

Distributed Control of Autonomous Swarms by Using Parallel Simulated Annealing Algorithm

Wei Xi and John S. Baras

Abstract—In early work of the authors, it was shown that Gibbs sampler based sequential annealing algorithm could be used to achieve self-organization in swarm vehicles based only on local information. However, long travelling time presents barriers to implement the algorithm in practice. In this paper we study a popular acceleration approach, the parallel annealing algorithm, and its convergence properties. We first study the convergence and equilibrium properties of the synchronous parallel sampling algorithm. A special example based on a battle field scenario is then studied. Sufficient conditions that the synchronous algorithm leads to desired configurations (global minimizers) are derived. While the synchronized algorithm reduces travelling time, it also raises delay and communication cost dramatically, in order to synchronize moves of a large group of vehicles. An asynchronous version of the parallel sampling algorithm is then proposed to solve the problem. Convergence properties of the asynchronous algorithm are also investigated.

I. INTRODUCTION

In recent years, with the rapid advances in sensing, communication, computation, and actuation capabilities, groups (or swarms) of autonomous unmanned vehicles (AUVs) are expected to cooperatively perform dangerous or explorative tasks in a broad range of potential applications [1]. Due to the large scales of vehicle networks and bandwidth constraints on communication, distributed control and coordination methods are especially appealing [2], [3], [4], [5].

A popular distributed approach is based on artificial potential functions (APF), which encode desired vehicle behaviors such as inter-vehicle interactions, obstacle avoidance, and target approaching [6], [7], [8], [9]. Despite its simple, local, and elegant nature, this approach suffers from the local minima entrapment problem [10]. Researchers attempted to address this problem by designing potential functions that have no other local minima [11], [12], or escaping from local minima using ad hoc techniques, e.g., random walk [13].

An alternative approach to dealing with the local minima problem was explored using the concept of Markov Random Fields (MRFs) and simulated annealing (SA) approach by Baras and Tan [14]. Traditionally used in statistical

mechanics and in image processing [15], MRFs were proposed to model swarms of vehicles. Similar to the APF approach, global objectives and constraints (e.g., obstacles) are reflected through the design of potential functions. The movement of vehicles is then decided using a Gibbs sampler based SA approach. The SA algorithm has also been adopted for UAV preposition in [16].

Theoretical studies and simulations have shown that, with a special sequential sampling, the global goals can be achieved despite the presence of local minima in the potentials [17], [18]. However, the maintenance of global indices, which is required for sequential sampling, in large vehicle networks, is difficult when there exist node failures. Moreover, long maneuvering time, which is due to the fact that only one vehicle moves at each time instance, presents difficulties in practice.

The above problems can be resolved using parallel sampling [14], i.e., each node in the vehicle swarm executes the local Gibbs sampling in parallel. Parallel sampling techniques have been studied for many years in order to accelerate the slow convergence rate of the sequential simulated annealing algorithm [19]. It is usually required that nodes update their locations at the same time clock (synchronously). However, synchronization causes communication cost and delay, which degrade performance. This can be resolved by using asynchronous parallel sampling, i.e., each vehicle uses its own clock to do the local sampling.

In this paper, we first investigate the convergence properties of a synchronous parallel sampling algorithm. In the analysis of the asynchronous parallel algorithm, the fact that there is a “time-varying” number of active nodes presents challenges. Fortunately, by applying a partially parallel model in [20], the asynchronous algorithm could be described by a homogeneous Markov chain. The convergence of the asynchronous parallel algorithm then follows. Finally, a special example based on a battle field scenario was investigated. Sufficient conditions that guarantee the optimality of the parallel sampling algorithm were analyzed.

II. REVIEW OF GIBBS SAMPLER BASED ALGORITHM

A. MRFs and Gibbs Sampler

One can refer to, e.g., [15], [21], for a review of MRFs. Let S be a finite set of cardinality σ , with elements indexed by s and called *sites*. For $s \in S$, let Λ_s be a finite set called the *phase space* for site s . A *random field* on S is a collection $X = \{X_s\}_{s \in S}$ of random variables X_s taking values in Λ_s . A *configuration* of the system is $x = \{x_s, s \in S\}$, where $x_s \in \Lambda_s, \forall s$. The product space $\Lambda_1 \times \cdots \times \Lambda_\sigma$ is

This research was supported by the Army Research Office under the ODDR&E MURI01 Program Grant No. DAAD19-01-1-0465 to the Center for Networked Communicative Control Systems (through Boston University), and under ARO Grant No. DAAD190210319.

W. Xi and J. S. Baras are with the Institute for Systems Research and the Department of Electrical & Computer Engineering, University of Maryland, College Park, MD 20742, USA. {wxi, baras}@isr.umd.edu

called the *configuration space*. A *neighborhood system* on S is a family $\mathcal{N} = \{\mathcal{N}_s\}_{s \in S}$, where $\forall s, r \in S$,

- $\mathcal{N}_s \subset S$,
- $s \notin \mathcal{N}_s$, and
- $r \in \mathcal{N}_s$ if and only if $s \in \mathcal{N}_r$.

\mathcal{N}_s is called the *neighborhood* of site s . The random field X is called a *Markov random field* (MRF) with respect to the neighborhood system \mathcal{N} if, $\forall s \in S$, $P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{N}_s)$.

A random field X is a *Gibbs random field* if it has the Gibbs distribution:

$$P(X = x) = \frac{e^{-\frac{U(x)}{T}}}{Z}, \quad \forall x,$$

where T is the temperature variable (widely used in simulated annealing algorithms), $U(x)$ is the potential (or energy) of the configuration x , and Z is the normalizing constant, called the *partition function*: $Z = \sum_x e^{-\frac{U(x)}{T}}$. One then considers the following useful class of potential functions $U(x) = \sum_{s \in \Lambda} \Phi_s(x)$, which is a sum of individual contributions Φ_s evaluated at each site. The Hammersley-Clifford theorem [21] establishes the equivalence of a Gibbs random field and an MRF.

The *Gibbs sampler* belongs to the class of *Markov Chain Monte Carlo* (MCMC) methods, which sample Markov chains leading to stationary distributions. The algorithm updates the configuration by visiting sites sequentially or randomly following a certain *proposal* distribution [15], and by sampling from the local specifications of a Gibbs field. A *sweep* refers to one round of sequential visits to all sites, or σ random visits under the proposal distribution.

The convergence of the Gibbs sampler was studied by D. Geman and S. Geman in the context of image processing [22]. There it was shown that as the number of sweeps goes to infinity, the distribution of $X(n)$ converges to the Gibbs distribution Π . Furthermore, with an appropriate cooling schedule, simulated annealing using the Gibbs sampler yields a uniform distribution on the set of minimizers of $U(x)$. Thus the global objectives could be achieved through appropriate design of the Gibbs potential function.

B. Problem Setup for Self-organization of Multiple Vehicles

Consider a 2D mission space (the extension to 3D space is straightforward), which is discretized into a lattice of cells. For ease of presentation, each cell is assumed to be square with unit dimensions. One could of course define cells of other geometries (e.g., hexagons) and of other dimensions (related to the coarseness of the grid) depending on the problems at hand. Label each cell with its coordinates (i, j) , where $1 \leq i \leq N_1, 1 \leq j \leq N_2$, for $N_1, N_2 > 0$. There is a set of vehicles (or *mobile nodes*) S indexed by $s = 1, \dots, \sigma$ on the mission space. To be precise, each vehicle s is assumed to be a point mass located at the center of some cell (i_s, j_s) , and the position of vehicle s is taken to be $p_s = (i_s, j_s)$. At most one vehicle is allowed to stay in each cell at any time instant.

The distance between two cells, (i_a, j_a) and (i_b, j_b) , is defined to be

$$R \triangleq \|(i_a, j_a) - (i_b, j_b)\| = \sqrt{(i_a - i_b)^2 + (j_a - j_b)^2}.$$

There might be multiple *obstacles* in the space, where an obstacle is defined to be a set of adjacent cells that are inaccessible to vehicles. For instance, a ‘‘circular’’ obstacle centered at $p^{ok} = (i^{ok}, j^{ok})$ with radius R^{ok} can be defined as $O \triangleq \{(i, j) : \sqrt{(i - i^{ok})^2 + (j - j^{ok})^2} \leq R^o\}$. The *accessible area* is the set of cells in the mission space that are not occupied by obstacles. An *accessible-area graph* can then be induced by letting each cell in the accessible area be a vertex and connecting neighboring cells with edges. The mission space is *connected* if the associated accessible-area graph is connected, which will be assumed in this paper. There can be at most one *target area* in the space. A target area is a set of adjacent cells that represent desirable destinations of mobile nodes. A ‘‘circular’’ target area with its center at p^g can be defined similarly as a ‘‘circular’’ obstacle. An example mission scenario is shown in Fig. 1.

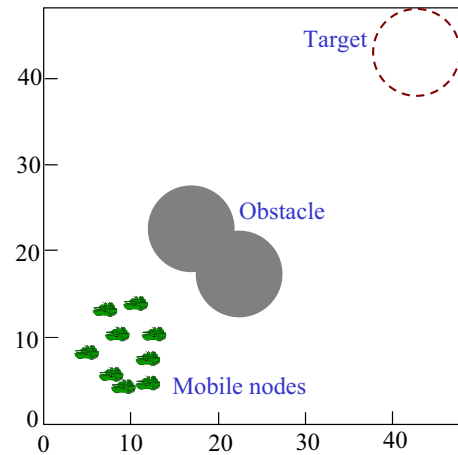


Fig. 1. An example mission scenario with a circular target and a nonconvex obstacle (formed by two overlapping circular obstacles). Note since the mission space is a discretized grid, a cell is taken to be within a disk if its center is so.

In this paper all vehicles are assumed to be identical. Each vehicle has a *sensing range* R_s : it can detect whether a cell within distance R_s is occupied by some node or obstacle through sensing or direct inter-vehicle communication. The motion decision of each node s depends on other nodes located within distance R_i ($R_i \leq R_s$), called the *interaction range*. These nodes form the set \mathcal{N}_s of *neighbors* of node s . A node can travel at most R_m ($R_m \leq R_s$), called *moving range*, within one move. See Fig. 2 for illustration of these range definitions.

The neighborhood system defined earlier naturally leads to a dynamic graph, where each vehicle represents a vertex of the graph and the neighborhood relation prescribes the edges between vehicles. An MRF can then be defined on the graph, where each vehicle s is a site and the associated phase space Λ_s is the set of all cells located within the

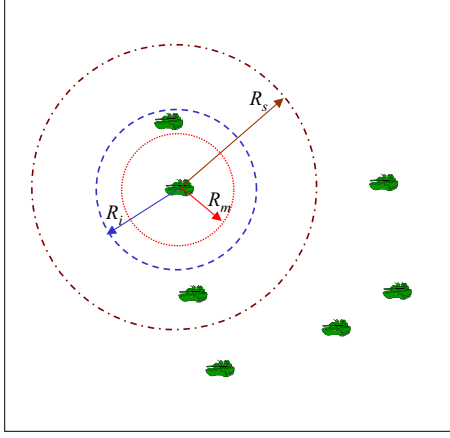


Fig. 2. Illustration of the sensing range R_s , the interaction range R_i , and the moving range R_m .

moving range R_m from location p_s and not occupied by obstacles or other vehicles. The configuration space of the MRF is denoted as \mathcal{X} .

The Gibbs potential $U(x) = \sum_s \Phi_s(x)$, where $\Phi_s(x)$ is considered to be a summation of all clique potentials $\Psi_c(x)$, and depends only on x_s and $\{x_r, r \in \mathcal{N}_s\}$. The clique potentials $\Psi_c(x)$ are used to describe local interactions depending on applications. Specifically,

$$\Phi_s(x) = \sum_{c \ni s} \Psi_c = \Psi_{\{s\}}(x_s) + \sum_{r \in \mathcal{N}_s} \Psi_{\{s,r\}}(x_s, x_r). \quad (1)$$

In the battle field scenario, the potential function $\Phi_s(x)$ consists of three terms each reflecting one goal or constraint.

$$\Phi_s(x) = \lambda_g J_s^g + \lambda_o J_s^o + \lambda_n J_s^n,$$

for the attraction from the target area, the repelling from obstacles, and the pulling force from neighbors, respectively, and $\lambda_g, \lambda_o, \lambda_n$ are the corresponding weighting coefficients for adjusting the potential surface. In particular, the following potential functions are used for each term:

$$\begin{aligned} J_s^g &= \|x_s - p^g\| \\ J_s^o &= \sum_{k=1}^K \frac{1}{\|x_s - p^{ok}\|} \\ J_s^n &= \sum_{t \in \mathcal{N}_s} \|x_s - x_t\|. \end{aligned} \quad (2)$$

There are important differences between a classical MRF introduced in Subsection II-A and the MRF defined for the vehicle networks. In a classical MRF, both the phase space Λ_s and the neighborhoods \mathcal{N}_s are time-invariant; however, for a vehicle network, both Λ_s and \mathcal{N}_s depend on the dynamic graph and therefore vary with time. These differences prevent the classical MRF theory from being adopted directly for convergence analysis.

III. CONVERGENCE ANALYSIS OF THE PARALLEL SAMPLING ALGORITHM

In [18], a two-step sequential sampling algorithm was proposed to coordinate the maneuvers of vehicle swarms to

desired configuration(s). As noted in section I, sequential location updating leads to an extremely slow convergence rate. Furthermore, global indexing is challenging for a large vehicle network when there exist node failures. It is natural to resolve these disadvantages and difficulties by adopting the parallel sampling algorithm, i.e., vehicles synchronously update their locations according to the local statistics.

However, synchronous parallel sampling requires an explicit global clock to have swarm vehicles move synchronously, which increases the delay and the communication cost dramatically for a large group of vehicles. Asynchronous parallel sampling is then adopted to avoid these penalties. In this section, the convergence properties of both synchronous and asynchronous algorithms are studied.

A. Synchronous parallel sampling algorithm

Using the synchronous parallel sampling algorithm, vehicles are synchronized to update their locations as follows:

- Step 1. Pick a cooling schedule $T(\cdot)$ and the total number N of annealing steps. Let annealing step $n=1$;
- Step 2. Conduct location updates for node 1 through S in *parallel* at the same time clock, where node $s, 1 \leq s \leq S$, performs the following:

- Determine the set L_s , of candidate locations, for the next move:

$$L_s = \Lambda_s \cap \{(i, j) : \sqrt{(i - i_s)^2 + (j - j_s)^2} \leq R_m\},$$

where Λ_s represents the set of cells not occupied by other vehicles or obstacles;

- When two neighboring vehicles ($s < s'$) have conflict in their candidate locations, i.e., $L_s \cap L_{s'} \neq \emptyset$, the vehicle with lower index updates its candidate locations to $L_s \cap L_{s'}^c$. Repeat this procedure until $L_s \cap L_{s'} = \emptyset$, for all $s \neq s'$.

- For each vehicle s evaluate the potential function for every $l \in L_s$,

$$\Phi_s(x_s = l, X(S \setminus s) = x(S \setminus s))$$

where $S \setminus s$ denotes the complement of s in S . Then update the location of each vehicle s in parallel by sampling the local distribution

$$p(z) = \frac{\exp(-\frac{\Phi_s(x_s=l, X(S \setminus s)=x(S \setminus s))}{T(n)})}{\sum_{l' \in L_s} \exp(-\frac{\Phi_s(x_s=l', X(S \setminus s)=x(S \setminus s))}{T(n)})}.$$

- Step 3. Let $n = n + 1$. If $n = N$, stop; otherwise go to Step 2.

For a fixed temperature T , the underlying mathematical model of the synchronous parallel sampling algorithm is a homogenous Markov chain defined by

$$\begin{aligned} &P_T(x(n+1)|x(n)) \\ &= \prod_{s \in S} (p(x_s = x_s(n+1) | x_{\mathcal{N}_s} = x_{\mathcal{N}_s}(n))) \\ &= \prod_{s \in S} \frac{e^{-\frac{\Phi_s(x_s=x_s(n+1), x_{\mathcal{N}_s}=x_{\mathcal{N}_s}(n))}{T(n)}}}{\sum_{l \in L_s(x(n))} e^{-\frac{\Phi_s(x_s=l, x_{\mathcal{N}_s}=x_{\mathcal{N}_s}(n))}{T(n)}}} \end{aligned} \quad (3)$$

where $\|x_s(n+1) - x_s(n)\| \leq R_m$ for all $s \in S$. $\Phi_s(x_s = x_s(n+1), x_{\mathcal{N}(s)} = x_{\mathcal{N}(s)}(n))$ is the local energy which could be evaluated by vehicle s with only local information.

Proposition 3.1: For a fixed temperature T , the homogeneous Markov chain (3) has a unique invariant distribution Π_T . From any initial distribution ν_0

$$\lim_{n \rightarrow \infty} \nu_0 P_T^n = \Pi_T \quad (4)$$

Sketch of Proof. Due to the connectivity of the accessible area, there exists at least one path between any two configurations x and y (i.e., a sequence of multiple moves $\{x, x_1, \dots, y\}$), and the shortest path is bounded by τ moves for some finite τ . This implies that P_T has a *strictly positive power* P_T^τ , i.e., the τ -step Markov chain reaches each state with positive probability from any state. The irreducibility and aperiodicity of the kernel then follows. Hence the Markov chain is ergodic and has a unique invariant distribution Π_T for a fixed temperature T [23]. \square

Picking an appropriate cooling schedule $T(n)$ and τ , the simulated annealing algorithm yields a unique distribution Π_∞ . This is made precise by the following theorem.

Theorem 3.1: Let $\tilde{U}(x, y) : X \times X \rightarrow \mathbb{R}$ be an induced energy function defined on the clique potentials

$$\tilde{U}(x, y) = \begin{cases} \sum_{s \in S} \sum_{c \ni s} \Psi_c(y_s, x_{S \setminus s}), & \text{when } y \in \mathcal{N}^m(x); \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $\mathcal{N}^m(x) = \{z \in X : \forall s, \|z_s - x_s\| \leq R_m\}$. Let $\tilde{\Delta}$ be:

$$\tilde{\Delta} \triangleq \max_{y, z \in \mathcal{N}^m(x)} |\tilde{U}(x, y) - \tilde{U}(x, z)|.$$

Let $T(n)$ be a cooling schedule decreasing to 0, so that eventually,

$$T(n) \geq \frac{\tau \tilde{\Delta}}{\ln n}.$$

Let $Q_n = P_T^\tau$. Then for any initial distribution ν ,

$$\lim_{n \rightarrow \infty} \nu Q_1 \cdots Q_n \rightarrow \Pi_\infty, \quad (6)$$

where Π_∞ is the limit distribution of (4) as T tends to zero. In particular,

$$\lim_{T \rightarrow 0} \Pi_T(x) = \Pi_\infty(x). \quad (7)$$

Proof. Let $\alpha_x = \min_{y \in \mathcal{N}^m(x)} \tilde{U}(x, y)$. From (3), we have

$$P_T(x, y) = \frac{\exp(-\frac{\tilde{U}(x, y) - \alpha_x}{T})}{\sum_{z \in \mathcal{N}^m(x)} \exp(-\frac{\tilde{U}(x, z) - \alpha_x}{T})} \leq \frac{e^{-\frac{\tilde{\Delta}}{T}}}{|\mathcal{N}^m(x)|},$$

where $|\mathcal{N}^m(x)|$ denotes the cardinality of the configuration space $\mathcal{N}^m(x)$. Following analogous arguments to those in the proof of Theorem 4.2 in [17], one can show

$$c(Q_n) \leq 1 - \lambda e^{-\frac{\tau \tilde{\Delta}}{T(n)}},$$

where $c(Q_n)$ denotes the contraction coefficient of Q_n , and $\lambda = \frac{|\mathcal{X}|}{|\mathcal{N}^m(x)|^\tau}$. Similarly, one can prove the claim (6). \square

Remark 3.1: For the parallel sampling algorithm, an explicit expression for the invariant distribution (4) is generally lacking. It is hard to analytically study the equilibrium properties. Here, we offer some brief comments.

Let Ω^0 be the set of limiting configuration(s) which is defined by

$$\Omega^0 \triangleq \{x : \Pi_\infty(x) > 0\}. \quad (8)$$

Let Ω^L be the set of all the local minima of U . Then we have $\Omega^0 \subset \Omega^L$. If the potential function U is ‘‘well behaved’’, i.e., $\{x^* : U(x^*) = \min_x U(x)\} \subset \Omega^0$, there is a positive chance that the parallel annealing algorithm leads the final configuration to x^* as temperature tends to zero, which is confirmed by extensive simulations in [14]. In section IV, we analytically study the limiting configurations for a special example.

B. Asynchronous parallel sampling algorithm

The asynchronous parallel sampling algorithm works similarly as the synchronous version, except each vehicle s makes moves independently by following its own time clock $t^s = \{t_1^s, t_2^s, \dots\}$. Thus, at one time instance n , only a subset of vehicles make a move. The transition probability from configuration $x(n)$ to $x(n+1)$ can be written down as follows

$$\begin{aligned} \tilde{P}_T(x(n+1)|x(n)) \\ = \prod_{s: n \in t^s} (p_T(x_s = x_s(n+1)|x_{\mathcal{N}(s)} = x_{\mathcal{N}(s)}(n))). \end{aligned}$$

Clearly this formulation leads to an inhomogeneous Markov chain. In general, an inhomogeneous Markov chain may not have a unique stationary distribution. This presents challenges in convergence analysis. To deal with this difficulty, we adopt the partial parallel model in [20] and model the asynchronous parallel algorithm as a hierarchical Markov chain.

Let $t = \cup_{s \in S} t^s$ denote the set of updating times for all vehicles. Clearly, t is a countable set. For each time instance $t_i \in t$, each vehicle s has a probability p_s to make a move, which is defined by

$$p_s \triangleq \lim_{|t| \rightarrow \infty} \frac{|t^s|}{|t|},$$

where $|t^s|$ and $|t|$ denote the cardinality of t^s and t respectively. For the synchronous case, $p_s \equiv 1$; whereas, for the asynchronous one $0 < p_s < 1$. Then, the associated Markov chain kernel P_T can be expressed as

$$\begin{aligned} \tilde{P}_T(x(n+1)|x(n)) \\ = \prod_{s \in S} ((1 - p_s) \mathbf{1}_{x_s(n+1) = x_s(n)} + p_s P_T(x(n+1)|x(n))) \end{aligned}$$

Since the kernel (9) defines a homogeneous Markov chain, it follows from proposition 3.1, that the Markov chain has a unique stationary distribution $\tilde{\Pi}_T$ for a fixed temperature. Then, using a similar argument as in theorem 3.1, with an appropriate cooling schedule, the asynchronous parallel annealing algorithm converges to a unique distribution $\tilde{\Pi}_\infty$, where $\tilde{\Pi}_\infty = \lim_{T \rightarrow \infty} \tilde{\Pi}_T$.

IV. EQUILIBRIUM ANALYSIS OF THE SYNCHRONOUS PARALLEL ALGORITHM IN AN EXAMPLE

In this section, an explicit Π_T is derived for a particular example based on the battle field scenario in section II. Sufficient conditions that guarantee the optimality of the parallel sampling algorithm are derived.

Proposition 4.1: For the synchronous Markov chain kernel of (3), suppose that $\tilde{U}(x, y)$ defined in (5) has a symmetric form, i.e., $\tilde{U}(x, y) = \tilde{U}(y, x)$ for all $x, y \in X$. For a fixed temperature T , the synchronous Markov chain has a unique stationary distribution Π_T given by

$$\Pi_T(x) = \frac{\sum_{z \in \mathcal{N}^m(x)} \exp(-\tilde{U}(x, z)/T)}{\sum_{y \in X} \sum_{z \in \mathcal{N}^m(x)} \exp(-\tilde{U}(y, z)/T)} \quad (9)$$

Proof. The existence and uniqueness of a stationary distribution follows from proposition 3.1. The Markov chain kernel (3) can be rewritten as

$$P_T(x, y) = \frac{\exp(-\frac{\tilde{U}(x, y)}{T})}{\sum_{z \in \mathcal{N}^m(x)} \exp(-\frac{\tilde{U}(x, z)}{T})} \quad (10)$$

(9) can then be verified since the balance equation is fulfilled, i.e., $\Pi_T(x) * P_T(x, y) = \Pi_T(y) * P_T(y, x)$ \square

In general, the symmetry of the energy function $\tilde{U}(x, y)$ does not hold. However, in some special cases, one can construct a symmetric energy function for the same parallel Markov chain kernel.

Theorem 4.1: Suppose that the Markov Random Field defined in section II consists only of singleton and pairwise cliques, and the neighborhood system is time-invariant, then there exist a symmetrized potential function \hat{U} which defines the same parallel Markov chain kernel defined by \tilde{U} in (10). Specifically,

$$\hat{U}(x, y) = \tilde{U}(x, y) + \sum_{s \in S} \Psi_{\{s\}}(x) \quad (11)$$

Proof. We first show the symmetry of \hat{U} .

$$\begin{aligned} \hat{U}(x, y) &= \tilde{U}(x, y) + \sum_{s \in S} \Psi_{\{s\}}(x_s) \\ &= \sum_{s \in S} \sum_{c \ni s} \Psi_c(y_s, x_{S \setminus s}) + \sum_{s \in S} \Psi_{\{s\}}(x_s) \\ &= \sum_{s \neq t} \Psi_{\{s, t\}}(y_s, x_t) + \sum_{s \in S} \Psi_{\{s\}}(y_s) + \sum_{s \in S} \Psi_{\{s\}}(x_s) \\ &= \sum_{s \neq t} \Psi_{\{s, t\}}(x_s, y_t) + \sum_{s \in S} \Psi_{\{s\}}(x_s) + \sum_{s \in S} \Psi_{\{s\}}(y_s) \\ &= \tilde{U}(y, x) + \sum_{s \in S} \Psi_{\{s\}}(y_s) = \hat{U}(y, x) \end{aligned}$$

Because the difference between $\hat{U}(x, y)$ and $\tilde{U}(x, y)$ depends only on the configuration x , the two potential functions actually define the same Markov chain kernel. More precisely, for any two configurations x and y , we

have

$$\begin{aligned} P_T(x, y) &= \frac{\exp(-\tilde{U}(x, y)/T)}{\sum_{z \in \mathcal{N}^m(x)} \exp(-\tilde{U}(x, z)/T)} \\ &= \frac{\exp(-\tilde{U}(x, y)/T - \sum_{s \in S} \Psi_{\{s\}}(x)/T)}{\sum_{z \in \mathcal{N}^m(x)} \exp(-\tilde{U}(x, z)/T - \sum_{s \in S} \Psi_{\{s\}}(x)/T)} \\ &= \frac{\exp(-\hat{U}(x, y)/T)}{\sum_{z \in \mathcal{N}^m(x)} \exp(-\hat{U}(x, z)/T)} \quad \square \end{aligned}$$

Let $\tilde{H}(x)$ be the induced energy from the invariant distribution $\Pi_{T(n)}$ of the Markov chain kernel $Q_{T(n)}$. Specifically,

$$\tilde{H}(x) = -\ln \left(\sum_{z \in \mathcal{N}^m(x)} \exp(-\hat{U}(x, z)) \right) \quad (12)$$

Picking an appropriate cooling schedule $T(n)$ as in theorem 3.1, one could conclude that the asymptotic configuration(s) Ω^0 of the parallel sampling algorithm are the minimizer(s) of $\tilde{H}(x)$. Next, one would like to study whether Ω^0 minimize the original configuration energy $U(x)$.

With the Gibbs potential function defined in (2), the induced energy function $\hat{U}(x, y)$ satisfies the following inequality

$$\begin{aligned} \hat{U}(x, y) &= \sum_{s \neq t} \lambda_n \|y_s - x_t\| \\ &+ \sum_{s \in S} (\lambda_g (J_s^g(x) + J_s^g(y)) + \lambda_o (J_s^o(x) + J_s^o(y))) \\ &\leq \sum_{s \neq t} \lambda_n (\|y_s - x_s\| + \|x_s - x_t\|) \\ &+ \sum_{s \in S} (\lambda_g (J_s^g(x) + J_s^g(y)) + \lambda_o (J_s^o(x) + J_s^o(y))) \\ &\leq \sum_{s \neq t} \lambda_n R_m + U(x) + U(y) \\ &\leq c_1 R_m + 2U(x) + \Delta \end{aligned} \quad (13)$$

where $\Delta = \max_{y \in \mathcal{N}^m(x)} |U(x) - U(y)|$ is the maximal local oscillation of the potential U . And $c_1 = \sum_{s \neq t} \lambda_n$

From (12) and (13), we have

$$\tilde{H}(x) \leq \overline{\mathcal{M}} (2U(x) + c_1 R_m + \Delta) \quad (14)$$

where $\overline{\mathcal{M}} = \max_x \ln |\mathcal{N}^m(x)|$. Let x^* be the minimizer of $U(x)$, i.e., $x^* = \arg \min_x U(x)$. Minimizing both side of (14), we have

$$\min_{x \in \mathcal{X}} \tilde{H}(x) \leq \overline{\mathcal{M}} (2U(x^*) + c_1 R_m + \Delta) \quad (15)$$

Similarly, since $\|x_s - x_t\| \leq R_m + \|y_s - x_t\|$, it can be shown that

$$\tilde{H}(x) \geq \underline{\mathcal{M}} (2U(x) - c_1 R_m - \Delta)$$

where $\underline{M} = \min_x \ln |\mathcal{N}^m(x)|$.

Lemma 4.1: Let the sets A, B be defined as $A = \{x : \tilde{H}(x) \leq \overline{M}(2U(x^*) + c_1 R_m + \Delta)\}$, and $B = \{x : U(x) - U(x^*) < \frac{\overline{M}}{\underline{M}}(c_1 R_m + \Delta)\}$. Then $B \supset A$

Proof. $\forall x \in \bar{B}$, we have

$$\begin{aligned} \tilde{H}(x) &\geq \underline{M}(2U(x) - c_1 R_m - \Delta) \\ &> \overline{M}(2U(x^*) + c_1 R_m + \Delta) \end{aligned}$$

which implies $x \in \bar{A}$. So, $\bar{B} \subset \bar{A}$, which is equivalent to $A \subset B$. \square

From the lemma, one could conclude that the minimizer of $\tilde{H}(x)$ lies in a ball Ω^B with radius $c_1 R_m + \Delta$ from the minimizer of $U(x)$. In section III, we have Ω^0 is a subset of local minima Ω^L . With lemma 4.1, we have

$$\Omega^0 \subset (\Omega^L \cap \Omega^B) \quad (16)$$

If $(\Omega^L \cap \Omega^B) = \{x^*\}$, the parallel algorithm minimizes the original potential function U and desired configuration(s) can be achieved. For many applications, goal configurations might not restrict to ones with minimum energy. If all configurations contained in $(\Omega^L \cap \Omega^B)$ are desired, then the parallel algorithm achieves the global goals for sure.

V. SUMMARY AND CONCLUSION

In this paper, convergence properties of a synchronous parallel simulated annealing algorithm were analyzed. The synchronous parallel sampling algorithm significantly reduces the maneuvering time. However, the delay and the communication cost introduced by the synchronization degrade the performance. An asynchronous parallel sampling algorithm was proposed to solve the problem. But the inhomogeneous nature of the underlying Markov chain at a fixed temperature presents difficulties in the convergence analysis. Applying the partial parallel model, a homogeneous Markov chain was constructed to model the evolution of the asynchronous parallel sampling algorithm. It was shown that the asynchronous algorithm will lead to a unique stationary distribution for a fixed temperature.

Although extensive simulations suggest that the parallel annealing algorithm leads swarms to the desired configuration(s), in general, parallel sampling algorithm might not achieve global goal. We studied a special example based on a battle field scenario to investigate sufficient conditions that guarantee optimality. By assuming that the Gibbs potential only consists of terms associated with pairwise and singleton cliques, an explicit equilibrium distribution was derived to investigate the limiting configurations. It was shown that the parallel algorithm achieves global goal if $\Omega^0 = (\Omega^L \cap \Omega^B)$ holds.

REFERENCES

- [1] D. A. Schoenwald, "AUVs: In space, air, water, and on the ground," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 15–18, 2000.
- [2] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.
- [3] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [4] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part I: Fixed topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 2010–2015.
- [5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [6] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, 2001, pp. 2968–2973.
- [7] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proceedings of the IEEE International Conference on Robots and Automation*, Washington, DC, 2002, pp. 1217–1222.
- [8] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [9] D. H. Kim, H. O. Wang, G. Ye, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, Atlantis, Paradise Island, Bahamas, 2004, pp. 159–164.
- [10] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 1398–1404.
- [11] R. Volpe and P. Khosla, "Manipulator control with superquadratic potential functions: Theory and experiments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1423–1436, 1990.
- [12] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.
- [13] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.
- [14] J. S. Baras and X. Tan, "Control of autonomous swarms using Gibbs sampling," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 4752–4757.
- [15] G. Winkler, *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods: A Mathematical Introduction*. New York: Springer-Verlag, 1995.
- [16] S. Salapaka, A. Khalak, and M. A. Daleh, "Capacity constraints on locational optimization problems," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Maui, 2003, pp. 1741–1746.
- [17] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based path planning for autonomous vehicles: Convergence analysis," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [18] —, "A stochastic algorithm for self-organization of autonomous swarms," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Seville, Spain, 2005.
- [19] R. Azencott, "Parallel simulated annealing: An overview of basic techniques," *Simulated Annealing: Parallelization Techniques*, pp. 37–47, 1992.
- [20] A. Trounev, "Massive parallelization of simulated annealing: A mathematical study," *Simulated Annealing: Parallelization Techniques*, pp. 145–162, 1992.
- [21] P. Bremaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues*. New York: Springer Verlag, 1999.
- [22] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and automation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
- [23] R. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1985.