

Evitando o congestionamento de grupos de robôs autônomos com alvos em comum

Gustavo P. Mitsuchi Oliveira¹, Douglas Macharet¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

{gustavo.oliveira,doug}@dcc.ufmg.br

Abstract. *The congestion problem within the mobile robotics field is a well known subject by the researchers. The problem occurs when a large number of autonomous robots tries to reach a target in the space at the same time. This paper addresses the congestion problem with subdivisions of small groups inside the large group. It is proposed two different algorithms to optimize some metrics related to this version of the congestion problem. The first one gives to each subgroup a priority number based on the average distance between the robots and the target. This priority will define which subgroup will move into the target first. The second method defines which subgroup will move based on a random variable.*

Resumo. *O problema do congestionamento dentro do campo da robótica móvel é um assunto bem conhecido pelos pesquisadores da área. O problema ocorre quando um grande número de robôs autônomos tenta alcançar um alvo no espaço ao mesmo tempo. Este artigo aborda uma versão de problema de congestionamento que possui subdivisões de pequenos grupos dentro do grande grupo de robôs. Propõe-se dois algoritmos diferentes para otimizar métricas relacionadas a esta versão do problema de congestionamento. O primeiro define a ordem de movimentação de cada subgrupo pela distância média entre os robôs e o alvo. O segundo método define qual subgrupo se moverá com base em uma variável aleatória.*

1. Introdução

No mundo atual, especialmente no contexto da tecnologia e inovação, um assunto que está cada vez mais sendo discutido e ganhando espaço no cotidiano é a robótica. A popularização desse tópico se deve ao fato de que incontáveis problemas de diversas áreas diferentes podem ser solucionados de alguma forma com o auxílio de robôs agindo individualmente ou em cooperação com outros robôs.

Porém, com os avanços na área, crescem os problemas e impedimentos relacionados às novas tecnologias e estratégias de resolução de problemas associados à robótica. Nesse trabalho será abordado o problema de congestionamento em situações envolvendo múltiplos robôs organizados em sub conjuntos.

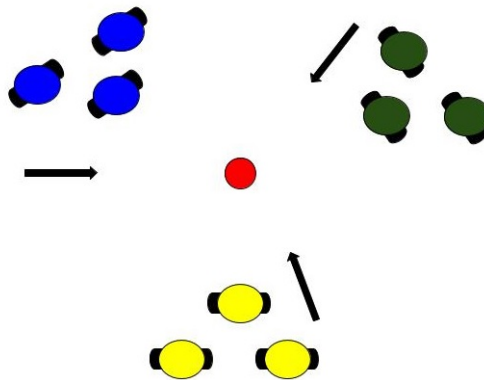


Figura 1. Robôs com alvo em comum agrupados

O problema consiste em evitar o congestionamento e a colisão de um conjunto de robôs autônomos que têm como objetivo chegar um ponto em comum. Dentro desse conjunto de robôs, os indivíduos estão agrupados em sub conjuntos que correspondem a times de robôs, podendo representar times de robôs do mesmo tipo ou com uma mesma função.

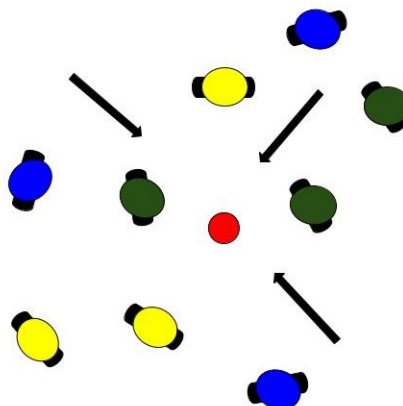


Figura 2. Robôs com alvo em comum desagrupados

As figuras 1 e 2 representam duas situações possíveis do problema do congestionamento. Em cada uma delas, o ponto vermelho representa o alvo em comum dos robôs e a cor de cada robô representa seu sub grupo. Na primeira, os robôs estão dispostos de modo que cada indivíduo do sub grupo fique próximo uns aos outros, já na segunda, não há nenhuma organização prévia.

É fácil visualizar o problema no mundo real, sendo possível citar como exemplo robôs tentando chegar a um ponto para realizar um carregamento de bateria, robôs mineradores autônomos tentando minerar uma mesma mina ou até mesmo quadro-rotorez que devem chegar a um determinado alvo para transportá-lo.

Esse trabalho tem como objetivo propor dois algoritmos que solucionam o problema de maneira descentralizada, ou seja cada indivíduo será responsável pela sua tomada de decisão, realizar uma série de experimentos utilizando cada um deles e apresentar de maneira comparativa os resultados. As métricas para avaliar o desempenho de cada um

deles será descrita em uma seção posterior.

2. Referencial teórico

O problema do congestionamento e tráfego de indivíduos já foi abordado por diversos autores, inclusive de outras áreas diferentes da robótica. Nessa seção serão citados trabalhos que abordam o problema específico da robótica.

Grossman (1988)[Grossman 1988], é um dos pioneiros a tratar do problema de tráfego de robôs, ele propõe uma solução centralizada, ou seja, uma unidade central é responsável pela tomada de decisão de cada indivíduo e cada robô age de maneira independente, não havendo sub-grupos, nem um alvo em comum para os robôs.

Em Hinoshi (2011)[Hoshino 2011], o problema do congestionamento é abordado de maneira simplificada, onde os robôs possuem um mesmo alvo em comum porém podem trafegar em apenas um determinado trajeto cada um, nesse projeto, além de estarem dispostos em sub grupos, os robôs poderão vir de qualquer direção e trafegar em todo o espaço.

Os algoritmos que serão construídos nesse projeto terão grande influência de Marcolino et al.(2009) [Marcolino and Chaimowicz 2009] e Marcolino et al.(2016)[Marcolino et al. 2016], em ambos trabalhos são propostos algoritmos descentralizados para o tratamento do problema e considerando que os indivíduos podem estar em qualquer posição do espaço de trabalho. Marcolino et al.(2009) [Marcolino and Chaimowicz 2009], apresenta um algoritmo probabilístico que trata o problema do congestionamento mas apresenta problemas no momento que os robôs estão deixando o alvo. No segundo, os autores propõem uma modificação nesse algoritmo a fim de resolver essa questão.

Vale ressaltar nessa seção que, não foi encontrado um trabalho prévio que trate do problema do congestionamento em grupos ou swarms de robôs subdivididos.

3. Objetivos

Os algoritmos foram desenvolvidos com o objetivo de otimizar dois aspectos específicos do problema que serão descritos a seguir. As métricas são atribuídas a cada um dos sub-grupos

1. **Tempo de Trânsito.** É o tempo que leva para que um indivíduo do subgrupo alcance a área objetivo, ou seja, o tempo gasto para que o grupo comece o processo de alimentação (entrada e saída da área objetivo). Durante esse período, é esperado que cada indivíduo do subgrupo realize movimentos que tendem a levá-lo a área objetivo.
2. **Tempo de alimentação** Essa métrica diz respeito ao tempo que um subgrupo leva para que todos os seus indivíduos alcancem a área objetivo. É esperado que durante esse período de tempo os indivíduos de um subgrupo entrem e saiam da área objetivo.
3. **Distância Percorrida** Essa métrica é a distância percorrida por todos os robôs, incluindo todos os grupos.

4. Algoritmo da Distância Média

O primeiro método desenvolvido foi o chamado Algoritmo da Distância Média. Nele, cada subgrupo de robôs possui uma prioridade P , que determina qual pode avançar em direção a área objetivo. A prioridade P de cada subgrupo é definida com base na média das distâncias entre cada robô j e a área objetivo, dentro de um grupo i com n robôs, como mostrado na equação 1. O subgrupo que possuir a menor distância média, possuirá a maior prioridade P e por consequência poderá se encaminhar em direção a área objetivo.

$$D_i = \frac{\sum_{j \in i} \text{dist}(j, \text{Alvo})}{n} \quad (1)$$

Os subgrupos com menor prioridade permanecem estáticos até que o último robô do subgrupo com maior prioridade entre e saia da área objetivo. Quando isso ocorre, o próximo subgrupo recebe permissão para se encaminhar a área objetivo.

Algorithm 1 Distância Média

```
for all  $i \in \text{grupos}$  do
   $D_i = \frac{\sum_{j \in i} \text{dist}(j, \text{Alvo})}{n}$ 
  if  $D_i \leq \text{min\_dist}$  then
     $\text{min\_dist} = D_i$ 
     $\text{grupo\_prioridade} = i$ 
while  $\text{existe\_grupo\_entrando}()$  do
  for all  $j \in \text{grupo\_prioridade}$  do
     $\text{move}(j)$ 
  if  $\text{grupo\_prioridade\_saindo}()$  then
    for all  $i \in \text{grupos\_restantes}$  do
       $D_i = \frac{\sum_{j \in i} \text{dist}(j, \text{Alvo})}{n}$ 
      if  $D_i \leq \text{min\_dist}$  then
         $\text{min\_dist} = D_i$ 
         $\text{grupo\_prioridade} = i$ 
```

Para a navegação, o método utiliza da técnica de campos potenciais, que será descrita em uma seção posterior, pois o outro algoritmo também faz uso desse tipo de navegação.

5. Algoritmo Probabilístico

O segundo método desenvolvido nesse trabalho foi livremente inspirado no trabalho de Marcolino et al.(2009) [Marcolino et al. 2016]. Nesse segundo algoritmo, a permissão de ir em direção a área alvo de cada grupo é dada por uma variável aleatória.

A cada n iterações, um valor aleatório é gerado e se o valor for menor que uma variável ρ , o grupo ganha a permissão de avançar, caso contrário, o grupo permanece estático.

No início da execução, todos os grupos estão no estado *esperando*, ao final de n iterações, cada um dos grupos tem chance ρ de transitar para o estado *entrando*. Quando

todos os robôs do grupo já entraram na área alvo e estão se movendo em direção oposta à mesma, o grupo está no estado *saindo*. Como no método anterior, a execução finaliza quando o último robô deixa a área objetivo.



Figura 3. Máquina de estados finitos do algoritmo probabilístico

Algorithm 2 Probabilístico

```

for all  $i \in \text{grupos}$  do
   $\text{random} = \text{random}()$ 
  if  $\text{random} \leq \rho$  then
     $\text{tem\_permissao}(i)$ 
  while  $\text{existe\_grupo\_entrando}()$  do
    for all  $j \in \text{grupo\_prioridade}$  do
       $\text{move}(j)$ 
       $\text{iteracoes}++$ 
    if  $\text{iteracoes} \% n == 0$  then
      for all  $i \in \text{grupos}$  do
         $\text{random} = \text{random}()$ 
        if  $\text{random} \leq \rho$  then
           $\text{tem\_permissao}(i)$ 
  
```

6. Prevenção de Colisões

A técnica escolhida para a navegação de cada robô em ambos os métodos desenvolvidos foi a modelagem por campos potenciais.

Nesse tipo de navegação, cada robô é modelado de modo a se comportar como cargas elétricas de mesmo sinal e a área objetivo como uma carga de sinal oposto. Isso faz com que exista um campo de força potencial ao redor de cada robô que "repele" outros indivíduos e o atrai para a área objetivo.

$$F_{att}(q) = -k_{att} \cdot (q - q_{objetivo}) \quad (2)$$

$$F_{rep}(q) = \begin{cases} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} & \text{se } \rho(q) \leq \rho_0 \\ 0 & \text{se } \rho(q) > \rho_0 \end{cases} \quad (3)$$

As equações 2 e 3 mostram como são calculadas as forças de atração e repulsão respectivamente. A incógnita q representa um robô e função $\rho(q)$ é a distância do robô para um obstáculo (outro robô).

De maneira bem simples, a força resultante em cada robô é calculada a cada iteração e sua nova posição é computada. A força resultante é dada pelo somatório das forças de repulsão mais a força de atração em direção a área objetivo.

A técnica é interessante para esse tipo de problema principalmente por funcionar em ambientes dinâmicos. Porém ainda podem ocorrer problemas de ótimo local quando as forças que atuam sobre um indivíduo resultam em um vetor nulo. A fim de tentar evitar esse problema, foi acrescentada uma variação aleatória na direção da força de repulsão em um robô. A direção que em princípio poderia somente ser totalmente oposta ao obstáculo, agora pode sofrer leves alterações, somando valores de $\frac{\pi}{2}$ até $\frac{\pi}{8}$.

7. Metodologia

Para a etapa de experimentação do trabalho, houve a implementação de ambos os métodos descritos anteriormente, utilizando a linguagem Python 3. A linguagem oferece uma biblioteca que permite a visualização gráfica de cada iteração do algoritmo, o que facilita a compreensão.

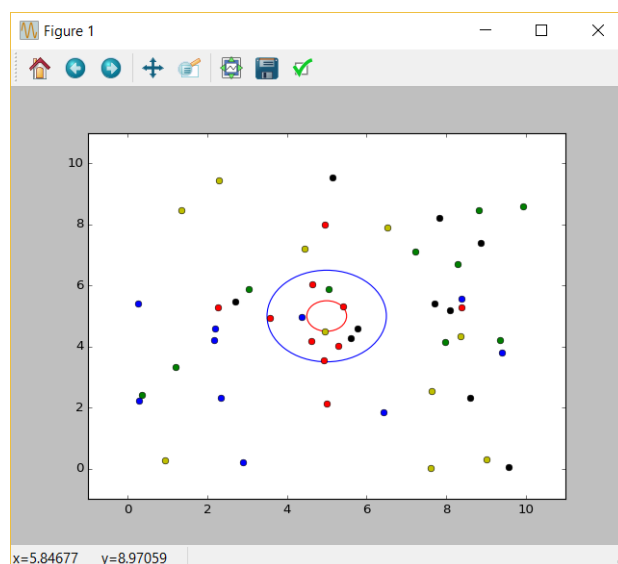


Figura 4. Ambiente de execução dos experimentos

Foram testadas as variações para o seguinte conjunto de parâmetros:

- **Quantidade de robôs em cada subgrupo**
- **Quantidade de subgrupo**
- **Alcance do sensor de obstáculos**

Os indivíduos em cada um dos experimentos aparecem aleatoriamente no mapa, portanto, foi necessário realizar cada um dos experimentos 10 vezes, para melhores resultados.

8. Experimentos

Nessa seção serão apresentados e analisados os resultados dos experimentos. Ela será dividida em sub seções que tratarão de experimentos com um determinado valor de alcance do sensor.

Antes de cada um dos experimentos, o ambiente de simulação é preparado com valores estáticos para alguns parâmetros, como é possível observar na tabela 1. Os valores utilizados surgiram de observações e experimentos prévios.

Parâmetro	Valor
Região de surgimento de robôs	[[0,10],[0,10]]
Posição da área objetivo	[5,5]
Raio da área objetivo	0.5
Força máxima (módulo)	0.08
K_ATT	0.1
K_REP	0.04

Tabela 1. Parâmetros de execução

Antes de apresentar os resultados dos experimentos, é interessante ressaltar que durante os testes percebeu-se que, no algoritmo da distância média, houveram execuções em que o grupo com maior prioridade levou um tempo muito maior que a média para alcançar a área objetivo. Isso aconteceu devido a geração inicial das posições dos robôs. Nessas execuções, alguns indivíduos tinham grande dificuldade para encontrar um caminho em meio aos outros robôs que estavam estáticos, o que levou a um aumento excessivo nos tempo de trânsito e alimentação, além da distância percorrida, já que o indivíduo tentava várias formas de desviar do obstáculo. Isso aconteceu cerca de uma vez a cada bateria de dez execuções. Os resultados foram retirados dos gráficos das execuções, uma vez que gerariam muita poluição. Essas execuções contribuíram para as conclusões formuladas sobre os algoritmos, como será visto em seções posteriores.

8.1. Alcance 2

Primeiramente será analisado o resultado da execução com sensor de obstáculos com alcance 2. A figura 8 mostra o gráfico com os resultados das execuções, exibindo a informação sobre tempo de trânsito. As informações que dizem respeito a tempo, sempre serão exibidas como número de iterações a fim de desconsiderar o tempo de processamento computacional.

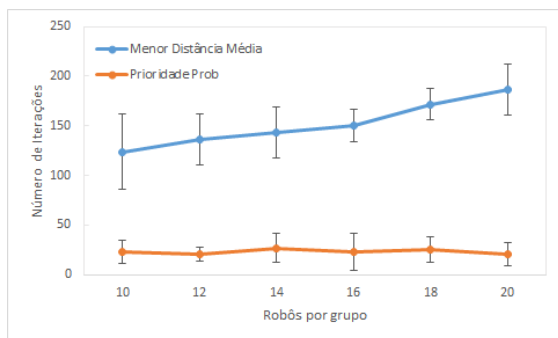


Figura 5. 3 grupos

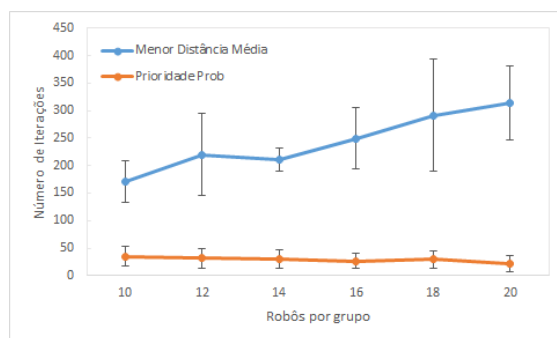


Figura 6. 4 grupos

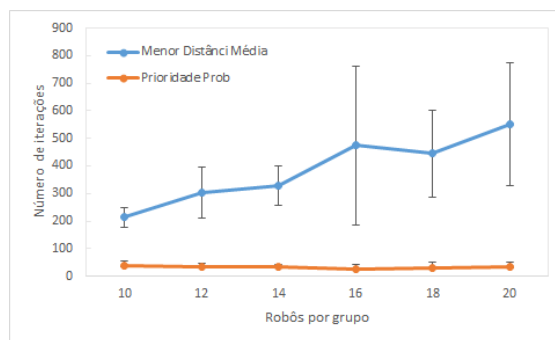


Figura 7. 5 grupos

Figura 8. Tempo de Trânsito

É possível notar que no algoritmo probabilístico, os grupos levam menos iterações para começar a chegar a área objetivo. Isso era esperado, já que no algoritmo da distância os grupos de menor prioridade ficam estáticos, esperando o grupo com maior prioridade terminar de atingir a área objetivo, ao contrário do probabilístico em que os grupos podem avançar ao objetivo a qualquer momento, o que faz com que o tempo médio de espera de cada grupo seja menor e consequentemente diminuindo o tempo de trânsito.

É necessário observar como o desvio padrão aumenta conforme o número total de robôs também aumenta, somente no caso do algoritmo da distância. Isso se deve ao problema da geração das posições iniciais que possibilita a existência de robôs em posições problemáticas causadas por outros robôs estáticos.

No quesito tempo de alimentação, percebe-se uma vantagem considerável do algoritmo da distância média. O que causa essa discrepância é que o congestionamento pode ocorrer no algoritmo probabilístico, já que mais de um grupo pode tentar atingir a área objetivo no mesmo período de tempo, fato que aumenta o tempo médio de alimentação. O

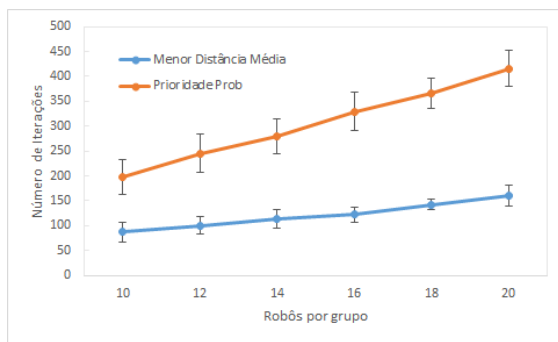


Figura 9. 3 grupos

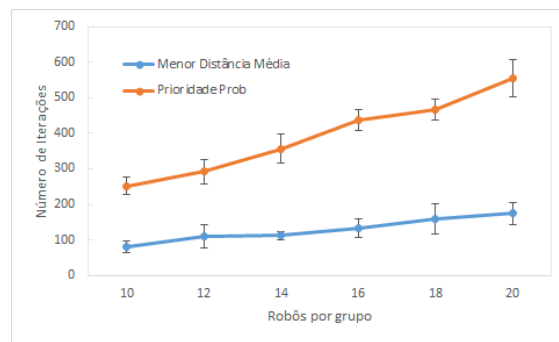


Figura 10. 4 grupos

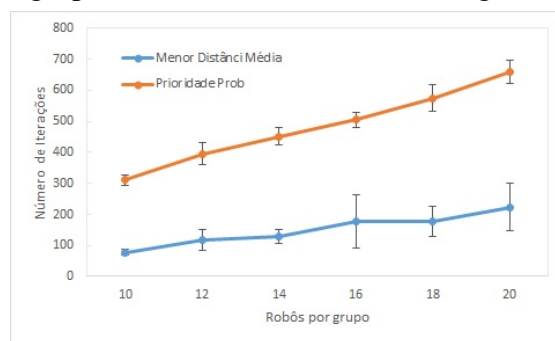


Figura 11. 5 grupos

Figura 12. Tempo de Alimentação

método da distância resolve isso melhor, já que impede os indivíduos de grupos diferentes de se moverem ao mesmo tempo.

Na métrica da distância percorrida pelos robôs, o método da distância média se mostrou mais eficiente, como se observa na figura 16. Esse comportamento se deve ao fato de que quando os grupos estão esperando a sua vez de ir em direção a área objetivo, os robôs desse grupo ficam estáticos. Isso aumenta o tempo de trânsito, como foi visto anteriormente, porém reduz movimentações que são causadas pelo efeito repelente de outros robôs muito próximos.

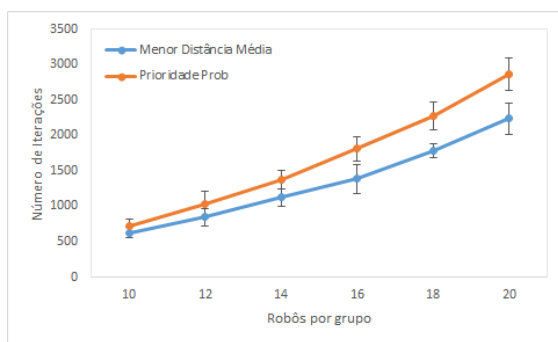


Figura 13. 3 grupos

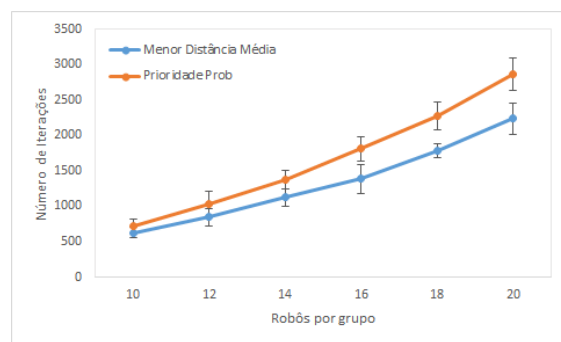


Figura 14. 4 grupos

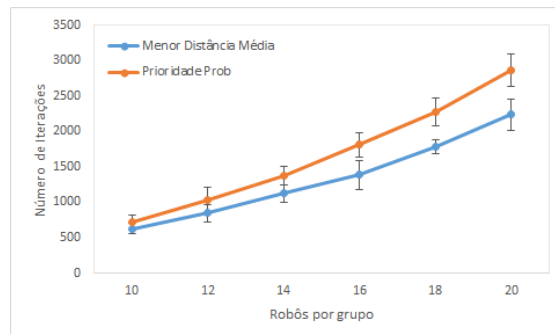


Figura 15. 5 grupos

Figura 16. Distância Percorrida

8.2. Variação do alcance do sensor

O aumento do alcance do sensor causou um impacto pequeno, porém negativo em todas as métricas como pode observado na figura 20. Entretanto, a alteração mais significativa foi nos desvios padrão do método da menor distância. Aumentando o alcance do sensor de obstáculos, o algoritmo se mostrou mais sensível à posição de robôs estáticos no mapa, o que aumentou a incidência dos casos em que um indivíduo tem dificuldades em encontrar o caminho até a área objetivo.

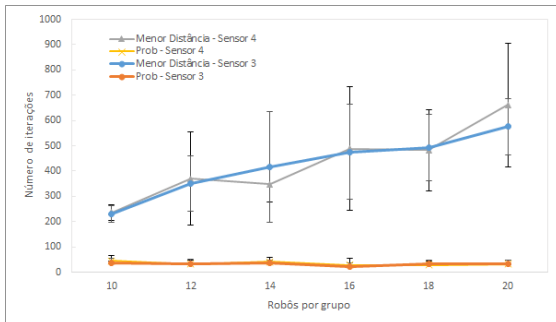


Figura 17. Tempo de trânsito

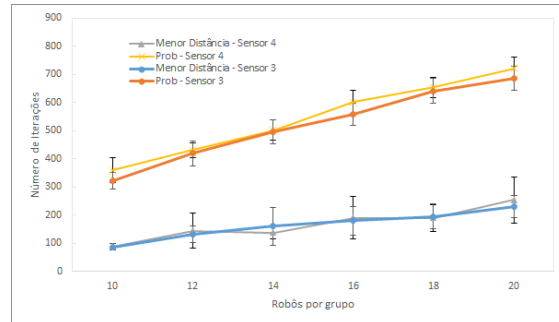


Figura 18. Tempo de alimentação

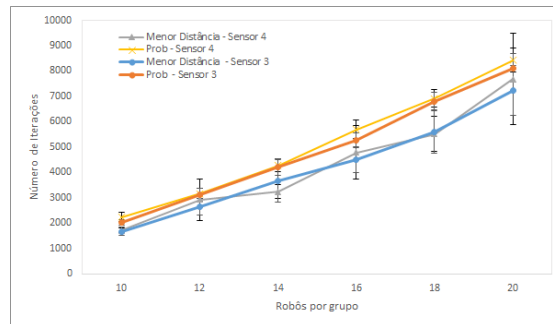


Figura 19. Distância percorrida

Figura 20. Comparação entre alcances - 5 Grupos

Esse "problema" com alcances maiores de sensor pode ser explicado com a equação 3. Essa equação define que todos os obstáculos (robôs) dentro do alcance do sensor vão influenciar na movimentação do robô. Apesar de ser proporcional ao inverso do quadrado da distância, essa mínima interferência é totalmente desnecessária, tendo em vista a escala do robô e gera uma dificuldade extra na definição do caminho até a área objetivo.

9. Conclusão

Os métodos aqui apresentados mostraram desempenhos um tanto diferentes. No método pirobalístico, o tempo gasto para o grupo começar a entrar na área alvo é bem pequeno, sendo pouco afetado pelo número total de robôs. Entretanto para o tempo total médio de alimentação, ele apresenta um desempenho bem ruim quando comparado com o método da distância média. Isso acontece porque o método do probabilístico permite o congestionamento, já que mais de um grupo pode tentar chegar ao mesmo tempo a área objetivo. Com o congestionamento, também aumenta a métrica da distância percorrida

pelos indivíduos, já que eles vão se movimentar devido a influência de outros robôs muito próximos.

O algoritmo da distância média se mostrou eficaz em evitar o congestionamento de robôs porém sofre com o problema da navegação quando existem outros robôs estáticos "bloqueando" o caminho até a área objetivo, principalmente quando o número total de robôs é aumentado revelando que o método é pouco escalável, já que provavelmente com um certo número de robôs podem haver casos em que o algoritmo não termina. Esse problema se deve muito à técnica de navegação por campos potenciais que sofre com o problema de ótimos locais. Uma abordagem alternativa que pode corrigir o problema é fazer com que os robôs com menor prioridade não fiquem estáticos e se afastem de outros robôs ou até mesmo utilizar outra técnica de navegação.

Além de corrigir esse problema, outra sugestão para trabalhos futuros é trabalhar com uma organização prévia dos grupos antes de que eles se dirijam a área objetivo. Como os robôs surgem no mapa desorganizados, talvez seja interessante agrupar os robôs do mesmo grupo.

Referências

- Grossman, D. D. (1988). Traffic control of multiple robot vehicles. *IEEE Journal on Robotics and Automation*, 4(5):491–497.
- Hoshino, S. (2011). Multi-robot coordination methodology in congested systems with bottlenecks. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2810–2816. IEEE.
- Marcolino, L. S. and Chaimowicz, L. (2009). Traffic control for a swarm of robots: Avoiding target congestion. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1955–1961. IEEE.
- Marcolino, L. S., dos Passos, Y. T., de Souza, Á. A. F., dos Santos Rodrigues, A., and Chaimowicz, L. (2016). Avoiding target congestion on the navigation of robotic swarms. *Autonomous Robots*, pages 1–24.