Cooperative Object Transportation using Gibbs Random Fields

Paulo Rezeck[1] and Renato M. Assunção[1,2] and Luiz Chaimowicz[1]

***Abstract*— This paper presents a novel methodology that allows a swarm of robots to perform a cooperative transportation task. Our approach consists of modeling the swarm as a *Gibbs Random Field* (GRF), taking advantage of this framework's locality properties. By setting appropriate potential functions, robots can dynamically navigate, form groups, and perform cooperative transportation in a completely decentralized fashion. Moreover, these behaviors emerge from the local interactions without the need for explicit communication or coordination. To evaluate our methodology, we perform a series of simulations and proof-of-concept experiments in different scenarios. Our results show that the method is scalable, adaptable, and robust to failures and changes in the environment.**

I. Introduction

Robotic swarms are composed of a large number of robots that generally rely on emergent collective behaviors to solve complex problems. Such systems present desirable characteristics, such as robustness, adaptability, simplicity, and scalability, which are important in different tasks [1], [2].

In particular, robotic swarms capable of cooperatively transporting objects may be suitable for many applications with high societal and economic impact potential. For instance, one may use robotic swarms for operations where the use of sophisticated robots is impossible or impractical, such as warehouse automation, waste disposal, and demining.

Despite several advantages of using robotic swarms robots for cooperative transportation, designing decentralized control methods for such application is not simple. Some of the main challenges consist of aligning and synchronizing the forces applied to sustain the transportation. The method must be robust to objects of different shapes and resilient to changes in the environment, such as surfaces with different coefficients of friction, and robot failures.

This work presents a novel stochastic and decentralized approach that allows a swarm of robots to navigate autonomously through a bounded environment and cooperatively transport an object toward its goal location, as shown in Fig. 1. Our approach consists of modeling the robotic swarm as a Gibbs Random Field (GRF) and defining its potential energy as a combination of the Coulomb-Buckingham potential and kinetic energy. The Coulomb-Buckingham potential enables the robots to aggregate, interact with the object by pushing it, and avoid obstacles in the environment. The kinetic energy allows the robots to reach a consensus on their relative velocities concerning their neighborhood and circulate the object looking for adequate pushing positions.

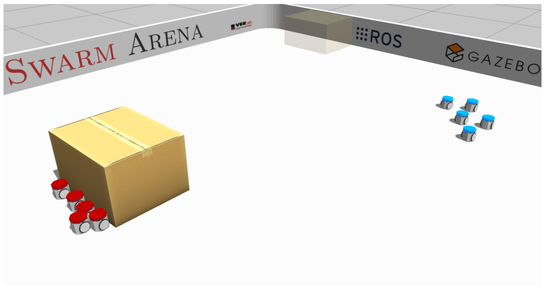


Fig. 1: Swarm of robots cooperatively transporting an object (solid cardboard) toward a goal (transparent cardboard).

Dynamically and autonomously adapting the potential and energy parameters, robots can navigate through the environment looking for the object to be transported, form groups, and push the object toward a goal location. These behaviors emerge from the local interactions without the need for explicit communication or coordination. The robots only need to be able to estimate the relative position and velocity of their neighbors and also distinguish between obstacles and the object detected within their sensing range. Moreover, the robots do not need any information about the object (i.e., location, size, mass, and shape), except for its goal location.

II. Related Work

Over the years, a wide range of control and coordination methods have been proposed to perform cooperative object transportation using multiple robots. Recently, Tuci et al. [8] presented a review of the state-of-the-art works in this area, which categorizes the most common transportation strategies into three types: pushing, caging, and grasping.

Pushing strategies, such as those used in this work, consider that the robots are not attached to the object. They provide a simple way of manipulating relatively large objects and do not require any sophisticated mechanism to apply forces to the object. Pushing is also interesting because, unlike other strategies, it allows robots to aggregate and apply forces to specific points on the object, enabling the transportation of large and heavy objects when the number of robots is increased. Moreover, it allows the use of simpler robots, providing a suitable scenario for robotic swarms.

While there is a large body of work in cooperative object transport [8], we focus our discussion on approaches that do not require centralized planners or direct communication among robots to push an object from a random location to a specific goal. Table I summarizes the main characteristics of these approaches, and in the following, we review and compare these works with ours, mainly discussing scalability, adaptability, and robustness issues. By adaptability,

Paulo Rezeck, Renato M. Assunção and Luiz Chaimowicz are with the Department of Computer Science, Universidade Federal de Minas Gerais, Brazil[1]. Renato M. Assunção is also with the ESRI Inc., Redlands, CA[2]. Email: {rezeck,assuncao,chaimo}@dcc.ufmg.br.
This work was partially supported by CAPES, CNPq, and Fapemig.

TABLE I: Main characteristics of research works using pushing-only transport strategy.

Studies	Control method	Experimental setup				Scalability	Adaptability test		Robustness test		
		Number of robots	Env. with obstacles	Transport direction	Random position		Robot failure	Change in the environment	Object mass	Object shape	Object size
Kube and Bonabeau, 2000 [3]	Behavior-based	2-20	no	towards a goal	no	high	no	no	no	no	yes
Yamada and Saito, 2001 [4]	Behavior-based	4	no	towards a goal	no	low	yes	yes	no	no	no
Fujisawa et al., 2013 [5]	Behavior-based	4-10	no	towards a goal	yes	high	yes	yes	yes	no	no
Jianing Chen et al., 2015 [6]	Behavior-based	20	yes	towards a goal	yes	high	no	no	yes	yes	yes
Alkilabi et al., 2018 [7]	Neural Networks	2-6	no	towards a goal	no	high	no	no	yes	no	yes
Ours	**GRFs**	**2-20**	**yes**	**towards a goal**	**yes**	**high**	**yes**	**yes**	**yes**	**yes**	**yes**

we consider the team's ability to work reliably even with individual robot failures or changes in the environment, while robustness is related to the power of the methodology to deal with objects of different mass, size, or shape.

Kube et al. [9] presented one of the first studies that formally dealt with the dynamics of cooperative transport. The authors demonstrated that coordinated efforts produced by a homogeneous group of simple robots pushing an object are possible without using a direct communication mechanism. Further, in Kube and Zhang [10], the authors extended the original work by constructing a new robot for experimentation. Also, In Kube and Bonabeau [3], they proposed the addition of a stagnation recovery strategy avoiding deadlock conditions in which the robots cancel the pushing forces on the object depending on where they are positioned around it. Moreover, they extended their approach, allowing the robots to push the object towards a fixed goal position and performed experiments to test their method's robustness and scalability.

Yamada and Saito [4] also presented theoretical and experimental analyses to support the assumption that robots can transport objects without using direct communication. Unlike previous works, the authors demonstrated that the use of indirect communication also allows the transport of an object towards a goal location considering a dynamic environment. Their approach consists of using an action selection method to design a behavior-based control method that is robust to a small change in the environment, such as increase the number of robots or the mass of the object. To evaluate the performance of their approach, the authors executed real experiments using four robots showing that the robots can operate in a simple environment where individual robots are required to push a light object or in complex environments where multiple robots are required to push a heavy object cooperatively. Although their approach allows adaptability to small changes in the environment, the authors did not evaluate the robustness of the method to objects with different shapes, mass, and sizes. Furthermore, the method does not seem to scale easily as the system grows.

Fujisawa et al., 2013 [5] presented a cooperative transport approach that uses an interesting mechanism for indirect communication via artificial pheromones as seen in ants. Such a mechanism allows the robots to sense and lay on the terrain a volatile alcohol substance, mimicking pheromone during trail formation. The authors also proposed a behavior-based algorithm using a deterministic finite automaton, which allows the robots to perform a random search to find a food item (i.e., a heavy object) and transport it to a goal location (i.e., the nest). To evaluate the efficiency of the proposed system, they perform experiments with up to 10 robots demonstrating the adaptability of their approach but did not evaluate the system's scalability and the robustness of the method as to the object's shape and size changes.

Jianing Chen et al. [6] proposed another behavior-based strategy for cooperative transport that deals with goal occlusion by keeping the robots pushing the object even when not detecting the goal. They performed experiments using twenty robots to transport objects of different shapes toward their goals and provided analytical proof of the method's effectiveness. Also, the authors demonstrated other interesting experiments in which they consider the goal location as a mobile robot remotely controlled by a human, showing potential applications in the context of human-robot interaction. The work was further extended with this focus in Kapellmann-Zafra et al. [11]. Although the authors demonstrated the system's scalability and robustness to different types of objects, tests evaluating the adaptability of the system to failures or changes in the environment were not performed.

Differently from previous approaches, Alkilabi et al. [12] proposed an approach based on recurrent neural networks and evolutionary algorithms, which allow a group of robots with a minimalist sensory apparatus to transport a heavy object by perceiving its movement. The best instance of the evolved controller was extensively tested on physical robots to transport objects of different sizes and masses in an arbitrary direction. Later on, Alkilabi et al. [7] presented a complementary study extending their neural-controller with mechanisms to direct the transport towards a specific target location. The authors showed that the transport strategies are scalable concerning the group size and robust enough to deal with boxes of various masses and sizes. Despite presenting interesting results in a large set of experiments, the strategy still requires robots to start with a direct view of the object. Also, it does not consider objects with different shapes or obstacles in the environment.

This paper presents an approach that allows a swarm of robots to coordinate their motion to navigate a bounded environment and cooperatively transport an object toward its goal location. We model the swarm as a Gibbs Random Fields (GRF) and take advantage of this framework's properties to control the robots in a completely decentralized fashion, relying only on local information. Different from other methods, we do not require pre-synthesized behaviors or automatic learning methods. Using appropriate potential functions, the desired behaviors emerge directly from local interactions, bringing scalability, robustness, and adaptability to our method.

III. Methodology

The proposed approach is based on our recent work [13] in which we proposed and evaluated a novel methodology that allows a swarm of heterogeneous robots to simultaneously achieve flocking and segregative behaviors. Here, we adapt and extend this methodology allowing a swarm of robots to cohesively navigate through a bounded environment and cooperatively transport an object toward its goal location.

The main idea behind our approach consists of modeling the swarm as a dynamic *Gibbs Random Field* (GRF) and then sampling velocities for each robot in a decentralized way, which leads the entire swarm to converge toward the global minimum of the potential. Thus, setting the potential functions accordingly, we can make the swarm execute the desired task in a completely decentralized fashion.

A. Formalization

Assume a set $\mathcal{R}$ of η homogeneous robots navigating through a bounded region within the two-dimensional Euclidean space. The state of the i-th robot at time step t is represented by its pose $\mathbf{q}_i^{(t)}$ and velocity $\dot{\mathbf{q}}_i{}^{(t)} = \mathbf{v}_i^{(t)}$, which is bounded by $||\mathbf{v}_i^{(t)}|| \leq v_{max}$. The robots are driven by a kinematic model with motion model defined as $\mathcal{K} : (\mathbf{q}_i^{(t)}, \mathbf{v}_i^{(t)}) \rightarrow (\mathbf{q}_i^{(t+1)})$ and have a circular sensing range of radius λ, enabling them to estimate the relative position and velocity of other robots as well as objects and obstacles within this range.

We represent an object as a finite set of points $\mathcal{O} = \{\mathbf{o}_1, ..., \mathbf{o}_n\}$ outlining the object perimeter. An object detected by the i-th robot within its circular sensing range consists of a subset of points $\mathcal{O}_i \subset \mathcal{O}$, where $\mathbf{o}_j \in \mathcal{O}_i \rightarrow ||\mathbf{o}_j - \mathbf{q}_i|| \leq \lambda$ and $||\mathbf{o}_j - \mathbf{q}_i||$ is the Euclidean norm between two points. The robot represents obstacles in the same way as the object. Thus, we require the i-th robot to be able to arrange the points detected on objects and obstacles at time t into two distinct sets, $\mathcal{O}_i^{(t)}$ and $\mathcal{W}_i^{(t)}$, respectively. In addition, we assume that the robots can track the goal location, $\mathbf{g}$, of object $\mathcal{O}$ from any point within the environment.

B. Applying GRFs concepts to swarm robotics

A GRF is a probabilistic graphical model that is a particular case of a *Markov Random Field* (MRF) when the joint probability density of the random variables is represented by the Gibbs Measure [14]. Such models are conditioned by the Markov properties [15], which are convenient to model robotic swarms. Such properties imply the conditional independence of information coming from outside a neighborhood system, supporting the requirement of local interactions.

1) Modeling a swarm of robots: assume an undirected graph $\mathbf{G} = (\mathcal{R}, \mathbf{E})$ with each vertex representing one of the robots. A random field on $\mathbf{G}$ consists of a collection of random variables $\mathbf{X} = \{X_i\}_{i \in \mathcal{R}}$ and, for each $i \in \mathcal{R}$, let Λ_i be a finite set called the phase space for the i-th robot. A configuration of the system $\mathbf{X}$ at time step t is defined as $\mathbf{x}^{(t)} = \{\mathbf{v}_1, ..., \mathbf{v}_\eta\}$, where $\mathbf{v}_i \in \Lambda_i$ and represents the velocities performed by each robot at that time step.

A neighborhood system on $\mathcal{R}$, given a configuration $\mathbf{x}^{(t)}$, is a family $\mathcal{N} = \{\mathcal{N}_i\}_{i \in \mathcal{R}}$, where $\mathcal{N}_i \subset \mathcal{R}$ is the set of neighbors for the i-th robot. The neighborhood is constrained by the sensing range λ satisfying $\mathcal{N}_i \triangleq \{j \in \mathcal{R} : j \neq i, ||\mathbf{q}_j - \mathbf{q}_i|| \leq \lambda\}$, and induces the configuration of the graph $\mathbf{G}$ by setting an edge between two robots if they are neighbors.

A random field $\mathbf{X}$ is called a GRF if the joint probability density of the system is represented by,

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} e^{-\frac{H(x)}{T}}, \text{ with } Z = \sum_z e^{-\frac{H(z)}{T}}, \quad (1)$$

where Z is a normalizing term; T is a constant interpreted as the temperature in the statistical physics context and taken as equal to 1 in this paper; $\frac{1}{Z} e^{-\frac{H(x)}{T}}$ is called the Gibbs distribution; and $H(x)$ is the potential energy of the system.

In short, the distribution function (1) establishes that the probability of the random field $\mathbf{X}$ assuming state $\mathbf{x}$ is proportional to the potential energy of such a state divided by the sum of the potential energy relative to all states z that the field can assume. It implies that the low energy states are more likely than those of higher energy.

The potential energy $H(x)$ consists of the summation of values produced by potential functions, $U_A : \Lambda \rightarrow \mathbb{R}$, which quantifies the state $\mathbf{x}$ of the swarm system concerning a physical property or behavior. Formally, it is defined by pairwise interactions between neighboring vertices as

$$H(x) = \sum_{i \in \mathcal{R}} U_{\{i\}}(\mathbf{v}_i) + \sum_{(i,j) \in \mathcal{R} \times \mathcal{R}, j \in \mathcal{N}_i} U_{\{i,j\}}(\mathbf{v}_i, \mathbf{v}_j), \quad (2)$$

where $U_{\{i\}}(\mathbf{v}_i)$ is interpreted as the potential for the i-th robot to reach the velocity $\mathbf{v}_i$, and $U_{\{i,j\}}(\mathbf{v}_i, \mathbf{v}_j)$ is the potential regarding the velocities of the neighboring $(\mathbf{v}_i, \mathbf{v}_j)$ pair of vertices.

2) Parallel Gibbs sampling: the probability function (1) enables one to compute the probability of the entire swarm reaching a certain configuration $\mathbf{x}$. However, what we require is the sampling of each robots' velocity in a decentralized way, given only the neighborhood of the robots.

Parallel Gibbs sampling is a strategy that simultaneously updates the robots' velocities based on their configuration at time t. Such strategy is possible due to the local nature of the potential energy (2), and it implies that we do not require the knowledge of the entire swarm to sample velocities for the i-th robot, but only information about its $\mathcal{N}_i$ neighbors. Indeed, by replacing (2) in (1), we obtain

$$P_i(\mathbf{v_i}, \bar{\mathbf{v}}_\mathbf{i} | \mathbf{x}) = \frac{e^{-\left(U_{\{i\}}(\bar{\mathbf{v}}_\mathbf{i}) + \sum\limits_{\forall j \in \mathcal{N}_i} U_{\{i,j\}}(\bar{\mathbf{v}}_\mathbf{i}, \mathbf{v_j})\right)}}{\sum\limits_{\mathbf{z_i} \in \mathbf{Z}_i(x)} e^{-\left(U_{\{i\}}(\mathbf{z_i}) + \sum\limits_{\forall j \in \mathcal{N}_i} U_{\{i,j\}}(\mathbf{z_i}, \mathbf{v_j})\right)}}, \quad (3)$$

where $\mathbf{Z}_i(\mathbf{x}) \triangleq \{\mathbf{z}_i : ||\mathbf{z}_i|| \leq v_{max}\}$ is a set of possible velocities for the i-th robot given the configuration $\mathbf{x}^{(t)}$, restricted by $\mathbf{Z}_i(\mathbf{x}) \subset \Lambda_i$; and $\bar{\mathbf{v}}_\mathbf{i}$ is a velocity sampled in $\mathbf{Z}_i(\mathbf{x})$ representing a likely velocity for the next time step, $\mathbf{v_i}^{(t+1)} = \bar{\mathbf{v}}_\mathbf{i}$.

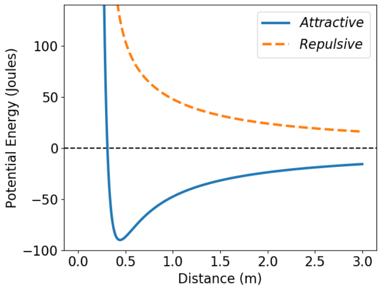


Fig. 2: The Coulomb-Buckingham potential function depends on the distance r among two particles i and j and their charge c_i and c_j. We set the charge to produce attractive behavior among the robots and with the object, or repulsive behaviors to avoid collision with obstacles [13].

We use (3) to sample appropriate velocities for each robot using only the current state and information about their neighborhood. A common approach to sampling velocities given the probability function (3) consists of using Markov Chain Monte Carlo (MCMC) methods. In this work, we use the Metropolis-Hastings algorithm [16].

3) Swarm behaviors and potential functions: one of the many advantages of modeling a robotic swarm as a GRF is the flexibility to implement different swarm behaviors, changing only the potential functions used by the potential energy (2). We extend the definition of the potential $U_{\{i\}}(\mathbf{v}_i)$ to allow the robots to interact with objects as well as to avoid collisions with obstacles within the environment. Also, the potential $U_{\{i,j\}}(\mathbf{v}_i, \mathbf{v}_j)$ is used to maintain the robots cohesively navigating through the environment. For this, we use the Coulomb-Buckingham potential in conjunction with a kinetic energy term.

The Coulomb-Buckingham potential [17] is a combination of the Lennard-Jones potential with the Coulomb potential. It describes the interaction among particles considering the distance between them and their charges. We take advantage of such a potential to model the cohesion among the robots as well as their interaction with objects and obstacles. Fig. 2 illustrates the Coulomb-Buckingham potential, given by:

$$\Phi(r) = \varepsilon \left(\frac{6}{\alpha - 6} e^{\alpha} \left(1 - \frac{r}{r_0} \right) - \frac{\alpha}{\alpha - 6} \left(\frac{r_0}{r} \right)^6 \right) + \frac{c_i c_j}{4\pi\varepsilon_0 r}, \tag{4}$$

where $r = ||\mathbf{q}_j - \mathbf{q}_i||$ is the euclidean distance between the particles i and j; ε is a constant representing the depth of the minimum energy; r_0 is the minimum energy distance; α is a free dimensionless parameter; c_i and c_j are the charges of the particles i and j; and ε_0 is an electric constant.

We assume classical mechanics to compute the kinetic energy regarding the robots and object motion. Kinetic energy allows the robots to maintain a consensus on their velocities while navigating as a group and also to move around the object. Formally, assume $\mathbf{V}$ as the sum of the velocity vectors and m is the mass of the system. The kinetic energy $\mathbf{E}$ is defined as

$$\mathbf{E}(\mathbf{V}) = \frac{1}{2} m (\mathbf{V} \cdot \mathbf{V}). \tag{5}$$

Fig. 3 presents a diagram illustrating how the desired behaviors emerge from the combination of Coulomb-Buckingham potential and kinetic energy. These behaviors are detailed in the next sections.

C. Cooperative transport

For the cooperative transportation task, we design the potential $U_{\{i\}}(\mathbf{v}_i)$ to simultaneously generate the move around and pushing behaviors. The pushing behavior comes directly from the Coulomb-Buckingham potential, and the move around comes from the combination of such potential with the kinetic energy. We note that increasing or decreasing the Coulomb-Buckingham potential implies favoring the pushing or moving around behavior. Thus, we propose a conditional factor: if the robot is in a position where it can push the object towards its goal, the robot increases the Coulomb-Buckingham potential. Otherwise, it decreases.

The *Move Around Behavior* comes from the combination of the Coulomb-Buckingham potential with the kinetic energy. The first one forces the robots to stay at a certain distance, δ, from the object avoiding missing or colliding with it. We set $r_0 = \delta$ and $c_i c_j < 0$ in (4). The second one indicates which velocity favors circumvent the object. The strategy for moving around consists of computing a gradient of the points detected on the surface of the object and using it to sample velocities that favor the robot to move around the object.

To compute the gradient, we arrange the points detected by the i-th robot on the $\mathcal{O}_i^{(t)}$ object on a clockwise or counterclockwise order. To decide in which order, we project a line segment ("front segment" shown in Fig. 4) and count the number of points to the left and to the right of that line segment. If there are more points to the left than right, we order clockwise; otherwise, we order them counterclockwise. This forces the robots to choose velocities that align them with the surface of the object. For now on, assume $\bar{\mathcal{O}}_i^{(t)}$ is the ordered set of points detected on the object.

Defining the order of the elements directly implies on the direction of the gradient, which in turn indicates the direction that the robot should move around the object. Given $\bar{\mathcal{O}}_i^{(t)}$, the gradient is calculated as,

$$\nabla_i^{(t)} = \nabla \bar{\mathcal{O}}_i^{(t)} = \{\mathbf{o}_{i+1} - \mathbf{o}_i : |\bar{\mathcal{O}}_i^{(t)}| \geq 2\}, \tag{6}$$

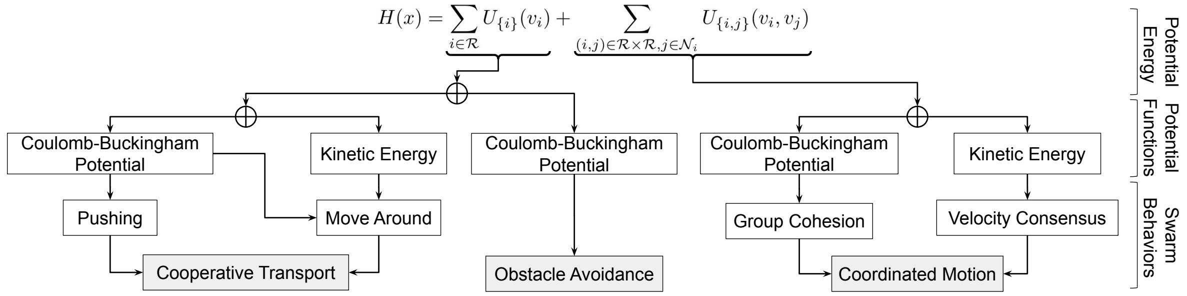
where $\nabla_i^{(t)}$ is a simplified notation and consists of a set with each element being a vector on the surface of the object.

To evaluate how much the velocity performed by the robot favors the move around behavior, we sum the differences between the gradient vectors and the robot velocity. That is, we compute the resulting vector,

$$\mathbf{Q}_i = \sum_{\forall \upsilon_{\mathrm{j}} \in \nabla_i^{(t)}} \upsilon_{\mathrm{j}} - \bar{\mathbf{v}}_i, \tag{7}$$

where $\bar{\mathbf{v}}_i$ is a sampled velocity for the i-th robot and $\mathbf{Q}_i$ is a vector that summarizes the mismatching among the sampled velocity and the gradient.

On the other hand, the *Pushing Behavior* is achieved simply by decreasing the δ distance applied to the Coulomb-Buckingham potential. This forces the robots to collide



$H(x) = \underbrace{\sum_{i\in\mathcal{R}} U_{\{i\}}(v_i)} + \underbrace{\sum_{(i,j)\in\mathcal{R}\times\mathcal{R}, j\in\mathcal{N}_i} U_{\{i,j\}}(v_i, v_j)}$
Potential Energy
Potential Functions
Swarm Behaviors
Coulomb-Buckingham Potential
Kinetic Energy
Coulomb-Buckingham Potential
Coulomb-Buckingham Potential
Kinetic Energy
Pushing
Move Around
Group Cohesion
Velocity Consensus
Cooperative Transport
Obstacle Avoidance
Coordinated Motion

Fig. 3: Diagram showing how the swarm behaviors emerge from the combination of the potential functions.

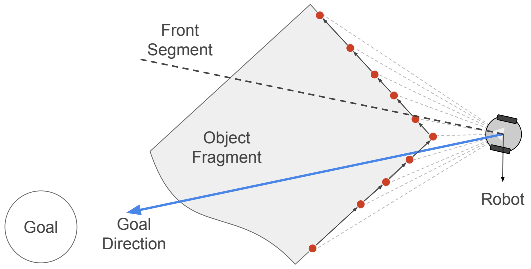


Fig. 4: Illustrative scheme showing the perception of the robot regarding the object and its goal.

with the object producing the pushing behavior. To achieve this behavior automatically, we use a similar mechanism presented previously to divide the points detected to the left or the right side regarding a line segment in the goal direction (shown as a blue arrow in Fig. 4). If there are points on both sides, the object is occluding the goal from the robot; otherwise, the robot has a direct line of sight to the goal. The robot decides to start pushing the object if there are points on both sides and the proportion of points from one side to the other is higher than a threshold ρ. To start pushing the object, the robot must decrease the δ distance such that it collides with the object. If such a proportion is lower than ρ, then it increases the δ distance so that the robots can circulate the object.

D. Obstacle avoidance

To avoid collisions with obstacles, we extend the potential $U_{\{i\}}(\mathbf{v}_i)$ with another Coulomb-Buckingham potential. To compute such potential, we use distance r between the points detected on the surface of obstacles and the robot's pose. By setting the charges in (4) to $(c_i c_j > 0)$, we note that the robots generate a repulsive behavior to obstacles avoiding collisions with them.

E. Cohesive motion

To cohesively navigate the swarm through the environment, we combine the Coulomb-Buckingham potential and the kinetic energy (see Fig. 3). The first one forces the robots to aggregate, while the second one drives the robots to reach a consensus on the group's velocity.

To keep the cohesion of the swarm, we set the charges in (4) to $(c_i c_j < 0)$ and use the relative distances between the robot and its neighbors.

The velocity consensus is reached by computing the kinetic energy from the relative velocities of the i-th robot neighbors. That is,

$$\mathbf{V}_i = \sum_{\forall \mathbf{v}_j \in \mathcal{N}_i} \mathbf{v}_j, \tag{8}$$

where $\mathbf{V}_i$ is defined as the sum of the relative velocities.

F. Combination

Formally, all the swarm behaviors described above and illustrated in Fig. 3 are represented by the potential energy

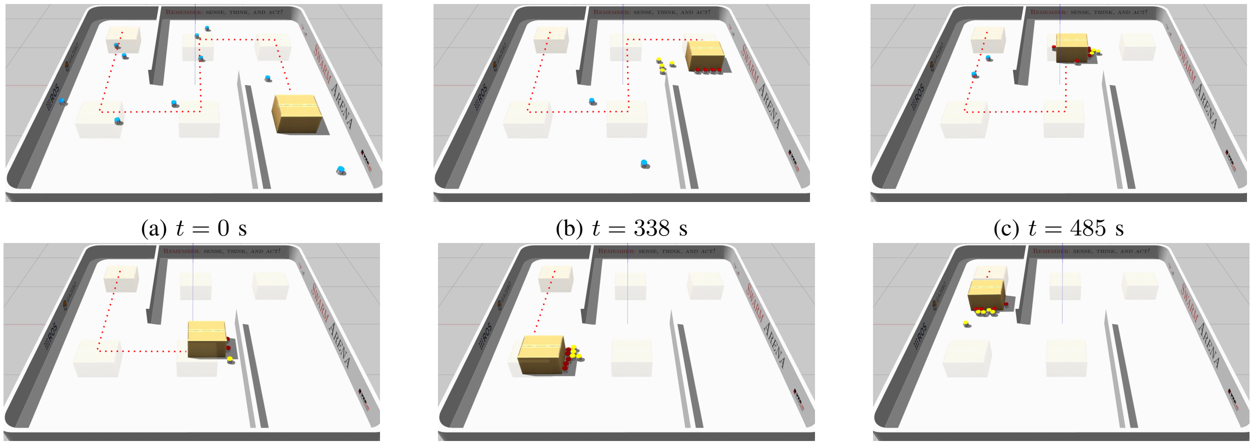
$$H_i(\mathbf{x}) = \left(\sum_{\forall \mathbf{o_j} \in \mathcal{O}_i} \Phi(||\bar{\mathbf{q}} - \mathbf{o_j}||) + \mathbf{E}(\mathbf{Q}_i) + \sum_{\forall \mathbf{w_j} \in \mathcal{W}_i} \Phi(||\bar{\mathbf{q}} - \mathbf{w_j}||) \right) + \left(\sum_{\forall \mathbf{v_j} \in \mathcal{N}_i} \Phi(||\bar{\mathbf{q}} - \mathcal{K}(\mathbf{q}_j, \mathbf{v}_j)||) + \mathbf{E}(\mathbf{V}_i) + \mathbf{E}(v_{max} - \bar{\mathbf{v}}_i) \right), \tag{9}$$

where $H_i(\mathbf{x})$ is the potential energy associated with the i-th robot concerning state $\mathbf{x}$ and a sampled velocity $\bar{\mathbf{v}}_i$. The first term in parenthesis defines, respectively, the cooperative transport and obstacle avoidance behaviors. The second one is related to the coordinated motion.

IV. Experiments and Results

We conduct several simulated experiments to evaluate our approach's performance regarding scalability, adaptability, and robustness. For this, we implemented our methodology using the ROS (Robot Operating System)[1] middleware and set up a simulated environment in Gazebo[2]. The simulated environment consists of a 4 by 4 meters arena, an object, and a group of differential drive robots. The simulated robots are a model of a physical robot that we are developing in our laboratory for experiments with swarm robotics [18]. It is equipped with a range sensor consisting of a set of infrared beams with $\lambda = 0.5$ meters. We consider that this sensor can distinguish between objects and obstacles. As mentioned, we also consider that robots can estimate neighbors' positions and velocities. The maximum speed reached by these robots is 0.12 m/s. The simulator's physics engine allows a single robot to start the movement of a 200 grams object if it collides with it at speeds above 0.10 m/s. It also allows it to keep pushing the object if it maintains a speed greater than 0.01 m/s. For each experiment, we performed 30 runs

[1] https://www.ros.org
[2] http://gazebosim.org



(a) $t = 0$ s

(b) $t = 338$ s

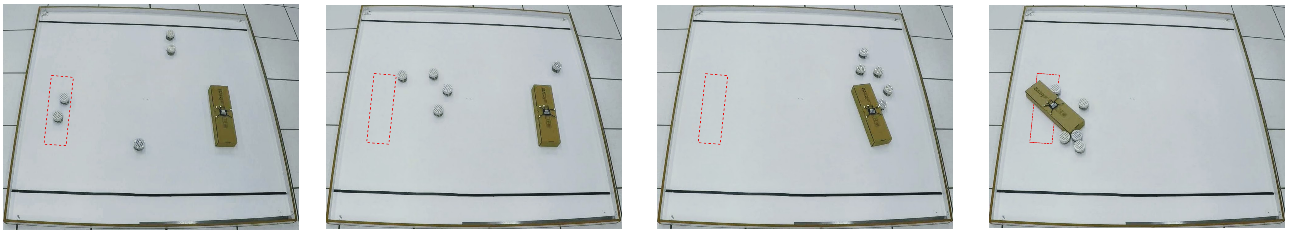
(c) $t = 485$ s

(d) $t = 642$ s

(e) $t = 805$ s

(f) $t = 919$ s

Fig. 5: Snapshots of an experiment showing 10 simulated robots transporting an object toward a sequence of goal locations in a complex environment.



(a) $t = 0$ s.

(b) $t = 65$ s.

(c) $t = 146$ s.

(d) $t = 307$ s.

Fig. 6: Snapshots of an experiment showing 5 e-puck robots transporting an object toward its goal. The red rectangle indicates the goal location.

and, for each run, the robots are randomly placed in the environment. The results are presented as an average value with a 95% confidence interval. A video of the experiments is available on Youtube[3] and the source code at Github[4].

A. Scalability

To evaluate the system scalability, we performed experiments increasing the number of robots: 2, 4, 10, 20. A rectangular object of 0.5 by 0.4 meters and mass equal to 200 grams was placed in the center of the arena and the goal is located 1.7 meters ahead. As metrics, we measure the velocity and current distance between the object and its goal location. We assume the robots stop pushing the object if it is less that 0.1 meters of its goal. Fig. 7 shows the results from this experiment and Table II summarizes the average time spent by the robots to transport the object to its goal.

As expected, increasing the number of robots reduces the time taken to find and transport the object to the goal location. With more robots searching, the time taken to find the object is smaller. Moreover, the object reaches higher speeds when pushed by more robots, also reducing the transport time.

However, we also observe that the gain is smaller for a large number of robots. For example, the gain of increasing from 2 to 4 robots is larger than from 10 to 20. When there are too many robots trying to push the object, the most distant ones cannot detect it due to sensing restrictions and will not contribute to the pushing. Thus, while our approach is scalable, the performance will depend on the size of the object and robots' sensing capabilities, and it may saturate for a large number of robots.

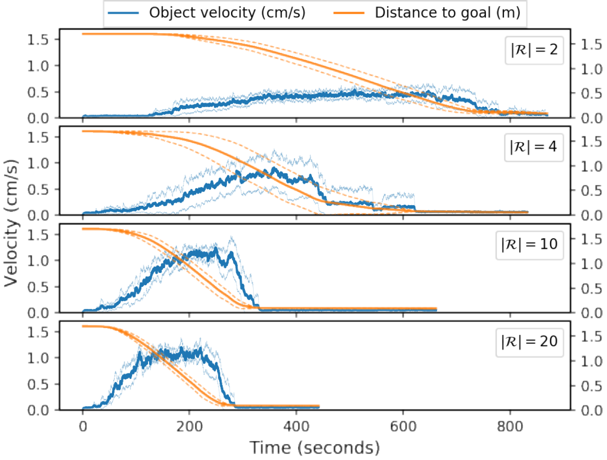


Fig. 7: Experiments on the scalability of the system when increasing the number of robots.

[3] https://youtu.be/hrkJKL3W3pQ

[4] https://github.com/rezeck/grf_transport

TABLE II: Average transportation time for 30 trials when the number of robots increases.

Number of robots	Transport time (seconds)
2	829 ± 74
4	593 ± 96
10	324 ± 18
20	278 ± 14

TABLE III: Average transportation time for 30 trials when the robots need to adapt to changes in the environment.

Scenario	Transport time (seconds)
Ideal	438 ± 13
Robot failure	596 ± 21
Goal change	449 ± 11

B. Adaptability

We propose two experiments to determine whether our approach allows the swarm to adapt to robot failures and changes in the environment. We use 10 robots in these experiments, and we move a rectangular object of 0.5 by 0.4 meters with a mass equal to 200 grams toward its goal, located 2.6 meters apart. Unlike the previous setup, we place the object such that it requires the robots to push it around corners.

In the first experiment, we assume that 4 robots stop working due to mechanical failures when they are pushing the object. We intend to evaluate the method's resilience when the number of robots performing the task abruptly decreases. Also, this experiment allows us to assess the robots' resilience to execute the task. Fig. 8 shows the average velocity and distance of the object from its goal. Table III summarizes the average time spent by the robots to transport the object.

In Fig. 8, we may observe that robot failures impact the duration of the task. Besides the impact of having fewer robots pushing the object, we noticed that in many runs, the "dead" robots obstructed the others and were also pushed together with the object, decreasing the object velocity. Despite that, we noticed that the swarm managed to successfully transport the object in all attempts.

In the second experiment, we evaluate how the swarm behaves when the goal location is changed. To assess this, we changed the goal when the object is less than 1.3 meters from its goal. We set the new goal location to 1.3 meters in the other direction so that the robots need to move around the object to continue the transport. The total distance with the change of the goal is 2.6 meters. In spite of that, Figure 8 shows that the robots are able to complete the task, and the time taken does not increase significantly, demonstrating the adaptability of the method.

C. Robustness

We conducted experiments with three objects of different shapes, sizes, and masses to assess the robustness of the method. To evaluate how robots behave when they detect objects with differently shaped corners, we used three objects: a rectangular prism (right angle), an octagonal prism (obtuse angle), and a triangular prism (acute angle). For each object, we experiment the effects of doubling their sizes and masses. In all experiments, we placed the object at a distance of 2.6 meters from its goal. Fig. 9 shows the time required to transport the objects to the goal.

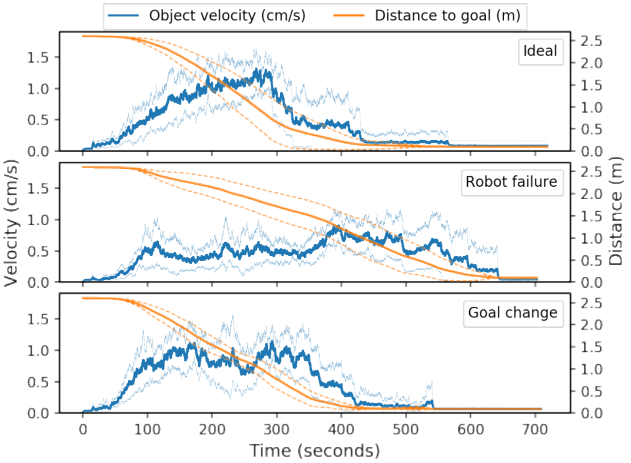


Fig. 8: Experiments on the adaptability of the swarm when robot failure occurs and when the goal location changes.

In this experiment, the robots were less efficient in pushing the triangular prism than the other two objects. When the robots start pushing the triangle at its acute corners, few robots are grouped to push, increasing the task time. As expected, increasing the mass of the object also increases the transport time. Although larger objects enable more robots to push it, they also cause an overall increase in transport time. Increasing the effective contact surface between the object and the ground increases the maximum intensity of the friction force, making the object more difficult for the robots to push. When we increase the size of the triangular prism, we observe that its impact over the transport time is relatively higher than increasing either the rectangular or the octagonal prism. We believe that objects with sharp corners decrease our method's performance since we assume our sensor to be radial, making detection difficult when the robot is pushing on a corner. Despite the disparity in the transport time for different objects, the robots successfully transported all objects to their goals.

D. Object transportation in complex environments

In more complex environments, pushing an object toward its goal location may be challenging due to obstructions caused by obstacles. Navigation strategies can help robots to move an object to its final location, passing through a series of waypoints.

Since our method allows robots to adapt to the change in goal location, we conducted experiments where the robots transport an object through a sequence of goals. We assume the robots are initially distributed around the environment, do not know the object's location, but have the sequence of waypoints beforehand. Fig. 5 displays a sequence of snapshots from one experiment and a video[5] is also available.

[5] https://youtu.be/a0eYZid3jQs

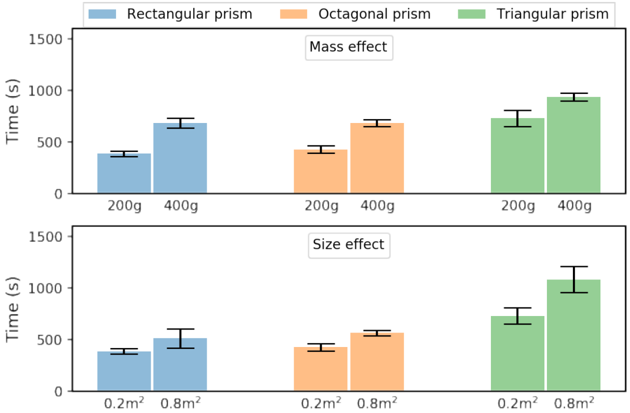


Fig. 9: Experiments on the robustness of the swarm regarding different objects with different sizes and masses.

E. Real robots

We also conducted proof-of-concept experiments to assess the applicability of our approach in a real environment. The environment consists of a bounded square area of 2 by 2 meters, an object of 0.14 by 0.48 meters that weighs 550 grams, and five e-puck robots [19]. The robots receive velocity commands from a remote server executing ROS, where each one of them is implemented as an independent node. Given that our robots are not equipped with sensors that enable them to process our method, we emulate such sensors using the information provided by an Optitrack motion capture system[6]. We set the sensing distance to $\lambda = 0.3$ and set the goal location 1.1 meters ahead of the object.

We performed 10 runs with a duration of 10 minutes each. As we expected, the robots were able to transport the object in all attempts, showing that the method is effective even with the uncertainties found in real scenarios. Overall, the robots took 448 ± 8 seconds to transport the object. Fig. 6 presents a sequence of snapshots from one experiment and a video[7] is also available.

V. Conclusion and Future Work

This paper presented a novel approach based on Gibbs Random Fields that allows a swarm of robots to navigate through an environment and cooperatively transport an object toward its goal location. The approach is fully decentralized and based on local information, being suitable for robotic swarms. To assess the scalability, adaptability, and robustness of our method, we conducted several experiments in different scenarios. Results showed that our method is scalable and supports the transportation of objects of different shapes, sizes, and masses. Moreover, it exhibited resilience to changes in goal location and robot failures. Proof-of-concept experiments using real robots showed the feasibility of our approach in a real-world environment. By setting a sequence of goals, we further demonstrated our method's application in more complex environments, where the robots push an object through an environment with obstacles. In future work, we intend to explore heterogeneity to allow part of the group to take responsibility for finding the goal location and indicating it to other robots in charge of transporting the object. Also, we plan to experimentally compare our work with others in the literature and investigate ways to allow our method to adjust the object's orientation regarding its goal.

References

[1] Danesh Tarapore, Roderich Groß, and Klaus-Peter Zauner. Sparse robot swarms: Moving swarms to real-world applications. *Front. Robot. AI 7: 83. doi: 10.3389/frobt*, 2020.

[2] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.

[3] C Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1-2):85–101, 2000.

[4] Seiji Yamada and Jun'ya Saito. Adaptive action selection without explicit communication for multirobot box-pushing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(3):398–404, 2001.

[5] Ryusuke Fujisawa, Hikaru Imamura, and Fumitoshi Matsuno. Cooperative transportation by swarm robots using pheromone communication. In *Distributed Autonomous Robotic Systems*, pages 559–570. Springer, 2013.

[6] Jianing Chen, Melvin Gauci, Wei Li, Andreas Kolling, and Roderich Groß. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, 31(2):307–321, 2015.

[7] Muhanad H Mohammed Alkilabi, Aparajit Narayan, Chuan Lu, and Elio Tuci. Evolving group transport strategies for e-puck robots: moving objects towards a target area. In *Distributed Autonomous Robotic Systems*, pages 503–516. Springer, 2018.

[8] Elio Tuci, Muhanad HM Alkilabi, and Otar Akanyeti. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers in Robotics and AI*, 5:59, 2018.

[9] C Ronald Kube and Hong Zhang. Collective robotics: From social insects to robots. *Adaptive behavior*, 2(2):189–218, 1993.

[10] C Ronald Kube. Task modelling in collective robotics. *Autonomous robots*, 4(1):53–72, 1997.

[11] Gabriel Kapellmann-Zafra, Jianing Chen, and Roderich Groß. Using google glass in human–robot swarm interaction. In *Annual Conference Towards Autonomous Robotic Systems*, pages 196–201. Springer, 2016.

[12] Muhanad H Mohammed Alkilabi, Aparajit Narayan, and Elio Tuci. Cooperative object transport with a swarm of e-puck robots: robustness and scalability of evolved collective strategies. *Swarm intelligence*, 11(3):185–209, 2017.

[13] Paulo Rezeck, Renato M. Assunção, and Luiz Chaimowicz. Flocking-segregative swarming behaviors using Gibbs random fields. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[14] Ross Kindermann. Markov random fields and their applications. *American mathematical society*, 1980.

[15] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[16] Peter H Peskun. Optimum monte-carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.

[17] Richard A Buckingham. The classical equation of state of gaseous helium, neon and argon. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 168(933):264–283, 1938.

[18] Paulo Rezeck, Hector Azpurua, and Luiz Chaimowicz. Hero: An open platform for robotics research and education. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, pages 1–6. IEEE, 2017.

[19] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. Instituto Politécnico de Castelo Branco, 2009.

[6] `http://optitrack.com`

[7] `https://youtu.be/1I9-hTQO8CU`