A Novel Stochastic Clustering Auction for Task Allocation in Multi-Robot Teams

Kai Zhang, Emmanuel G. Collins, Adrian Barbu

Abstract— **It has been shown that the global cost of the task allocations obtained with fast greedy algorithms can be improved upon by using a class of auction methods called Stochastic Clustering Auctions (SCAs). SCAs use stochastic transfers or swaps between the task clusters assigned to each team member, allow both uphill and downhill cost movements, and rely on simulated annealing. The choice of a key annealing parameter and turning the uphill movements on and off enables the converged solution of a SCA to slide in the region between the global optimal performance and the performance associated with a random allocation. The first SCA, called here GSSCA, was based on a Gibbs sampler, which constrained the stochastic cluster reallocations to simple single transfers or swaps. This paper presents a new and more efficient SCA, called SWSCA, based on the generalized Swendsen-Wang method that enables more complex and efficient movements between clusters by connecting tasks that appear to be synergistic and then stochastically reassigning these connected tasks. For centralized auctioning, extensive numerical experiments are used to compare the performance of SWSCA with GSSCA in terms of costs and computational and communication requirements. Distributed SWSCA is then compared with centralized SWSCA using communication links between robots that were motivated by a generic topology called a "scale free network."**

I. INTRODUCTION

Auction methods are an effective approach to task allocation for heterogeneous robot teams. They are generally presented as either centralized auctions that involve a central auctioneer that determines the task allocation based on the task bids provided to it by each team member [1], [2], [3], [4], [5], [6] or distributed auctions that involve peer-to-peer redistribution of plans between given subsets of robots, where one of the robots serves as the auctioneer [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. However, this distinction is not strong since centralized auction approaches lead to distributed auction approaches using the concept of "opportunistic centralization" [12] in which the centralized auction algorithm is applied regionally. This concept is used in this paper in Section II-D. Opportunistic centralization is inherent in all of the distributed auction approaches. (One way to see this is that each of the distributed auction methods corresponds to a centralized auction method when the auction simultaneously involves each of the robots, i.e., a regional auction is the global auction.)

K. Zhang and E. G. Collins 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

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

Distributed auctions can generally be divided into three classes. The first set [8], [11], [14], [15], [17], [20] uses greedy auctioning, which is inherently suboptimal. The second set [9], [10], [12], [13], [16] uses the deterministic heuristics in [6] to limit the combinations considered in combinatorial auctioning. A current limitation of these methods is that the deterministic heuristics assume that the triangle inequality is preserved for the metric cost space [21], which does not apply to cost functions that can be used to represent minimum time objectives (see (2) below). The third set of auction methods [7], [18], [19] is closely related to the method developed here. This set uses a deterministic synthesis of single transfer, swap and multi-party exchange movements between the clusters assigned to the robots. However, a limitation of all the approaches in these three classes is that they do not provide a mechanism to avoid local minima [7], [21].

An additional limitation of the previously developed auction methods is that they do not provide a mechanism for using computational and communication requirements to enable the performance obtained after the algorithm convergence to slide in the region between the globally optimal performance and the performance associated with some random allocation, as illustrated in Fig. 1. In particular, once these algorithms converge for a given problem they converge to a single cost. However, it may be desirable to specify that one is willing to increase (or decrease) computational and communication requirements in order to increase (or decrease) the allocation performance by decreasing (or increasing) the converged cost.

The first stochastic clustering auction based on global optimization, in this case simulated annealing, is presented in [22] and is called here the Gibbs sampler Stochastic Clustering Auction (GSSCA) since the underlying optimization algorithm is a Gibbs sampler in the class of probabilistic algorithms called Markov Chain Monte Carlo [23]. GSSCA alternates with equal probabilities between transfer and swap moves and allows not only downhill movements, but also uphill movements, which can enable it to escape local minima. The team performance obtained after algorithm convergence can slide in the region between the global optimal performance and the performance of a random allocation by tuning the annealing suite and turning the uphill movements on and off [22].

However, the difficulty of approaching optimal clustering using a Gibbs sampler is well reflected in a simple Ising and Potts model [24]. In Fig. 2 the tasks are indicated by the shaded circles and aggregated into two clusters, each cluster

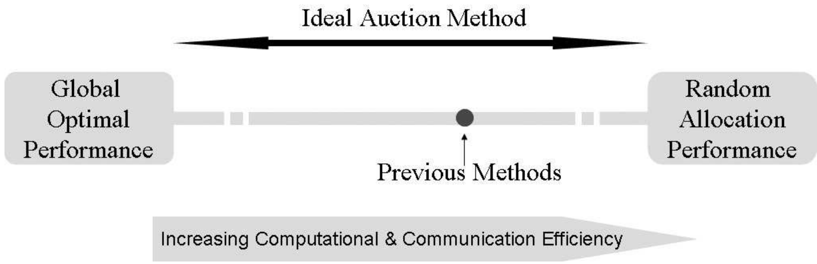


Fig. 1. Illustration of the ability of an "ideal auction method" to trade off computational and communication requirements so that the converged performance lies anywhere in the performance spectrum

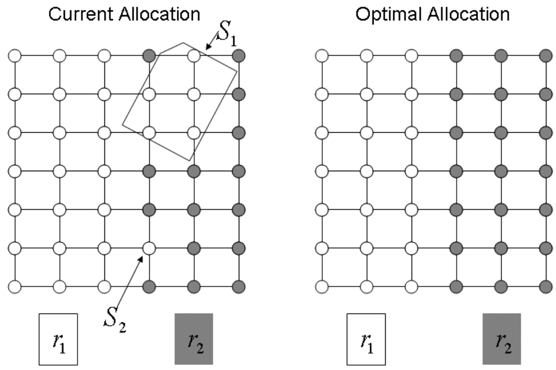


Fig. 2. Illustration of a reallocation task for two homogeneous robots r_1 and r_2 that is difficult to accomplish efficiently using a clustering algorithm based on a Gibbs sampler

corresponding to the robot of the identical shading. The Gibbs sampler requires flipping two connected set of tasks, S_1 and S_2, from light circles in the current allocation to dark circles in the optimal allocation. The probability of flipping a task from light to dark is $p_0 = \frac{1}{2}$. Thus, the expected number of steps N needed to flip the 5 tasks in the set S_1 from light to dark is $N = \frac{1}{(1/p_0)^5} = 2^5$. This illustrates that the expected number of steps required to switch M tasks is exponential in M. Intuitively, it is desirable to flip an entire set of interconnected tasks such as S_1 in one step. This paper presents a stochastic algorithm that enables these types of complex movements for optimal task allocation based on an algorithm that uses a generalized Swendsen-Wang method and has been successfully applied to shape clustering and segmentation in computer vision [24].

The remainder of this paper is organized as follows. Section II formulates the basic optimization problem for task allocation, provides a description of the Swendsen-Wang Stochastic Clustering Auction (SWSCA) and discusses how the algorithm may be used for both centralized and distributed auctioning. Section III considers centralized auctioning and presents simulation results from random scenarios with a focus on comparing the results of the SWSCA and GSSCA algorithms with and without uphill movements. 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 comparing the performance achieved with distributed and centralized SWSCA. Finally, Section V presents conclusions and future work.

II. SWENDSEN-WANG STOCHASTIC CLUSTERING AUCTION

This section first presents the basic problem statement. It then describes the Swendsen-Wang Stochastic Clustering Auction (SWSCA). Furthermore, a generic framework for Stochastic Clustering Auctions (SCAs) is given and used to describe greedy and non-greedy versions of an SCA. After introducing the concept of regional cost, it is shown that when a distributed auctioneer reduces the corresponding regional cost, the global cost will either decrease or remain the same. Hence SWSCA is proposed to optimize the regional cost in a distributed auction.

A. Notation and Problem Statement

Let $\mathcal{R}$ denote a set of k heterogeneous robots, and $\mathcal{T}$ denote a set of n tasks, i.e. $\mathcal{R} = \{r_1, r_2, \ldots, r_k\}$ and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$. Also, let $\mathcal{A}$ denote the allocation, $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, where $\bigcup_{i=1}^{k} a_i = \mathcal{T}$, $a_i \subseteq \mathcal{T}$ and the cluster a_i is assigned to robot r_i. Each a_i is decomposed into n_i conected components a_{ij} such that $a_i = \bigcup_{j=1}^{n_i} a_{ij}$. $\mathcal{CP}$ denote the entire set of connected components such that $\mathcal{CP} = \{a_{ij} : i = 1, 2, \ldots, k; j = 1, \ldots, n_i\}$. $A(a_{ij}) = p$ is used to denote the allocation of the connected component a_{ij} from robot i to robot p. The cost associated with $\mathcal{A}$ is given by either

$$C(\mathcal{A}) = \sum_{i=1}^{k} c_i(a_i), \tag{1}$$

or

$$C(\mathcal{A}) = \max_i c_i(a_i), \tag{2}$$

where $c_i(a_i)$ is the minimum cost for robot i to complete the set of tasks a_i. The individual cost function $c_i(\cdot)$ 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 $\min_{\mathcal{A}} C(\mathcal{A})$. In practice the cost function in (1) might be used to represent the total distance traveled or the total energy expended by the robots while the cost function in (2) might be used to represent the maximum time taken to accomplish the tasks.

B. Description of the Swendsen-Wang Stochastic Clustering Auction (SWSCA)

In the auction framework [13], SWSCA attempts to minimize the cost $C(\mathcal{A})$ using a Markov chain search process in the space of possible allocations. It is assumed that the robots are cooperative, and that collusion, shilling and other cheating mechanisms are not allowed [14]. The basic algorithm was originally developed in [24]. The essential mechanism of SWSCA is to start with an allocation $\mathcal{A}$ for k clusters and to reduce or probabilistically hillclimb $C(\mathcal{A})$ by rearranging the tasks $\mathcal{T}$ in connected components among the clusters. The rearrangement is performed in a stochastic fashion using transfer and swap moves. These moves are performed with probabilities proportional to the negative exponential of the costs $C(\mathcal{A})$ of the resulting allocations $\mathcal{A}$ (see (3) and (7)). SWSCA is always guaranteed to result in an allocation that has a cost less than or equal to the cost of the initial allocation. The actual algorithm is described below.

1) For a given set of tasks $\mathcal{T}$ construct an adjacency graph $G_0 = \langle \mathcal{T}, \mathcal{E}_0 \rangle$ where $\mathcal{E}_0$ is the edge set of $\mathcal{T}$.
2) Partition $\mathcal{T}$ into k clusters to form an initial allocation $\mathcal{A}^{(0)} = \{a_1^{(0)}, a_2^{(0)}, \ldots, a_k^{(0)}\}$, where each cluster $a_i^{(0)}$ 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 during the iterations that stores the lowest cost.)
3) Construct a graph $G(\mathcal{A}) = \langle \mathcal{T}, \mathcal{E}(\mathcal{A}) \rangle$ based on the current allocation $\mathcal{A}$, where $\mathcal{E}(\mathcal{A}) \in \mathcal{E}_0$ has all the edges of $\mathcal{E}_0$ except those connecting the tasks belonging to different robots.
4) Each robot $r_i \in \mathcal{R}$ ($i = 1, 2, \ldots, k$) uses a "constrained Prim's Algorithm"[1] (a greedy algorithm) to efficiently approximate the cost $c_i(a_i)$ and submits its cost to the auctioneer. In this bid valuation stage, each cluster a_i becomes an ordered subset of $\mathcal{T}$. The auctioneer computes the global cost $C(\mathcal{A})$ using (1) or (2) and sets a high temperature T.
5) For $e \in \mathcal{E}(\mathcal{A})$, turn the edge e off with a probability $1 - p_e$, e.g. $p_e = \frac{\mathcal{D}_{min}}{d(e)}$ where $\mathcal{D} = \{||d(e)|| : \ e \in \mathcal{E}(\mathcal{A})\}$, $|| \cdot ||$ denotes Euclidean distance, and $\mathcal{D}_{min} = \min \mathcal{D}$.
6) For $i = 1, 2, \ldots, k$ turning the edge off divides each $a_i \in \mathcal{A}$ into n_i connected components such that $a_i = \bigcup_{j=1}^{n_i} a_{ij}$ by each robot.
7) Collect all the connected components in the set $\mathcal{CP}$.
8) The auctioneer rearranges the clusters using either a single move or a dual move among $\mathcal{CP}$.

   Single Move (Connected Component Transfer): Select a connected component $a_{si} \in \mathcal{CP}$ from robot r_s with a probability $q(a_{si}|\mathcal{CP})$, e.g., $q(a_{si}|\mathcal{CP}) = \frac{1}{||\mathcal{CP}||}$ in a uniform distribution. Assume that a_{si} is reassigned to robot r_t with a probability $q(A(a_{si}) = t|a_{si}, \mathcal{A})$, e.g., $q(A(a_{si}) = t|a_{si}, \mathcal{A}) = \frac{1}{k}$ 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 r_s computes[3] $c_s(a_s^{(-i)})$ and for $t = 1, 2, \ldots, k$ ($t \neq s$) robot r_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 of the acceptance of the transfer of the connected component a_{si} from robot r_s to robot r_t is given by [24]

$$\alpha_S(\mathcal{A} \to \mathcal{A}_i^{(s,t)}) = \min(1, \alpha_S^1 \cdot \alpha_S^2 \cdot \alpha_S^3), \qquad (3)$$

where

$$\alpha_S^1 = \frac{\prod_{e \in \mathcal{E}(a_{si}, a_t)}(1 - p_e)}{\prod_{e \in \mathcal{E}(a_{si}, a_s - a_{si})}(1 - p_e)}, \qquad (4)$$

$$\alpha_S^2 = \frac{P(A(a_{si}) = s|a_{si}, \mathcal{A}_i^{(s,t)})}{P(A(a_{si}) = t|a_{si}, \mathcal{A})}, \qquad (5)$$

$$\alpha_S^3 = \frac{\exp(-C^{(2)}/T)}{\sum_{t=1, t\neq s}^{k} \exp(-C(\mathcal{A}_i^{(s,t)})/T)} \qquad (6)$$

and $C^{(2)} = C(\mathcal{A}_i^{(s,t)})$, the cost after transferring.

   Dual Move (Connected Component Swap): Select two connected components in a_s and a_t, one connected component a_{si} from robot r_s with a probability $q(a_{si}|\mathcal{CP})$ and the other connected component a_{tj} from robot r_t with a probability in $q(a_{tj}|\mathcal{CP})$, 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 r_s computes[3] $c_s(a_s^{(-i,+j)})$ and robot r_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 of swapping the two connected components is given by

$$\alpha_D(\mathcal{A} \to \mathcal{A}_{i,j}^{(s,t)}) = \min(1, \alpha_D^1 \cdot \alpha_D^2), \qquad (7)$$

where

$$\alpha_D^1 = \frac{\prod_{e \in \mathcal{E}(a_{si}, a_t)}(1 - p_e) \cdot \prod_{e \in \mathcal{E}(a_{tj}, a_s)}(1 - p_e)}{\prod_{e \in \mathcal{E}(a_{si}, a_s - a_{si})}(1 - p_e) \cdot \prod_{e \in \mathcal{E}(a_{tj}, a_t - a_{tj})}(1 - p_e)}, \qquad (8)$$

$$\alpha_D^2 = \frac{\exp(-C^{(2)}/T)}{\sum_{p=1}^{2} \exp(-C^{(p)}/T)}, \qquad (9)$$

and $C^{(1)} = C(\mathcal{A})$, the cost before swapping while $C^{(2)} = C(\mathcal{A}_{i,j}^{(s,t)})$, the cost after swapping. (The proof of (7) is a generalization of the proof of (3), given in [24].)

[1]This algorithm fixes the initial vertex with a single edge in Prim's Algorithm [25], and hence, unlike Prim's algorithm, is not guaranteed to be optimal.

[2]Each cluster is treated as an unordered subset and is ordered in a later bid valuation stage.

[3]This cost is computed using the constrained Prim's algorithm during bid valuation stages.

9) If $C^{(2)} < C(\mathcal{A}^*)$, then $\mathcal{A}^*$ is updated (to $\mathcal{A}_i^{(s,t)}$ or $\mathcal{A}_{i,j}^{(s,t)}$).
10) If $C^{(2)} < C^{(1)}$, where $C^{(1)} = C(\mathcal{A})$, the cost before transferring or swapping, or otherwise $\alpha_S(\mathcal{A} \rightarrow \mathcal{A}_i^{(s,t)})$ or $\alpha_D(\mathcal{A} \rightarrow \mathcal{A}_{i,j}^{(s,t)})$ falls into acceptance probability, the auctioneer accepts the proposal so that $\mathcal{A}$ is updated and the cost $C(\mathcal{A})$ is put on log. Otherwise, the auctioneer declines the proposal and the auctioneer reserves the current configuration and goes back to Step 5).
11) If the auction evolution termination criteria 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 $C(\mathcal{A}^*) < C(\mathcal{A}^{(0)})$. If the criteria is not satisfied, reduce T, using $T \leftarrow T/\beta$ where $\beta > 1$ and go to Step 5).

In the implementation of SWSCA used in this study, the algorithm alternates with equal probabilities between single and dual moves. Simulation results (omitted for brevity) showed that when SWSCA alternates with equal probabilities between single and dual moves it is more efficient than using exclusively single moves or dual moves.

In order to search for the global optimum, a simulated annealing method has been adopted. Similar to the seminal work in [26], SWSCA starts with a high value of T and gradually reduces it in order to to make small variations in the task allocation while searching for the optimal allocation in $\mathcal{T}$. Although the random search in simulated annealing helps SWSCA avoid local minimum, simulated annealing algorithms are only guaranteed to converge to the global optimum if the annealing temperature T is sufficiently small [23]. However, although SWSCA relies on simulated annealing, even for small T, the use of an internal greedy algorithm (see Step 4 above) can prevent it from converging to a globally optimal solution. Hence, the primary practical value of using simulated annealing is to enable the algorithm to yield high performance solutions with reasonably fast execution times rather than guarantee asymptotic convergence to a global optimum.

C. Non-Greedy and Greedy Stochastic Clustering Auctions

Algorithm 1 describes the generic structure of the non-greedy and greedy versions of a Stochastic Clustering Auction (SCA). Both the GSSCA of [22] and the SWSCA of this paper fit in this framework. These algorithms primarily differ in line 3, 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, SWSCA is based on transfers and swaps of inter-connected tasks. All SCA algorithms can be made greedy by not allowing the uphill movements of line 8, which enable the algorithm to escape local minima. The ability to intialize and update the annealing suite in lines 1 and 10 and turn the uphill movements on and off in lines 6 and 8 provides SCA with the ability to tradeoff the converged algorithm cost with computational and communication efficiency, a novel feature of SCA. The authors' experience is that when a mission is being planned and more time is available it may be advisable to use the non-greedy version of an SCA. However, during a mission, the speed of the greedy version of an SCA may be needed.

Algorithm 1 principal mechanisms for the non-greedy and greedy versions of a Stochastic Clustering Auction

```
1:  Initialize the annealing suite.
2:  repeat
3:     Propose a reclustering.
4:     Decide whether to accept the proposed cluster.
5:     if the solution is better then
6:        Accept.
7:     else
8:        Accept with an acceptance probability (for SCA
          only). {This uphill movement is turned off for
          gSCA.}
9:     end if
10:    Update the annealing suite.
11: until The termination is reached
```

D. Use of a SWSCA for Distributed Task Allocation

If all the mission tasks are given in $\mathcal{T}$, then a SWSCA is a centralized auction. A centralized auction may make sense at the beginning of a mission, but it may not be feasible during the mission due to limited communication and the computational cost of a centralized auction. Hence, once the mission begins, it is assumed that clustering must be performed in a distributed fashion in which each robot sequentially in a given (possibly random) order becomes the auctioneer. If the distributed auction is based on optimizing the regional cost, the new global cost will be at least as small as the global cost of the initial global allocation, which motivates basing distributed SWSCA on the optimization of regional costs. In particular, each robot, sequentially or in random order, calls and clears one auction. Rounds are held repeatedly until a stable solution is reached. The auctioning process can recommence when a new task is obtained or when there is a substantial change in the existing costs.

III. EXPERIMENTAL RESULTS FOR CENTRALIZED IMPLEMENTATION OF SWSCA

This section provides simulation results for SWSCA using the multi-robot routing problem, which is a standard test domain for robot coordination using auctions [4], [7], [9], [10], [11], [12], [13], [16], [18], [19]. The task allocation is time-extended assignment such that all tasks are assigned to robots before the assignments are carried out [27]. 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 SWSCA task allocations are compared with those obtained using the Sequential (single-item) Auction (SA) and the Parallel Auction (PA), which are standard auction methods in the existing literature [13], [14], [16], [20], and their variants, the Look-Back Sequential (single-item) Auction (LBSA) and the Look-Back Parallel Auction (LBPA), which take into account the previous bids when considering the cost of the current bid in comparison with SA and PA. LBSA and LBPA sometimes yield better performance than their better known respective counterparts, SA and PA, while having similar computational requirements.

For each simulation the stochastic random scenario appears in a $10000m \times 10000m$ area. The initial robot positions were evenly distributed along one edge of the area and the speeds for each of the robots were assumed to be constant and were chosen randomly from the interval $(0m/s, 20m/s]$ assuming a uniform distribution. The cost function is a MinSum cost function in (2) corresponding to the total distance traveled or the total energy expended. Also, for each simulation the following SWSCA parameters were used: initial temperature, $T = 1000$; and termination temperature, $T_{cut} = 20$.

The communication complexity of SWSCA is measured by the number of *auction cycles* (*ACs*). Formally, an AC is one bid evaluation cycle corresponding to Steps 5-7 of Section II-B. In addition, to evaluate the performance of SWSCA the concept of *Mean Cost Improvement (MCI)* is introduced as given by Definition 1.

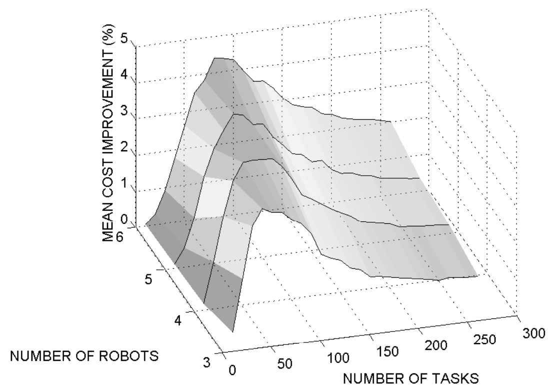
Definition 1 *For m stochastic scenarios let $\{C^{SWSCA}(i) : i = 1, \cdots, m\}$ denote the set of m costs resulting from SWSCA and let $\{C^{BestGreedy}(i) : i = 1, \cdots, m\}$ denote the set of minimum costs achievable with the greedy algorithms: SA, LBSA, PA and LBPA. The* ***Mean Cost Improvement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

$$MCI \triangleq \frac{\sum_{i}^{m} \left(\frac{C^{BestGreedy}(i) - C^{SWSCA}(i)}{C^{BestGreedy}(i)} \right)}{m}. \quad (10)$$

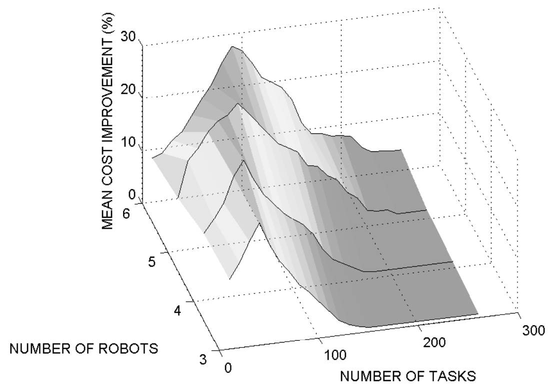
Previous studies [22], [24] reveal the performance and algorithm convergence benefits of initializing an SCA with an allocation obtained from a greedy algorithm as opposed to initializing them with a random allocation. Thus, the lowest cost allocation from the set of greedy auctions {SA,LBSA,PA,LBPA} is used to initialize SWSCA. This section studies the performance of centralized SWCA (cSWSCA), greedy centralized SWSCA (gcSWSCA), centralized GSSCA (cGSSCA), greedy centralized (gcGSSCA), and the four greedy auctions using simulations involving 1000 random scenarios for a given number of tasks and robots.

A. Simulation Results for cSWSCA and gcSWSCA with 3 Robots

The initial simulations were restricted to 3 robots with the number of tasks ranging from 5 to 100 in increments of 5. The algorithms cSWSCA and gcSWSCA were evaluated for 3 cooling schedule ratios β, representing slow (β=1.001), medium (β=1.01), and fast (β=1.1) algorithm convergence. For the smallest cooling schedule ratio, $\beta = 1.001$, the MCI of cSWSCA is in the interval [12.2%, 40.9%] and (for the same number of tasks) is always greater than the MCI for gcSWSCA, which is in the interval [6.9%,36.6%] with a maximum increase of 7.4%. In contrast, for the larger cooling schedule ratio, $\beta = 1.01$, the MCI of gcSWSCA is in the interval of [0.4%,23.3%] and is always greater than the MCI for cSWSCA, which is in the interval of [0.28%,21.2%] with a maximum increase of 3.34%. For cSWSCA the auction cycles interval changes from [3,99] to [1,4] with a maximum decrease of 96%. For medium to fast annealing, e.g. $\beta = 1.01$ or $\beta = 1.1$, gcSWSCA tended to converge faster than cSWSCA in terms of ACs by an order of magnitude, while actually exceeding cSWSCA in MCI. The reason for this is that the uphill random walk that is a part of cSWSCA is inefficient when the annealing is sufficiently fast. The purpose of the uphill random walk is to enable the optimization to escape local minimum. However, when the annealing is fast, the optimization will usually converge to a local minimum and the uphill movement simply makes the optimization less efficient.



(a) gcGSSCA

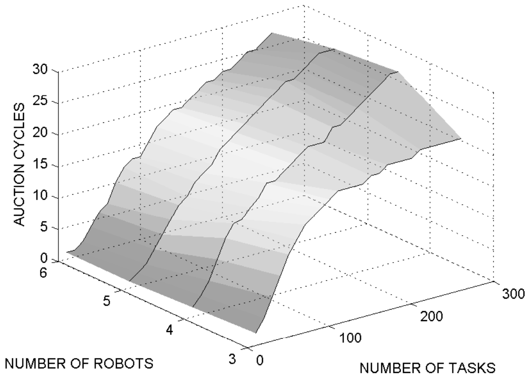


(b) gcSWSCA

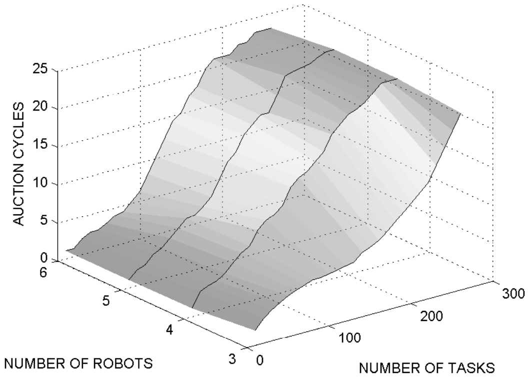
Fig. 3. Mean cost improvement (MCI) vs. the number of robots and the number of tasks for gcGSSCA and gcSWSCA for $\beta = 1.01$

B. Simulation Results for gcSWCA and gcGSSCA with a Varying Number of Robots

In the subsequent simulations the number of robots was added as a variable in the random simulations. In particular, 1000 random scenarios were again studied for a given number of robots and tasks with the number of robots now ranging from 3 to 6 and the number of tasks ranging from 10 to 260 in increments of 10. These results showed that the relationship between the MCIs and ACs of gcSWSCA observed before for 3 robots extend to an arbitrary number of robots. (The detailed results are omitted for brevity.) Second, they were used to provide a fairly comprehensive comparison of gcSWSCA and gcGSSCA (see Section III-B.1). Third, they were used to generate curves that can be used to determine the number of robots needed for a mission that is specified by some number (or range of numbers) of possible tasks in a specified region (see Section III-B.2).



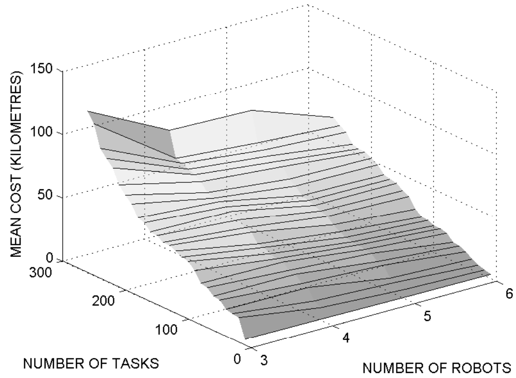
(a) gcGSSCA



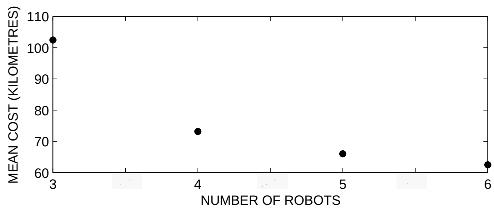
(b) gcSWSCA

Fig. 4. Average numbers of auction cycles to converge vs. the number of robots and the number of tasks for gcGSSCA and gcSWSCA for $\beta = 1.01$

1) Comparison of gcSWSCA with gcGSSCA: Fig. 3 shows that the maximum MCI for gcSWSCA is 27.7% while the maximum MCI for gcGSSCA is 4.3%. Also, Fig. 4 shows that the maximum AC for gcSWSCA is 24 while the maximum AC for gcGSSCA is 30. In general gcSWSCA is able to achieve a higher MCI (i.e., performance improvement) than gcGSSCA in a comparable number of ACs (auction cycles), revealing the efficiency of the synergic task coupling in SWSCA. It should be noted that an AC in SWSCA is more computationally expensive than an AC in GSSCA. Future work will quantify this extra expense.



(a) Mean cost vs. the number of robots and the number of tasks for gcSWSCA



(b) Frontal plane of Fig. 5(a) in 260 tasks

Fig. 5. Mean cost vs. the number of robots and the number of tasks, and its "slice" in a frontal plane for gcSWSCA with $\beta = 1.01$

2) Evaluation of gcSWSCA: Fig. 5 displays the costs (in this case for gcSWSCA 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., distance savings) can be achieved by adding a small number of robots. For example in Fig. 5(b), which shows the costs for 260 tasks, the cost corresponding to 2 robots is 102.5 km, while the costs with 4 robots improves to 73.1 km. In general for a fixed number of tasks, the corresponding "slice" of a 3-D curve such as Fig. 5(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.

IV. Experimental Results for Distributed Implementation of SWSCA

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 SWSCA (dSWSCA) approach described in Section II-D. As in Section III, random scenarios were simulated in a $10000m \times 10000m$ area and the speeds for each of the robots were assumed to be constant and were chosen randomly from the interval ($0m/s$,$20m/s$] assuming a uniform distribution. The cost function is a MINSUM cost function as in (1). The SWSCA parameters used were as before: initial temperature, $T = 1000$; termination temperature, $T_{cut} = 20$; and the cooling schedule ratio, $\beta = 1.01$.

A. Two Metrics for Evaluation of dSWSCA

The efficacy of dSWSCA is measured by comparing the resultant global cost with the corresponding gcSWSCA global cost. This leads to the following definition for *optimization efficiency*.

Definition 2 *The* ***optimization efficiency for scenario j*** *is denoted by* $\eta_j \in (0, 1]$ *and defined by* $\eta_j \triangleq \frac{C_j^*}{C_j}$*, where* C_j^* *is the global cost resulting from the application of gcSWSCA and* C_j *is the global cost resulting from the application of dSWSCA.*

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 robots.

Definition 3 *The* ***Tournament Participation Index (TPI)***[4] *for* k *robots is denoted by* $\zeta(k) \in (0, 1]$ *and defined by* $\zeta(k) \triangleq \frac{\frac{\sum_{i=1}^{k} b^2(i)}{k}}{k^2} = \frac{\sum_{i=1}^{k} b^2(i)}{k^3}$*, where* $b(i)$ *is the number of robots that participate in the regional auction in which robot* r_i *is the auctioneer. Hence* $\zeta(k)$ *is the mean of* $b^2(i)$ *for the* k *robots, normalized so that it lies in the interval* $(0, 1]$.

Note the TPI $\zeta(k) = 1$ corresponds to full communication between each of the robots.

B. Evaluation of Distributed SWSCA Using Scale Free Networks

A key issue is how to evaluate distributed SWSCA (dSWSCA) using simulations. In this section we base the simulations on robots whose communication links are determined according the topology of a scale free network (SFN) [28]. As a robot is added to a SFN, the communication links with other robots is determined probabilistically using "growth" and "preferential attachment" laws [28]. The resulting SFN networks tend to have some robots that have sparse communication links while others have more dense communication links.

A SFN network for 11 robots is illustrated in Fig. 6. This circular network was used in the simulations and it was assumed that each robot sequentially takes a turn as an auctioneer in the pattern $r_1 \to r_2 \to r_3 \to \ldots$. 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 $5000m$. The auctioning process for dSWSCA was initialized using the set of fast greedy algorithms {SA,LBSA,PA,LBPA}. In the dynamic scenarios 300 tasks were randomly given at the outset of auctioning and subsequently no tasks were changed or added. However, a new robot r_{12} is added after robot r_{11} first serves as the auctioneer. For a given number of tasks each simulation involved 1000 random scenarios. Fig. 7 showed that the mean optimization efficiency actually decreased after the new robot r_{12} was introduced due to the TPI decreasing from 0.0744 to 0.0689. However, the distributed auctioning accommodated the new robot and increased the optimization efficiency as the tournaments progressed.

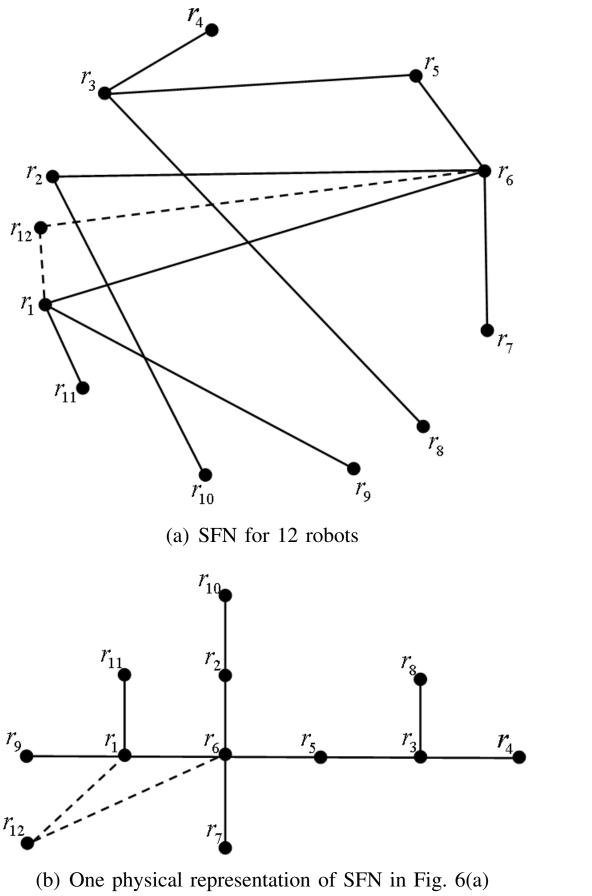


Fig. 6. A SFN communication pattern for 11 robots (The solid line represent communication links among the 11 robots while the dashed lines represent the communication links determined by the growth and preferential attachment laws when a 12th robot is added to the network.)

V. CONCLUSIONS AND FUTURE WORKS

This paper presented a novel Stochastic Clustering Auction (SCA) based on the generalized Swendsen-Wang method. The new algorithm is called the Swendsen-Wang SCA (SWCCA) and unlike the previous Gibbs Sampler SCA (GSSCA) it enables the transfer and swapping of tasks that have been connected. SCA algorithms are based on simulated annealing and have the ability to avoid local minima via uphill moves. However, for faster convergence the uphill

[4]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 $b(i)$ counts the robots instead of the links.

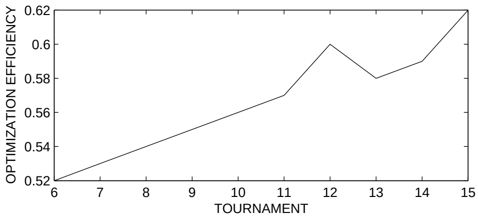


Fig. 7. Mean optimization efficiency vs. number of tournaments for a SFN of abstract auction rotation patterns in Fig. 6: 300 tasks with a new robot introduced after Tournament 12

movements may be turned off resulting in a greedy SCA. The experiences of the authors is that when a mission is first planned (and more time is available) the uphill movements should be included to increase performance. However, during a mission it may be more appropriate to use a greedy SCA.

A series of random simulations showed that SWSCA was able to obtain significantly greater cost improvements than GSSCA for both the greedy and non-greedy cases. Distributed SWSCA, denoted as dSWSCA, was based on applying the greedy SWSCA regionally and enabling each robot to serve as the auctioneer in a rotation pattern. The performance of dSWCA was evaluated in random simulations using communication links derived from a scale free network. The simulation results showed that dSWSCA continuously improved the global performance each time one of the robots completed its tournament (i.e., its auction process).

The current Swendsen-Wang SCA is only valid for task allocation using MinSum cost functions. Future work will develop an algorithm that can also be used for problems with MinMax cost functions.

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] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[2] S. de Vries and R. V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, Summer 2003.

[3] H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-2000)*, pages 22–29, 2000.

[4] 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.

[5] M. Mito and S. Fujita. On heuristics for solving winner determination problem in combinatorial auctions. *Journal of Heuristics*, 10(5):507–523, 2004.

[6] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, February 2002.

[7] 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.

[8] S. S. C. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1234–1239, 1999.

[9] L. Brunet. Consensus-based auctions for decentralized task assignment. Master thesis, Massachusetts Institute of Technology, June 2008.

[10] H. Choi, L. Brunet, and J. P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 25:912–926, August 2009.

[11] C. M. Clark, R. Morton, and G. A. Bekey. Altruistic relationships for optimizing task fulfillment in robot communities. In Hajime Asama, Haruhisa Kurokawa, Jun Ota, and Kosuke Sekiyama, editors, *Distributed Autonomous Robotic Systems 8*, pages 261–270, Tsukuba, Ibaraki, Japan, 2008.

[12] 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.

[13] 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.

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

[15] M. Golfarelli, D. Maio, and S. Rizzi. A task-swap negotiation protocol based on the contract net paradigm. Technical Report 005-97, 1997. CSITE (Research Centre for Informatics and Telecommunication Systems, associated with the University of Bologna, Italy).

[16] 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.

[17] J. McLurkin and D. Yamins. Dynamic task assignment in robot swarms. In *Robotics: Science and Systems*, pages 129–136, 2005.

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

[19] 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.

[20] 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.

[21] E. H. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley and Sons, London, 1997.

[22] 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 Hajime Asama, Haruhisa Kurokawa, Jun Ota, and Kosuke Sekiyama, editors, *Distributed Autonomous Robotic Systems 8*, pages 345–354, Tsukuba, Ibaraki, Japan, 2008.

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

[24] A. Barbu and S. Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1239–1253, 2005.

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

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

[27] 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.

[28] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.