robots needed to execute a given range of tasks. This information can be an important component of a systematic engineering approach to designing multi-robot systems.

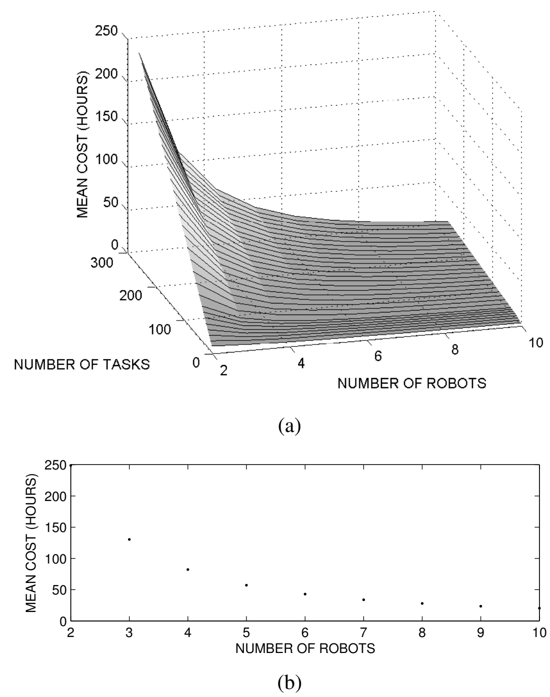


Fig. 2. (a) Mean cost vs. the number of robots and the number of tasks for SW^{2}SCA with $\beta = 1.01$ and the RTVs in Table II; (b) The "slice" of (a) corresponding to 260 tasks

IV. EXPERIMENTAL RESULTS FOR DISTRIBUTED APPLICATION OF SW^{2}SCA

As previously discussed, distributed auctions are needed due to limited communication between robots. This section uses numerical experiments to evaluate the efficacy of the distributed auction of SW^{2}SCA.

For each simulation the stochastic random scenario appears in a $10000m \times 10000m$ area involving 1000 random scenarios for a given number of tasks and robots. The initial robot positions were evenly distributed along one edge of the area. As in Section III-B, the robots were assumed to differ in terms of their speed of travel. In particular, $at(1)$ traveled at $20m/s$, $at(2)$ at $10m/s$ and $at(3)$ at $1m/s$. The type of each robot was chosen randomly from the set $\{at(1), at(2), at(3)\}$. RTV follows Table II in Section III-B. The cost function is a MINMAX cost function Equation (2), representing time to mission completion. The SW^{2}SCA parameters used were as before: initial temperature, $T = 1000$; termination temperature, $T_{cut} = 20$; and the cooling schedule ratio, $\beta = 1.01$.

The efficacy of distributed SW^{2}SCA is measured by comparing the resultant global cost with the corresponding SW^{2}SCA global cost. This leads to the following definition for *optimization efficiency*.

Definition 4 *The **optimization efficiency for scenario r** is denoted by $\eta_r \in (0, 1]$ and defined by*

$$\eta_r \triangleq \frac{C_r^*}{C_r}, \tag{4}$$

where C_r^* is the global cost resulting from the application of SW^{2}SCA and C_r is the global cost resulting from the application of distributed SW^{2}SCA.

A *tournament* corresponds to one round of distributed auctioning in which one of the robots serves as the auctioneer and leads an auction with the robots that are within communication range. To quantify the extent of robot interaction in the tournaments of the distributed auctioning the concept of *tournament participation index* is introduced in the following definition. Increasing values of this index corresponds to increasing communication between the agents.

Definition 5 *The **Tournament Participation Index (TPI)**[3] for k robots is denoted by $\zeta(k) \in (0, 1]$ and defined by*

$$\zeta(k) \triangleq \frac{\frac{\sum_{p=1}^{k} b^2(p)}{k}}{k^2} = \frac{\sum_{p=1}^{k} b^2(p)}{k^3}, \tag{5}$$

where $b(p)$ is the number of robots that participate in the regional auction in which robot h_p is the auctioneer. Hence $\zeta(k)$ is the mean of $b^2(p)$ for the k robots, normalized so that it lies in the interval $(0, 1]$.

A. Evaluation of Distributed SW^{2}SCA Using Scale Free Networks

A key issue is how to evaluate distributed SW^{2}SCA (dSW^{2}SCA) using simulations. Prior work [4] performed evaluation of a SCA using four benchmark auction communication scenarios that were derived from fundamental network topologies. This methodology was useful in demonstrating that an SCA can perform well in distributed communication conditions. However, in this section the simulations are based on robots whose communication links are determined according to the topology of a scale free network (SFN) [20]. The resulting SFN tends to have some robots that have sparse communication links while others have more dense communication links. Hence, the SFN may be viewed as combining aspects of all of the previously used fundamental network topologies [4].

The circular network was used in the simulations and it was assumed that each robot sequentially takes a turn as an auctioneer. The origin of the circle defining the positions of the robots was at the center of a $10000m \times 10000m$ area and the diameters of the circles were chosen to be $5000m$. Each simulation involved 1000 random scenarios for a given number of tasks and robots.

[3]TPI is similar to but different than the degree or connectivity in networks or graph theory since there are no redundant connections between two robots and $b(p)$ counts the robots instead of the links.

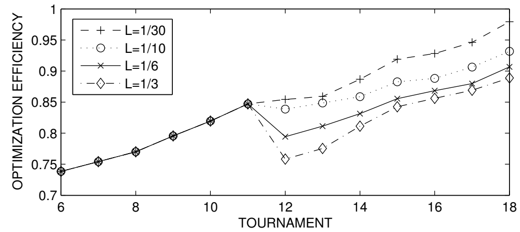


Fig. 3. Mean optimization efficiency vs. number of tournaments and the ratio (L) of new tasks to initial tasks for a SFN of abstract auction rotation patterns in [5]: 300 initial tasks with a new robot and new tasks introduced between Tournament 11 and Tournament 12, which cause a decrease in TPI

In the dynamic scenarios 300 tasks were randomly given at the outset of auctioning and after a certain number of tournaments a new robot is introduced and new tasks are added in some ratio L. Subsequently, the tournaments continue and assign the new tasks to the appropriate robots. The number of new tasks introduced varies with each simulation and is some ratio L times the number of the initial tasks.

A SFN network for 11 robots is illustrated in prior research [5] with $M(11) = [\ 3 \quad 3 \quad 5\ ]^T$. The sequence of the robot types starting from h_1 is ordered as follows: $at(1) \to at(2) \to at(3) \to at(3) \to at(2) \to at(2) \to at(2) \to at(3) \to at(1) \to at(2) \to at(1)$. However, a new robot h_{12} in $at(1)$ is added and a number of new tasks are introduced after robot h_1 serves as the auctioneer for the second time. Figure 3 showed that the mean optimization efficiency actually decreased after the new robot h_{12} was introduced due to the TPI decreasing from 0.0744 to 0.0689. However, the distributed auctioning accommodated the new robot and additional tasks as the tournaments progressed; the optimization efficiency increased as the tournaments progressed and the optimization efficiency intervals increased from [76%,85%] to [88%,97%] with the ratio L increased.

V. Summary and Final Discussions

This paper presented a novel and efficient Stochastic Clustering Auction (SCA), called called the SW^{2}SCA, based on the modified Swendsen-Wang method. Unlike the previously developed generalized Swendsen-Wang SCA (SW^{1}SCA), it allows each robot reconstruct the tasks that have been connected and applies to heterogeneous teams.

Using homogeneous teams, GSSCA, SW^{1}SCA, and SW^{2}SCA were compared with each other and with OCCSA, which has been shown to yield the best cost improvements in the fewest steps of all the contracts sequencing algorithm based on a subset of the OCSM contract types. It was seen that for homogeneous teams centralized SW^{2}SCA can be far more computationally efficient than OCCSA at obtaining very similar performance, is similar to SW^{1}SCA, and is superior to centralized GSSCA. However, whereas the performance and computations of OCCSA are fixed, the SCAs can sacrifice performance for computations.

The random simulation for centralized SW^{2}SCA also resulted in 3-D curves that show cost vs. number of robots and number of tasks. Given a fixed number of tasks, a 2-D slice of the 3-D curve shows the tradeoff between cost and number of robots. This enables an appropriate choice of the number of robots for a given mission that has an expected number of tasks. The important point to note is that the improvement from adding robots is better than linear. Using the given metric, a solution would improve linearly just by adding robots and not reallocating any tasks.

Distributed SW^2SCA, denoted as dSW^2SCA, was based on using SW^2SCA in regional auctions. Its performance was evaluated in random simulations using communication links derived from a scale free network. The simulation results showed that dSW^2SCA continuously improved the global performance each time one of the robots completed its tournament (i.e., its auction process) and that dSW^2SCA can sucessfully accommodate a new robot and new tasks. However, as expected, the costs achieved with decentralized auctioning were worse than those achieved with centralized auctioning.

VI. ACKNOWLEDGMENTS

This work was sponsored, in part, by the Army Research Office under grant W911NF-09-1-0568. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. 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 ARL or the US Government.

REFERENCES

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

[2] R. Michael Zlot and A. Stentz. Market-based multirobot coordination for complex tasks. *International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics*, 25(1):73–101, January 2006.

[3] K. Zhang, E. G. Collins, D. Shi, X. Liu, and O. Chuy. A stochastic clustering auction for centralized and distributed task allocation in multi-agent teams. In H. Asama, H. Kurokawa, J. Ota, and K. Sekiyama, editors, *Distributed Autonomous Robotic Systems 8*, pages 345–354, Tsukuba, Ibaraki, Japan, 2008.

[4] K. Zhang, E. Collins, and D. Shi. Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. *ACM Transactions on Autonomous and Adaptive Systems (to appear)*.

[5] K. Zhang, E. Collins, and A. Barbu. A novel stochastic clustering auction for task allocation in multi-robot teams. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 3300–3307, Taipei, Taiwan, October 18-22 2010.

[6] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[7] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[8] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):590–602, 2005.

[9] T. Sandholm. Contract types for satisficing task allocation: I theoretical results. In *AAAI Spring Symposium: Satisficing Models*, 1998.

[10] M. Andersson and T. Sandholm. Contract type sequencing for reallocative negotiation. In *Proceedings of the The 20th International Conference on Distributed Computing Systems (ICDCS 2000)*, pages 154–160, Washington, DC, USA, 2000. IEEE Computer Society.

[11] V. Jarnick. Ojistám problému minimálním. *Axta Societatis Natur Moravicae*, 6:57–63, 1930.

[12] R. C. Prim. Shortest connection networks and some generalisations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[13] D. Henderson, S. H. Jacobson, and A. W. Johnson. *Handbook of Metaheuristics*, volume 57, chapter The Theory and Practice of Simulated Annealing, pages 287–319. Kluwer Academic Publishers, Boston, MA, 2003.

[14] M. B. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '02)*, volume 3, pages 2714 – 2720, September 2002.

[15] S. Koenig, C. A. Tovey, M. G. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. J. Kleywegt, A. Meyerson, and S. Jain. The power of sequential single-item auctions for agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1625–1629. AAAI Press, 2006.

[16] X. Zheng and S. Koenig. K-swaps: Cooperative negotiation for solving task-allocation problems. In Craig Boutilier, editor, *International Joint Conference on Artificial Intelligence*, pages 373–379, 2009.

[17] B. P. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[18] B. P. Gerkey and M. J. Mataric. Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[19] S. Koenig, C. A. 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 (IJCAI)*, pages 1359–1365, 2007.

[20] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.