XLIII CILAMCE
21–25, November, 2022
FOZ DO IGUAÇU – PR – BRAZIL

Cooperative transport of objects by multi-robots

Gustavo B. Ferreira[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
gustavobueno181@gmail.com, nadia@eng.uerj.br
[2]*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. Coordinated transport of objects by robotic swarms can be advantageous when the object to be transported is too large and/or too heavy to be effectively handled by a single robot alone. This type of operation requires coordination and synchronization of the pushing forces, which are to be exerted by the robots in order for the object to be transported successfully. In this paper, a new algorithm called Cooperative Object Transport is proposed. The algorithm has four stages. The first stage implements the object search done by the swarm robots. The second stage of the algorithm starts when a robot finds the object. It then recruits the other robots. During the third stage, each robot first computes and then goes towards the location where it must position itself around the object to cooperate in the pushing actions required for executing the transport properly. When the positioning of all robots around the object is completed, the fourth stage of the algorithm begins. From then on, the robots start alternating between pushing and re-positioning actions in order to keep the object moving along the expected transport trajectory. This procedure is executed until the object is actually homed. In this work, the implementation of the object search stage is inspired by the strategy used by ant colonies during their foraging for food sources, depositing pheromones along the traversed paths. The proposed algorithm is implemented in GRITSBot swarm robots, using Robotarium. We evaluate the impact of ant foraging strategy on the overall collective transport task in swarm robotics, comparing it to a random object-seeking strategy. The achieved results prove that the proposed algorithm is effective yet efficient in finding the shortest path while looking for an object. This allows for an efficient execution of the transport stage of the sought object from its original location to the intended warehouse.

Keywords: Cooperative transport, Swarm robotics, Pheromone, Swarm intelligence, Pushing object.

1 Introduction

The cooperative transport of objects by robots is a topic that has been studied for a long time. It is addressed in many research areas, promoting different approaches such as transport by caging Wang et al. [1], transport by grasping Habibi et al. [2] and transport by pushing Chen et al. [3]. To carry out an object transport cooperatively, the idea of dividing it into a series of simple resolution stages is proposed. The coordinated execution of the transport stage allows the execution of the transport of the object. Stages are dynamic processes, as they need to be continually adjusted in response to changes regarding the orientation of the object to be transported. An immediate solution to this problem is to make all robots push in the same direction Fujisawa et al. [4]. The distribution of robots around the object presents a similar approach in Fujisawa et al. [4]. This distribution of robots in relation to the object increases the difficulty of the problem, since the robot does not have a complete view of the environment. Communication takes place indirectly through the position of the object.

A variety of cooperative transport approaches in the literature were reviewed as part of this work. In Wang et al. [1], they describe a variable internal force control algorithm to guide a group of three omnidirectional robots needed to transport an object in the shape of a cube. Only the leader pushes the object, while the followers hold the object's sides tightly. In Habibi et al. [2], they present four motion controllers distributed to allow a group of robot manipulators to collectively carry out the transport of a large object. While others map the environment in order to guide the transport robots towards the destination. In Fujisawa et al. [4] the cooperative transport is based on the behavior of ants, in which they use indirect communication via artificial pheromones. The task requires the robots to perform a random search to find a food, in this case an object, and transport it to a destination location.

In this work, an algorithm for cooperative object transport is proposed. Reference points are coordinates (x, y). At each iteration, each robot chooses a reference point from the neighborhood where it is currently located to perform a search for the object. After the first robot gets close to the object, it recruits the others, who position themselves and then carry out the transport of the object to its final destination. The proposed algorithm allows efficient task execution by robots with limited resources to carry out the transport. This algorithm is tested on GRITSBot-type robots Pickem et al. [5] to demonstrate its effectiveness and efficiency.

The rest of this paper is organized into six sections. Initially, in the section 2 presents the optimization based on an ant foraging strategy. Next, in Section 3, the problem of cooperative object transport is presented. Later, in Section 4, the proposed algorithm is detailed. Section 5, the implementation of the proposed algorithm in virtual robots is described and presents and discusses the experimental results obtained. Finally, in the section 6, conclusions and future work are presented.

2 Ant Colony System

Real ants are able to find the shortest path from a food source to their nest without using visual cues by exploiting pheromones information. As they walk, ants deposit pheromone on the ground and probably follow the pheromones previously deposited by other ants Dorigo and Gambardella [6].

One version of these algorithms is called the ant colony system (ACS). Each ant builds a route by repeatedly applying a stochastic state transition rule. When building its route, an ant also modifies the amount of pheromone at visited reference points by applying the local update rule. The effect of this rule is to make the reference points preference dynamically change. Every time an ant visits a reference point, it becomes a little less desirable as it loses some of its pheromone. In this way, the ants make better use of the pheromone information, as without local updating, all ants would search in a close neighborhood of the previous best route. Once all the ants have built their route, the amount of pheromone in the reference points is modified again, applying the global pheromone update rule. The ants are guided during the construction of their routes both by heuristic information and by pheromone information. A reference point with a lot of pheromone is a very desirable choice. Pheromone update rules are designed so that they tend to provide more pheromones to reference points that should be visited by ants. Intensification and diversification probabilities are used to implement to select the ant next destination.

3 Cooperative Transport Problem

Cooperative Object Transport (COT) is the process that manages and organizes a swarm of robots to correctly execute a set of stages, aiming at a single objective. This process consists of searching for the object, recruiting the other robots, positioning the robots, and transporting the object in an organized and distributed way.

In order to provide a formal definition of the COT problem, let $S = \{0, 1, \ldots, \omega-1\}$ be the set of ω reference points and $R = \{0, 1, \ldots, \rho - 1\}$ the set of ρ robots. These variables are independent. The position of robot i is $p_i = [x_i, y_i, \beta]^T$, where x_i and y_i denote the position of the robot's centroid and β its bearing. We assume that the calculation between the reference points takes place along the Euclidean distance. So we have a distance matrix of $\omega \times \omega$. The neighborhood of the reference points is defined as a matrix of the same size as the matrix of the distance between the reference points, defined as V. The initial value of the pheromone matrix τ is $\tau = 0.1 \times (\omega \times \omega)$.

The first stage of the process is the search for the object. In this work, the search is performed using ACS. All robots start around the reference point w_0, to which they will have to take the object. The robot i chooses a reference point in its neighborhood. This stage is performed until a robot reaches the reference point w^*. Note that w^* is the identifier of the reference point where the object is. The robot that finds the object first has its traveled route C_i chosen for the following stages, being it the shortest route C^*. The next stage consists of recruiting the remaining robots to the object. In this work, we use direct communication between robots for recruitment. They have as their primary reference C^*. In this way, the remaining robots rely on it to obtain the new route and arrive at the reference point w^*. The penultimate and last stages consist of the initial positioning and transport of the object. In this work, we developed, the robots position themselves behind the object and transport it along the route C^*.

4 Cooperative Object Transport Algorithm

The COT implementation is structured into five stages, as shown in Algorithm 1. Take note that W represents the set of reference points, P represents the set of robot positions, and α represents the angle between the horizontal line that passes through the center of the object and the horizontal line that passes through the center of the penultimate reference point of the route C^*. The choice of reference point w^* can be done randomly, except for the initial reference point w_0.

Algorithm 1 COT

Require: ρ, ω, V;
1: Initialize arena(ρ, ω);
2: Search for object by robot $i(W, P, \tau, V)$;
3: Recruit robot $i(p_i, C_i, C^*, W, \tau, V)$;
4: Position robots(w^*, C^*, P);
5: Transport object(w^*, w_0, α, P, C^*);

In the arena initialization stage, the positions of the reference points W and the initial positions of the robots P are determined, both of which are chosen randomly. In the object search stage, at each iteration, the search for the reference point w^* is performed. In this stage, it is verified whether the desired reference point w^* is reached, through the difference between the position of the robot i (p_i) and the distance sensor of the robot i, defined as $sensor_i$, and this sensor is emulated. If not, the path to the last chosen reference point w_u is performed. The next stage is recruitment, which takes place after one of the robots has reached w^* and the rest of the robots have reached their respective reference points. After recruitment has been achieved, the robots position themselves to start the transport stage. Finally, the robots transport the object to its final destination.

4.1 Object Search Stage

The object search stage is implemented as shown in Algorithm 2. During this stage, the robot i uses the state transition rule to select a reference point g. This step is described in Algorithm 3. Robot i, at its current reference point w_i, chooses to move to the reference point in its neighborhood $V_i^{w_i}$ that has not been visited yet. Variables Pr_{int} and Pr_{div} represent the intensification and diversification probabilities, respectively. The pheromone matrix τ is used in the calculation of probabilities. The parameters al and be are used to determine the influence of pheromone and heuristic information, q_0 is a value between 0 and 1. If $q \leq q_0$ then the best reference point according to the probability of intensification is chosen, otherwise a reference point is chosen according to the probability of diversification. This way, the reference point g is assigned to the last position u_i of your route C_i. On the way to the reference point g, the pheromone is updated by the local update rule. Where, τ_0 and ξ are parameters determined in empirical ways.

The robot that finds the object first has its route chosen as the shortest route C^*. The route C^* is one of the entries in the recruiting stage. The remaining robots each head towards the last chosen reference point. After each robot i builds its route C_i, the pheromone is updated by the global update rule. Where, p is the evaporation rate and Q is a constant. Evaporation and pheromone deposition only occur at reference points on the route C^*.

4.2 Recruitment Stage

The Algorithm 4 describes the proposed method to implement the recruitment of robots from the swarm to arrive at the object. This stage is responsible for taking the robots, which did not find the object, to the same, by the route C^* and, if the robot i is already close to the object, it remains there. Note that b is a variable for the last reference point of the route C^* and that g is a variable for a given reference point.

Initially, the comparison between the last reference point of the route C_i and the route C^* is performed. In this comparison, four situations are possible:

1. $C_i[u_i] = w^*$, indicates that the last reference point of your route C_i is the reference point where the object is located w^*. As a result, the robot i remains in its current position p_i.
2. $(C_i[u_i] \in C^*) \wedge (C_i[u_i] \neq w^*)$, indicates that the last reference point of your route C_i belongs to the route C^*, but this reference point is not the reference point w^*. Therefore, the robot i starts from this reference point in route C^* until it completes the entire route C^*.
3. $C_i[u_i] \notin C^*$, indicates that the last reference point of your route C_i does not belong to route C^*, there are two cases here. In the first case, if there is a reference point in the vicinity of this last reference point of its route C_i, in route C^*, the robot i moves to this point and then the robot i follows the previous item. The neighborhood is $V_i^{C_i[u_i]}$. In the second case, if there is no reference point in neighborhood $V_i^{C_i[u_i]}$, in route C^*, the robot i chooses the next reference point in similar way as during the search stage. There are two cases. In the first case, robot i reaches directly a reference point in route C^*. In the second case, the robot takes an alternative path to the object because its neighborhood does not include any reference point that belongs to route C^*.

Algorithm 2 Search for object by robot i

Require: W, P, τ, V;
Ensure: C_i, C^*, p_i, w^*, τ;
1: $var_i \leftarrow false$; $obj\_enc \leftarrow false$; $u_i \leftarrow 0$; $mc \leftarrow \infty$, {mc is the length of the shortest route}
2: **while** $\neg obj\_enc$ **do**
3: $g \leftarrow$ Choose the next position of the robot $i(W, \tau, V, w_i)$; $C_i[u_i] \leftarrow g$;
4: **while** $sensor_i > 0.5 \wedge p_i \neq g$ **do**
5: robot i advances towards g;
6: **end while**;
7: $\tau_{u_{i-1},u_i} \leftarrow (1-\xi) \times \tau_{u_{i-1},u_i} + \xi \times \tau_0$; $u_i \leftarrow u_i + 1$;
8: **if** $sensor_i \leq 0.5$ **then**
9: $w^* \leftarrow r$; $oe_i \leftarrow true$; {found object variable}
10: **end if**
11: **for** $i := 0 \rightarrow \rho - 1$ **do**
12: **if** $oe_i = true$ **then**
13: $obj\_enc \leftarrow true$; $z \leftarrow fc(C_i)$; {fc calculate the length of the route C_i}
14: **if** $z < mc$ **then**
15: $mc \leftarrow z$; $C^* \leftarrow C_i$; $u^* \leftarrow u_i$;
16: **end if**
17: **end if**
18: **end for**
19: **end while**;
20: **for** $j := 0 \leftarrow u^* - 1$ **do**
21: $\tau_{C_j^*,C_{j+1}^*} \leftarrow (1-p) \times \tau_{C_j^*,C_{j+1}^*} + p \times (Q/mc)$
22: **end for**

Algorithm 3 Choose the next position of the robot i

Require: W, τ, V, w_i;
Ensure: g;
1: $sum \leftarrow 0$; $p_{ot} \leftarrow 0$; {p_{ot} the best probability}; $ot \leftarrow 0$; {p_{ot} the best probability index}
2: **for** $j \in V_i^{w_i}$ **do**
3: $Pr_{div}[j] \leftarrow (\tau_{w_i,j})^{al} \times (1/V_{w_i,j})^{be}$; $sum \leftarrow sum + Pr_{div}[j]$; $Pr_{int}[j] \leftarrow (\tau_{w_i,j}) \times (1/V_{w_i,j})^{be}$;
4: **if** $p_{ot} < Pr_{int}[j]$ **then**
5: $p_{ot} \leftarrow Pr_{int}[j]$; $ot \leftarrow j$;
6: **end if**
7: **end for**
8: $Pr_{div} \leftarrow Pr_{div}/sum$;
9: **Generate** randomly a value between 0 and 1;
10: **if** $q \leq q_0$ **then**
11: $g \leftarrow ot$;
12: **else**
13: $g \leftarrow roleta(Pr_{div})$;
14: **end if**

4.3 Initial Positioning Stage

Algorithm 5 describes how the robots position themselves behind the object towards the penultimate reference point of the route C^*. Note that u^* is the last reference point of the route C^*.

The positions are calculated in Algorithm 6, considering the angle α, the diameter of the robot d, the diameter of the object D and the distribution of robots behind the object. The distribution depends on the number of robots and the angular difference between each robot, represented by the angle θ. Each robot has a line that passes through the center of it in the direction of its motion, which passes through the center of the object. The angle θ is formed when these lines intersect at the center of the object. If the number of robots is even, the robot with an even number is shifted to the right of the angle α. However, if the robot number is odd, the robot is offset to the left of the angle α. If the number of robots is odd, only robot 0 is positioned in the same direction of the angle α, and the other robots are positioned in the same way as in the previous case. The angle β is the one at which the robot i initially positions itself in relation to the horizontal line of the object to start the transport.

Algorithm 4 Recruit robot i

Require: $p_i, C_i, C^*, W, \tau, V$;
Ensure: p_i
1: $rota\_enc = false$;
2: **if** $C_i[u_i] \in C^*$ **then**
3: Be $b | C^*[b] = C_i[u_i]$; $rota\_enc \leftarrow true$;
4: **end if**
5: **while** $sensor_i > 0.5$ **do**
6: **if** $rota\_enc = true$ **then**
7: $g \leftarrow C^*[b]$; $b \leftarrow b + 1$;
8: **else**
9: **if** $\exists b \in V_i^{C_i[u_i]} | b \in C^*$ **then**
10: $g \leftarrow b$; $rota\_enc \leftarrow true$;
11: **else**
12: $g \leftarrow$ Choose the next position of the robot $i(W, \tau, V, w_i)$;
13: **end if**
14: **end if**
15: **while** $p_i \neq g$ **do**
16: robot i advances towards g;
17: **end while**
18: **end while**

Algorithm 5 Position robots

Require: w^*, C^*, P;
Ensure: P;
1: **for parallel** $i := 0 \rightarrow \rho - 1$ **do**
2: $p_i^+ \leftarrow$ Initial position of robot $i(w^*, p_i, C^*, u^*)$;
3: **while** $p_i \neq p_i^+$ **do**
4: robot i advances towards p_i^+;
5: **end while**
6: $P \leftarrow P \cup p_i^+$
7: **end for parallel**

Algorithm 6 Initial position of robot i

Require: w^*, p_i, C^*, u^*;
Ensure: p_i^+, α;
1: $u \leftarrow u^* - 1$; $w_u \leftarrow C^*[u]$; $\alpha := \arctan\left((w_{uy} - w_y^*)/(w_{ux} - w_x^*)\right)$
2: **if** $\rho \mod 2 = 0$ **then**
3: **if** $i \mod 2 = 0$ **then**
4: $\beta \leftarrow \alpha + \theta \times (((i \textbf{ div } 2) + 1) - 0,5)$;
5: **else**
6: $\beta \leftarrow \alpha + \theta \times ((-(i + 1) \textbf{ div } 2) + 0,5)$;
7: **end if**
8: **else**
9: **if** $i \mod 2 = 0$ **then**
10: $\beta \leftarrow \alpha + \theta \times (i \textbf{ div } 2)$;
11: **else**
12: $\beta \leftarrow \alpha - \theta \times ((i + 1) \textbf{ div } 2)$;
13: **end if**
14: **end if**
15: $x_i := w_x^* + ((d + D)/2) \times \cos(\beta + \pi)$; $y_i := w_y^* + ((d + D)/2) \times \sin(\beta + \pi)$; $p_i^+ \leftarrow [x_i, y_i, \beta]^T$;

4.4 Transport Stage

The Algorithm 7 describes the execution of object transport by all robots. This algorithm runs until the robots travel the entire route C^* in the opposite direction. Updating the robots' position after each push is performed

by Algorithm 8. The distribution of robots behind the object is explained in the initial positioning stage. After each push, a new angle α^+ is evaluated. Its calculation is the same as the one used for the angle α in the initial positioning stage. If there is a deviation greater than the angular error $\Delta\theta_0$, the robots position is calculated again with the angle α^+. Otherwise, angle α is used. After the push, the position of the target is updated (line 8).

Algorithm 7 Transport object

Require: w^*, w_0, α, P, C^*;
1: **repeat**
2: **for parallel** $i := 0 \rightarrow \rho - 1$ **do**
3: $p_i^+ \leftarrow$ New position for robot $i(w^*, \alpha, p_i, C^*, u^*)$;
4: **while** $p_i \neq p_i^+$ **do**
5: robot i advances towards p_i^+ pushing the target;
6: **end while**
7: **end for parallel**
8: $w_x^* := w_x^* + v_x \times t + a_x \times \frac{\Delta t^2}{2}$; $w_y^* := w_y^* + v_y \times t + a_y \times \frac{\Delta t^2}{2}$; $w^* \leftarrow [w_x^*, w_y^*]^T$ {new target position};
9: **until** $w_u = w_0$

Algorithm 8 New position for robot i

Require: $w^*, \alpha, p_i, C^*, u^*$;
Ensure: α, p_i^+
1: $u^* \leftarrow u^* - 1$; $w_u \leftarrow C^*[u^*]$; $\alpha^+ := \arctan\big((w_{uy} - w_y^*)/(w_{ux} - w_x^*)\big)$; $\Delta\theta_1 := |\alpha^+ - \alpha|$;
2: **if** $\Delta\theta_1 > \Delta\theta_0$ **then**
3: $\alpha \leftarrow \alpha^+$
4: **end if**
5: **if** $\rho \mod 2 = 0$ **then**
6: **if** $i \mod 2 = 0$ **then**
7: $\beta \leftarrow \alpha + \theta \times (((i \textbf{ div } 2) + 1) - 0,5)$;
8: **else**
9: $\beta \leftarrow \alpha + \theta \times ((-(i+1) \textbf{ div } 2) + 0,5)$;
10: **end if**
11: **else**
12: **if** $i \mod 2 = 0$ **then**
13: $\beta \leftarrow \alpha + \theta \times (i \textbf{ div } 2)$;
14: **else**
15: $\beta \leftarrow \alpha - \theta \times ((i+1) \textbf{ div } 2)$;
16: **end if**
17: **end if**
18: $x_i := w_x^* + ((d + D)/2) \times \cos(\beta + \pi)$; $y_i := w_y^* + ((d + D)/2) \times \sin(\beta + \pi)$; $p_i^+ \leftarrow [x_i, y_i, \beta]^T$;

5 Performance Results

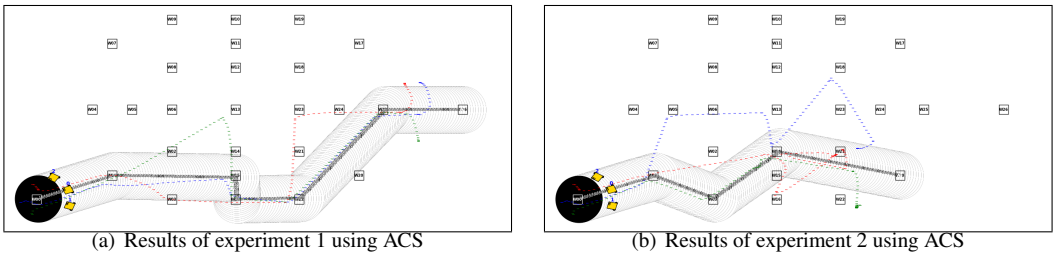
The proposed algorithm was implemented in a swarm of robots of the GRITSBot type. Robots are currently not equipped with sensor boards as distance detection can be emulated via the back-end server which tracks the positions of all robots. The global position of all robots is retrieved through an aerial tracking system that uses a single webcam together with ArUco tags for Pickem et al. [5] tracking. Simulations are done in the Robotarium, which is a remotely accessible multi-robot research center Pickem et al. [5]. The Robotarium uses Safety Barrier Certificates to ensure collision-free behavior of all robots. Direct communication is only emulated when the first robot finds the reference point w^*. Thus, the other robots are aware of the route C^*. For the transport phase, communication is indirect, using the object for communication, similar to the one used in Wang et al. [1]. Security Barrier Certificates are used at all stages. However, they are not used to the object, as the robots have to push the object.

The constants used in the COT are specific to the Robotarium. The robot's diameter d is 0.11 m. The robot's mass mr is 0.06 kg. The time step t is the inverse of the Robotarium update frequency (3 Hz). The speed v represents the maximum speed that can be reached by the robot, in this case it is 0.2 m/s. Acceleration a denotes the robot's speed variation, going progressively from 0 to v.

The analysis of the results makes it possible to evaluate the effectiveness of the COT, as well as its efficiency, in achieving the main objective of transporting the object, through the fulfillment of all the corresponding stages.

The parameters of the experiments are presented below. The number of robots used ρ is 3. The number of reference points ω is 27, with point 0 being the target and point 26 where the object is at the beginning of the first experiment. In the second experiment, the object is at point 20. The object's mass m is 1 kg, in this case 16 times greater than that of the robot and its diameter D is 0.6 m, in this case 5 times greater than that of the robot. The angular difference that one robot need to the other is 45° in the initial positioning step. During the transport step, this angular difference is maintained as far as possible. The angular error $\Delta\theta_0$ represents the angular deviation of the robot front from the straight line formed by the center of the transported object and the next reference point. The tolerated angular error is 0.03. The angular error value is defined by simulations. For values greater than $\Delta\theta_0$, the object's center would deviate greatly from the straight line connecting it to the next reference point, while smaller values would cause a constant repositioning of the object. More weight is given to heuristic information al than to pheromone influence be, as greater diversification is needed in the search for the object in the experiments is al 0.1 and be is 0.9. The evaporation rate is 0.1, the pheromone decay coefficient ξ is 0.1. Minimum pheromone level τ_0 is 0.01 and constant Q is 5 and were determined empirically. The probability q_0 is 0.4, to have greater diversification in the search for the object.

Figures 1(a) and 1(b) illustrate the object transport for both experiments using the ACS method. Figures 1(c) and 1(d) illustrate the object transport for both experiments using a random search.



(a) Results of experiment 1 using ACS

(b) Results of experiment 2 using ACS

(c) Results of experiment 1 using random search

(d) Results of experiment 2 using random search

Figure 1. Simulation results of the collective object transport

The search stage time is the time that a robot takes to find an object and, for the other robots, it is the time that each robot takes to reach the last reference point chosen through the state transition rule. The waiting time is the difference between the time the robot arrived at its last reference point chosen by the state transition rule and the time that the robot completed the search stage. The time to reach the object is the time of the recruitment stage. The time to position themselves is the time used by each robot in the initial positioning stage. Finally, the time to transport the object is the time for the robots to transport the object along the route C^* in reverse.

The duration of the stages performed by each robot in the first and second experiments using the ACS method are shown in Table 1 and for the same experiments using the random search method are shown in Table 2.

In the search stage, in relation to the two experiments in Table 1, the robots find the object faster than using the random search method in Ferreira et al. [7], Table 2 presents its results. As the robots find the object faster using the ACS method, the time to transport it ends up being affected. The reduction for transporting the object to the target is 64% in the first experiment and 82.5% in the second experiment.

6 Conclusion

In this work, an algorithm was proposed to transport a circular object to its final destination using robots. Experimental results prove the effectiveness of the algorithm in transporting the object along the shortest route found by the robots. The method used by the robots to choose the reference points in the search stage didn't provide an efficient route from the object to the target, as the experiments showed. However, the results are better

Table 1. Results of the experiments using ACS

	Experiment 1			Experiment 2		
Robots	1	2	3	1	2	3
Time of search stage (s)	56.96	60.26	56.21	46.76	33.63	33.75
Time of wait (s)	3.3	0	4.05	0	13.13	13.01
Time to reach the object (s)	56.96	71.38	56.21	63.88	33.63	68.36
Time to position themselves (s)	36.66	36.7	36.75	33.4	33.45	33.5
Time to transport the object (s)	2521.58			1532.4		
Route C^*	[0,1,15,16,22,25,26]			[0,1, 3, 14, 20]		
Length of route C^* (m)	6.61			4.61		

Table 2. Results of the experiments using random search

	Experiment 1			Experiment 2		
Robots	1	2	3	1	2	3
Time of search stage (s)	128.85	71.76	111.1	53.77	117.16	103.15
Time of wait (s)	0	57.09	17.75	63.39	0	14.01
Time to reach the object (s)	244.54	71.76	203.62	53.77	150.35	187.50
Time to position themselves (s)	36.92	36.85	37.05	33.25	33.28	33.30
Time to transport the object (s)	4135.18			2796.70		
Route C^*	[0,1,16,20,23,25,26]			[0,3, 6, 15, 20]		
Length of route C^* (m)	7.61			6.11		

when compared with those obtained using a random choice during the search stage. In future work, we intend to investigate other transport methodologies.

References

[1] Z. Wang, Y. Takano, Y. Hirata, and K. Kosuge. A pushing leader based decentralized control method for cooperative object transportation. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1035–1040, Sendai, Japan. IEEE, 2004.

[2] G. Habibi, Z. Kingston, W. Xie, M. Jellins, and J. McLurkin. Distributed centroid estimation and motion controllers for collective transport by multi-robot systems. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1282–1288, Seattle, WA, USA. IEEE, 2015.

[3] 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.

[4] R. Fujisawa, H. Imamura, and F. Matsuno. *Cooperative transportation by swarm robots using pheromone communication*, pp. 559–570. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[5] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1699–1706, 2017.

[6] M. Dorigo and L. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, vol. 1, n. 1, pp. 53–66, 1997.

[7] G. B. Ferreira, N. Nedjah, and L. M. Mourelle. Transporte cooperativo de objeto por multi-robôs. submitted, 2022.