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

Kai Zhang, Emmanuel G. Collins, Adrian Barbu

Abstract—It will be 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, allowing 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 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 methods 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-E. 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] (see Section II-C).

An additional limitation of the previously developed auction methods is that they do not provide a mechanism to enable the performance after 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] (see Sections II-D and III-A).

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 corresponding to the robot of the identical shading.
The Gibbs Sampler requires flipping two connected sets of tasks, $S_1$ and $S_2$, from light circles in the current allocation to red circles in the optimal allocation. The probability of flipping a task from light to red 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 red 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 image segmentation and stereo 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 $R$ denote a set of $k$ heterogeneous robots, and $T$ denote a set of $n$ tasks, i.e. $R = \{r_1, r_2, \ldots, r_k\}$ and $T = \{t_1, t_2, \ldots, t_n\}$. Also, let $A$ denote the allocation, $A = \{a_1, a_2, \ldots, a_k\}$, where $a_s$ is a cluster of tasks, $\bigcup_{s=1}^{k} a_s = T$, $a_s \cap a_t = \emptyset, (s \neq t)$ and the cluster $a_s$ is assigned to robot $r_s$. Each $a_s$ is decomposed into $n_s$ connected components $a_{si}$ such that $a_s = \bigcup_{i=1}^{n_s} a_{si}$. Let $CP$ denote the entire set of connected components such that $CP = \{a_{si} : s = 1, 2, \ldots, k; i = 1, \ldots, n_s\}$. $A(a_{si}) = t$ is used to denote the allocation of the connected component $a_{si}$ from robot $r_s$ to robot $r_t$. The cost associated with $A$ is given by either

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

or

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

where $c_s(a_s)$ is the minimum cost for robot $r_s$ 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 $C(A)$. 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 $C(A)$ 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 $A$ for $k$ clusters and to reduce or probabilistically hillclimb $C(A)$ by rearranging the tasks $T$ 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 $C(A)$ of the resulting allocations $A$ (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 algorithm is described below:

1) The auctioneer partitions $T$ into $k$ clusters to form an initial allocation $A^{(0)} = \{a^{(0)}_1, a^{(0)}_2, \ldots, a^{(0)}_k\}$, where each cluster $a^{(0)}_s$ is an unordered subset of $T$. Let $A = A^{(0)}$ and $A^* = A^{(0)}$. ($A$ is the current algorithm allocation, while $A^*$ is the allocation that has the lowest cost.)

2) Each robot $r_p \in R$ ($p = 1, 2, \ldots, k$) uses a “constrained Prim’s Algorithm” (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 $T$. The auctioneer computes the global cost $C(A)$ using (1) or (2) and sets a high temperature $T$.

3) Each robot $r_p \in R$ ($p = 1, 2, \ldots, k$) constructs an adjacency graph $G_0(a^{(0)}_p) = (T(a^{(0)}_p), \mathcal{E}_0(a^{(0)}_p))$ for each cluster $a^{(0)}_p$, where $\mathcal{E}_0(a^{(0)}_p)$ is the edge set of $\mathcal{T}(a^{(0)}_p)$.

4) Each robot $r_p \in R$ ($p = 1, 2, \ldots, k$) submits $l_{\min}(a^{(0)}_p)$ to the auctioneer, where $l(a^{(0)}_p) = \min_{e \in \mathcal{E}(a^{(0)}_p)} l(e)$, and $l(e)$ denotes the Euclidean length of the edge $e$.

5) For each $e \in \mathcal{E}(A)$, the auctioneer turns the edge $e$ off with a probability $1 - p_e$, e.g., $p_e = \frac{l_{\min}(a^{(0)}_p)}{l(e)}$, where $l_{\min} = \min\{l_{\min}(a^{(1)}_1), l_{\min}(a^{(2)}_2), \ldots, l_{\min}(a^{(k)}_k)\}$. For $s = 1, 2, \ldots, k$ this results in the decomposition of $a_s \in A$ into $n_s$ connected components $a_{si}$ such that $a_s = \bigcup_{i=1}^{n_s} a_{si}$.

6) The auctioneer collects all the connected components in the set $CP$.

7) The auctioneer rearranges the clusters with equal probabilities selecting either a single move (a) or a dual move (b) among $CP$ after randomly selecting two clusters $a_s$ and $a_t$ for task transfers or swaps.

a) Single Move (Connected Component Transfer): Select a connected component $a_{si} \in CP$ from robot $r_s$ with a probability in $q(a_{si}|CP)$, e.g., $q(a_{si}|CP) = \frac{1}{|CP|}$ in a uniform distribution. Assume that $a_{si}$ is reassigned to robot $r_p$ with a probability in $q(A(a_{si}) = p(a_{si}, A)$ for $p = 1, 2, \ldots, k$ $(p \neq s)$, e.g., $q(A(a_{si}) = p(a_{si}, A) = \frac{1}{k}$ in a uniform distribution, resulting in the new allocation $A_{s;p} = \{a_{si}^{(0)}, a_{p}^{(0)}\}$ that has two modified clusters $a_{si}^{(-i)}$ and $a_{p}^{(+i)}$. Assume that robot $r_s$ computes $c_s(a_{si}^{(-i)})$ and robot $r_p$ computes $c_p(a_{p}^{(+i)})$, which the auctioneer uses to compute the corresponding cost $C(A_{s;p}^{(s,t)})$ (based on (1) or (2)). The probability $\alpha_S(A \rightarrow A_{s;i}^{(s,t)})$ of the acceptance of the transfer of the connected component $a_{si}$ from robot $r_s$ to robot $r_t$ is given by [24]

$$\alpha_S(A \rightarrow A_{s;i}^{(s,t)}) = \min(1, \alpha_S^3 \cdot \alpha_S^2 \cdot \alpha_S^3),$$

where

$$\alpha_S^3 = \frac{\prod_{e \in \mathcal{E}(a_{si}, a_{p})}(1 - p_e)}{\prod_{e \in \mathcal{E}(a_{si}, a_{-si})}(1 - p_e)},$$

$$\alpha_S^2 = \frac{P(A(a_{si}) = s|a_{si}, A) \cdot \exp(-C(A^{(t)}_s)/T)}{P(A(a_{si}) = t|a_{si}, A)},$$

$$\alpha_S^3 = \frac{\exp(-C(A^{(t)}_s)/T)}{\sum_{p=1, p \neq s} \exp(-C(A_{s;p}^{(s,t)})/T)}.$$
(The proof of (7) can be generalized from the proof of (3) in [24].) If \( C(A_{t,j}^{(s,t)}) \) is less than \( C(A^*) \), then the new \( A^* \) is updated to \( A_{t,j}^{(s,t)} \).

8) The auctioneer accepts the proposal with probability \( \alpha_G(A \rightarrow A_i^{(s,t)}) \) or \( \alpha_D(A \rightarrow A_{t,j}^{(s,t)}) \) so that \( A \) is updated and the cost \( C(A) \) is put on log. Otherwise, the auctioneer declines the proposal and the auctioneer keeps the current configuration and goes back to Step 5.

9) 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 \( A^* \) with final cost \( C(A^*) \leq C(A^{(0)}) \). If the criterion is not satisfied, reduce \( T \), using \( T \leftarrow T/\beta \) where \( \beta > 1 \) and go to Step 5.

C. Further Discussion of GSSCA and SWSCA

In the implementation of GSSCA in [22] and SWSCA used in this study, the algorithms alternate with equal probabilities between single and dual moves. Simulation results (omitted for brevity) showed that when a SCA 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 [25], a SCA starts with a high value of \( T \) and gradually reduces it in order to make small variations in the task allocation while searching for the optimal allocation in \( \mathcal{T} \). Although simulated annealing and the random nature of the search help in avoiding local minima, the convergence to a global minimum is difficult to establish. As described in [23], the output of this algorithm is a Markov chain that is neither homogeneous nor convergent to a stationary chain. If the temperature \( T \) is decreased slowly, then the chain is guaranteed to converge to a global minimum. Note that the use of an internal greedy algorithm (see Step 2 above) is likely to prevent the computation from converging to desired global optimum even if the annealing procedure converges. Hence, the primary purpose 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.

D. 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). The greedy version of SCA is called here gSCA. Both the GSSCA of [22] and the SWSCA of this paper fit in Algorithm 1. 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 interconnected tasks. All SCA algorithms can be made greedy by not allowing the uphill movements associated with line 8, which enable the algorithm to escape local minima. The ability to initialize and update the annealing suite in lines 1 and 10 and turn the uphill movements on and off in line 6 provides SCA with the ability to trade off the converged algorithm cost with computational and communication efficiency. This is 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 a SCA. However, during a mission, a greedy 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 (for gSCA only) then
6: Accept.
7: else
8: Accept with an acceptance probability.
9: end if
10: Update the annealing suite.
11: until The termination is reached

E. 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 [26]. 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 x 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 \( \text{MINSUM} \) cost function in (1) 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, \( T_{\text{cut}} = 20 \).

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)-8) of Section II-B. In addition, to evaluate the performance of SWSCA the concept of \textit{Mean Cost Improvement (MCI)} is introduced as given by Definition 1.

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

\[
MCI = \frac{1}{m} \sum_{q=1}^{m} \left( \frac{C_{\text{BestGreedy}}^{\text{BestGreedy}}(q) - C_{\text{SWSCA}}^{\text{SWSCA}}(q)}{C_{\text{BestGreedy}}^{\text{BestGreedy}}(q)} \right).
\]

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 \( \{\text{SA}, \text{LBSA}, \text{PA}, \text{LBPA}\} \) is used to initialize SWSCA. This section studies the performance of centralized SWCA (cSWSCA), greedy centralized SWSCA (gcSWSCA), centralized GSSCA (cgGSSCA), greedy centralized (gcGSSCA), and the four greedy auctions using simulations involving 1000 random scenarios for a given number of tasks and robots.

\textbf{A. Simulation Results for cSWSCA and gcSWSCA with 3 Robots}

Among the annealing suite the cooling schedule rate \( \beta \) is used to control the annealing speed, i.e. how fast the auction process terminates. 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 \( \beta \), representing slow (\( \beta = 1.001 \)), medium (\( \beta = 1.01 \)), and fast (\( \beta = 1.1 \)) 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 [11.05%, 36.6%] with a maximum increase of 21.63%. In contrast, for the larger cooling schedule ratio, \( \beta = 1.01 \), the MCI of gcSWSCA is in the interval of [10.51%, 23.3%] and is always greater than the MCI for cSWSCA, which is in the interval of [10.25%, 22.2%] with a maximum increase of 13.47%. 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. These general trends were also seen when the number of robots was varied. In the subsequent sections, therefore, the greedy version of SWSCA was utilized for
\( \beta = 1.01 \) since it is shown that good performance is achieved in medium annealing with a reasonable number of ACs.

B. Simulation Results for gcSWSCA 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 2 to 10 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).

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 127.1 km, while the costs with 6 robots improves to 96.2 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-E. 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 \(q\) is denoted by \(\eta_q \in (0, 1]\) and defined by 

\[
\eta_q = \frac{C_{q}^g - C_{q}^{D}}{C_{q}^g}
\]

where \(C_{q}^g\) is the global cost resulting from the application of gcSWSCA and \(C_{q}^{D}\) 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)\(^5\) for \(k\) robots is denoted by \(\zeta(k) \in (0, 1]\) and defined by

\[
\zeta(k) = \frac{\sum_{p=1}^{k} b^2(p)}{k^2} = \frac{\sum_{p=1}^{k} b^2(p)}{k^2}
\]

where \(b(p)\) is the number of robots that participate in the regional auction in which robot \(r_p\) 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 \((0, 1]\).

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 dSWSCA using simulations. In this section we base the simulations on robots whose communication links are determined according to the topology of a scale free network (SFN) [27]. As a robot is added to a SFN, the communication links with other robots is determined probabilistically using “growth” and “preferential attachment” laws [27]. 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 \(r_1\) serves as the auctioneer for the second time. 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.

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 (SWSCA) 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 movements may be turned off resulting in a greedy SCA.

\(^4\)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 \(b(p)\) counts the robots instead of the links.
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 dSWSCA 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 homogeneous teams. Future work will develop an algorithm that uses concepts from SWSCA to achieve greater efficiency than GSSCA in allocating tasks for heterogeneous teams.

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


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, which causes a decrease in TPI

![Graph showing mean optimization efficiency vs. number of tournaments](image-url)