Decentralized Coordination of Autonomous Swarms using Parallel Gibbs Sampling

Paulo Rezeck

November 2019

Abstract

In this document, I proposed the study and implementation of the work of *Tan et al.* [1] to understand the applications of Markov Random Fields (MRFs) and Gibbs sampling theories for the coordination of autonomous swarm robots.

1 Introduction

This work deals with the self-organization of a large number of robotic swarms conducting a wide-area search. Given an initial configuration of the robots, one wishes to find a distributed maneuver strategy under which the robots move to a new desired configuration.

Inspired by the emergent behaviors demonstrated by swarms of bacteria, insects, and animals, control methods that yield desired collective behaviors based on simple local interactions have received great interest in academia. In addition, a multitude of applications can benefit from using studies like these, ranging from surveillance and reconnaissance, to search and rescue, to transport, and to foraging.

In recent years significant advances have been made in collaborative control of autonomous swarms, where tools in optimization, control, dynamical systems, and algebraic graph theory are applied to formally analyze or synthesize interaction rules for groups of mobile robots. Artificial potential functions have often been involved in these methods, where the motion of the robots is determined by the gradient flow. An essential problem with such approaches, however, is that the system dynamics may get trapped at local minima of the potential function. In addition, many of the studies have considered the requirement of distributed, local interactions among robots, which is dictated by the otherwise prohibitive cost for centralized coordination of large-scale networks, and by the need to ensure robustness against single-node failures.

The work proposed by *Tan et al.* [1] consist of a systematic approach to coordinate an autonomous swarm based on the theory of Markov Random Fields (MRFs) and Gibbs sampling. The approach aims to achieve global objectives

(without being trapped at local minima) using primarily local interactions together with limited global interactions and obstacle avoidance. A discrete-time path planning setting is considered, where robots are allowed to move on a discretized grid in a 2D space. A swarm is modeled as an MRFs on a graph, where the robots and their communication/sensing links constitute the vertices and the edges of the graph, respectively. As in the artificial potential approach, global objectives and constraints are reflected in potential functions—in this case, Gibbs potentials. The movement of vehicles is decided using simulated annealing based on the Gibbs sampler.

The remainder of the document is organized as follows. In Section 2, the background on MRFs is briefly reviewed and the application of MRFs to modeling of autonomous swarms is described. The analysis of the algorithm is carried out in Section 3. Simulation results are presented in Section 4. Finally, Section 5 provides concluding remarks.

2 Background

2.1 Markov Random Fields

Following the references presented in the paper, let S be a finite set of cardinality α, 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 $X = Xs_{s \in S}$ 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 $\Lambda \equiv \Lambda_1 \times ... \times \Lambda_\alpha$ is called the configuration space. A neighborhood system on S is a family $\Gamma = \{\}_s$, where $\forall s, r \in S$,

- $\Gamma_s \subset S$,
- $s \notin \Gamma_s$, and
- $r \in \Gamma_s$ if and only if $s \in \Gamma_r$.

The neighborhood system induces an undirected graph with vertices $s \in \alpha$, where an edge exists between vertices s and r if and only if $r \in \Gamma_s$. A set $C \subset$ is called a *clique* if all elements of C are neighbors of each other. A random field X is called a Markov random field (MRF) with respect to the neighborhood system Γ if, $\forall s \in \alpha$,

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \Gamma_s) \tag{1}$$

A potential U is a family $\{U_A : A \subset \alpha\}$ of functions on the configuration space χ, where $U_A : \chi \to R$, and $U_A(x)$ depends only on $x_A \equiv \{x_s : s \in A\}$. In other words, U_A is only a function of the values at the sites contained in the set A. For convenience, we will denote $U_A(x)$ as $U_A(x_A)$ from here on. If $U_A \equiv 0$ whenever A is not a *clique* or a singleton, U is called a nearest-neighbor potential. If $U_A \equiv 0$ whenever A is not a pair or a singleton, U is called a

pairwise potential. U is called a pairwise, nearest-neighbor potential if it is both a pairwise potential and a nearest-neighbor potential.

Given a potential U, the potential energy $H(x)$ for configuration x is defined as

$$H(x) = \sum_{A \subset \alpha} U_A(x_A). \tag{2}$$

In particular, for a pairwise, nearest-neighbor potential U, we can write H as

$$H(x) = \sum_{s \in \alpha} U_{\{s\}}(x_s) + \sum_{(s,t) \in \alpha \times \alpha, t \in \Gamma_s} U_{\{s,t\}}(x_s, x_t). \tag{3}$$

A random field X is called a Gibbs random field (GRF) if

$$P(X = x) = \frac{e^{\frac{-H(x)}{T}}}{Z}, \text{ with } Z = \sum_z e^{\frac{-H(z)}{T}}. \tag{4}$$

The probability measure shown in 4 is called a Gibbs distribution, and the underlying potential U is called a Gibbs potential. T has the interpretation of temperature in the context of statistical physics. Eq. 4 implies that a higher-energy state has a lower probability, and that the influence of energy on probability increases as T decreases.

2.2 Gibbs Sampler

In statistical physics, a GRF is often used to describe the distribution of system configurations at the thermodynamic equilibrium. However, direct evaluation of eq. 4 and related ensemble averages is often impossible due to the high cardinality of the configuration space (the latter rendering the computation of Z intractable). Markov Chain Monte Carlo (MCMC) methods, such as the Metropolis algorithm and the Gibbs sampler, can generate Markov chains on the configuration space, with eq. 4 as the limiting probability measure. Next we illustrate such a process with the example of sequential Gibbs sampling.

Given a configuration $x = (x_1, ..., x_\alpha)$ at time instant n, one can update it to a different configuration y by updating the values at each site sequentially. For example, for site s, one can update x_s to some value y_s at time $n+1$ based on the following probability (this is what is meant by sampling): for $y_s \in \Lambda_s$,

$$P(X_s(n+1) = y_s | X_{\alpha \backslash s}(n) = x_{\alpha \backslash s}) = \frac{e^{\frac{-H(y_s, x_{\alpha \backslash s})}{T}}}{\sum_{z_s \in \Lambda_s} e^{\frac{-H(z_s, x_{\alpha \backslash s})}{T}}} \tag{5}$$

In eq. 5, $\alpha \backslash s$ denotes the set of all sites other than s:
$\alpha \backslash s \equiv \{r \in \alpha : r \neq s\}$,

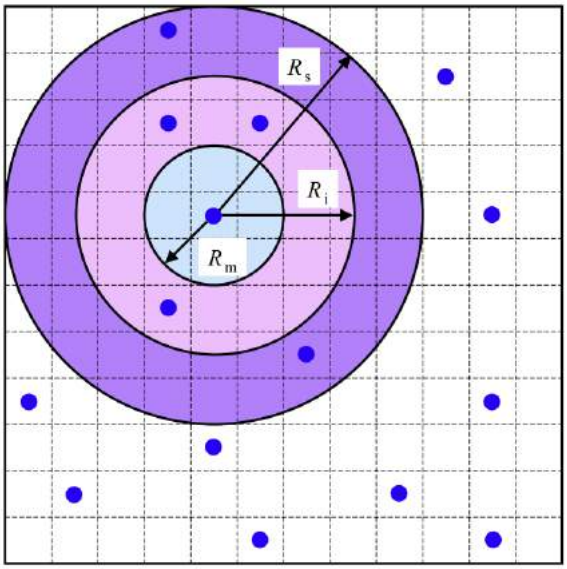

Figure 1: Illustration of the sensing range R_s, the interaction range R_i, and the moving range R_m on a 2D grid.

and $x_{\alpha\backslash s}$ represents the components of x corresponding to the sites in $\alpha\backslash s$: $x_{\alpha\backslash s} \equiv \{x_r : r \in \alpha\backslash s\}$. The notation $(y_s, x_{\alpha\backslash s})$(likewise for $(z_s, x_{\alpha\backslash s})$) represents a configuration where site s takes the value y_s while other sites take the values $x_{\alpha\backslash s}$. Note that the right-hand side of eq. 5 is precisely the conditional probability of X_s given the values at other sites for a Gibbs distribution 4. It can be verified easily that the evaluation of eq. 5 involves only $\{x_r : r \in \Gamma_s\}$ for a Gibbs field with a nearest-neighbor potential, and thus can be performed efficiently.

2.3 Extension to Swarming Control

Assuming the assumptions considered by the authors, we have a group of mobile robots moving in a bounded region within the two-dimensional (2D) or three-dimensional (3D) space. The region is discretized into a lattice, and for ease of presentation, each cell is assumed to be square with unit dimensions. A mobile robot is assumed to be a point that moves from the center of one cell to that of another. Each robot has a sensing range R_s: it can sense the locations of obstacles and other robots within distance Rs. It also has an interaction range $R_i \leq R_s$: the moving decision of a robot is only influenced by robots within the distance R_i, which form its set of neighbors. In addition, each robot can move by at most $R_m \leq R_s$ within each time step. Fig. 1 illustrates the definitions of the three ranges in a 2D grid. The distances on the lattice are defined using the Euclidean norm based on the center locations of the cells.

The R_i-neighborhood relations induce a graph structure, where the robots form the vertices of the graph and an edge exists between two robots if and only if they are neighbors of each other. A random field is then defined on this graph, where each robot is considered as a site and the random variable associated with this site represents the location of the robot. In particular, we will denote the set of robots as $\alpha = \{1, 2, ..., N\}$, where N is the total number of robots. X_s (or x_s) will denote the center location of the cell in which node s resides. Similar notation, such as y_s, z_s, etc., will also be used in the later discussion. We will use $x = (x_1, ..., x_N)$ to denote the configuration of the swarm. Given x, the set of neighbors $\Gamma_s(x)$ is defined as

$$\Gamma_s(x) \equiv \{r \in \alpha : r \neq s, ||x_r - x_r|| \leq R_i\}. \tag{6}$$

The set of lattice cells within R_m from node s form the phase space Λ_s. Unlike in the classical MRF case, the phase space Λ_s here will vary with x_s, the location of node s. A suitable potential U can be defined to reflect the swarm coordination objectives, from which the potential energy $H(x)$ can be evaluated.

3 Algorithm

Let n denote the index of time steps. Let $X(n) = x = (x_1, ..., x_N)$ be the swarm configuration at time n. Let $F_s(x) \equiv \{z_s : ||z_s - x_s|| \leq R_m\}$ be the set of accessible cell locations for robot s given the configuration x, determined by the mobility constraint. Let $F(x)$ be the set of configurations that are accessible from x within one time step:

$$F(x) \equiv \{z = (z_1, ..., z_N) : ||z_s x_s|| \neq R_m, s \in \alpha\}. \tag{7}$$

Under parallel Gibbs sampling, all robots will simultaneously update their locations based on the configuration x at time n; in particular, the robot s will move from x_s to y_s at time $n + 1$ with probability

$$P_{T,s}(x_s, y_s|x) = \begin{cases} \frac{e^{\frac{-H(y_s, x_{\alpha \setminus s})}{T}}}{\sum_{z_s \in F_s} e^{\frac{-H(z_s, x_{\alpha \setminus s})}{T}}}, & \text{if } y_s \in F_s(x) \\ 0 & \text{if } y_s \notin F_s(x). \end{cases} \tag{8}$$

For simulated annealing, the temperature variable T will be a function of the time step n, defined by

$$T(n) = \frac{T_0}{ln(n)}, \tag{9}$$

where T_0 is the initial temperature and it is chosen empirically since the analytically determined values are found to be too conservative in simulation.

In order to define $H(\cdot)$, the following assumptions are made:

- **($\mathbf{A}_1$) The total number of lattice cells is bounded**: requires that the robots move in a bounded region, which is a reasonable assumption. It will allow us to establish the ergodicity of the Markov chain induced by Gibbs sampling under a constant temperature, and consequently the convergence of the chain to a unique stationary distribution;
- **($\mathbf{A}_2$)** $R_i + R_m \leq R_s$: implies that a robot s at x_s is able to evaluate the set of new neighbors should it move to $y_s \in F_s(x)$ while other robots stay put;
- **($\mathbf{A}_3$)** U **is a pairwise, nearest-neighbor potential**: implies that the corresponding potential energy $H(x)$ for configuration x can be written as

$$H(x) = \sum_{s \in \alpha} U_{\{s\}}(x_s) + \sum_{(s,t) \in \alpha \times \alpha,\ t \in \Gamma_s(x)} U_{\{s,t\}}(x_s, x_t).$$

We can show that (A_2) and (A_3) together ensure the local computability of eq. 8 by node s. In particular, for $y_s, z_s \in F_s(x)$,

$$H(y_s, x_{\alpha \setminus s}) = U_{\{s\}}(y_s) + \sum_{t \in \Gamma_s(y_s, x_{\alpha \setminus s})} U_{\{s,t\}}(y_s, x_t) + \epsilon, \tag{10}$$

and

$$H(z_s, x_{\alpha \setminus s}) = U_{\{s\}}(z_s) + \sum_{t \in \Gamma_s(z_s, x_{\alpha \setminus s})} U_{\{s,t\}}(z_s, x_t) + \epsilon, \tag{11}$$

where ϵ consist of terms not involving node s. Since the robots make independent moving decisions at time n for given x, the kernel $P_T(x, y) \equiv Prob(X(n+1) = y | X(n) = x)$ for the parallel Gibbs sampling-induced Markov chain can be obtained evaluating the equations 8,10 and 11. Stationary distribution evidence is detailed in the article and will not be detailed in this document for space limitations.

4 Simulations

Simulation has been further performed to corroborate the analysis and verify the effectiveness of the proposed algorithm. For convenience and performance, the algorithm was implemented in python and is available in Github[1]

The authors presents two examples as proof-of-concept: (1) rendezvous, and (2) line formation, both on a 50×50 square lattice. T_0 is chosen empirically since the analytically determined values are found to be too conservative in simulation.

[1] `https://github.com/rezeck/gibbs_swarm`.

4.1 Rendezvous

In the rendezvous problem, the potential is designed as, $U_{\{s\}}(x_s) = 0, \forall s \in \alpha$, and for $t \in \Gamma_s(x)$,

$$U_{\{s,t\}}(x_s, x_t) = \begin{cases} 10 & \text{if} ||x_s - x_t|| = 0 \\ -\frac{1}{||x_s - x_t||} & \text{otherwise.} \end{cases} \tag{12}$$

The equation $U_{\{}s\}(x_s) = 0$ implies that there is no pre-specified gathering point. By setting the potential of an overlapping pair to be high in eq. 13, we discourage multiple robots from occupying the same cell and thus avoid overcrowding. Figs. 2, 3 and 4 show the snapshots of different swarm configurations at different times for the sampling algorithm. The initial configurations for both experiments were the set by a random seed. The number of robots was set to $N = 20$, $N = 40$ and $N = 200$, respectively, and the parameters used in simulation were: $R_s = 5\sqrt{2} + 2$, $R_i = 5\sqrt{2}$ for the first experiment and $R_s = 13\sqrt{2} + 2$, $R_i = 13\sqrt{2}$, $R_m = 2$, and $T_0 = 5$. The respectively videos for the three set of experiments is available in YouTube[2][3][4].

4.2 Line Formation

The robots are required to form a line that makes a 45 angle with respect to the horizontal axis. The potential is designed as, $U_{\{}s\}(x_s) = 0, \forall s \in \alpha$, and for $t \in \Gamma_s(x)$,

$$U_{\{s,t\}}(x_s, x_t) = \begin{cases} 0 & \text{if} ||x_s - x_t|| = 0 \\ -\frac{|\langle x_s - x_t, [1,1]^T \rangle|}{||x_s - x_t||} & \text{otherwise.} \end{cases} \tag{13}$$

where $\langle \cdot \rangle$ indicates the inner product. The potential is essentially a measure for the distance between 45 and the angle made by the line connecting a pair of neighboring robots with respect to the horizontal line. The additive form of the potential energy thus encourages robots to have more neighbors with desired angles, leading to the formation of a line; overlapping robots, however, are discouraged since a connecting line is not well defined in that case. Figs. 5 and 6 show the snapshots of swarm configurations under the Gibbs sampling algorithm. Here 50 robots were simulated, with $R_s = 10\sqrt{2} + 3$, $R_i = 10\sqrt{2}$, $R_m = 3$, and $T_0 = 1$. The respectively videos for two set of experiments is available in YouTube[5][6].

[2] https://youtu.be/kOG6lDmS7xE
[3] https://youtu.be/OzqZm7QrbOo
[4] https://youtu.be/THkGc90ST2c
[5] https://youtu.be/cn0V3Fs7nUY
[6] https://youtu.be/eCXb_7-PmQE

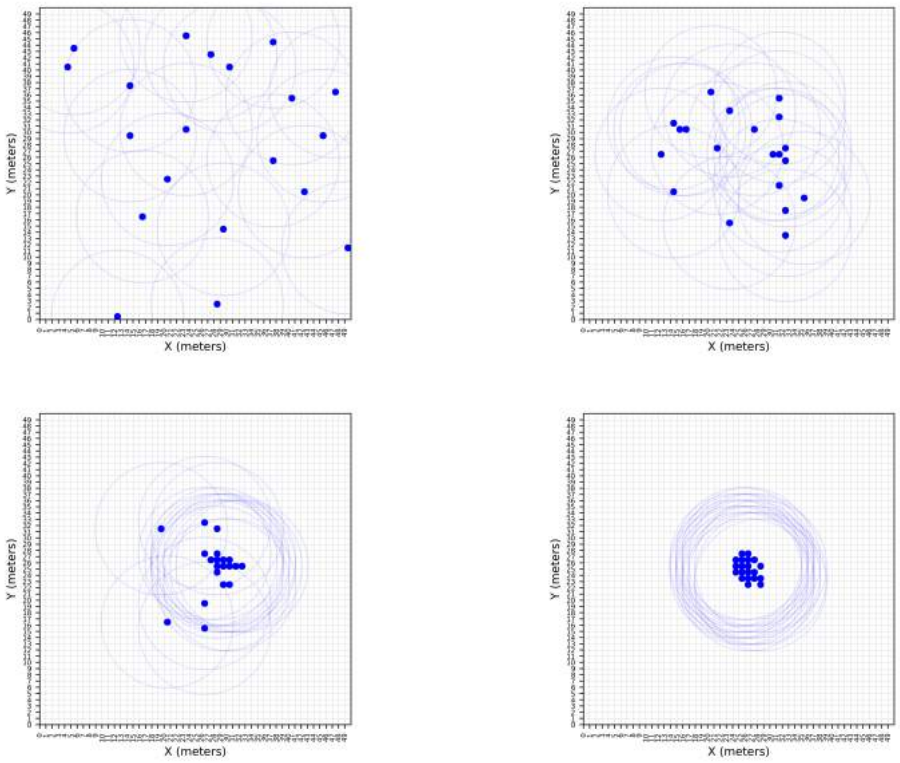

Figure 2: Snapshots of the first experiment considering a swarm of 20 robots during rendezvous under the parallel Gibbs sampling algorithm: (a) Initial configuration; (b) $n = 63$; (c) $n = 141$; (d) $n = 183$.

5 Conclusion and Discussions

In this document, it was presented a stochastic algorithm for coordination of autonomous swarms. The algorithm consists of a systematic approach to coordinate an autonomous swarm based on the theory of Markov Random Fields(MRFs) and Parallel Gibbs sampling. The approach aims to achieve global objectives (without being trapped at local minima) using primarily local interactions together with limited global interactions and obstacle avoidance.

In this work, the algorithm was implemented and a set of experiments was carried out in order to validate the practice of some topics presented in the discipline of probabilistic graphical models. The experiments performed are similar to those presented by the author of the paper, both in time of convergence and in terms of the characteristics. Although the experiment with rendezvous control was extremely similar to the ones presented in the paper, the experiments with line formation control did not have similar formation, even though it was

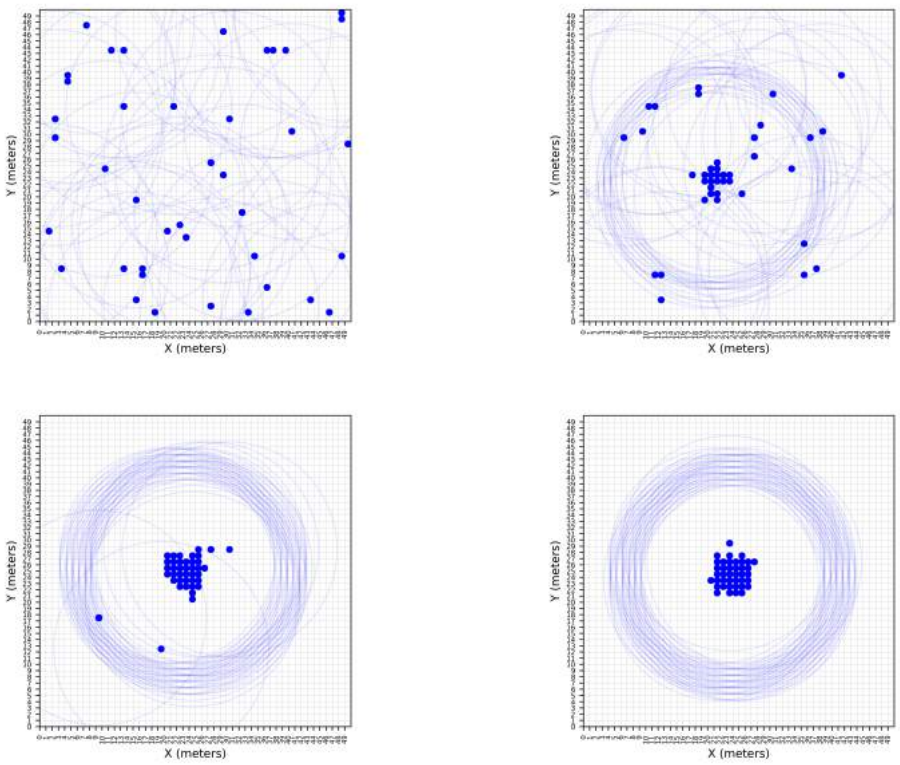

Figure 3: Snapshots of the second experiment considering a swarm of 40 robots during rendezvous under the parallel Gibbs sampling algorithm: (a) Initial configuration; (b) $n = 56$; (c) $n = 100$; (d) $n = 200$.

carefully implemented. Apparently, the author uses some repulsive field for obstacles, and this was not detailed. Another note for line formation control is that if you implement $U(x)$ as presented in the paper the robots try to form a line, but not "stretched" as presented in the article.

Finally, this work was chosen as a project for the discipline because it addresses a similar methodology that I intend to study during my Phd. And having the algorithm ready implemented for simulation will be interesting for future comparisons. Moreover, this work could be extended in various ways such as generalization to the continuous domain and heterogeneous groups of robots, where it is expected to observe different clusters.

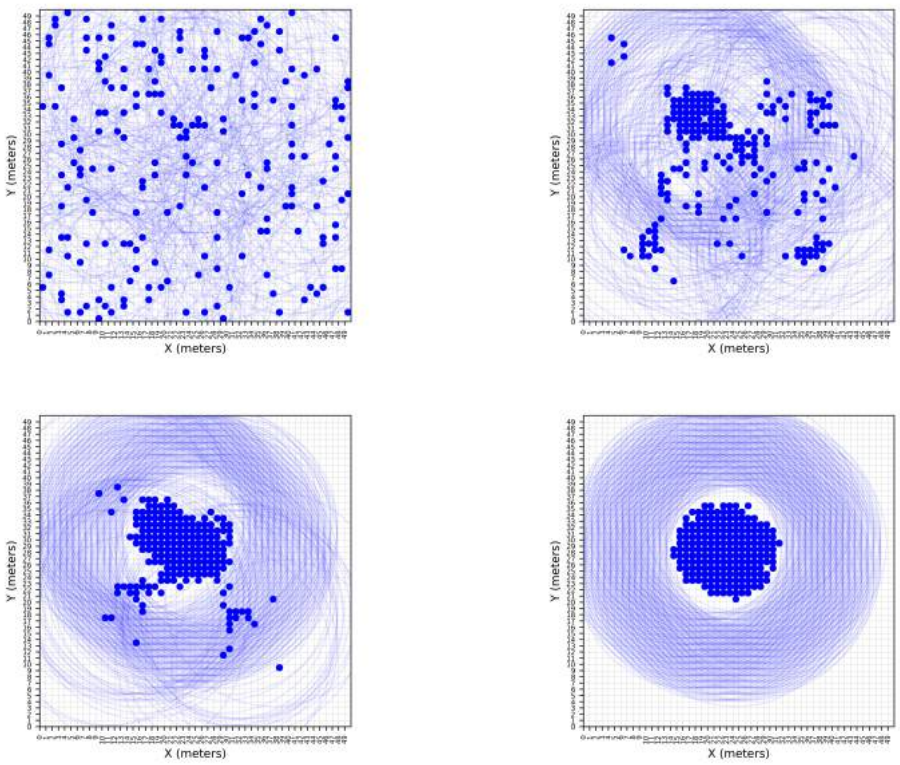

Figure 4: Snapshots of the third experiment considering a swarm of 200 robots during rendezvous under the parallel Gibbs sampling algorithm: (a) Initial configuration; (b) $n = 27$; (c) $n = 38$; (d) $n = 58$.

References

[1] Xiaobo Tan, Wei Xi, and John S Baras. Decentralized coordination of autonomous swarms using parallel gibbs sampling. *Automatica*, 46(12):2068–2076, 2010.

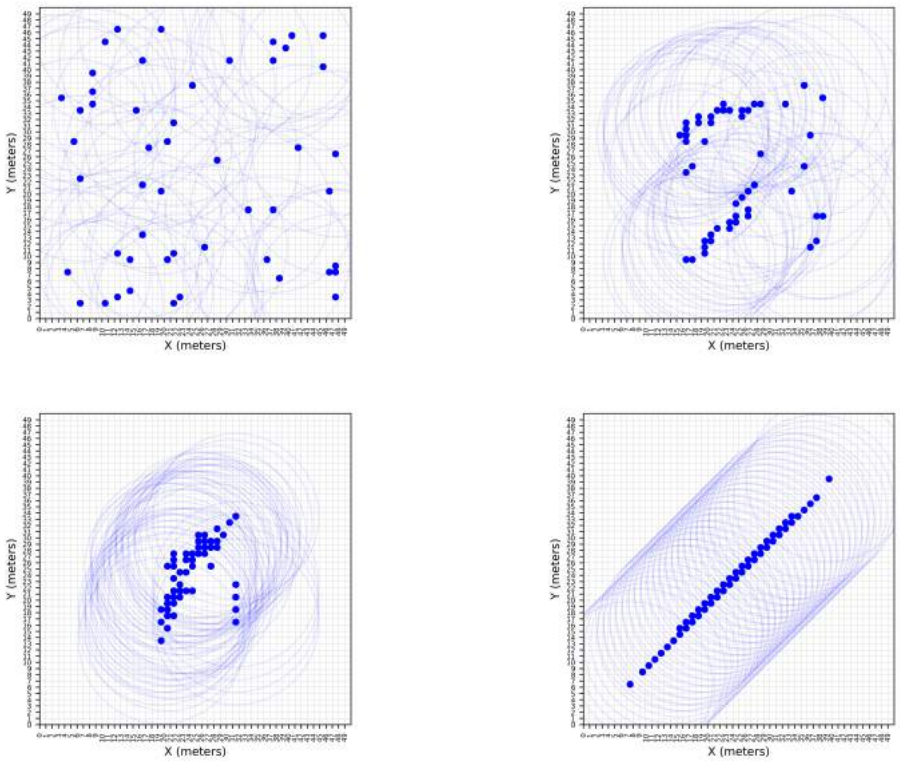

Figure 5: Snapshots of the first experiment considering a swarm of 50 robots during line formation control under the parallel Gibbs sampling algorithm: (a) Initial configuration; (b) $n = 7$; (c) $n = 10$; (d) $n = 30$.

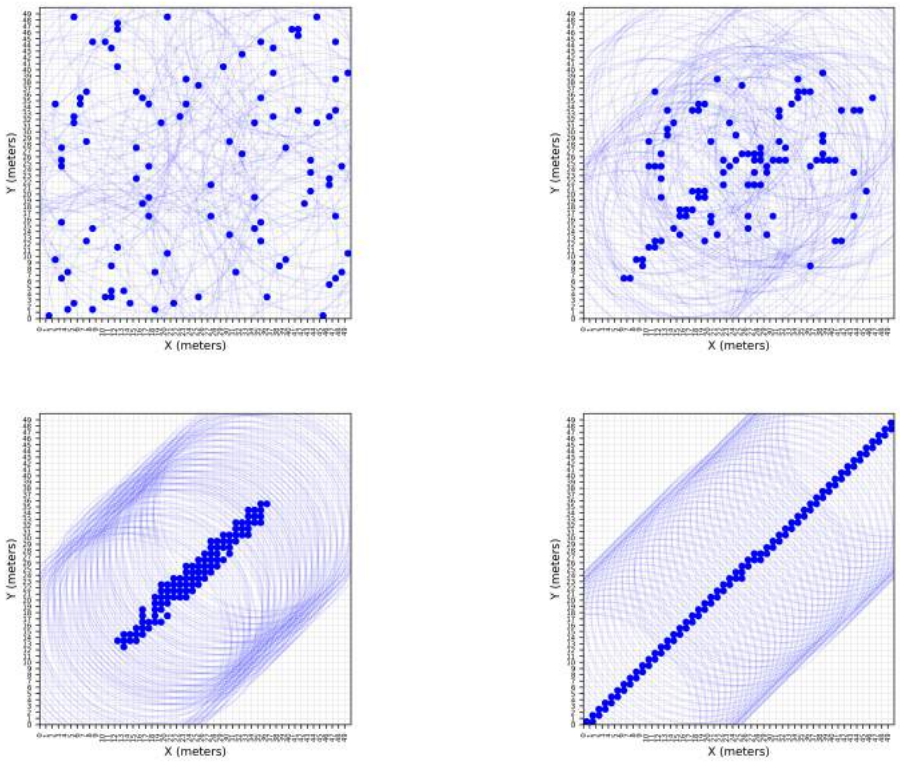

Figure 6: Snapshots of the second experiment considering a swarm of 100 robots during line formation control under the parallel Gibbs sampling algorithm: (a) Initial configuration; (b) $n = 7$; (c) $n = 10$; (d) $n = 30$.