Collective transport by caging in swarm robotics

Karen S. Cardoso[1], Nadia Nedjah[1], Luiza M. Mourelle[2]

[1]*Dept. de Engenharia Eletrônica e Telecomunicações, Universidade do Estado do Rio de Janeiro*
Rua São Francisco Xavier n° 524, 20550-900, Maracanã, Rio de Janeiro, Brazil
cardosokaren.ck@gmail.com, nadia@eng.uerj.br
[3]*Dept. de Engenharia de Sistemas e Computação, Universidade do Estado do Rio de Janeiro*
Rua São Francisco Xavier n° 524, 20550-900, Maracanã, Rio de Janeiro, Brazil
ldmm@eng.uerj.br

Abstract. Collective transport, in swarm robotics, is based on the transportation of objects by swarms of robots characterized by having significantly smaller dimensions compared to the object to be transported. The set of robots of the swarm generally have the same architecture. Individually, the robots are able to perform simple functions, such as sensing, locomotion, and basic communication. However, cooperatively, they can perform complex task. Collective transport has diverse applications that encompass the transportation of both large-scale objects and nanoscale objects. Consequently, numerous studies have been driven by this topic, highlighting three transport strategies: pushing, grasping, and caging. In this work, the caging strategy is adopted. It can be defined as the complete enclosure of the object. Allowing it a certain degree of freedom but preventing it from escaping the formation of robots around it. The main advantage of this strategy is the coordinated and cohesive progression of the transport since the forces applied to the object by the swarm complement each other and prevent significant deviations in its trajectory towards the planned destination. This work proposes a method to approach collective transport by caging, which operates in three stages: the search for the object by the swarm, the recruitment of swarm robots, aiming to position the robots around the object and enclose it, and the transport stage. The implementation of the proposed strategy is carried out in the CoppeliaSim platform, wherein the swarm is composed of Khepera III-type robots. This robot is chosen due to the arrangement of its sensors along its perimeter, which allows for a wide view of the environment. The arena, which is the scenario where the simulations are conducted is discretized into different configurations. This space discretization has a significant impact on the performance of the search and transport stages. The performance evaluation of the search and recruitment stages are based on the execution time, while for the transport stage, in addition to the execution time, the normalized error of the object's trajectory is evaluated for different arena discretization configurations and path lengths. The proposed method proves to be effective, as the caging is maintained throughout the entire path, thus ensuring the uniform transport of the object.

Keywords: Swarm robotics, Collective Transport, Caging.

1 Introduction

Collective transportation in swarm robotics is defined as a group of robots capable of carrying an object with dimensions significantly larger than those of a single robot. The swarm of robots is inspired by the behavior of insects and animals that exhibit collective intelligence, Berman et al. [1], Vardharajan et al. [2]. Due to its broad applicability, numerous studies are dedicated to collective transportation, which includes three strategies: pushing, grasping, and caging, Vardharajan et al. [2], Wang and Kumar [3], Chen et al. [4], Sudsang and Ponce [5]. In this work, caging is chosen as the strategy, characterized by the complete enclosure of the object by a swarm of robots. Its main advantage lies in its coordinated and cohesive progression in transportation. Applications include the transport of fragile objects, structure assembly, manipulation of objects at the nano/micro scale, logistics optimization in confined spaces, and medical applications involving the controlled transport of samples, medications, and medical devices, Becker et al. [6]. The caging transportation problem involves essential processes: object search, swarm recruitment, initial positioning (caging), and the actual transportation.

This work proposes the transportation of a cylindrical object, discretizing its path into landmarks corresponding to the reverse path taken by the detecting robot. The paper is organized into six sections: Section 2 presents related works, Section 3 defines the problem, Section 4 describes the main stages and algorithms, Section 5 presents the results analysis, and Section 6 provides conclusions and outlines future work.

2 Related works

The implementation of collective transportation through caging offers various methods. By analyzing related works, these methods have been grouped into four categories: distributed coordinated transportation, machine learning-based transportation, leadership-based transportation, and claw-based caging. The distributed coordinated transportation method stands out for its focus on robot communication, allowing the robots to autonomously detect the position of the object and align themselves around it to create a cage. This object position identification may stem from direct interactions or from data shared by other robots that have received the object position cordinates, Vardharajan et al. [2]. The machine learning-based transportation approach is characterized by incorporating machine learning techniques into the control of robots responsible for caging and transporting objects, Gross and Dorigo [7]. The leadership-based method is a caging transportation method that is globally controlled by a leader. This leader can be a robot or even a human who holds overall control of the system and provides instructions to be followed by the robots, Becker et al. [6]. The claw-based transportation method involves caging the object with claws in a way that prevents it from escaping, Rimon and Blake [8]. The distributed coordinated transportation method is chosen for this work, drawing inspiration from the approach presented in Vardharajan et al. [2]. A similar caging strategy is adopted, where the first robot to detect the object, positions itself around it, and subsequently, other robots are progressively recruited and positionned one by one until the cage is established.

3 Collective transportation problem

In this work, the collective transport problem is defined by a swam of robots, foraging for a cylindrical target object placed in the arena, wherein the simulation takes place. In order to guide the robot in their foraging for the target object, a set of Λ landmarks $L = \{\lambda_0, \lambda_1, ..., \lambda_{\Lambda-1}\}$ are placed in the arena. For each landmark λ_i, a neighbohood $\eta(\lambda_i)$ is defined as a set of neighboring landmarks. The robot swarm is defined as ρ robots, identified as $R = \{0, 1, ..., \rho - 1\}$. Position $p_i^t = (x_i^t, y_i^t, \beta_i^t)$ defines the positional and orientational information of robot i at any time t. Each robot i has a total of s sensors, and these are structured as follows: $sensors\langle detect, id\rangle$, where $detect$ holds a boolean value, which is true whenever an obstacle, identified as id, is detected. Robot can communicate via messages. These messages are structured as four-element records: $msg\ \langle\ type$, $origin$, $destination$, $payload\ \rangle$, where the first element contains the message type, the second the identifier of the robot that sent the message, third specifies robot to which the message is destined, and the fourth contains the useful information. Message types are integers in the range of $[0, 4]$. For each robot i, there is an associated vector μ_i that defines the order of robot placement around the object during recruitment whenever robot i is the recruiter. The first robot to detect the object is referred to as the recruiter $r^* \in R$. Let the radius of the target object be $r_\oplus$ and its position in the are an $C_\oplus = (x_\oplus, y_\oplus)$. The minimum required number of robots for the transport of a target object is defined in eq. (1):

$$\rho_{min} = \frac{M \times A}{m \times a}, \tag{1}$$

where M is the objet mass and A is the object acceleration. Constants m and a represent the mass and acceleration of the pusushing robots, respectively. The minimum angle θ_{min}, formed by the orientations of adjacent robots allows determining the maximum number of robots that should be used for object caging. It is defined in eq. (2):

$$\arctan\left(\frac{\theta_{min}}{2}\right) = \frac{(w/2 + \varepsilon/2)}{\sqrt{(w/2 + r_\vartheta)^2 - (w/2 + \varepsilon/2)^2}}, \tag{2}$$

where, w is the robot width, and ε is an predefined constant representing the minimum space between adjacent robots. The positioning of the robots around the object during caging is determined using angle $\phi \geq \theta_{min}$. This angle is $\phi = 360°/\rho$.

4 Proposed algorithm for collective transportation by caging

The proposed algorithm for collective transport by obejct caging is structured into three stages, as shown in Algorithm 1: arena initialization, search and recruitment and transport.

Algorithm 1 Transport by Caging

Arena initialization;
for parallel $i := 0 \to \rho - 1$ **do**
  Search & Recruitment in Robot i; // Search and recruitment stage
end for parallel
for parallel $i := 0 \to \rho - 1$ **do**
  Transport by robot i; // Transport stage
end for parallel

The arena initialization stage is described in Algorithm 2. This step is characterized by: defining landmarks along the entire arena, determining the neighborhood of each landmark by calculating the function $\eta(\lambda_i)$, randomly determining the initial position of the center of mass of the object $C_\oplus$ and establishing the starting position of the robots in the arena p_i^0. The starting positions p_i^0 of robot i in the arena are defined by radomically.

Algorithm 2 Arena initialization

Ensure: $C_\oplus, p_i^0, \mu_i$;
for $i := 0 \to \Lambda - 1$ **do**
  Generate randolmly λ_i in the arena;
end for
for $i := 0 \to \Lambda - 1$ **do**
  Define $\eta(\lambda_i)$;
end for
Generate randolmly $C_\oplus = (x_\oplus, y_\oplus)$;
for $i := 0 \to \rho - 1$ **do**
  Generate randolmly p_i^0;
  for $j := 0 \to \rho - 2$ **do**
    Define μ_{ij};
  end for
end for

4.1 Tartget object search

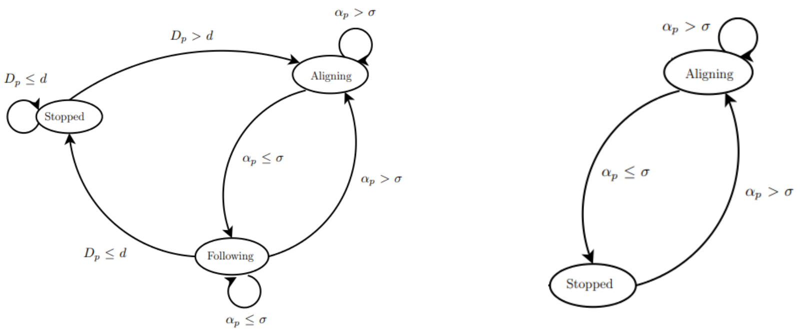
Algorithm 3 describes the object search performed by robot i culminated by the recruitment stage triger. Initially, the first landmark to which robot i will move is randomly chosen. From this setup, the robot departs from its current position towards the selected landmark. During the search stage, the robot movement from its current position to a select landmark λ is prescribed by the state machine of Fig. 1a and two PID controllers based on $D_p = D_\lambda$ and $\alpha_p = \alpha_\lambda$. This angle is formed by the robot orientation and the that to reach landmark λ. Distance D_p is defined as in eq. (3):

$$D_p = \sqrt{(x - x_i^t)^2 + (y - y_i^t)^2}. \tag{3}$$

The angle α_λ is computed as the normalized difference between the current orientation of the robot and the arc-tangent of the distance between the center of mass of the robot and that of the landmark. The normalization is performed so that α remains in $[-\pi, \pi]$. This angle is defined in eq. (4):

$$\alpha_p = |\arctan(D_p) - \beta_i^t|. \tag{4}$$

In the state machine of Fig. 1a, d is a predefined error threshold regarding the distance and σ a predefined error threshold regarding orientation. At each simulation step, if a sensor on robot i detects the object, it stops moving and declares itself as the recruiter r^*, triggering the recruitment stage. This occurs through the sending of messages from the recruiter robot to the swarm. Upon detecting the object, the recruiter sends messages of type-0, which contains the position of center of mass of the object and the trajectory of the recruiter T_{r^*} as payload. However, if the robot reaches landmark λ without detecting the target object, the current landmak is appended to the robot trajectory T_i and landmark within the current neighborhood is selected. This process is iterated either by robot i the target is detected or a message is received, announcing that a recruiter has been identified. The action to be executed depends on whether robot i is the recruiter or a recruited.



(a) Movement FSM (b) Orientation FSM

Figure 1. State machines for the movement and orientation of robots

Algorithm 3 Search & Recruitment in Robot i

$found := false$; $r^* := \infty$; $msg := \infty$; $n_i := 0$; $t := 0$; **Select randomly** $\lambda \in L$;
while $\neg found$ and $r^* = \infty$ **do**
  Calculate D_λ using Equation (3); **Calculate** α_λ using Equation (4);
  Move towards position λ with orientation α_λ using FSM of Fig. 1a;
  for $k := 0 \rightarrow s - 1$ **do**
    if $sensores[k].detect \wedge sensores[k].id = id_\oplus$ **then**
      $found := true$; $r^* := i$; **Send** $msg\langle 0, r^*, j, T_i\ C_\oplus\rangle$, $j = 0, \ldots \rho - 1$ and $j \neq r^*$;
    end if
  end for
  if $\neg found$ and $r^* = \infty$ **then**
    Receive msg;
    if $msg \neq \infty$ **then**
      $r^* := msg.origin$; $C_\oplus := msg.payload$;
    else
      if $p_i^t \approx \lambda$ **then**
        Append λ to T_i; $n_i := n_i + 1$; **Select randomly** $\lambda \in \eta(\lambda)$; $p_i^t := \lambda$;
      end if
    end if
  end if
  $t := t + 1$;
end while
if $i = r^*$ **then**
  Recruiter($C_\oplus$) in robot i;
else
  Recruited($C_\oplus$) in robot i;
end if

4.2 Recrutement

Algorithm 4 describes the actions undertaken by the recruited robot. Based on the type-0 message received by the recruiter at the end of the search stage, robot i moves to a predefined object approach distance Δ. The robot movement is carried out based on the state machine presented in Fig. 1a and two PID controllers based on $D_p = D_\oplus$ and $\alpha_p = \alpha\oplus$. $D_\oplus$ represents the distance between p_i^t and the current position of the center of mass $C_\oplus$ of the object, computed as in eq. (3), and $\alpha_p = \alpha_\oplus$, computed as in eq. (4). Afterward, the robot waits for a type-1 message, which signals that its initial positioning around the object should begin. Upon the reception of such a message, robot i calculates its initial position around the target object, using eq. (5):

$$\begin{aligned} x_i^* = &\quad x_\oplus + ((r_\oplus + w) + c) \times \cos(i \times \phi); \\ y_i^* = &\quad y_\oplus + ((r_\oplus + w) + c) \times \sin(i \times \phi), \end{aligned} \tag{5}$$

where c is a constant, added to ensure that the robot does not touch the object when initially positioned. Recall that w denotes the robot width. Again, the robot moves towards p_i^* based on the state machine shown in Fig. 1a, with $D_p = D^*$, which represents the distance between p_i^t and its initial position p_i^* in the cage, and $\alpha_p = \alpha_\oplus$. As before, the distance is also calculated as in eq. (3), and the angle is computed as in eq. (4). After completing the positioning, the robot sends a type-2 message to the swarm to indicate its completion.

Algorithm 4 Recruited($C_\oplus$) in robot i

Calculate $D_\oplus$ using Equation (3); **Calculate** $\alpha_\oplus$ using Equation (4);
while $D_\oplus > \Delta \vee \neg msg$ **do**
  Move towards position $C_\oplus$ with orientation $\alpha_\oplus$ using FSM of Fig. 1a;
  Calculate $D_\oplus$ using Equation (3); **Calculate** $\alpha_\oplus$ using Equation (4); **Receive** msg;
end while
if $D_\oplus \leq \Delta$ **then**
  Receive msg;
end if
if $(msg.type = 1) \wedge (msg.origin = r^*)$ **then**
  Calculate p_i^* using Equation (5); **Calculate** $\alpha_\oplus$ using Equation (4);
  while $\neg(p_i^t \approx p_i^*)$ **do**
    Move towards position $C_\oplus$ with orientation $\alpha_\oplus$ using FSM of Fig. 1a;
    Calculate p_i^* using Equation (5); **Calculate** $\alpha_\oplus$ using Equation (4);
  end while
  Send $(msg\langle 2, i, r^*, \emptyset\rangle)$, $j = 0, \ldots \rho - 1$ and $j \neq i$;
end if

Algorithm 5 describes the actions executed by the recruiting robot. Upon finding the object, this robot calculates its own position around the object using eq. (5). Upon reaching $p^*_{r^*}$, the recruiter r^* summons the first robot as prescribed in μ_{r^*} and waits for it to reach the desired position in the initial cage. This process is repeated with subsequent robots, according to the order in μ_{r^*} until the entire swarm is positioned, completing the object caging. The Note that the recruitment is finished when the object caging process is concluded.

Algorithm 5 Recruiter($C_\oplus$) in robot i

Calculate p_i^* using Equation (5); **Calculate** $\alpha_\oplus$ using Equation (4);
while $\neg(p_i^t \approx p_i^*)$ **do**
  Move towards position $C_\oplus$ with orientation $\alpha_\oplus$ using FSM of Fig. 1a;
  Calculate p_i^* using Equation (5); **Calculate** $\alpha_\oplus$ using Equation (4);
end while
for $j := 0 \to \rho - 2$ **do**
  $r := \mu_{ij}$; **Send** $(msg\langle 1, i, r, \emptyset\rangle)$;
  repeat
    Receive msg;
  until $(msg.type = 2) \wedge (msg.origin = r)$;
end for

4.3 Transport

Algorithm 6 describes the actions undertaken by robot i during the transport stage. First, the robot waits for type-2 messages from all robots, signaling the completion of caging and the start of the transport state. The trajectory T_{r^*} to be used to transport the object is the same as the trajectory followed by the recruiting robot r^*, but in the reverse order. Note that, the first landmark in $\lambda = T_{r^*1}$ holds the position of that where the object has been detected, while the last position has the initial position of robot r^* when it started the search for the target object. In this case, the robot orientation must be deduced from the state machine presented in Fig. 1b and using a PID controller with $\alpha_p = \alpha_\lambda$ computed as in eq. (4). Robot i spins towards the correct direction and informs via a type-3 message other robots of the completion. Then it awaits confirmation from the swarm through the same type of message, indicating the completion of their orientation process. When all robots are all aligned with the direction where the landmark λ is, they move together causing the object transport. The movement of robot i during transport is controlled by a state machine similar to the one shown in Figure Fig. 1a. However, in addition to analyzing D_λ and α_λ, distances D_r, D_ℓ, and $D_\oplus$, as well as angles θ_r and θ_ℓ are also considered by this state machine. Hence, this state machine employs seven PID controllers, one for each analyzed variable. Variables D_r and D_ℓ represent the distances between p_i^t and the current position of the center of mass of its right and left neighbors, respectively. These distances is also calculated using eq. (3). Variables θ_r and θ_ℓ represent the angles between the that of robot i and the current orientations of its right and left neighbors, respectively. It is worth noting that the ideal value for θ_r and θ_ℓ is ϕ, as expalined in Section 3. The robot continue moving in the same direction until it reaches the position p_i^+. Position p_i^+ is calculated similarly to p_i^*, as presented in eq. (5). Note that in this case, $p_\oplus$ coincides with that of λ. Upon reaching p_i^+, robot i notifies the swarm, confirming its arrival at the landmark, through type-4 messages. Robot i then waits for the confirmation from the other robots, indicating that all of them have also reached their respective new positions in the cage. This process is repeated until the entire trajetory T_r^* has been traversed, thus concluding the collective transport.

Algorithm 6 Transport by robot i

$j := 0$;
while $j < \rho$ **do**
    Receive msg;
    if $(msg.type = 2) \wedge (msg.origin = j) \vee j \neq i$ **then**
        $j := j + 1$;
    end if
end while
for $j := n - 1 \to 0$ **do**
    $\lambda := T_{ij}$;
    repeat
        Calculate α_λ using Equation (4); **Spin** towards α_λ using FSM of Fig. 1b;
    until $\beta_i^t \approx \alpha_\lambda$
    Send $msg\langle 3, i, j, \emptyset\rangle, j = 0, \dots \rho - 1 \wedge j \neq i; k := 0$;
    while $k < \rho - 1$ **do**
        Receive msg;
        if $(msg.type = 3) \wedge (msg.origin = k) \vee k \neq i$ **then**
            $k := k + 1$;
        end if
    end while
    repeat
        Calculate D_λ, D_r, D_ℓ and $D_\oplus$ using Equation (3); **Calculate** α_λ, θ_r and θ_ℓ;
        Move towards λ with orientation α_λ with maintaining $D_\oplus \pm d$, $D_r \pm d$, $D_\ell \pm d$, θ_r= θ_ℓ= $\phi \pm \sigma$.
    until $p_i^t \approx p_i^+$
    Send $msg\langle 4, i, j, \emptyset\rangle, j = 0, \dots \rho - 1; k := 0$;
    while $k < \rho - 1$ **do**
        Receive msg;
        if $(msg.type = 4) \wedge (msg.origin = k) \vee (k \neq i)$ **then**
            $k := k + 1$;
        end if
    end while
end for

.

5 Performance results

The implementation of this work is done via CoppeliaSim simulator[1], using a swarm of Khepera-III robots[2]. For the analysis of the search and recruitment, an arena with dimensions of $10m \times 10m$ is defined, using two different discretization steps. For the step $0.5\ m$, , we have $\Lambda = 255$, while for $1\ m$, $\Lambda = 72$. Table 1 presents the simulation parameters and average execution time of search and recruitment stages. It is notable that, in the search stage, the execution time is inversely proportiional to the size of the diccretization step. This occurs because a smaller distance between the robot and the landmark leads to a lower robot velocity to ensure better route adjustment. Therefore, a smaller discretization step results in lower speeds. Recruitment, on the other hand, is independent of this parameter. The only factor affecting this stage is the distance between the swarm and the object at the moment it is detected.

For the analysis of the transport peformance, two case studies are conducted, varying the swarm and the object sizes. The first case study investigates the impact of the swarm size on the transport performance, with a target object of radius $r_\oplus = 0.3\ m$ and mass $M = 0.3\ kg$. The second case investigates the impact of the target object size, using a swarm of examines this performance for a swarm of 8 robots. The simulations for the two case studies can be accessed at `https://linktr.ee/transportbycaging`. The obtained results are shown in Table 2. The error is the absolute difference between the path minimum length and that of the path traversed by the object center of mass. For all cases, the average error in the trajectory is around 3.8%. In addition to the error, the average execution time is $247.5s$. The impact on the execution time is due to the distance between the swarm and the object at the moment that recruitment begins. As for the trajectory error, it is evident that this error remains low and constant, highlighting the robustness of the proposed methodology.

[1]CoppeliaSim: Robotics Simulation Plataform, `https://www.coppeliarobotics.com`, Last acessed: August 18th, 2023.
[2]Khepera-III: User Manual, `https://ftp.k-team.com/KheperaIII`, Last acessed: August 18th, 2023.

Table 1. Simulation parameters and Search and recruitment results

Simulation parameters					
PIDs	α_p	D_λ	D_ℓ, D_r	θ_ℓ , θ_r	$D_\oplus$
k_p	1	1	0.3	0.3	1
k_i	0.01	0.01	0.01	0.01	0.01
k_d	0.01	0.01	0.1	0.1	0.1
$d = 0.05\ m$		$\sigma = 3°$	$c = 0.01\ m$	$\Delta = (r_\oplus + w/2 + 0.6)\ m$	

Search and recruitment results		
	$Step\ (m)$	$Time\ (s)$
Search	0.5	222.165
Search	1	197.535
Recruitment	0.5	164.481
Recruitment	1	164.481

Table 2. Simulation results for case studies 1 and 2

Case 1: $r_\oplus = 0.3\ m$; $M = 0.3\ kg$			
ρ	$length\ (m)$	$error\ (m)$	$Time\ (s)$
6	4.3284	0.1778	215.07
7	4.7426	0.1894	239.35
8	3.9142	0.1526	244.79

Case 2: $\rho = 8$				
$r_\oplus\ (m)$	$M\ (kg)$	$length\ (m)$	$error\ (m)$	$Time\ (s)$
0.30	0.3	3.9142	0.1526	244.79
0.40	0.7	4.3284	0.1686	282.47
0.45	1.0	4.7426	0.1474	258.62

6 Conclusion

In this work, a method is proposed for the implementation of collective transport by caging a cylindrical object. The method is distributed and coordinated by message passing among the swarm robots. The obtained results demonstrate that it allows the caging to be maintained throughout the transport trajectory. By discretizing the path traveled by the object, it is possible to analyze the error regarding the position of its center of mass. This analysis shows that by respecting the maximum and minimum of swarm size limits based on the physical and dynamic characteristics of the object and robots, it is possible to transport an object with swarms of different sizes. Additionally, the error shows little variation, regardless of the swarm size and/or object characteristics.

The proposed approach for the search phase does not guarantee that the first robot to detect the object has taken the shortest path between its initial position and the whereabout of the target object. As this path is traversed in reverse during the actual transport, it is necessary to optimize it. Therefore, the exploitation of path optimization algorithms, ensuring an optimized path. The incorporation of more complex arena topologies is intrinsically linked to the progression of the project, aiming to validate the robustness of the proposed method. The inclusion of static or dynamic obstacles in the arena is one another considered extension of this work. Another possible direction for future work is expanding the scope of the project through the implementation of a collaborative approach involving multiple and foraging for multiple target obejcts.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. C. Pratt. Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proceedings of the IEEE*, vol. 99, n. 9, pp. 1470–1481, 2011.

[2] V. S. Vardharajan, K. Soma, and G. Beltrame. Collective transport via sequential caging. In F. Matsuno, S.-i. Azuma, and M. Yamamoto, eds, *Distributed Autonomous Robotic Systems*, pp. 349–362, Cham. Springer International Publishing, 2022.

[3] Z. Wang and V. Kumar. Object closure and manipulation by multiple cooperating mobile robots. vol. 1, pp. 394–399, 2002.

[4] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, vol. 31, pp. 307–321, 2015.

[5] A. Sudsang and J. Ponce. On grasping and manipulating polygonal objects with disc-shaped robots in the plane. vol. 3, pp. 2740–2746, 1998.

[6] A. Becker, C. Ertel, and J. McLurkin. Caging and swarm manipulation experiments: A massive user study with large swarms of simple robots, 2014.

[7] R. Gross and M. Dorigo. Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, vol. 1, n. 1-2, pp. 1–13, 2009.

[8] E. Rimon and A. Blake. Caging 2d bodies by 1-parameter two-fingered gripping systems. vol. 2, pp. 1458–1464, 1996.