

# Planejamento Hierárquico Para Enxames Robóticos Baseado na Exploração do Espaço de Configurações

Vinicius Graciano Santos e Luiz Chaimowicz

<sup>1</sup> Laboratório de Visão e Robótica (VeRLab)

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{vgs, chaimo}@dcc.ufmg.br

**Abstract.** *This work proposes to model a robotic swarm using a hierarchical abstraction that reduces the dimensionality of the planning and control problem. Thus, it is possible to use it in order to solve the path planning problem by exploring its configuration space, which is independent of the number of robots. We perform simulated and real experiments. Results show that our approach allows the swarm to navigate without congestions in a smooth and coherent fashion, being suitable for large groups of robots.*

**Resumo.** *Este trabalho propõe modelar um enxame robótico através de uma abstração hierárquica que reduza a dimensionalidade do problema de planejamento e controle. Dessa maneira, é possível utilizá-la para resolver o problema do planejamento de caminhos a partir da exploração do seu espaço de configurações, que é independente do número de robôs. Experimentos foram realizados em simulação e ambientes reais e os resultados mostram que a abordagem permite a navegação sem congestionamentos para um enxame robótico de forma suave e coerente, sendo adequada para grandes grupos de robôs.*

## 1. Introdução

O uso de grandes grupos de agentes têm recebido muita atenção em anos recentes. Muitas vezes inspirados na biologia, tais sistemas utilizam um enxame de agentes simples com o objetivo de executar tarefas complexas, como a exploração e o mapeamento de ambientes desconhecidos, o transporte de grandes objetos, manipulação distribuída e no auxílio em cenários de resgate.

Não somente limitado à robótica, o estudo de algoritmos para o controle de enxames também possui um grande impacto em outras áreas como realidade virtual, jogos digitais e simulações de multidões. Normalmente, os ambientes criados nessas aplicações são povoados por entidades que devem comportar-se como um grupo coerente em vez de indivíduos únicos. Por exemplo, dado o modelo virtual de um prédio, é desejável simular o comportamento de uma multidão evacuando o local em caso de incêndios.

Um problema comum na maioria dessas tarefas é a navegação. Como a dimensão do espaço de configurações do sistema aumenta exponencialmente com o número de agentes, o uso de algoritmos tradicionais para resolver o problema do planejamento de caminhos torna-se muito caro, mesmo para um número pequeno de robôs. Possíveis soluções incluem planejar separadamente um caminho para cada agente, considerando-se

apenas o espaço de configurações individual, ou definir campos vetoriais globais para controlar o enxame como um todo. Tais soluções geralmente levam a conflitos nas trajetórias dos robôs, que devem ser resolvidos. Um exemplo é o congestionamento de tráfego, que geralmente ocorre quando um grande número de robôs movimentam-se para uma mesma região do ambiente durante o mesmo intervalo de tempo.

Nos últimos anos, novas abordagens tentam modelar o enxame através de estruturas virtuais que reduzem a dimensionalidade do problema de controle. De forma geral, tais estruturas definem regras de formação para os agentes de modo a agrupá-los hierarquicamente, introduzindo um novo nível de abstração para o controle do enxame, pois apenas a estrutura virtual precisa ser explicitamente controlada.

A presente contribuição explora o uso de paradigmas hierárquicos em conjunto com algoritmos de controle de tráfego para possibilitar a navegação de enxames em ambientes com obstáculos, de forma a evitar congestionamentos entre diferentes grupos. O objetivo foi aproveitar-se das ideias e soluções robustas que têm sido desenvolvidas em ambas as áreas de pesquisa para poder solucionar ambos os problemas de forma eficiente.

O artigo está organizado como se segue: a Seção 2 apresenta alguns conceitos fundamentais e discute trabalhos relacionados na área; a Seção 3 apresenta a metodologia utilizada para definir a abstração e também controladores hierárquicos com aplicações na área de controle de tráfego e planejamento de caminhos. Resultados experimentais em ambientes simulados e reais são mostrados na Seção 4, enquanto a Seção 5 apresenta as conclusões e diretrizes para trabalhos futuros.

## 2. Referencial Teórico

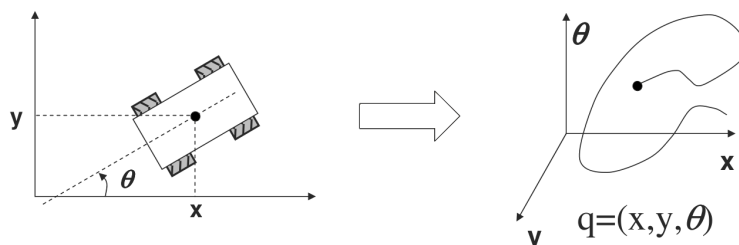
### 2.1. Espaço de configurações

O problema do planejamento de caminhos consiste em encontrar um caminho contínuo no espaço de configurações de um sistema robótico entre uma configuração inicial e final [LaValle 2006]. Para se planejar um caminho para um robô é preciso especificar completamente sua localização, visto que é necessário garantir que nenhuma colisão com algum obstáculo ocorra. Com isso, é desejável que tal especificação seja formada por um conjunto mínimo de parâmetros que possuam propriedades relevantes de acordo com a geometria do problema. Para tornar a discussão mais precisa, de acordo com [Choset et al. 2005], temos as seguintes definições:

- A *configuração*  $\mathbf{q}$  de um sistema robótico é uma especificação completa da posição de cada ponto desse sistema.
- O *espaço de configurações*  $\mathbb{Q}$  do sistema robótico é o espaço de todas as possíveis configurações.
- O número de *graus de liberdade* do sistema robótico é a dimensão do seu espaço de configurações.

Para ilustrar essas definições, considere um robô que possa efetuar translações e rotações no plano. Assim, serão necessários três parâmetros para especificar completamente os pontos desse robô: sua posição e orientação no plano. Logo, é fácil ver que  $\mathbf{q} = [x, y, \theta]$  e consequentemente  $\mathbb{Q} = \mathbb{R}^2 \times S^1$ . Como  $S^1$  é uma variedade (*manifold*) unidimensional, dado que existe um homeomorfismo local desse espaço para a reta euclidiana  $\mathbb{R}$ , é possível localmente visualizar  $\mathbb{R}^2 \times S^1$  no espaço  $\mathbb{R}^3$ , como é mostrado na

Figura 1. Tal representação é interessante por mostrar que mesmo estando o robô preso a uma *espaço de trabalho* bidimensional, seu espaço de configurações é tridimensional.



**Figura 1. Representação local em  $\mathbb{R}^3$  do espaço de configurações  $\mathbb{R}^2 \times S^1$  de um robô móvel  $\mathbf{q} = [x, y, \theta]^T$ .**

A principal vantagem dessa modelagem relaciona-se com o fato de ser mais fácil planejar o movimento para um único ponto do que para um corpo rígido complexo. Entretanto, ao considerar-se sistemas com múltiplos robôs, o espaço de configurações é obtido através do produto cartesiano do espaço de configurações de cada agente. Logo, dado um crescimento linear no número de graus de liberdade do sistema, a representação completa do respectivo espaço de configurações tende a um crescimento exponencial. Por esse motivo, o principal objetivo do trabalho é encontrar uma representação com poucos graus de liberdade para um grupo formado por múltiplos robôs.

## 2.2. Campos potenciais

Uma *função potencial* é uma função diferenciável  $U : \mathbb{R}^m \rightarrow \mathbb{R}$  cujo gradiente é um vetor  $\nabla U(\mathbf{q}) = \left[ \frac{\partial U}{\partial q_1}(\mathbf{q}), \dots, \frac{\partial U}{\partial q_m}(\mathbf{q}) \right]^T$  que aponta na direção que maximiza  $U$  localmente. O método dos campos potenciais artificiais [Khatib 1985] utiliza o campo vetorial definido por  $\nabla U$  para dirigir o robô como se fosse uma partícula. O gradiente pode ser visto de forma intuitiva como forças atuando em uma partícula carregada que é atraída para uma posição objetivo e repelida de obstáculos próximos. Tal processo é feito seguindo-se o gradiente negado da função potencial, uma técnica conhecida como *descida do gradiente*.

Uma das funções mais simples e mais utilizadas na área é o potencial atrativo/repulsivo [Choset et al. 2005]. A função potencial pode ser construída como a soma dos potenciais atrativos e repulsivos

$$U(\mathbf{q}) = U_{atr}(\mathbf{q}) + U_{rep}(\mathbf{q}).$$

O potencial atrativo considera um crescimento quadrático conforme o aumento da distância do robô com relação a sua posição objetivo. Seja  $\mathbf{q}_{obj}$  tal posição, então

$$\begin{aligned} U_{atr}(\mathbf{q}) &= \frac{1}{2} \lambda \|\mathbf{q} - \mathbf{q}_{obj}\|^2 \\ \nabla U_{atr}(\mathbf{q}) &= \lambda(\mathbf{q} - \mathbf{q}_{obj}). \end{aligned}$$

Um potencial repulsivo evita a colisão do robô com os obstáculos do ambiente e a magnitude da sua força de repulsão depende de quão próximo o agente está do obstáculo. Seja

$d_i(\mathbf{q})$  a menor distância de  $\mathbf{q}$  a um objeto  $i \in O$ , com  $\nabla d_i(\mathbf{q})$  apontando na direção que maximiza a distância ao objeto. Assim, define-se o potencial repulsivo como

$$U_{rep_i}(\mathbf{q}) = \begin{cases} \frac{1}{2}\eta \left( \frac{1}{d_i(\mathbf{q})} - \frac{1}{d^*} \right)^2, & d_i(\mathbf{q}) \leq d^*, \\ 0, & d_i(\mathbf{q}) > d^*, \end{cases}$$

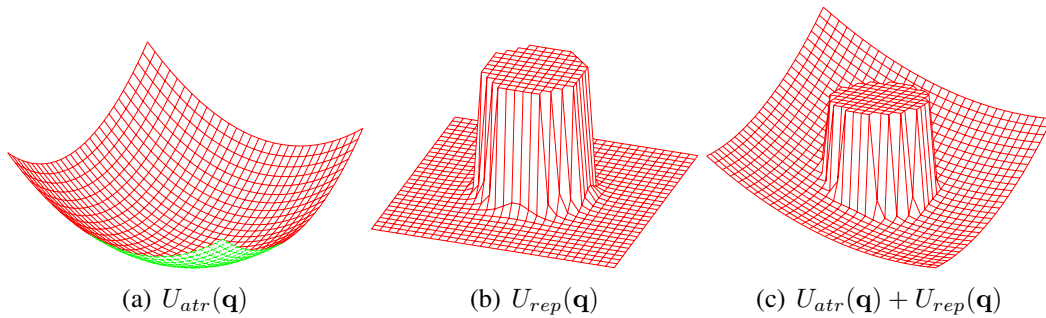
cujo gradiente é

$$\nabla U_{rep_i}(\mathbf{q}) = \begin{cases} \eta \left( \frac{1}{d^*} - \frac{1}{d_i(\mathbf{q})} \right) \frac{1}{d_i(\mathbf{q})^2} \nabla d_i(\mathbf{q}), & d_i(\mathbf{q}) \leq d^*, \\ 0, & d_i(\mathbf{q}) > d^*. \end{cases}$$

Note que  $d^* \in \mathbb{R}$  é um limiar que permite ignorar obstáculos localizados suficientemente longe e as constantes  $\lambda, \eta > 0$  são escalares de ganho que permitem afinar o controle do módulo dos vetores. Exemplos de campos potenciais são expostos na Figura 2.

Modelando-se o robô como uma partícula e definindo-se o modelo cinemático  $\dot{\mathbf{q}} = \mathbf{v}$ , sendo  $\mathbf{v}$  a velocidade do robô, a lei de controle baseada no método dos campos potenciais é dada por

$$\mathbf{v} = -\nabla U_{atr}(\mathbf{q}) - \sum_{i \in O} \nabla U_{rep_i}(\mathbf{q}).$$



**Figura 2. Exemplos de campos potenciais.**

### 2.3. Mapas de rotas

Um planejador deve encontrar um caminho a partir de uma configuração inicial até uma configuração final. Caso seja desejável que vários caminhos sejam computados sobre o mesmo ambiente, então é interessante construir uma única vez uma estrutura de dados que facilite os planejamentos subsequentes. Tal estrutura é denominada um *mapa* e três modelos prevalecem na literatura: mapas topológicos, geométricos e de grades [Choset et al. 2005, LaValle 2006].

Mapas topológicos modelam o ambiente através de grafos cujos nós representam pontos de interesse e arestas definem relações de adjacência. Por exemplo, em um escritório, interseções de corredores podem especificar nós, enquanto as arestas correspondem a comportamentos específicos ou comandos de movimento que permitem mover o robô de um lugar ao outro. Modelos geométricos utilizam primitivas geométricas para representar o ambiente, como retas, curvas, polígonos e poliedros. Por fim, mapas baseados em

grades são estruturas que discretizam o ambiente em células, cujos valores relacionam-se com a probabilidade da respectiva porção do ambiente conter um obstáculo. Como exemplo, um mapa de duas dimensões poderia ser modelado a partir de uma imagem na qual cada *pixel* corresponde a uma célula discreta.

Um *Mapa de Rotas* (MR) é um mapa topológico embutido no subconjunto  $\mathbb{Q}_{livre} \subset \mathbb{Q}$  das configurações livres de colisões. Os nós desse grafo correspondem a locações específicas sem obstáculos e as arestas representam um caminho livre entre nós vizinhos. Dessa forma, é dito que o mapa captura a topologia do espaço livre do ambiente. Formalmente, um mapa de rotas é a união de curvas unidimensionais pertencentes a  $\mathbb{Q}_{livre}$  que podem ser conectadas por um caminho entre todo par  $(\mathbf{q}_{inicial}, \mathbf{q}_{final})$ , também respeitando as seguintes propriedades:

1. **Acessibilidade:** existe um caminho de  $\mathbf{q}_{inicial} \in \mathbb{Q}_{livre}$  para algum  $\mathbf{q}'_{inicial} \in \text{MR}$ ;
2. **Departabilidade:** existe um caminho de  $\mathbf{q}'_{final} \in \text{MR}$  para algum  $\mathbf{q}_{final} \in \mathbb{Q}_{livre}$ ;
3. **Conectividade:** existe um caminho em MR entre  $\mathbf{q}'_{inicial}$  e  $\mathbf{q}'_{final}$ .

A utilização de um mapa de rotas é semelhante a de um sistema de rodovias, pois se existe uma rodovia que permite acesso a uma cidade destino, basta encontrar o caminho a partir de uma cidade origem até a rodovia para garantir que exista um caminho entre as cidades.

## 2.4. Trabalhos relacionados

Um dos primeiros algoritmos para controlar um enxame de agentes foi proposto em [Reynolds 1987] com o objetivo de gerar um padrão de movimento realístico para a simulação de um bando de pássaros. De forma resumida, interações locais entre agentes dentro de uma mesma área de vizinhança definem um comportamento emergente para todo o bando. Tais interações podem ser modeladas como um caso especial do método dos campos potenciais sociais [Reif and Wang 1999], que utiliza uma técnica de descida do gradiente para gerar forças virtuais que levam os agentes ao seu objetivo enquanto desviam localmente de obstáculos [Khatib 1985].

Uma diferente abordagem considera o grupo como uma entidade única, modelando-o a partir de uma *estrutura virtual* [Tan and Lewis 1996]. O movimento desejado é comandado à estrutura, que controla implicitamente o enxame robótico. Baseado no mapeamento do espaço de configurações do enxame para uma variedade de menor dimensão, uma abstração formal foi desenvolvida em [Belta and Kumar 2004] que permite o controle da pose e do formato de um time de robôs localizados em um plano.

No contexto da navegação de enxames, o problema de controle de tráfego é um importante tópico de pesquisa, sendo caracterizado como um problema de conflito de recursos. Em geral, trabalhos nessa área assumem que os robôs estão contidos em um ambiente estruturado, onde navegam em vias delimitadas que encontram-se em cruzamentos, geralmente o lugar no qual o controle de tráfego é aplicado. Tal tarefa pode ser realizada utilizando-se um único agente administrador [Dresner and Stone 2005] ou uma robusta rede distribuída de sensores [Viswanath and Krishna 2007].

Em [Marcolino and Chaimowicz 2009] um algoritmo para o controle de congestionamento em ambientes não estruturados foi apresentado. Especificamente, a pesquisa lidou com o problema de grupos movendo-se em direções opostas, condição que pode ser considerada um dos piores casos de conflito de tráfego. O presente trabalho explora o uso

de hierarquias para promover uma nova solução para esse problema, além de possibilitar em conjunto a aplicação de planejadores de caminho baseados em amostragem.

### 3. Metodologia

Considera-se um conjunto de robôs completamente atuados com modelo dinâmico dado por  $\dot{\mathbf{q}}_i = \mathbf{v}_i$ ,  $\dot{\mathbf{v}}_i = \mathbf{u}_i$ , sendo  $\mathbf{q}_i = [x_i, y_i]^T$  a pose do robô  $i$ ,  $\mathbf{v}_i$  sua velocidade e  $\mathbf{u}_i$  sua entrada de controle. Tais robôs são agrupados em um conjunto  $\Gamma$  de grupos, no qual cada grupo  $j$  é modelado por um par  $(g_j, s_j)$  que compreende a pose e a forma do grupo, respectivamente. Assim, a abstração  $a_j$  é definida por:

$$\begin{aligned} a_j &= (g_j, p_j) \\ g_j &= (\mu_j^x, \mu_j^y, \theta_j) \\ p_j &= (s_j, t_j). \end{aligned} \quad (1)$$

Tal abstração pode ser vista como uma elipse definida implicitamente por  $c_j(x, y) = 0$ , centrada em  $(\mu_j^x, \mu_j^y)$ , com orientação  $\theta_j$  e eixos principais dados por  $s_j, t_j > 0$ .

#### 3.1. Agrupamento

Um grupo  $j \in \Gamma$  é formado pelos robôs que satisfazem a restrição  $c_j(\mathbf{q}_i) < 0$ . Portanto, a curva  $c_j(x, y) = 0$  pode ser vista como uma borda que limita e define um grupo. Com o intuito de simplificar as próximas equações, considere que  $\mathbf{q}_i$  é dado em relação ao eixo de referência ortonormal definido pelos componentes de  $g_j$ .

Deseja-se criar um campo vetorial que sirva de controle para os agentes, de forma que um robô  $i$  será agrupado se  $c_j(\mathbf{q}_i) < 0$  e repellido no caso contrário. Para definir-se um campo potencial cujo gradiente possui tais características, considere a seguinte função:

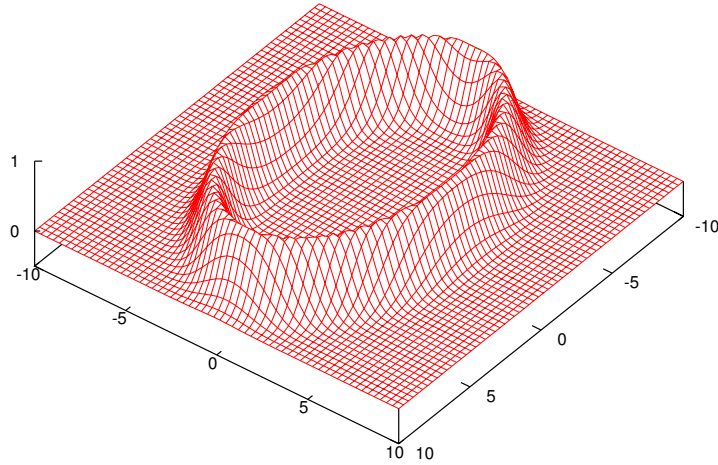
$$\phi(\mathbf{q}_i, a_j) = \|\mathbf{q}_i\| \left( 1 - \frac{s_j t_j}{\sqrt{(x_i t_j)^2 + (y_i s_j)^2}} \right). \quad (2)$$

A equação (2) mapeia  $\mathbf{q}_i$  para a sua distância radial com relação à curva  $c_j(x, y) = 0$ , não sendo definida quando o robô encontra-se na origem. Logo, a função

$$U(\mathbf{q}_i, a_j) = \begin{cases} 0, & \text{se } x_i = 0 \wedge y_i = 0 \\ e^{-\gamma \phi^2(\mathbf{q}_i, a_j)}, & \text{se } x_i \neq 0 \vee y_i \neq 0 \end{cases} \quad (3)$$

produz um campo potencial artificial cujo máximo está localizado na curva  $c_j(x, y) = 0$ , formando uma superfície similar a uma bacia, com  $\gamma$  sendo inversamente proporcional a espessura de sua parede, como é exposto na Figura 3. Assim, a seguinte lei de controle é proposta:

$$\mathbf{u}_i = -k_1 \nabla U(\mathbf{q}_i, a_j) - k_2 \dot{\mathbf{q}}_i + \sum_{k \in N_i} F_r(\mathbf{q}_i, \mathbf{q}_k). \quad (4)$$



**Figura 3. Campo potencial artificial  $U(\mathbf{q}_i, a_j)$  com  $\gamma = 1$ ,  $s_j = 5$  e  $t_j = 8$ .**

As constantes  $k_1$  e  $k_2$  são positivas. O primeiro termo é a força responsável por agrupar os agentes que estão dentro do grupo e repelir aqueles que não estão; o segundo termo é uma força dissipativa de energia, usada para melhorar a estabilidade; e o último representa forças locais de repulsão, que previnem colisões entre os robôs vizinhos [Choset et al. 2005]. O conjunto  $N_i$  consiste em cada agente  $k$  que está a um certo limite  $\delta$  de distância do robô  $i$ .

Note que controladores lineares simples poderiam ser aplicados aos parâmetros de  $a_j$ , tal que, em conjunto com (4), uma hierarquia é definida na qual os robôs são controlados implicitamente de acordo com as abstrações de alto nível.

### 3.2. Controle de congestionamento

A ideia geral do algoritmo para evitar congestionamentos é aproveitar-se das propriedades geométricas da estrutura virtual  $c_j(x, y) = 0$  para gerar forças repulsivas entre grupos com o objetivo de desviá-los de áreas onde um cenário de congestionamento poderia ocorrer.

Dados dois grupos  $m, n \in \Gamma$  e seja  $\mathbf{p}$  o centroide dos  $z$  pontos de interseção entre  $c_m(x, y) = 0$  e  $c_n(x, y) = 0$ , com  $z \in \{1, 2, 3, 4, \infty\}$ . A força repulsiva  $F_g$  entre grupos deve ser diretamente proporcional a profundidade de penetração de  $\mathbf{p}$  com relação às elipses. Tal profundidade pode ser calculada por  $\phi(\mathbf{p}, a_j)$ . Então, especifica-se a força atuando sobre o grupo  $m$  como

$$F_g(m, n) = \begin{cases} 0, & \text{para } z = 0 \\ \phi(\mathbf{p}, a_m)(\hat{\mathbf{g}}_m - \hat{\mathbf{p}}), & \text{para } z > 0, \end{cases} \quad (5)$$

com  $\mathbf{g}_m = [\mu_m^x, \mu_m^y]^T$  e  $\hat{\mathbf{p}}$  denotando o vetor unitário na direção  $\mathbf{p}$ .

A equação (5) apenas tenta minimizar a área de interseção entre os dois grupos envolvidos, mas não cumpre o objetivo de desviá-los. Portanto, também é necessário especificar uma força de rotação  $R_g$  de forma que  $F_g \cdot R_g = 0$ . Como (5) define um campo vetorial  $F_g = (F_x, F_y)$ , então simplesmente definindo-se  $R_g = (-F_y, F_x)$  é possível satisfazer a restrição de ortogonalidade. Assim, a força resultante que desvia o grupo  $m$  de  $n$  é dada por

$$D(m, n) = F_g(m, n) + R_g(m, n). \quad (6)$$

Considerando-se o modelo dinâmico  $\ddot{\mathbf{g}}_m = \mathbf{u}_m^g$  para os grupos, seu controlador é especificado como

$$\mathbf{u}_m^g = k_3 \sum_{n \in \Gamma} D(m, n) - k_4 \dot{\mathbf{g}}_m + F_{obj}(\mathbf{g}_m), \quad (7)$$

sendo  $F_{obj}(\mathbf{g}_m)$  uma força atrativa que leva o grupo  $m$  na direção da sua posição objetivo [Choset et al. 2005]. O comportamento resultante de (7) é mostrado na Figura 4.

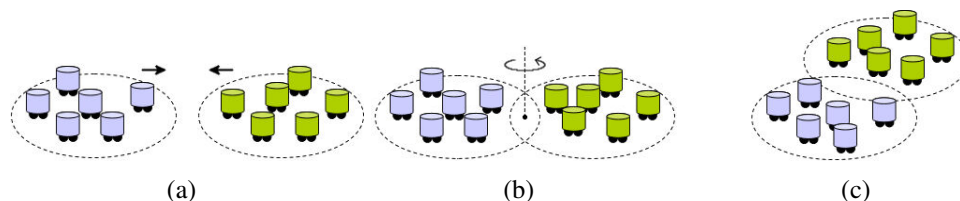


Figura 4. Passos de execução do algoritmo para controle de congestionamento.

### 3.3. Planejamento de caminhos

Como citado na Seção 2, a representação explícita do espaço de configurações de um exame robótico torna-se impraticável devido ao seu crescimento exponencial. A abstração  $a_j$  permite o uso de cinco parâmetros pertencentes ao espaço  $\mathbb{Q} = \mathbb{R}^4 \times S^1$  para representar um conjunto arbitrário de robôs. No entanto, especificar completamente tal espaço em memória ainda possui um custo elevado.

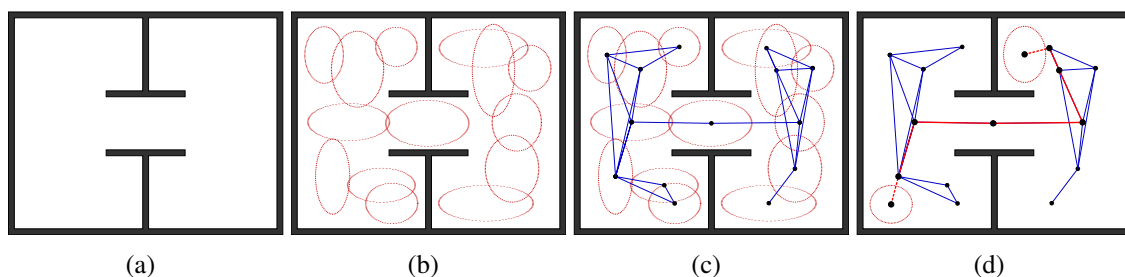
O método do *Mapa de Rotas Probabilístico* [Kavraki et al. 1996] explora o relativo baixo custo necessário para verificar se uma determinada configuração é válida, isto é, sem colisões com obstáculos. O algoritmo emprega a amostragem das configurações com o objetivo de determinar um mapa topológico aproximado. A característica mais importante da abordagem é que uma representação explícita do espaço de configurações não é construída. Em vez disso, o procedimento depende de testes de interseção específicos entre o sistema robótico e os obstáculos do ambiente.

O mapa de rotas é representado por um grafo  $G = (V, A)$ , em que  $V$  denota um conjunto de configurações válidas e uma aresta  $(\mathbf{q}_1, \mathbf{q}_2) \in A$  corresponde a um caminho livre de colisões entre  $\mathbf{q}_1$  e  $\mathbf{q}_2$ . Inicialmente, o grafo  $G$  está vazio e diversas configurações são amostradas em  $\mathbb{Q}$ , sendo as válidas adicionadas em  $V$ . O processo é repetido até se obter um número predefinido de vértices. Logo após, para cada  $\mathbf{q} \in V$ , um planejador local tenta conectar  $\mathbf{q}$  aos seus  $k$  vizinhos mais próximos, acrescentando arestas no grafo conforme a validade desses caminhos. Finalizado o processo de construção, qualquer algoritmo de busca em grafos pode ser aplicado para determinar um caminho entre duas configurações. A Figura 5 ilustra a execução do algoritmo.

Especificamente, o algoritmo desenvolvido recebe como entrada um mapa do espaço de trabalho e um procedimento para testar a interseção da elipse com os obstáculos desse ambiente. O planejador local, responsável por conectar dois vértices distintos, consiste em verificar se as configurações contidas na combinação linear convexa dos parâmetros da abstração são livres de colisões. Por fim, o algoritmo de busca  $A^*$  é utilizado para determinar o menor caminho no grafo.

O algoritmo  $A^*$  [Choset et al. 2005, LaValle 2006] é amplamente utilizado para resolver problemas genéricos de busca em grafos devido ao seu ótimo desempenho pe-





**Figura 5. Construção e utilização de uma Mapa de Rotas Probabilístico. (a) Mapa do espaço de trabalho. (b) Amostragem de configurações válidas. (c) Arestas são adicionadas entre pares de configurações cuja combinação convexa seja livre de colisões. (d) Caminho retornado entre uma configuração inicial e final.**

rante outros algoritmos similares, sendo diretamente uma extensão do clássico algoritmo de Dijkstra. A característica mais importante do  $A^*$  é o uso de uma heurística que tende a minimizar o número de vértices avaliados. A otimalidade do algoritmo é garantida se a heurística empregada for admissível, isto é, se o custo estimado para alcançar o estado final não for maior do que custo ótimo, para todos os estados da busca.

#### 4. Resultados

Uma série de simulações e experimentos reais foram executados com o intuito de estudar a viabilidade da abstração proposta. A implementação foi feita utilizando-se a plataforma *Player/Stage* [Gerkey et al. 2003], um arcabouço de controle e simulação de robôs muito utilizado na comunidade científica. Os agentes virtuais foram baseados no modelo *P2DX*, um robô diferencial equipado com *lasers* para a estimação da distância a obstáculos. Os experimentos reais foram conduzidos com um conjunto de robôs *e-puck* [Mondada et al. 2009], um pequeno robô diferencial equipado com oito sensores infravermelho para sensoriamento de proximidade, com uma interface *bluetooth* que permite a comunicação local entre robôs, assim como o controle através de um computador remoto. Além disso, um processador *dsPIC* é responsável pelo processamento local. Para controlar os robôs diferenciais, a equação (4) foi adaptada para considerar as restrições não-holonômicas de acordo com [De Luca and Oriolo 1994].

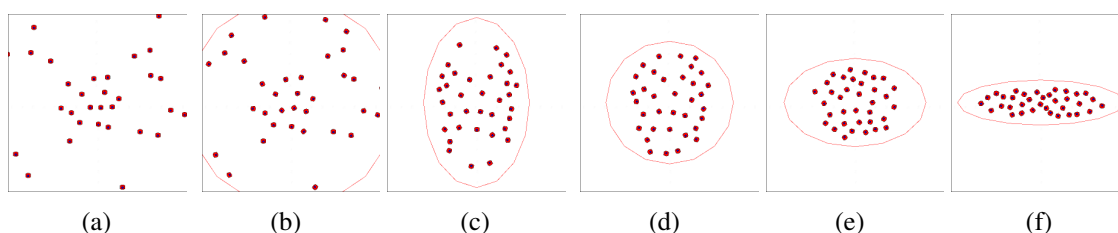


**Figura 6. Doze robôs *e-puck* utilizados nos experimentos.**

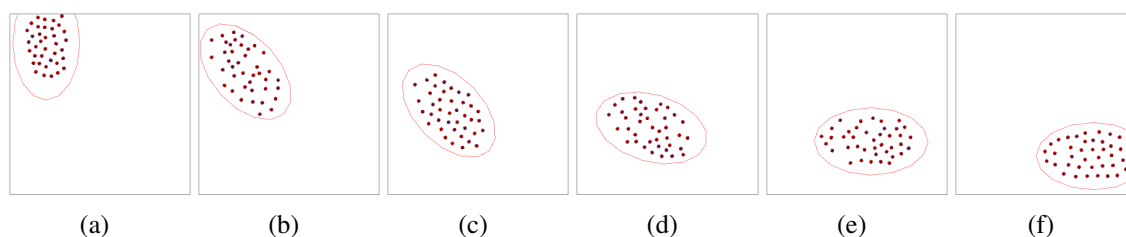
##### 4.1. Simulações

Quatro diferentes tipos de experimentos foram executados: agrupamento, navegação sem obstáculos, desvio de grupos e navegação com obstáculos. Durante o trabalho de iniciação científica, diversas instâncias desses experimentos foram analisadas. Nesta seção, uma amostra de cada tipo de experimento é apresentada.

O primeiro experimento consiste em um ambiente contendo robôs espalhados de acordo com uma distribuição normal (Figura 7(a)) e os componentes de  $p_j$  passam por transformações de modo a agrupar os agentes em diferentes formações elípticas, como mostrado nas Figuras 7(c) a 7(f). No segundo experimento, os robôs são posicionados em uma área predefinida e um caminho é especificado manualmente através de uma sequência de configurações. A Figura 8 exemplifica tais experimentos, apresentando um grupo de 40 robôs percorrendo um caminho definido por um arco de círculo. Como pode ser visto, os resultados indicam que a abstração é viável para controlar um grande número de agentes. Além disso, a movimentação dos robôs mostrou-se suave com relação às mudanças dos parâmetros abstratos.



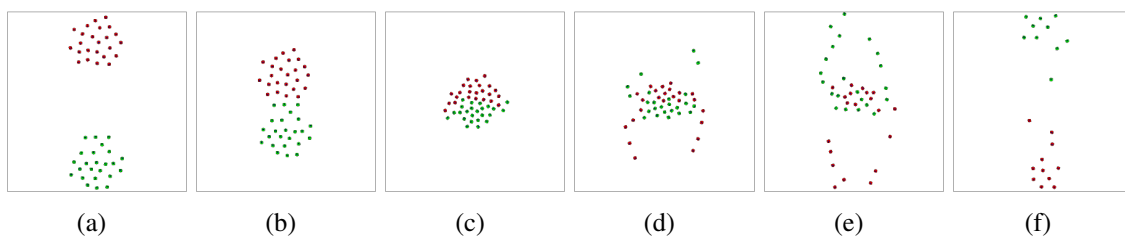
**Figura 7. Execução do experimento de agrupamento em simulação.**



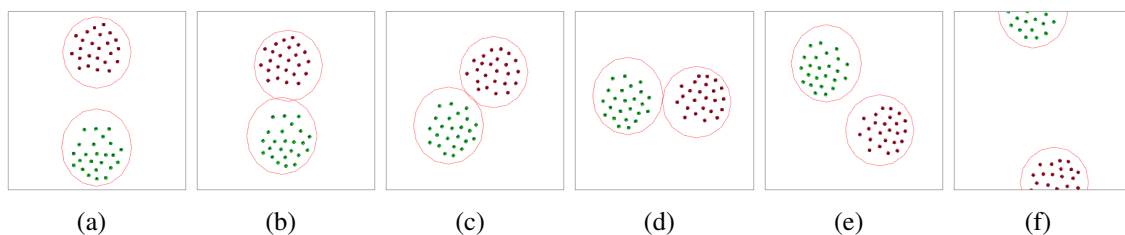
**Figura 8. Translações e rotações do enxame em um ambiente sem obstáculos.**

As Figuras 9 e 10 retratam um ambiente contendo 48 robôs virtuais divididos igualmente em dois grupos que movem-se em direções opostas. Na primeira simulação, apenas forças locais de repulsão são utilizadas para evitar colisões entre os robôs; enquanto na segunda, o algoritmo de controle de congestionamento é empregado. Através de uma inspeção visual, é fácil ver que o movimento dos robôs é mais suave e estável aplicando-se o algoritmo proposto. Ambos os grupos comportam-se de forma mais coesa durante a navegação e um cenário com congestionamentos foi completamente evitado. Por outro lado, a simulação que se baseou apenas em forças repulsivas apresentou conflitos de tráfego no momento em que os grupos colidiram e misturaram-se de forma não coesa. Adicionalmente, experimentos com mais de dois grupos também foram realizados. Um exemplo contendo oito grupos é exibido na Figura 11. Detalhes sobre esses experimentos podem ser encontrados em [Santos and Chaimowicz 2011a].

Uma série de simulações foi executada para avaliar a eficiência e a escalabilidade da abordagem. Para um número variado de robôs, o tempo de execução da simulação foi medido. Basicamente, para cada número fixo de agentes a mesma simulação foi executada 10 vezes e a média e o desvio padrão dos resultados foram calculados. Os agentes foram divididos igualmente em dois grupos e posicionados de acordo com uma distribuição normal em uma área específica. Os resultados são apresentados na Figura 12.



**Figura 9. Execução em simulação sem o algoritmo hierárquico.**



**Figura 10. Execução em simulação com o algoritmo hierárquico.**

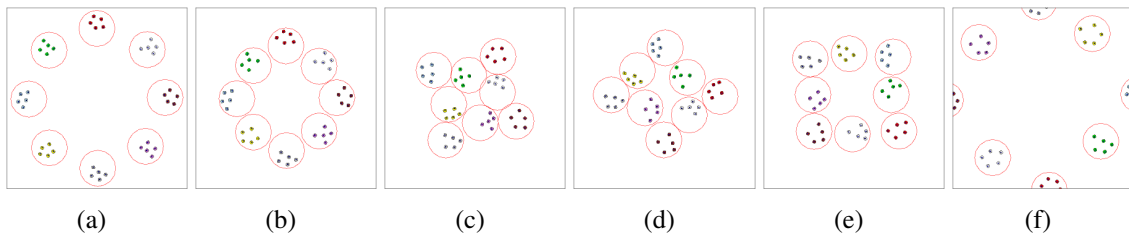
Como pode ser observado, o tempo de execução de ambos os algoritmos cresce monotonamente com o número de robôs. Entretanto, a abordagem proposta demonstra um desempenho melhor, visto que o tempo de execução foi significativamente reduzido. Por exemplo, para 100 robôs, o mecanismo de coordenação mostrou ganhos médios de 67%. Também, o método mostrou-se escalável devido a tendência linear da curva tracejada. Além do mais, o comportamento geral dos grupos tornou-se mais previsível devido ao aumento da estabilidade e coerência. Tal fato pode ser entendido através do pequeno desvio padrão apresentado nos resultados.

O experimento de planejamento de caminhos envolveu a aplicação do método *Probabilistic Roadmaps* [Kavraki et al. 1996] em conjunto com o algoritmo de busca  $A^*$ , como citado na Seção 3. Na Figura 13, uma amostra da navegação de um único grupo é exposta. Os experimentos mostraram que a abstração permite o uso direto de planejadores baseados em amostragem para efetuar a navegação de um enxame robótico de forma eficaz e eficiente. Para mais detalhes, consulte [Santos and Chaimowicz 2011b].

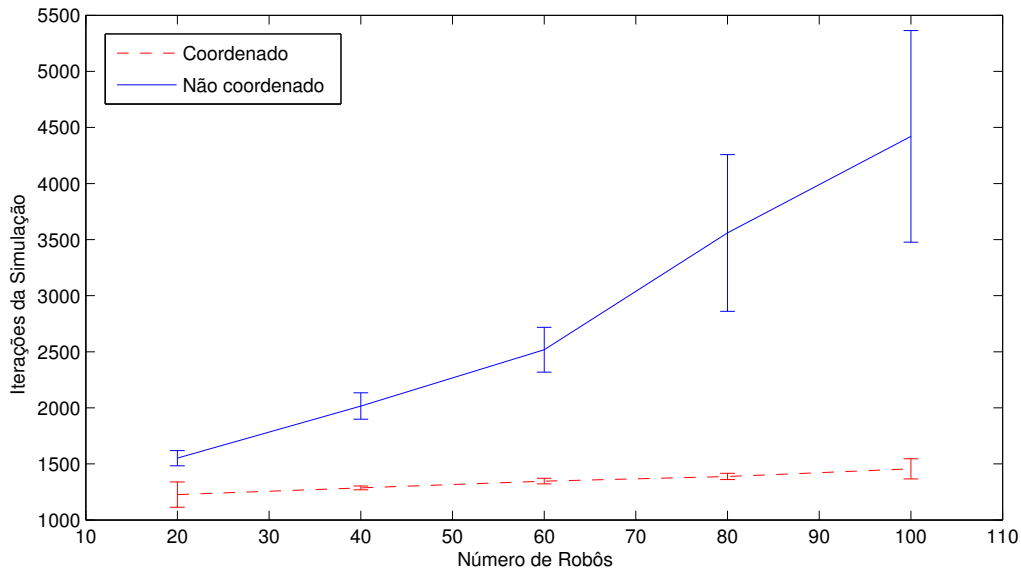
## 4.2. Robôs reais

Experimentos reais são importantes para demonstrar a viabilidade do algoritmo em situações nas quais as incertezas causadas por erros de sensoriamento e atuação podem ter grande influência sobre os resultados. Tais experimentos foram conduzidos em laboratório, utilizando-se um arcabouço de experimentação para exames robóticos desenvolvido no Laboratório de Visão e Robótica (VeRLab) [Garcia et al. 2007]. O arcabouço é composto por doze robôs *e-puck* (Figura 6), que comunicam-se remotamente com uma máquina hospedeira do sistema *Player*, e um conjunto de câmeras com visão aérea do ambiente, que localizam marcros fiduciais acoplados a cada robô com o objetivo de estimar a sua respectiva configuração.

Como o *e-puck* não foi projetado para atingir velocidades altas, por simplicidade, um modelo cinemático  $\dot{\mathbf{q}}_i = \mathbf{u}_i$  foi utilizado em vez do modelo dinâmico proposto. Em conformidade, o termo dissipativo de (4) foi removido. Também, como os seus sensores



**Figura 11. Execução do algoritmo hierárquico utilizando-se oito grupos.**



**Figura 12. Tempo de execução dos algoritmos.**

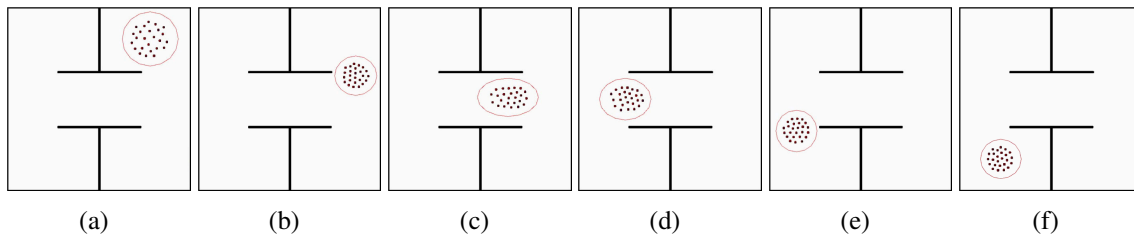
infravermelho possuem um alcance pequeno, um sensor virtual foi implementado com base no sistema de localização citado.

Os resultados do experimento de agrupamento são mostrados na Figura 14, enquanto as Figuras 15 e 16 apresentam as execuções do algoritmo de desvio para dois e quatro grupos, respectivamente. Tais experimentos indicam que o algoritmo funciona bem para coordenar um enxame de robôs, permitindo-lhes navegar com segurança enquanto evitam congestionamentos com eficácia.

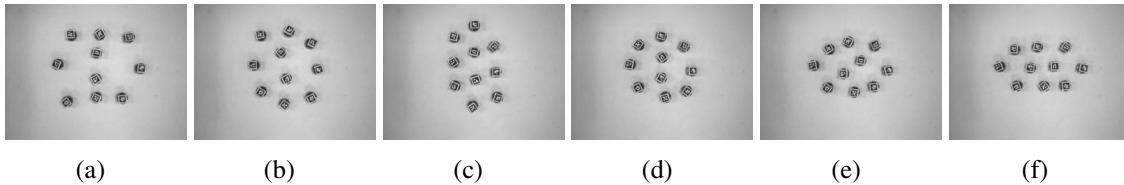
## 5. Conclusões e Trabalhos Futuros

Neste trabalho foi proposto um algoritmo para o controle da navegação de um enxame robótico que além de evitar congestionamentos de tráfego também permite o uso de planejadores de caminho baseados em amostragem. O trabalho juntou dois tópicos distintos que são amplamente estudados na área de robótica móvel: comportamentos de enxames e controle de tráfego.

O algoritmo utiliza uma abstração hierárquica formulada através da técnica dos campos potenciais artificiais. Dessa maneira, robôs individuais são controlados implicitamente através da variação dos parâmetros abstratos. Considerando-se as características geométricas da abstração, um algoritmo para controle de tráfego foi desenvolvido. Além disso, as mesmas características foram exploradas para permitir o planejamento de cami-



**Figura 13. Navegação de um grupo robótico em um ambiente com obstáculos.**



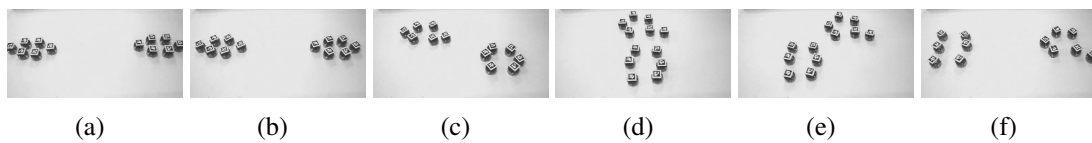
**Figura 14. Execução real do experimento de agrupamento.**

nhos para um número arbitrário de robôs de forma coesa e eficiente. Diversos experimentos foram realizados em ambientes virtuais e reais, cujos resultados demonstraram a viabilidade da abordagem.

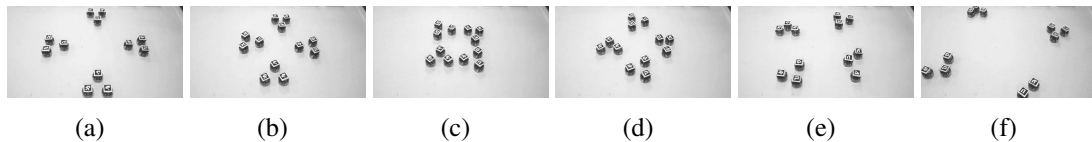
Como trabalho futuro, pretende-se explorar o uso de diferentes algoritmos de planejamento baseados em amostragem, além de unir os experimentos de planejamento com o desvio de grupos, permitindo a navegação de múltiplos grupos em ambientes com obstáculos. Explorando-se essas e outras possibilidades será possível desenvolver um sistema de navegação hierárquico para enxames robóticos mais completo e robusto.

## Referências

- Belta, C. and Kumar, V. (2004). Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875.
- Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.
- De Luca, A. and Oriolo, G. (1994). Local incremental planning for nonholonomic mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 104–110.
- Dresner, K. M. and Stone, P. (2005). Multiagent traffic management: an improved intersection control mechanism. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477.
- Garcia, R., Shiroma, P., Chaimowicz, L., and Campos, M. (2007). A framework for swarm localization. In *Proceedings of VIII SBAI - Brazilian Symposium on Intelligent Automation*.
- Gerkey, B. P., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323.



**Figura 15. Execução real para dois grupos com o algoritmo hierárquico.**



**Figura 16. Execução real para quatro grupos com o algoritmo hierárquico.**

- Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.
- Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 500–505.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K. Disponível em <http://planning.cs.uiuc.edu/>.
- Marcolino, L. and Chaimowicz, L. (2009). Traffic control for a swarm of robots: Avoiding group conflicts. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1949–1954.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65.
- Reif, J. H. and Wang, H. (1999). Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27(3):171–194.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, pages 25–34.
- Santos, V. and Chaimowicz, L. (2011a). Hierarchical congestion control for robotic swarms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4372–4377.
- Santos, V. and Chaimowicz, L. (2011b). Uso de hierarquias no controle de enxames robóticos. In *Anais do X Simp. Brasileiro de Automação Inteligente*, pages 557–562.
- Tan, K.-H. and Lewis, M. (1996). Virtual structures for high-precision cooperative mobile robotic control. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 132–139.
- Viswanath, K. and Krishna, M. (2007). Sensor network mediated multi robotic traffic control in indoor environments. In *Proc. of the Int. Conf. on Advanced Robotics*, pages 1220–1225.