

Transporte cooperativo de objeto por multi-robôs

Gustavo Bueno Ferreira* Nadia Nedjah**
Luiza de Macedo Mourelle***

* Departamento de Engenharia Eletrônica e Telecomunicações,
Universidade do Estado do Rio de Janeiro, RJ (e-mail:
gustavobueno181@gmail.com).

** Departamento de Engenharia Eletrônica e Telecomunicações,
Universidade do Estado do Rio de Janeiro, RJ (e-mail:
nadia@eng.uerj.br)

*** Departamento de Engenharia de Sistemas e Computação,
Universidade do Estado do Rio de Janeiro, RJ, (e-mail:
ldmm@eng.uerj.br)

Abstract: The transport of object by groups of robots can be advantageous when an object is too large or too heavy to be effectively handled by a single robot alone. It requires the coordination and synchronization of pushing forces to be exercised by the robots. In this paper, we propose a new algorithm called Cooperative Transport of Object (CTO). The algorithm has four stages. The first one implements the search for object. The second stage of the algorithm start when a robot finds the object. It then recruits the other robots. During the third stage each robot first computes then goes to the position it must be located at to execute the pushing actions. When the positioning of all robots around the circular object is completed, the fourth stage commences. Therein, the robots start alternating between pushing and positioning actions to maintain the object within the expected transport trajectory. This is performed until the object is homed. The algorithm is implemented in GRITSBot swarm robots, using Robotarium. The algorithm is effective to carry out the stages and the corresponding performance is promising.

Resumo: O transporte de objetos por grupos de robôs pode ser vantajoso quando um objeto é muito grande ou muito pesado para ser efetivamente manipulado por um único robô sozinho. Este requer a coordenação e sincronização de forças de empurre a serem exercidas pelos robôs. Neste artigo, é proposto um novo algoritmo chamado Transporte Cooperativo de Objeto (TCO). O algoritmo possui quatro etapas. A primeira implementa a busca pelo objeto. A segunda etapa do algoritmo começa quando um robô encontra o objeto. Em seguida, recruta os demais robôs. Durante a terceira etapa cada robô primeiro calcula e depois vai para a pose em que deve estar localizado para executar as ações de empurre. Quando o posicionamento de todos os robôs ao redor do objeto circular é concluído, a quarta etapa começa. A partir de então, os robôs começam a alternar entre ações de empurre e posicionamento para manter o objeto dentro da trajetória de transporte esperada. Isso é executado até que o objeto seja alojado. O algoritmo é implementado em robôs de enxame GRITSBot, utilizando o Robotarium. O algoritmo é eficaz para realizar as etapas e o desempenho correspondente é promissor.

Keywords: Cooperative transport, Swarm robots, Algorithm, Swarm intelligence, Pushing object.

Palavras-chaves: Transporte Cooperativo, Enxame de robôs, Algoritmo, Inteligência de enxame, Empurrando objeto.

1. INTRODUÇÃO

O transporte cooperativo de objeto por robôs é um tema que vem sendo estudado há muito tempo. Este tema é abordado em muitas áreas de investigação, promovendo diferentes abordagens, como transporte por enjaulamento [Wang et al. (2004)], transporte por agarramento [Habibi et al. (2015)] e transporte por impulsionamento [Chen et al. (2015)]. Os algoritmos para cada abordagem apresentam dificuldades distintas. Em todas as abordagens, as etapas de busca e recrutamento do objeto podem ser semelhantes, a distinção estando na etapa de transporte.

Para realizar um transporte de objeto cooperativamente, propõe-se a ideia de dividi-lo em uma série de etapas de resolução simples. A execução coordenada principalmente da etapa de transporte permite a execução do transporte do objeto. As etapas são processos dinâmicos, pois precisam ser continuamente ajustadas em resposta às mudanças no ângulo do objeto a ser transportado para o próximo ponto e/ou no desempenho dos robôs. Uma solução imediata para resolver este problema é fazer todos os robôs empurrarem na mesma direção [Fujisawa et al. (2013)]. A distribuição dos robôs ao redor do objeto apresenta uma abordagem semelhante em [Fujisawa et al. (2013)]. Essa distribuição

dos robôs em relação ao objeto aumenta a dificuldade do problema, pois, o robô não possui uma visão completa do ambiente. A comunicação se dá de forma indireta através da posição do objeto.

Diversas aplicações de transporte, que utilizam enxame de robôs, requerem o uso de cooperação coordenada. Por exemplo, em situações que são impraticáveis por humanos ou representam risco [Dai et al. (2015); Stroupe et al. (2005)].

Neste artigo, é proposto um algoritmo para transporte cooperativo de objeto. Os pontos de referência são coordenadas (x, y) . A cada iteração, cada robô escolhe um ponto de referência da adjacência, em que se encontra no momento, para realizar uma busca pelo objeto. Após o primeiro robô chegar perto do objeto, ele recruta os demais, que se posicionam e, então, executam o transporte do objeto até seu destino final.

O algoritmo proposto permite a execução por robôs com recursos limitados para realizar o transporte, por exemplo, garras para puxar ou levantar o objeto. Esse algoritmo foi testado em robôs do tipo GRITSBot [Pickem et al. (2017)] para demonstrar sua eficácia e eficiência.

O restante deste artigo está organizado em sete seções. Inicialmente, na Seção 2, são apresentados trabalhos relacionados. A seguir, na Seção 3, é apresentado o problema do transporte cooperativo de objeto. Posteriormente, na Seção 4, o algoritmo proposto é detalhado. Na Seção 5, é descrita a implementação do algoritmo proposto em robôs virtuais. A Seção 6 apresenta e discute os resultados experimentais obtidos. Por fim, na Seção 7, são apresentadas as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

Uma variedade de abordagens de transporte cooperativo na literatura foi revisada como parte deste trabalho. Em [Wang et al. (2004)] descrevem um algoritmo de controle de força interna variável para guiar um grupo de três robôs omnidirecionais necessários para transportar um objeto no formato de um cubo. Apenas o líder empurra o objeto, enquanto os seguidores seguram os lados do objeto com força. A comunicação indireta se dá através do objeto e é suficiente para permitir aos seguidores manterem a formação e contribuir para o transporte. Em [Fink et al. (2007)] os robôs estimam localmente o fechamento do objeto com base na comunicação direta a respeito de sua pose em relação a objeto. Controlados por uma arquitetura com três comportamentos, abordagem, rodeamento e transporte. Durante o comportamento de transporte, cada robô ajusta sua velocidade dependendo das posições dos vizinhos e da trajetória desejada do objeto. Em [Dai et al. (2015)] é proposto um algoritmo para calcular os pontos apropriados para os robôs. Eles devem evitar colisões entre si, quando se aproximam dos pontos adequados para prender o objeto. Quando todos os robôs alcançarem os pontos adequados atingindo a formação de enjaulamento, os robôs transportam o objeto para a posição de destino. Em [Habibi et al. (2015)] apresentam quatro controladores de movimento distribuídos para permitir que um grupo de robôs manipuladores realizem o transporte coletivamente de um objeto grande. Enquanto, outros mapeiam o ambiente de

modo a guiar os robôs de transporte na direção do destino, evitando obstáculos. Em [Stroupe et al. (2005)] dois robôs heterogêneos se comunicam para sincronizarem a apreensão, levantamento e colocação dos objetos na construção de uma estrutura em um ambiente realista. Em [Tuci et al. (2005)] os robôs se conectam fisicamente, para lidar com as condições ambientais que os impedem de realizar sua tarefa individualmente. O primeiro conjunto de experimentos é um estudo sobre a utilidade de robôs de automontagem para lidar com cenários relativamente complexos, como o transporte cooperativo de objetos. O segundo conjunto de experimentos é para aumentar a adaptabilidade do sistema multi-robô. Nele mostram que é possível sintetizar, por meio de técnicas de computação evolutiva, redes neurais artificiais que integram tanto os mecanismos de coordenação sensório-motora quanto de tomada de decisão exigidos pelos robôs no contexto de automontagem. Em [Chen et al. (2015)] os robôs apenas empurram o objeto em posições onde a linha de visão direta para o destino é obstruída pelo objeto. Este comportamento simples resulta no transporte do objeto em direção ao destino sem usar nenhuma forma de comunicação direta. Em [Wang and de Silva (2006)], os robôs transportam cooperativamente uma caixa, removendo obstáculos que obstruem o caminho para o destino final. Um agente supervisor tem uma visão global do ambiente para gerar posições e coordenadas de orientação de todos os robôs, o objeto e os obstáculos. Um agente de aprendizagem é responsável por gerar planos de cooperação com base em uma abordagem de otimização que integra aprendizagem por reforço e um algoritmo genético que permite a integração ao agente de tomada de decisão. Dois robôs físicos que executam o plano gerado pelos agentes de aprendizagem. O plano pode exigir que um robô deixe o transporte para remover obstáculos que obstruem o caminho até o destino final do transporte. Em [Fujisawa et al. (2013)] o transporte cooperativo é baseado no comportamento das formigas, na qual usam uma comunicação indireta via feromônio artificial. A tarefa requer que os robôs realizem uma busca aleatória para encontrar um alimento, neste caso, um objeto pesado e transportá-lo para um local de destino, ou seja, o ninho. Em [Eoh and Park (2021)], é apresentada uma técnica usando aprendizado por reforço profundo baseado. Propõem dois métodos para o aprendizado do transporte de objetos: crescimento de região e único robô para multi-robô. Durante o processo de aprendizagem, o método de crescimento da região gradualmente se estende a uma região na qual um objeto foi inicializado. Esse aprendizado visa aumentar a probabilidade de sucesso do transporte de objetos. Vários robôs podem aprender facilmente uma nova política, explorando a política pré-treinada de um único robô. Este método de um a vários robôs pode ajudar os robôs a aprender um método de transporte com tentativa e erro.

3. TRANSPORTE COOPERATIVO DE OBJETO

Transporte Cooperativo de Objeto (TCO) é o processo que gerencia e organiza um enxame de robôs para executarem corretamente um conjunto de etapas, visando um objetivo único. Esse processo consiste em procurar o objeto, recrutar os demais robôs, posicionar os robôs e transportar o objeto, de forma organizada e distribuída. A etapa de transporte deve ser dinâmica, pois precisa

ser continuamente ajustada em resposta às mudanças no ângulo do objeto a ser transportado para o próximo ponto de referência.

A fim de fornecer uma definição formal do problema TCO, seja $S = \{0, 1, \dots, \omega - 1\}$ o conjunto de ω pontos de referência e $R = \{0, 1, \dots, \rho - 1\}$ o conjunto de ρ robôs. As duas variáveis são independentes entre si. A Figura 1 ilustra a pose do robô i determinada por $p_i = [x_i, y_i, \beta]^T$, onde x_i e y_i denotam a posição do baricentro do robô e β a orientação do mesmo.

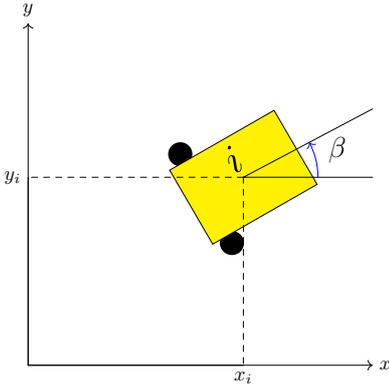


Figura 1. Posicionamento do robô i

Assumimos que o cálculo entre os pontos de referência se dê através da distância euclidiana. Logo, temos uma matriz da distância de $\omega \times \omega$. A adjacência dos pontos de referência consiste em uma matriz do mesmo tamanho da matriz da distância entre os pontos de referência, definida como A .

A primeira etapa do processo consiste na busca do objeto circular. Existem algumas formas distintas de planejamento de busca pelo objeto, iniciando perto do objeto [Alkilabi et al. (2018)], ou utilizando busca aleatória [Chen et al. (2015)]. No nosso trabalho, a busca é realizada de forma aleatória. Todos os robôs começam ao redor do ponto de referência w_0 , para a qual terão que levar o objeto. O robô i escolhe um ponto de referência pertencente à sua adjacência. Essa etapa é realizada até um robô alcançar o ponto de referência w^* . Note que w^* é o identificador do ponto de referência onde o objeto se encontra. Cada robô i tem a sua rota percorrida C_i , formada pelas pontos de referência que o robô i atravessou até o instante que este encontre o objeto. A menor rota usada pelos robôs que encontraram o objeto primeiro é denotada C^* .

A próxima etapa consiste no recrutamento dos robôs restantes até o objeto. Existem algumas maneiras do robô recrutar outros robôs para transportarem o objeto, por exemplo, utilizando uma comunicação indireta, através de feromônio [Fujisawa et al. (2013)]. Nesse trabalho, utilizamos comunicação direta entre os robôs para o recrutamento. Eles têm como referência principal a C^* . Deste modo, os robôs restantes baseiam-se nela para obter a nova rota e chegar no ponto de referência w^* .

A penúltima e a última etapas consistem no posicionamento inicial e no transporte do objeto. Existem propostas diferentes para os robôs se posicionarem e executarem o transporte, por exemplo, utilizando a oclusão [Chen et al.

(2015)]. Em outro exemplo, os robôs se posicionam um atrás do outro, em linha reta com o destino do transporte [Fujisawa et al. (2013)]. No trabalho que desenvolvemos, os robôs se posicionam atrás do objeto e o transportam percorrendo a rota C^* , neste caso, o caminho inverso dela.

4. ALGORITMO DE TRANSPORTE COOPERATIVO DE OBJETO

A implementação do TCO está estruturada em cinco etapas principais, conforme mostrado no Algoritmo 1: *inicializar arena, procurar objeto, recrutar robôs, posicionar robôs e transportar objeto*. Note que W é o conjunto posições dos pontos de referência, que P é o conjunto da pose dos robôs e que α é o ângulo entre a linha horizontal, que passa no centro do objeto, e a linha horizontal, que passa no centro do penúltimo ponto de referência da rota C^* . A escolha do ponto de referência w^* pode ser realizada de forma randômica, exceto pelo ponto de referência inicial w_0 . Inicialmente, os robôs não possuem conhecimento sobre a localização do ponto de referência w^* .

Algoritmo 1 TCO

requer ρ, ω, A ;
 1: Inicializar Arena(ρ, ω);
 2: Procura do objeto pelo robô $i(W, P)$;
 3: Recrutar robô $i(p_i, C_i, C^*)$;
 4: Posicionar Robôs(w^*, C^*, P);
 5: Transportar Objeto(w^*, w_0, α, P, C^*);

Na etapa de procura do objeto, a cada iteração é realizada a procura do ponto de referência w^* . Nesta etapa, é verificado se o ponto de referência desejado w^* é alcançado, através da diferença entre a pose do robô i (p_i) e o sensor de distância do robô i , definido como $sensor_i$, sendo que este sensor é emulado. Em caso negativo, é realizado o caminho até o último ponto de referência escolhido w_u . A próxima etapa a ser executada é o recrutamento, que acontece após um dos robôs ter alcançado w^* e o restante dos robôs atingir seu respectivo ponto de referência. Após o recrutamento ter sido alcançado, os robôs se posicionam para iniciarem a etapa de transporte. Por fim, os robôs transportam o objeto até seu destino final.

4.1 Etapa de Inicialização da Arena

A etapa de inicialização da arena é implementada conforme mostrado no Algoritmo 2. Nesta etapa, são determinadas as posições dos pontos de referência e as poses iniciais dos robôs.

Algoritmo 2 Inicializar Arena

requer ρ, ω ;
garante W, P ;
 1: **para** $k := 0 \rightarrow \omega - 1$ **faça**
 2: **Gera** w_k na arena;
 3: **fim para**
 4: **para** $i := 0 \rightarrow \rho - 1$ **faça**
 5: **Gera** randomicamente p_i ao redor do w_0 ;
 6: **fim para**

4.2 Etapa de Procura do Objeto

A etapa de procura do objeto é implementada conforme mostrado no Algoritmo 3. Nesta etapa, o robô i escolhe aleatoriamente um ponto de referência r , pertencente a adjacência do ponto de referência atual w_i , onde este se encontra e que não tenha sido escolhido anteriormente, essa adjacência é denominada como $A_i^{w_i}$. Dessa forma, o ponto de referência r é atribuído a última posição u_i da sua rota C_i .

O robô que encontra o objeto primeiro, tem sua rota escolhida como a menor rota C^* . A rota C^* é uma das entradas na etapa de recrutamento. Os robôs restantes, cada um se encaminha para o último ponto de referência escolhido aleatoriamente r .

Algoritmo 3 Procura do objeto pelo robô i

```

requer  $W, P$ ;
garante  $C_i, C^*, p_i, w^*$ ;
1:  $var_i \leftarrow false$ ;
2:  $obj\_enc \leftarrow false$ ;
3:  $mc \leftarrow \infty$ ; { $mc$  é o comprimento da menor rota}
4:  $u_i \leftarrow 0$ ;
5: enquanto  $\neg obj\_enc$  faça
6:   Gera aleatoriamente  $r \in A_i^{w_i}$ ;
7:    $C_i[u_i] \leftarrow r$ ;
8:    $u_i \leftarrow u_i + 1$ ;
9:   enquanto  $sensor_i > 0.5 \wedge p_i \neq r$  faça
10:    robô  $i$  avança na direção de  $r$ ;
11:   fim enquanto;
12:   se  $sensor_i \leq 0.5$  então
13:      $w^* \leftarrow r$ ;
14:      $oe_i \leftarrow true$ ; {variável objeto encontrado}
15:   fim se
16:   para  $i := 0 \rightarrow \rho - 1$  faça
17:     se  $oe_i = true$  então
18:        $obj\_enc \leftarrow true$ ;
19:        $z \leftarrow fc(C_i)$ ; { $fc$  calcula o comprimento da  $C_i$ }
20:       se  $z < mc$  então
21:          $mc \leftarrow z$ ;
22:          $C^* \leftarrow C_i$ ;
23:       fim se
24:     fim se
25:   fim para
26: fim enquanto;

```

4.3 Etapa de Recrutamento

O Algoritmo 4 descreve o método proposto para implementar o recrutamento dos robôs do exame para chegarem no objeto. Existem duas opções: os robôs irem direto ao ponto de referência w^* ou percorrerem o sub-caminho na rota C^* que leva o robô da sua posição corrente até w^* . Neste trabalho, a segunda opção é explorada. Esta etapa é responsável por levar os robôs, que não encontraram o objeto, até o mesmo, pela rota C^* e, caso o robô i já se encontre perto do objeto, que ele permaneça lá. Note que q é uma variável para o último ponto de referência da rota C^* e que g é uma variável para um dado ponto de referência.

Inicialmente, é realizada a comparação entre o último ponto de referência da rota C_i e a rota C^* . Nesta com-

paração, três situações são possíveis, sendo que na última, dois casos são possíveis:

- (1) $C_i[u_i] = w^*$, indica que o último ponto de referência da sua rota C_i é o ponto de referência onde se encontra o objeto w^* . Logo, o robô i permanece em sua pose p_i .
- (2) $(C_i[u_i] \in C^*) \wedge (C_i[u_i] \neq w^*)$, indica que o último ponto de referência da sua rota C_i pertence à rota C^* , porém este ponto de referência não é o ponto de referência w^* . Logo, o robô i parte deste ponto de referência pertencente à rota C^* até concluir toda a rota C^* .
- (3) $C_i[u_i] \notin C^*$, indica que o último ponto de referência da sua rota C_i não pertence à rota C^* , havendo aqui dois casos. No primeiro caso, se existir um ponto de referência na adjacência deste último ponto de referência da sua rota C_i , pertencente à rota C^* , o robô i se desloca para este ponto e, depois o robô i segue o item anterior. A adjacência é denominada como $A_i^{C_i[u_i]}$. No segundo caso, se não houver um ponto de referência na adjacência $A_i^{C_i[u_i]}$, pertencente à rota C^* , o robô i escolhe aleatoriamente um ponto de referência pertencente à adjacência $A_i^{C_i[u_i]}$, havendo aqui dois casos. No primeiro caso, o robô i pode alcançar um ponto de referência que na adjacência deste ponto de referência exista um ponto de referência, pertencente à rota C^* . No segundo caso, o robô faz um caminho alternativo até o objeto, pois na adjacência dos pontos de referência que foi passando, não havia um ponto de referência pertencente a rota C^* .

Algoritmo 4 Recrutar robô i

```

requer  $p_i, C_i, C^*$ ;
garante  $p_i$ 
1:  $rota\_enc = false$ ;
2: se  $C_i[u_i] \in C^*$  então
3:   Seja  $q | C^*[q] = C_i[u_i]$ ;
4:    $rota\_enc \leftarrow true$ ;
5: fim se
6: enquanto  $sensor_i > 0.5$  faça
7:   se  $rota\_enc = true$  então
8:      $g \leftarrow C^*[q + +]$ ;
9:   senão
10:    se  $\exists q \in A_i^{C_i[u_i]} | q \in C^*$  então
11:       $g \leftarrow q$ ;
12:     $rota\_enc \leftarrow true$ ;
13:    senão
14:      Gera aleatoriamente  $g \in A_i^{C_i[u_i]}$ ;
15:    fim se
16:    fim se
17:    enquanto  $p_i \neq g$  faça
18:      robô  $i$  avança na direção de  $g$ ;
19:    fim enquanto
20: fim enquanto

```

A ação de identificar o último ponto de referência da rota C_i , em relação à rota C^* , baseia-se no compromisso de reduzir o percurso até o ponto de referência w^* . Isto vem em decorrência do mesmo percurso ser usado de forma inversa para alcançar o ponto de referência w_0 na etapa de transporte.

4.4 Etapa de Posicionamento Inicial

O Algoritmo 5 descreve o método para os robôs se posicionarem para iniciarem o transporte. Os robôs se posicionam atrás do objeto em direção ao penúltimo ponto de referência da rota C^* . Note que u^* é o último ponto de referência da rota C^* .

As poses são calculadas no Algoritmo 6, considerando o ângulo α , o diâmetro do robô d , o diâmetro do objeto D e a distribuição dos robôs atrás do objeto. A distribuição depende do número de robôs e da diferença angular entre cada robô, representado pelo ângulo θ . Cada robô possui uma linha que passa no centro dele, no sentido da sua orientação, que passa o centro do objeto. O ângulo θ é formado quando essas linhas se cruzam no centro do objeto. A Figura 2(a) ilustra o caso onde o número de robôs é par, o robô com número par fica deslocado à direita do ângulo α . Entretanto, se o número do robô for ímpar, o robô fica deslocado à esquerda do ângulo α . A Figura 2(b) ilustra o caso onde o número de robôs é ímpar, apenas o robô 0 se posiciona na mesma direção do ângulo α e os demais robôs se posicionam da mesma forma que o caso anterior. O ângulo β é aquele em o robô i se posiciona inicialmente em relação à linha horizontal do objeto para iniciar o transporte.

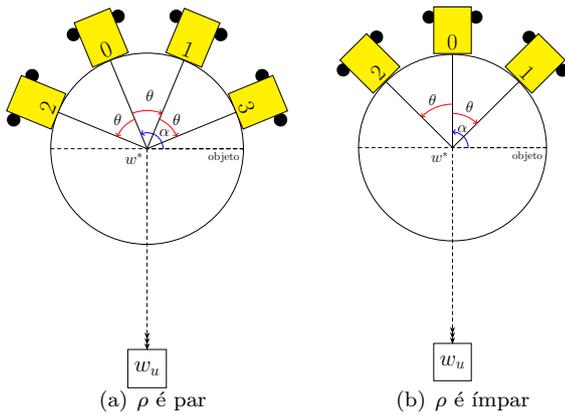


Figura 2. Ilustração do posicionamento inicial dos robôs

Algoritmo 5 Posicionar robôs

```

requer  $w^*, C^*, P$ ;
garante  $P$ ;
1: para paralelo  $i := 0 \rightarrow \rho - 1$  faça
2:    $p_i^+ \leftarrow$  Pose inicial do robô  $i(w^*, p_i, C^*, u^*)$ ;
3:   enquanto  $p_i \neq p_i^+$  faça
4:     robô  $i$  avança na direção de  $p_i^+$ ;
5:   fim enquanto
6:    $P \leftarrow P \cup p_i^+$ 
7: fim para paralelo

```

4.5 Etapa de Transporte

O Algoritmo 7 descreve a execução do transporte do objeto por todos os robôs. Este algoritmo é executado até os robôs percorrerem toda a rota C^* no sentido contrário.

A atualização da pose dos robôs, após cada empurrão, é executada pelo Algoritmo 8. A distribuição dos robôs, atrás do objeto, é explicada na etapa de posicionamento

Algoritmo 6 Pose inicial do robô i

```

requer  $w^*, p_i, C^*, u^*$ ;
garante  $p_i^+, \alpha$ ;
1:  $u \leftarrow u^* - 1$ ;
2:  $w_u \leftarrow C^*[u]$ ;
3:  $\alpha := \arctan((w_{uy} - w_y^*) / (w_{ux} - w_x^*))$ 
4: se  $\rho \bmod 2 = 0$  então
5:   se  $i \bmod 2 = 0$  então
6:      $\beta \leftarrow \alpha + \theta \times (((i \text{ div } 2) + 1) - 0, 5)$ ;
7:   senão
8:      $\beta \leftarrow \alpha + \theta \times ((-(i + 1) \text{ div } 2) + 0, 5)$ ;
9:   fim se
10: senão
11:   se  $i \bmod 2 = 0$  então
12:      $\beta \leftarrow \alpha + \theta \times (i \text{ div } 2)$ ;
13:   senão
14:      $\beta \leftarrow \alpha - \theta \times ((i + 1) \text{ div } 2)$ ;
15:   fim se
16: fim se
17:  $x_i := w_x^* + ((d + D) / 2) \times \cos(\beta + \pi)$ 
18:  $y_i := w_y^* + ((d + D) / 2) \times \sin(\beta + \pi)$ 
19:  $p_i^+ \leftarrow [x_i, y_i, \beta]^T$ 

```

inicial. Após cada empurrão, é avaliado um novo ângulo α^+ . Seu cálculo é igual ao utilizado para o do ângulo α na etapa de posicionamento inicial. Caso haja um desvio maior do que o erro angular $\Delta\theta_0$, calcula-se novamente a pose dos robôs com o ângulo α^+ . Caso contrário, utiliza-se o ângulo α . A Figura 3(a) ilustra um posicionamento errado dos robôs, após um empurrão. O objeto se desviou da trajetória esperada. Já a Figura 3(b) ilustra o reposicionamento correto dos robôs. Os robôs se reposicionam utilizando o ângulo α^+ para transportar corretamente o objeto até o ponto de referência w_u .

Após o empurrão, no Algoritmo 9, a posição do alvo é atualizada.

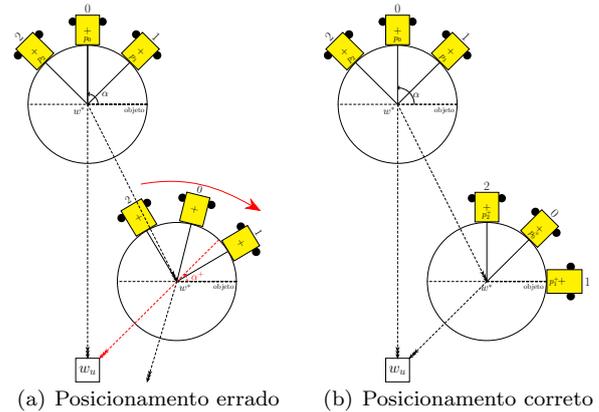


Figura 3. Ilustração do reposicionamento

5. ASPECTOS DE IMPLEMENTAÇÃO

O algoritmo proposto foi implementado em um enxame de robôs do tipo GRITSBot. Os robôs atualmente não estão equipados com placas de sensores, pois a detecção de distância pode ser emulada através do servidor de *back-end* que rastreia as posições de todos os robôs. A pose global de todos os robôs é recuperada através de um sistema

Algoritmo 7 Transportar objeto

```

requer  $w^*, w_0, \alpha, P, C^*$ ;
1: repita
2:   para paralelo  $i := 0 \rightarrow \rho - 1$  faça
3:      $p_i^+ \leftarrow$  Nova pose do robô  $i(w^*, \alpha, p_i, C^*, u^*)$ ;
4:     enquanto  $p_i \neq p_i^+$  faça
5:       robô  $i$  avança na direção de  $p_i^+$  empurrando o
           alvo;
6:     fim enquanto
7:   fim para paralelo
8:   Atualizar posição do alvo  $w^*$ ;
9: até  $w_u = w_0$ 
    
```

Algoritmo 8 Nova pose do robô i

```

requer  $w^*, \alpha, p_i, C^*, u^*$ ;
garante  $\alpha, p_i^+$ 
1:  $u^* \leftarrow u^* - 1$ ;
2:  $w_u \leftarrow C^*[u^*]$ ;
3:  $\alpha^+ := \arctan((w_{uy} - w_y^*) / (w_{ux} - w_x^*))$ 
4:  $\Delta\theta_1 := |\alpha^+ - \alpha|$ 
5: se  $\Delta\theta_1 > \Delta\theta_0$  então
6:    $\alpha \leftarrow \alpha^+$ 
7: fim se
8: se  $\rho \bmod 2 = 0$  então
9:   se  $i \bmod 2 = 0$  então
10:     $\beta \leftarrow \alpha + \theta \times (((i \bmod 2) + 1) - 0, 5)$ ;
11:   senão
12:     $\beta \leftarrow \alpha + \theta \times ((-i + 1) \bmod 2 + 0, 5)$ ;
13:   fim se
14: senão
15:   se  $i \bmod 2 = 0$  então
16:     $\beta \leftarrow \alpha + \theta \times (i \bmod 2)$ ;
17:   senão
18:     $\beta \leftarrow \alpha - \theta \times ((i + 1) \bmod 2)$ ;
19:   fim se
20: fim se
21:  $x_i := w_x^* + ((d + D)/2) \times \cos(\beta + \pi)$ 
22:  $y_i := w_y^* + ((d + D)/2) \times \sin(\beta + \pi)$ 
23:  $p_i^+ \leftarrow [x_i, y_i, \beta]^T$ 
    
```

Algoritmo 9 Atualizar posição do alvo

```

requer  $w^*$ ;
garante  $w^*$ ;
1:  $w_x^* := w_x^* + v_x \times t + a_x \times \frac{\Delta t^2}{2}$ 
2:  $w_y^* := w_y^* + v_y \times t + a_y \times \frac{\Delta t^2}{2}$ 
3:  $w^* \leftarrow [w_x^*, w_y^*]^T$  {nova posição do alvo}
    
```

de rastreamento aéreo e utiliza uma única *webcam* em conjunto com as *tags* ArUco para rastreamento [Pickem et al. (2017)]. As simulações são feitas no Robotarium, que é um centro de pesquisa multi-robô acessível remotamente [Pickem et al. (2017).] O Robotarium usa Certificados de Barreira de Segurança para garantir um comportamento livre de colisões de todos os robôs. Uma comunicação direta é emulada apenas quando o primeiro robô encontra o de referência w^* . Assim, os outros robôs têm conhecimento da rota C^* . Para a fase de transporte, a comunicação é indireta, usando-se o objeto para comunicação, semelhante à utilizada em [Wang et al. (2004)]. Os Certificados de Barreira de Segurança são usados em todas as etapas.

Entretanto, não são utilizados com o objeto, pois os robôs têm que empurrar o objeto.

Na Tabela 1, estão as constantes utilizadas no TCO e são próprias do Robotarium. O passo de tempo t é o inverso da frequência de atualização do Robotarium (3 Hz). A velocidade v representa a velocidade máxima que pode ser atingida pelo robô. A aceleração a denota a variação de velocidade do robô, indo progressivamente de 0 a v .

Tabela 1. Constantes do Robotarium

	Símbolo	Valor
Diâmetro do robô	d	0,11 m
Massa do robô	pr	0,06 kg
Passo de tempo	t	0,333 s
Velocidade do robô	v	0,2 m/s
Aceleração do robô	a	0,6 m/s ²

6. RESULTADOS DE DESEMPENHO

A análise dos resultados permite avaliar a eficácia do TCO, bem como sua eficiência, em atingir o objetivo principal de transportar o objeto, através do cumprimento de todas as etapas correspondentes. Após um robô encontrar o objeto, ele deve recrutar os demais robôs. Em seguida, se posicionam para iniciarem o transporte e então executam o transporte até seu destino final.

Os parâmetros dos experimentos estão apresentados na Tabela 2. O número de robôs usados é 3. O número de pontos de referência é 27, sendo o ponto 26 o alvo e o ponto 0 onde se encontram os robôs no começo dos experimentos. Já no segundo experimento, o objeto se encontra no ponto 20. Note que a massa m do objeto é 16 vezes maior do que a do robô e seu diâmetro D é 5 vezes maior do que o do robô. A diferença angular que um robô deve ter para o outro é de 45° na etapa de posicionamento inicial. Durante a etapa de transporte, esta diferença angular é mantida na medida do possível. O erro angular $\Delta\theta_0$ representa o desvio angular da frente do robô com relação à linha reta formada pelo centro do objeto transportado e o próximo ponto de referência, *i.e.* aquele para o qual o objeto deve ser transportado. O erro angular tolerado é 0,03. O valor do erro angular é definido a partir de simulações. Para valores maiores do que $\Delta\theta_0$, o centro do objeto desviaria muito da linha reta que o liga ao próximo ponto de referência, enquanto valores menores ocasionariam um reposicionamento constante do objeto.

Tabela 2. Constantes dos experimentos

	Símbolo	Valor
Número de robôs	ρ	3
Número de pontos de referência	ω	27
Massa do objeto	m	1 kg
Diâmetro do objeto	D	0,6 m
Diferença angular	θ	45°
Erro angular	$\Delta\theta_0$	0,03 m

Nos dois experimentos, os robôs foram inicializados com alocações iniciais aleatórias ao redor do ponto de referência 0. A primeira etapa consiste em procurar o objeto. A

Figura 4 mostra a adjacência de cada ponto de referência. Cada robô só pode ir para o próximo ponto de referência, conforme explicado na etapa de procura do objeto. A Figura 5(a) mostra o início desta primeira etapa. Já a Figura 5(b) retrata a procura do objeto. A Figura 5(c) retrata o instante em que o robô 1 encontra o objeto e os outros robôs seguem para seus pontos de referência escolhidos ultimamente. Por fim, a Figura 5(d) mostra quando estes robôs chegaram em seus pontos, sendo o fim da etapa de procura. As imagens são do primeiro experimento.

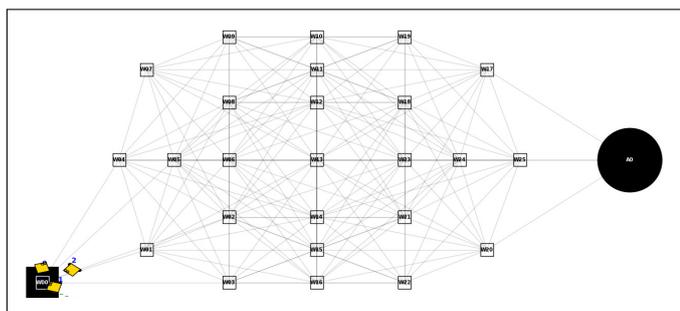


Figura 4. Configuração da arena: ponto de partida dos robôs, pontos de referência intermediários e ponto onde se encontra o objeto

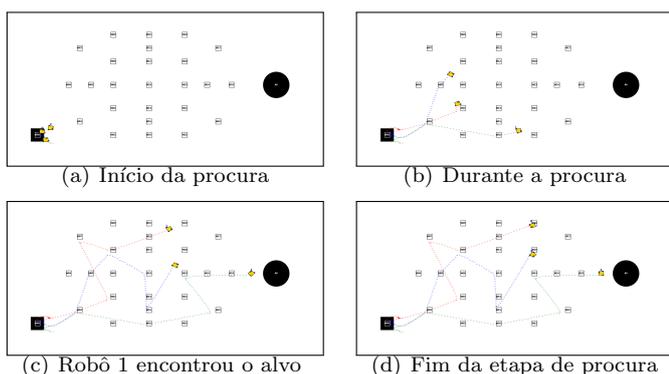


Figura 5. Etapa de procura do 1° experimento

A próxima etapa consiste no recrutamento dos robôs que não encontraram o objeto. A Figura 6(a) mostra o início do recrutamento. A Figura 6(b) ilustra a etapa do recrutamento. O final do recrutamento do primeiro experimento é mostrado na Figura 6(c). A etapa seguinte consiste no posicionamento inicial para começarem a transportar o objeto, conforme a Figura 6(d), do primeiro experimento.

A última etapa consiste no transporte do objeto pela rota C^* , fazendo ela inversamente. A Figura 7(a) ilustra o início do transporte. Já a Figura 7(b) ilustra a metade do percurso percorrido, durante o transporte do objeto no primeiro experimento. A Figura 7(c) ilustra o objeto chegando no penúltimo ponto de referência da menor rota. Por fim, a Figura 7(d) ilustra o término deste transporte. Em comparação, a Figura 8(a) ilustra o final da procura no segundo experimento. Já a Figura 8(b) ilustra o término do recrutamento. A Figura 8(c) ilustra o posicionamento inicial do transporte. Por último, a Figura 8(d) retrata o término do transporte no segundo experimento.

A duração das etapas, executadas por cada robô no primeiro experimento, são apresentadas na Tabela 3. A Ta-

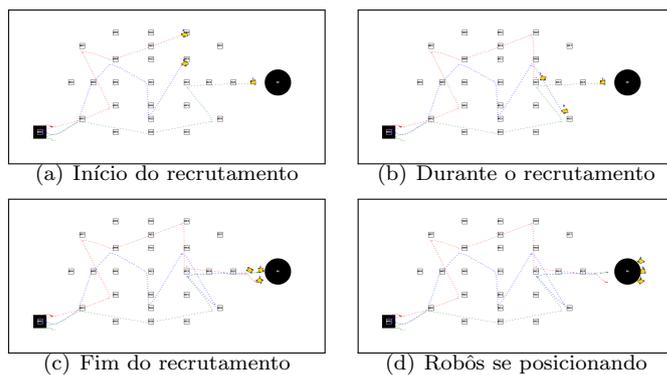


Figura 6. Etapa de recrutamento e posicionamento inicial do 1° experimento

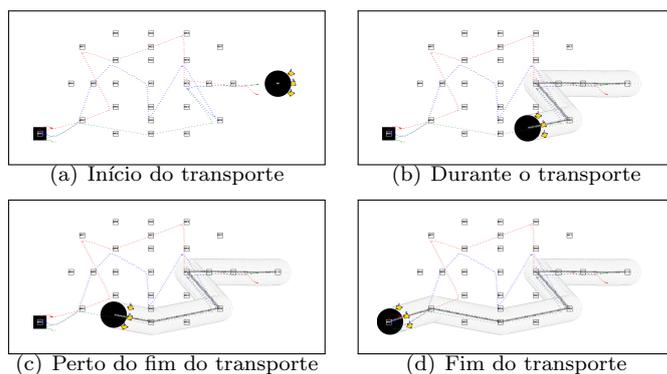


Figura 7. Etapa do transporte do objeto do 1° experimento

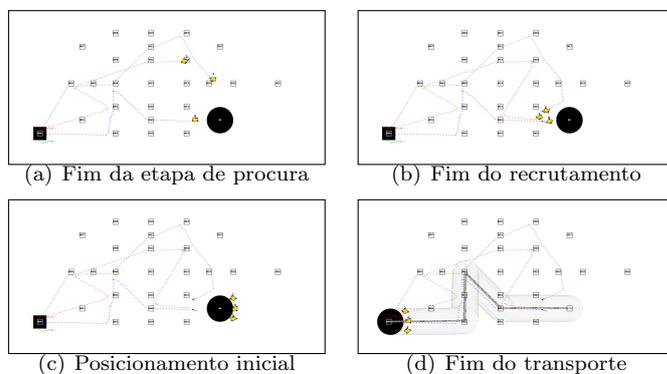


Figura 8. Etapas do 2° experimento

bela 4 mostra a duração das etapas para o segundo experimento.

O tempo da etapa de procura é o tempo que um robô demora para encontrar o objeto e, para os outros robôs, é o tempo que cada robô demorou para alcançar o último ponto de referência escolhido aleatoriamente. Já o tempo de espera é a diferença entre o tempo do robô que chegou no seu último ponto de referência escolhido aleatoriamente e o tempo que aquele robô que concluiu a etapa de procura. O tempo para chegar no objeto é o tempo da etapa de recrutamento. O tempo para se posicionarem é o tempo utilizado por cada robô na etapa de posicionamento inicial. Por fim, o tempo para transportar o objeto é o tempo para os robôs transportarem o objeto pela rota C^* inversamente.

Na etapa de procura, em relação ao primeiro experimento, o robô 2 encontra o objeto primeiro, conforme o tempo da etapa de procura, mostrado na Tabela 3. Já para o segundo experimento, o robô 1 é o que encontra o objeto primeiro, conforme o tempo da etapa de procura na Tabela 4.

No tempo de espera, para o primeiro experimento, o robô 2 é o que espera por mais tempo, pois foi ele que encontrou o objeto, o robô 1 foi o que concluiu a etapa de procura por último, conforme mostra a Tabela 3. Já no segundo experimento, o robô 2 foi o que concluiu a etapa de procura por último e o robô 1 o que esperou por mais tempo, conforme mostra a Tabela 4.

No primeiro experimento, o robô 1 demorou mais tempo para chegar ao objeto, conforme mostrado na Tabela 3. Para o robô 2, como ele encontrou o objeto primeiro, o tempo utilizado para chegar ao objeto é o tempo que foi utilizado na etapa de procura. Já no segundo experimento, o robô 3 demorou mais tempo para chegar ao objeto, conforme mostrado na Tabela 4. Para o robô 1, como ele encontrou o objeto primeiro, o tempo utilizado para chegar ao objeto é o tempo que foi utilizado na etapa de procura.

O tempo que cada robô utilizou para se posicionar está demonstrado nas Tabelas 3 e 4. O que distinguiu cada robô foi a pose que ele parou na etapa de recrutamento. O tempo para os robôs transportarem o objeto até o alvo, conforme mostrado nas Tabelas 3 e 4, foi menor no segundo experimento. Isto ocorre, em parte, pela localização do objeto ser diferente e, em parte, porque, para cada experimento, a rota C^* , encontrada para os robôs transportarem o objeto, foi distinta.

Tabela 3. Resultados do experimento 1

Robôs	Experimento 1		
	1	2	3
Tempo da etapa de procura	128.85 s	71.76 s	111.1 s
Tempo de espera	0 s	57.09 s	17.75 s
Tempo para chegar ao objeto	244.54 s	71.76 s	203.62 s
Tempo para se posicionarem	1.52 s	1.55 s	1.56 s
Tempo para transportar o objeto	4135.18 s	4135.18 s	4135.18 s

Tabela 4. Resultados do experimento 2

Robôs	Experimento 2		
	1	2	3
Tempo da etapa de procura	53.77 s	117.16 s	103.15 s
Tempo de espera	63.39 s	0 s	14.01 s
Tempo para chegar ao objeto	53.77 s	150.35 s	187.50 s
Tempo para se posicionarem	1.25 s	1.28 s	1.30 s
Tempo para transportar o objeto	2796.70	2796.70	2796.70

7. CONCLUSÃO

Neste artigo, foi proposto um algoritmo para transportar um objeto circular até o seu destino final, utilizando robôs. Resultados experimentais comprovam a eficácia do algoritmo em transportar o objeto pela menor rota encontrada pelos robôs. Entretanto, o método utilizado, para os robôs escolherem os pontos de referência na etapa de procura, não proporcionou uma rota eficiente do objeto até o alvo, conforme mostraram os experimentos. Em um futuro próximo, pretendemos utilizar outro método na etapa de procura, o que ajudaria na comparação do melhor método para obter uma menor rota até o objeto, assim como a inclusão de obstáculos na arena.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e da Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) - (Proc. E-26/210.200/2020). Os autores agradecem o apoio financeiro.

REFERÊNCIAS

- Alkilabi, M.H.M., Narayan, A., Lu, C., and Tuci, E. (2018). Evolving group transport strategies for epuck robots: moving objects towards a target area. In *Distributed Autonomous Robotic Systems*, volume 6 of *Springer Proceedings in Advanced Robotics*, 503–516.
- Chen, J., Gauci, M., Li, W., Kolling, A., and Groß, R. (2015). Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, 31, 307–321.
- Dai, Y., Kim, Y.G., Lee, D.H., and Lee, S. (2015). Symmetric caging formation for convex polygon object transportation by multiple mobile robots. In *Proceedings of the 2015 IEEE International Conference on Advanced Intelligent Mechatronics*, 595–600. IEEE, Busan, South Korea.
- Eoh, G. and Park, T.H. (2021). Cooperative object transportation using curriculum-based deep reinforcement learning. *Sensors*, 21(14).
- Fink, J., Michael, N., and Kumar, V. (2007). Composition of vector fields for multi-robot manipulation via caging. In *Conference Robotics: Science and Systems*, volume 3. Atlanta, GA, USA.
- Fujisawa, R., Imamura, H., and Matsuno, F. (2013). *Cooperative transportation by swarm robots using pheromone communication*, 559–570. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Habibi, G., Kingston, Z., Xie, W., Jellins, M., and McLurkin, J. (2015). Distributed centroid estimation and motion controllers for collective transport by multi-robot systems. In *IEEE International Conference on Robotics and Automation*, 1282–1288. IEEE, Seattle, WA, USA.
- Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M. (2017). The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1699–1706.
- Stroupe, A., Huntsberger, T., Okon, A., Aghazarian, H., and Robinson, M. (2005). Behavior-based multi-robot collaboration for autonomous construction tasks. In *2005 RSJ Int. Conference on Intelligent Robots and Systems*, 1495–1500. IEEE, Edmonton, Canada.
- Tuci, E., Gross, R., Trianni, V., Mondada, F., Bonani, M., and Dorigo, M. (2005). Cooperation through self-assembly in multi-robot. *TAAAS*, 1, 115–150.
- Wang, Y. and de Silva, C.W. (2006). Cooperative transportation by multiple robots with machine learning. In *2006 IEEE International Conference on Evolutionary Computation*, 3050–3056. IEEE, Vancouver, BC, CA.
- Wang, Z., Takano, Y., Hirata, Y., and Kosuge, K. (2004). A pushing leader based decentralized control method for cooperative object transportation. In *Proceedings of the 2004 RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1035–1040. IEEE.