

**UM ARCABOUÇO ESTOCÁSTICO PARA COLETA  
DE DADOS EM RSSF UTILIZANDO UM GRUPO DE  
ROBÔS MÓVEIS COOPERATIVOS**



MARCELO BORGHETTI SOARES

**UM ARCABOUÇO ESTOCÁSTICO PARA COLETA  
DE DADOS EM RSSF UTILIZANDO UM GRUPO DE  
ROBÔS MÓVEIS COOPERATIVOS**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Departamento de Ciência da Computação. como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

**ORIENTADOR: MARIO FERNANDO MONTENEGRO CAMPOS**

Belo Horizonte

15 de dezembro de 2011

© 2011, Marcelo Borghetti Soares.  
Todos os direitos reservados.

S676a Soares, Marcelo Borghetti  
Um arcabouço estocástico para coleta de dados em  
RSSF utilizando um grupo de robôs móveis cooperativos /  
Marcelo Borghetti Soares. — Belo Horizonte, 2011.  
xviii, 88 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas  
Gerais. Departamento de Ciência da Computação.  
Orientador: Mario Fernando Montenegro Campos.

1. Computação - Teses. 2. Robótica - Teses. 3. Redes de  
Sensores - Teses. I. Orientador. II. Título.

CDU 519.6\*82 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

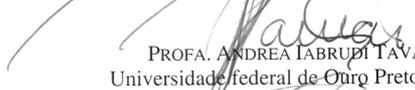
## FOLHA DE APROVAÇÃO

Um arcabouço estocástico para coleta de dados em RSSF utilizando um grupo  
de robôs móveis cooperativos

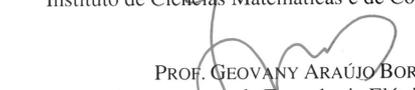
**MARCELO BORGHETTI SOARES**

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

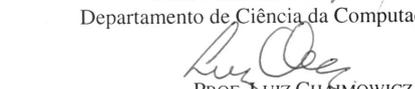
  
PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROFA. ANDRÉA IABRUDI TAVARES  
Universidade Federal de Ouro Preto - UFOP

  
PROF. DENIS FERNANDO WOLF  
Instituto de Ciências Matemáticas e de Computação - USP

  
PROF. GEOVANY ARAÚJO BORGES  
Departamento de Engenharia Elétrica - UnB

  
PROF. GERALDO ROBSON MATEUS  
Departamento de Ciência da Computação - UFMG

  
PROF. LUIZ CHAIMOWICZ  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 08 de abril de 2011.



# Agradecimentos

Mais do que um título, o doutorado é uma prova de vida. As lições aprendidas são muitas, fruto de uma jornada extensa, cansativa e incerta, mas recompensadora. Antes, a empolgação, a ânsia pela contribuição científica. Depois, a constatação de que existem muitas outras coisas igualmente importantes que ficam marcadas em nossa personalidade. Não é fácil descrever em palavras o amadurecimento adquirido. Mas, junto com a finalização dessa etapa, é indispensável deixar registrada a minha gratidão. E devo isso a muitas pessoas.

Ao professor Mario Campos, meu orientador, a quem agradeço a oportunidade de ser orientado e com quem aprendi uma das lições mais importantes: tudo pode ser alcançado, se houver dedicação, persistência e responsabilidade.

À minha família, principalmente à minha mãe, a quem devo tudo o que sou. A sua preocupação e carinho sempre foi sentida, mesmo à distância.

À Andréa Iabrudi, pelas valiosas contribuições antes e depois da minha defesa.

Ao Roberto, por estar presente em muitos momentos dessa jornada, como espectador e incentivador e muitas vezes acreditando mais em mim do que eu mesmo. Obrigado!

Ao pessoal do Verlab, pelo companheirismo em todos esses anos, em especial: ao Erickson pelas nossas acaloradas discussões sobre "Ace Ventura" e afins, ao Vilar, que sempre esteve disposto a ajudar quando acontecia algum pepino incompreensível, ao Luiz Cantoni que sentou ao meu lado algumas vezes e ajudou a resolver grandes problemas, ao Dimas e ao Víctor Costa que contribuíram significativamente nas bases do trabalho.

Ao pessoal da Ivision, em especial ao Luiz Fernando pela compreensão e flexibilidade com que tratou o meu doutorado. A Ivision foi muito importante nesse processo, certamente! Não menos importante foram as pessoas com quem trabalhei lado a lado e aprendi incontáveis coisas: Cádson, Eduardo Diego, Édilla, Lucas, Fernanda. Fiz bons amigos esses anos.

E, por fim, agradeço a todos os meus amigos, que direta ou indiretamente participaram do processo pelo qual passei, ouvindo minhas dúvidas, sugerindo caminhos, incentivando e visualizando o quão importante esse título representaria na minha vida.

Valeu gente!

# Resumo

Este trabalho tem por objetivo apresentar um arcabouço para cooperação entre robôs móveis em tarefas de coleta de dados, considerando-se incerteza de atuação e observação dos robôs, comunicação limitada e incerteza sobre a posição dos nós sensores. A metodologia consiste na criação de dois tipos de robôs: 1) Robôs Tarefa que realizam a visitação aos nós sensores e 2) Robôs Exploradores que exploram o ambiente. A cooperação entre os robôs é modelada por meio de Processos Descentralizados de Markov com Observação Parcial. A utilização desses robôs é investigada por meio da análise de várias métricas, tais como o Reforço Médio. Essa abordagem foi testada sobre vários cenários nos quais nós sensores foram depositados em posições desconhecidas. Experimentos foram realizados para demonstrar a validade e o funcionamento da metodologia proposta.



# Abstract

The aim of this work is to propose an approach to multi-robot cooperation in data collection tasks, considering actuation and observation errors, limited communication and uncertainty about sensor node position. The methodology consists in two types of robots: 1) Task Robots that perform the designed tasks and 2) Scout Robots that explore the environment. The cooperation among the robots is modeled with Decentralized Partially Observable Markov Decision Process. The use of these robots is investigated analyzing several metrics, such as Expected Reward. This approach has been applied to several scenarios with sensor nodes deployed in unknown positions. Experiments were conducted to validate the proposed methodology.



# Sumário

<b>Agradecimentos</b>	<b>vii</b>
<b>Resumo</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xv</b>
Geral . . . . .	xvii
Problema do Roteamento de Veículos . . . . .	xvii
Tomada de Decisão . . . . .	xviii
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Definição do problema . . . . .	5
1.2.1 Mobilidade no ambiente . . . . .	8
1.2.2 Comunicação . . . . .	10
1.3 Escopo da abordagem . . . . .	10
1.4 Objetivos . . . . .	11
1.5 Contribuições . . . . .	12
1.6 Organização do texto . . . . .	12
<b>2 Trabalhos Relacionados</b>	<b>13</b>
2.1 Mobilidade em RSSF . . . . .	13
2.1.1 Relação entre os trabalhos e esta tese . . . . .	14
2.1.2 Solução de problemas de otimização para controle de entidades móveis . . . . .	16
2.1.3 Integração e comunicação entre entidades móveis e nós sensores estáticos . . . . .	19

<b>3</b>	<b>Fundamentos</b>	<b>25</b>
3.1	Tomada de Decisão em Cenários Dinâmicos e Estocásticos . . . . .	25
3.1.1	Redes Bayesianas Dinâmicas . . . . .	27
3.1.2	Atuação e observação . . . . .	29
3.1.3	Tratamento de Erros de Atuação . . . . .	30
3.1.4	Tratamento de Erros de Observação . . . . .	32
3.1.5	Tratamento de Erros para Times de Robôs . . . . .	34
3.2	Hierarquia de Problemas de Decisão . . . . .	35
<b>4</b>	<b>Metodologia</b>	<b>37</b>
4.1	Cooperação em RSSF utilizando robôs móveis . . . . .	37
4.2	Modelo . . . . .	38
4.3	Abordagem Heurística Adotada . . . . .	40
4.3.1	Robôs Tarefa . . . . .	41
4.3.2	Robôs Exploradores . . . . .	42
4.4	Comunicação . . . . .	44
4.5	Gerador de Planos Locais . . . . .	45
4.6	Estimador de Posição Local . . . . .	48
4.7	Tomador de Decisão . . . . .	50
4.7.1	Tomada de Decisão em Resposta a Erros de Atuação . . . . .	50
4.7.2	Tomada de Decisão em Resposta a Erros de Observação . . . . .	53
4.7.3	Tomada de Decisão Descentralizada . . . . .	55
<b>5</b>	<b>Resultados Experimentais</b>	<b>59</b>
5.1	Metodologia de Testes . . . . .	61
5.2	Comportamento do Sistema com Erros de Observação e Atuação . . . . .	63
5.3	Comunicação para Cooperação . . . . .	66
5.4	Avaliação em Cenários Diversos . . . . .	69
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>75</b>
6.1	Conclusões . . . . .	75
6.2	Trabalhos Futuros . . . . .	77
	<b>Referências Bibliográficas</b>	<b>79</b>

# Lista de Figuras

1.1	(a) Ambiente discreto com 2 robôs holonômicos, $R_1$ e $R_2$ e com 2 nós sensores, $s_1$ e $s_2$ . (b) Os robôs navegam pelo ambiente com o intuito de visitar os nós sensores. Observe que os robôs possuem uma estimativa sobre as suas posições, $s'_1$ e $s'_2$ . (c) Os robôs podem estimar as novas posições dos nós sensores quando a comunicação é estabelecida, realizando um mapeamento do ambiente. . . . .	4
1.2	Procedimento de Trilateração. (a) Robô $R_1$ navega em ambiente com um nó sensor $s_1$ . (b) $R_1$ entra no raio de comunicação de $s_1$ . O nó sensor $s_1$ pode estar em qualquer posição do raio de comunicação. (c) Com uma segunda leitura a posição do nó sensor é reduzida para duas possibilidades, dadas pela intersecção de duas circunferências. (d) Finalmente, a uma terceira leitura estabelece a posição do nó sensor, dada pela intersecção de três circunferências. . . . .	7
1.3	Eixos de incerteza em relação aos nós sensores $S$ , $\hat{S}$ , $\bar{S}$ e em relação ao robô $R$ , $\hat{R}$ , $\bar{R}$ . . . . .	11
3.1	Exemplo de Rede Bayesiana Dinâmica. . . . .	27
3.2	Esquema e hierarquia dos Problemas de Decisão de <i>Markov</i> . . . . .	36
4.1	Esquema de um Robô Tarefa. . . . .	41
4.2	Esquema de um Robô Explorador. . . . .	42
4.3	Troca de informações entre os robôs. . . . .	43
4.4	Campos das mensagens de “Presença” e “Atualização de Posição”. . . . .	45
4.5	Estimação da posição dos nós sensores. . . . .	46
4.6	Trajetórias em um ambiente com 60 nós sensores . . . . .	48
4.7	Estimação de posição local. . . . .	49
4.8	Ambiente mostrando a política de atuação obtida por um MDP . . . . .	50
4.9	Política de atuação para MDP. . . . .	52

4.10	Decisão entre navegação e comunicação. . . . .	53
4.11	Observação parcial do ambiente. . . . .	54
4.12	Estimação da célula de encontro de um Robô Tarefa feito por um Robô Explorador. . . . .	57
5.1	Ambiente de testes. . . . .	59
5.2	Ambiente de simulação. . . . .	60
5.3	(a) Reforço médio e (b) Erro médio de Posicionamento dos Sensores em função da probabilidade de erro de observação. . . . .	64
5.4	Comparação entre o RM e o EMPS obtidos com a variação do erro de observação. . . . .	65
5.5	(a) Reforço médio e (b) Erro Médio de Posicionamento dos Sensores em função da probabilidade de erro de atuação. . . . .	66
5.6	Comparação entre o RM e o EMPS obtidos com a variação do erro de atuação. . . . .	67
5.7	Variação do Reforço Médio em função do Custo de Comunicação em função de (a) probabilidade de erro de observação (o erro de atuação foi de 20%, ou seja $p(x x, u) = 0.8$ ) e (b) probabilidade de erro de atuação (sem erro de observação, ou seja $p(o e) = 1$ ). . . . .	68
5.8	(a-b) Resultados mostrando o desempenho do <i>Método Rendezvous</i> comparado ao <i>Método 1-Tarefa</i> para Reforço Médio and Erro Médio da Posição dos Sensores. (c) Reforço Médio em função do Número de Robôs Exploradores. . . . .	70
5.9	(a-b) Resultados mostrando o desempenho do <i>Método Rendezvous</i> comparado ao <i>Método 1-Tarefa</i> para Reforço Médio and Erro Médio da Posição dos Sensores. (c) Mapa do ambiente onde as simulações foram realizadas. . . . .	71
5.10	Cenários 1-6 utilizados na Seção 5.4. . . . .	72
5.11	Cenários 7-12 utilizados na Seção 5.4. . . . .	73

# Lista de Símbolos

## Geral

$\mathcal{R}$	Conjunto de robôs.
$\mathcal{S}$	Conjunto de nós sensores.
$R_i$	$i$ -ésimo robô.
$s_i$	$i$ -ésimo nó sensor.
$n$	Número de nós sensores.
$m$	Número de robôs do time.
$t$	Ciclo atual.
$i, j, k$	Índices.
$v_{R_i}$	Velocidade do robô $R_i$ .
$g_i$	Taxa de aquisição de dados do $i$ -ésimo nó sensor.
$h_i$	Taxa de transferência de dados do $i$ -ésimo nó sensor.
$r_i$	Raio de comunicação do $i$ -ésimo robô.
$d_i^t$	Variação da quantidade de dados no $i$ -ésimo nó sensor. Quando o nó sensor não está no raio de comunicação de nenhum robô, $d_i^t = g_i$ .
RSSI	Received Signal Strength Indication.
RSSF	Rede de Sensores sem Fio.
RM	Reforço Médio.
EMPS	Erro Médio de Posicionamento dos Sensores.

## Problema do Roteamento de Veículos

PRV	Problema do Roteamento de Veículos.
PCV	Problema do Caixeiro Viajante.
$f_{ijk}$	Variável que assume o valor 1 se, e somente se, o trecho que liga os nós $i$ e $j$ pertencer à rota do veículo $k$
$Q$	Capacidade máxima do robô.
$d_i$	Demanda do $i$ -ésimo robô.
$D_i$	Capacidade de armazenamento do sensor $s_i$ .
$w_{ij}$	Custo associado à transição entre os nós $s_i$ e $s_j$ .
$W$	Matriz de Custos.

## Tomada de Decisão

MDP	Markov Decision Process (Processos de Decisão de Markov).
POMDP	Partially Observed Markov Decision Process (Processos de Decisão de Markov com Observação Parcial).
DEC-POMDP	Decentralized Partially Observed Markov Decision Process (Processos de Decisão de Markov com Observação Parcial).
$X$	Conjunto de estados do sistema.
$x_i$	$i$ -ésimo estado do ambiente.
$U$	Conjunto de ações que podem ser realizadas pelo sistema.
$u_i$	$i$ -ésima ação.
$\delta$	Função de transição probabilística $X X \times U \rightarrow [0, 1]$ .
$\gamma$	Função de reforço: $X \times U \leftarrow R$ aplicada ao robô .
$X_i^{[t]}$	Variável de estado $i$ não diretamente observável no ciclo $t$ .
$O_{[t]}^i$	Variável de estado $i$ observável no ciclo $t$ .
$V_i(x)$	Utilidade acumulada associada ao estado $x_i$ ( <i>payoff acumulado</i> ).
$\pi_i(x)$	Política de atuação executada pelo robô $R_i$ no estado $x$ .
$\beta$	Distribuição de probabilidades sobre os estados.
$\varphi$	Função de probabilidades de observações $E X \times U \rightarrow [0, 1]$ .
$a$	Fator de recompensa utilizado na função de reforço.
$b$	Fator de penalização utilizado na função de reforço.
$c$	Parâmetro de comunicação utilizado na função de reforço.
$\alpha$	Fator de desconto.
$C_m$	Custo de Movimentação.
$C_c$	Custo de comunicação.
$L_k$	Seqüência de estados entre os estados $x_i$ e $x_j$ .

# Capítulo 1

## Introdução

Este trabalho tem por objetivo apresentar um arcabouço para cooperação entre robôs móveis em redes de nós sensores sem fio (RSSF), considerando-se incerteza na atuação, na observação dos robôs e incerteza sobre a posição dos nós sensores. A motivação para este trabalho deve-se ao crescente interesse no estudo de RSSF, fomentado principalmente por elas proverem, de maneira simples, soluções para problemas relacionados ao monitoramento e sensoriamento de grandezas em ambientes diversos.

O emprego de robôs móveis nesses ambientes é um procedimento relativamente novo, apresentando questões interessantes a serem respondidas, por exemplo, relacionadas à cooperação entre elementos móveis e estáticos em um ambiente não controlado. Esse tipo de abordagem, que contrasta com a abordagem estática das RSSF, foi alavancado pela busca de soluções para um problema clássico: gasto de energia das RSSF. Em redes estáticas, o roteamento constitui a principal estratégia empregada para que os dados coletados pelos sensores sejam transmitidos para uma unidade com maior capacidade de processamento e armazenamento, tipicamente denominada “nó sorvedouro”. Nesse contexto, a mobilidade surge como uma alternativa interessante à realização total ou parcial da coleta dos dados. Há uma série de vantagens que essa estratégia apresenta em relação ao *roteamento multi-saltos*<sup>1</sup> tradicional:

1. É vantajosa em ambientes muito acidentados ou ruidosos nos quais exista dificuldade de propagação do sinal [Wang et al., 2005a] ou quando a rede

---

<sup>1</sup>No roteamento multi-saltos o envio de pacotes de dados acontece em múltiplas etapas, partindo do emissor até o nó receptor, passando pelos nós sensores vizinhos que encontram-se no caminho entre ambos. É diferente da transmissão em um salto, onde os dados são enviados diretamente do emissor para o receptor.

apresenta-se desconectada devido à falha de sensores. Esse tipo de ambiente também representa um desafio para a robótica e muitos estudos vem sendo realizados em diversos algoritmos de exploração [Thrun et al., 2005].

2. Tarefas como fusão de dados [Wu et al., 2004], monitoramento do ambiente [Li et al., 2003; Corke et al., 2004] ou recalibração de nós sensores podem ser facilmente adaptadas a partir de algoritmos de navegação na rede por meio de robôs móveis;
3. Existe a possibilidade de construir uma série de arquiteturas e métodos cooperativos entre redes de nós sensores e robôs móveis: por exemplo, robôs móveis podem utilizar-se da capacidade perceptiva (e onipresente) dos sensores estáticos e os sensores estáticos podem utilizar-se da mobilidade dos robôs móveis (por exemplo, não necessitando enviar dados a longas distâncias)[Batalin & Sukhatme, 2004; Batalin et al., 2004];
4. O tempo de vida da rede pode ser significativamente aumentado [Somasundara et al., 2006], uma vez que a diminuição ou ausência de roteamento evita que os sensores desperdicem grandes quantidades de dados, aproveitando o tempo de vida em tarefas de monitoramento do ambiente.

## 1.1 Motivação

Conforme mencionado anteriormente, a interação entre robôs móveis e redes de nós sensores possibilita que ambos tirem proveito das respectivas funcionalidades. Embora não se possa considerar como regra, um robô móvel atuando nesse tipo de ambiente precisará implementar alternativas apropriadas para problemas de navegação, exploração e planejamento de trajetórias. Da mesma forma, a cooperação entre robôs móveis ou entre robôs móveis e sensores estáticos pode ser empregada, na medida em que o conhecimento adquirido localmente poderá ser utilizado para beneficiar todos os elementos do sistema.

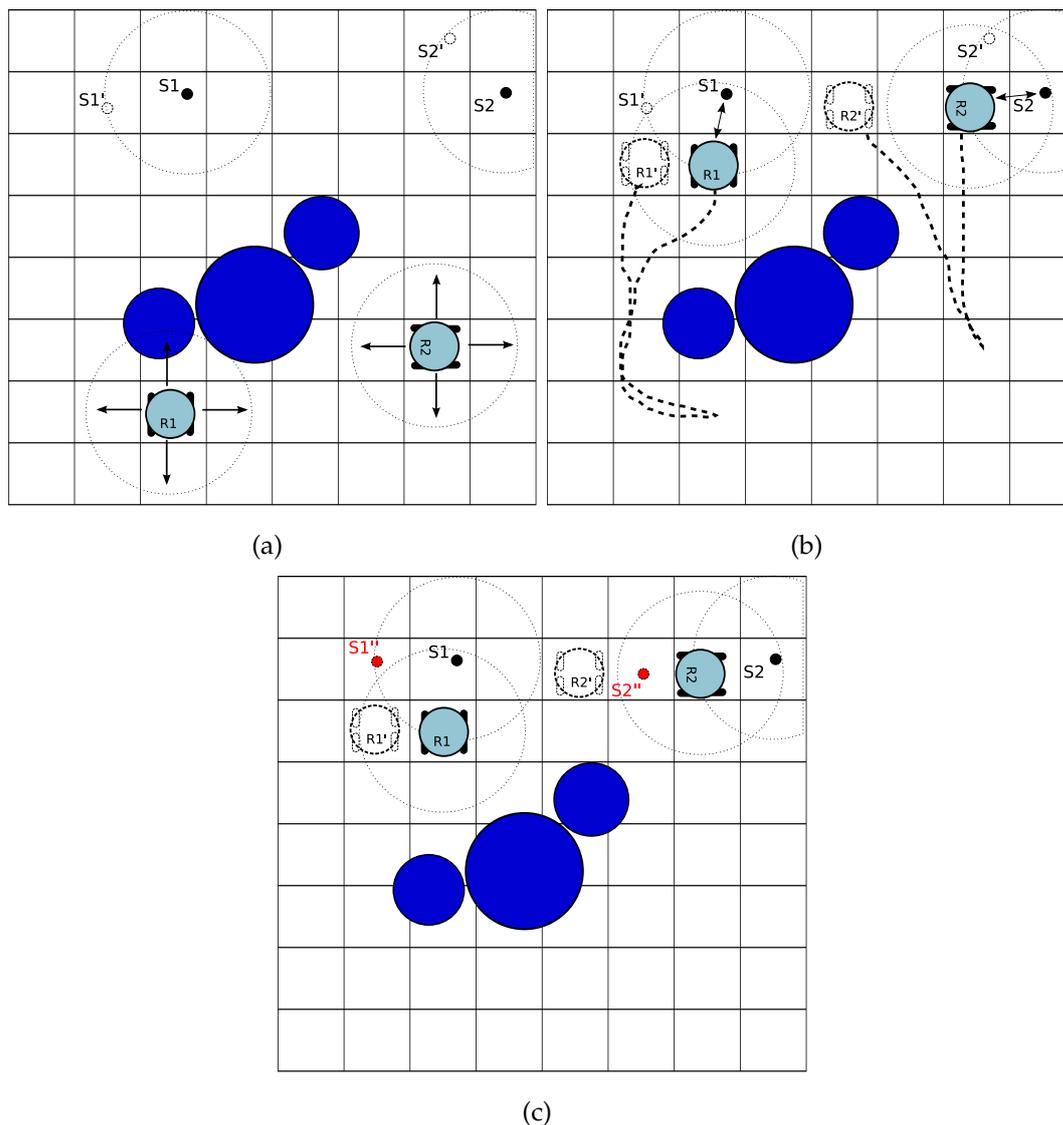
Serão mostradas algumas aplicações nas quais a rede de nós sensores e os robôs móveis podem ser utilizados conjuntamente. Essas aplicações relacionam-se fortemente com o tema desta tese, uma vez que a informação coletada pelos nós sensores estáticos é transmitida para o robô móvel (ou robôs móveis) e representa o aspecto mais importante da interação entre ambas as partes. Em algumas ocasiões, os robôs móveis são empregados como *sorvedouros móveis*, sendo designados unicamente para a tarefa de coleta dos dados armazenados nos nós sensores estáticos. A

realização eficiente das tarefas depende fundamentalmente de como o robô móvel “visualiza” a rede de nós sensores, com todos os aspectos de incerteza associados à percepção e atuação no ambiente e de como, diante dessa incerteza, o robô escolhe a melhor ação a ser realizada. Dentre as possíveis aplicações deste trabalho pode-se listar:

**Monitoramento do ambiente.** Os nós sensores podem ser depositados, por exemplo, em uma floresta ou próximo a um vulcão e coletar informações como temperatura, umidade, qualidade do ar, etc. Tais medições podem ser úteis na previsão do tempo, detecção de incêndios, abalos sísmicos, erupções vulcânicas, etc. Nesse contexto, é importante que sejam consideradas alternativas interessantes para navegação em meio à rede. Dentre essas alternativas, pode-se citar a criação de algoritmos diferenciados para geração das rotas a serem seguidas pelos robôs. Esses métodos têm ainda a vantagem de não se restringirem a um único problema, podendo ser empregados com as devidas modificações em tarefas de deposição, recalibração de sensores ou recarregamento de baterias dos nós. É interessante perceber que a utilização conjunta de robôs e nós sensores estáticos pode ser mais interessante do que empregar somente robôs móveis para sensoriamento do ambiente. Nós sensores podem coletar dados do ambiente constantemente em todos os locais de deposição (onipresença), ao passo que um robô não pode estar em todas as posições de interesse ao mesmo tempo. Além disso, já que a comunicação entre os nós diminuiria consideravelmente, os nós sensores poderiam ser depositados em posições afastadas umas das outras, caso não haja restrição na amostragem da grandeza sensorizada, de forma a abranger uma região maior.

**Exploração.** A exploração de ambientes remotos pode criar uma grande gama de aplicações que utilizam mobilidade em RSSF. Em ambientes totalmente desconhecidos e inóspitos, o emprego de nós sensores espalhados pelo ambiente pode ser valioso, pois a partir das informações coletadas por esses nós, um robô móvel poderia navegar de forma mais eficiente. Os nós sensores poderiam informar ao robô sobre a presença de obstáculos ou a ocorrência de eventos de interesse na região. Da mesma forma, a deposição dos nós poderia ser feita pelos próprios robôs, levando-se em consideração, por exemplo, regiões do ambiente ainda não visitadas. Esse tipo de aplicação ganhou evidência, nos últimos anos, devido à intensificação da exploração interplanetária por meio de robôs móveis [NASA, 2011].

De uma maneira geral, pode-se dizer que as aplicações citadas anteriormente levantam questões e desafios específicos que apresentam relação direta com o grau



**Figura 1.1.** (a) Ambiente discreto com 2 robôs holonômicos,  $R_1$  e  $R_2$  e com 2 nós sensores,  $s_1$  e  $s_2$ . (b) Os robôs navegam pelo ambiente com o intuito de visitar os nós sensores. Observe que os robôs possuem uma estimativa sobre as suas posições,  $s'_1$  e  $s'_2$ . (c) Os robôs podem estimar as novas posições dos nós sensores quando a comunicação é estabelecida, realizando um mapeamento do ambiente.

de interação entre a rede de nós sensores e os robôs móveis. Essa integração terá influência sobre a eficiência alcançada na solução dos problemas. Tais aspectos serão abordados nas seções seguintes.

## 1.2 Definição do problema

Na Figura 1.1(a) vê-se dois robôs,  $R_1$  e  $R_2$ , em um ambiente discreto com obstáculos. Esses robôs podem se mover nas direções norte, sul, leste e oeste. Além dos robôs existem dois nós sensores,  $s_1$  e  $s_2$ . No início, ambos os robôs possuem estimativas sobre a posição desses nós sensores, representadas por  $s'_1$  e  $s'_2$ . Esses nós sensores estão previamente alocados para os robôs, de forma que  $s_1$  deverá ser visitado pelo robô  $R_1$  e  $s_2$  deverá ser visitado pelo robô  $R_2$ . Como os robôs possuem incertezas de atuação e observação, os seus estados são representados por  $R'_1$  e  $R'_2$  (robôs tracejados). Quando os robôs comunicam com os nós sensores é possível estimar suas posições, mostradas como  $s''_1$  e  $s''_2$  em (c).

Os aspectos fundamentais para a definição do problema tratado nesta tese são descritos a seguir:

- Conforme pode ser visto na Figura 1.1 o ambiente é discreto, dinâmico e com mapa conhecido. O dinamismo do ambiente é caracterizado pela mobilidade dos robôs, pela mudança na quantidade de dados armazenados em cada nó sensor. O mapa, porém, é fixo e conhecido previamente.

$$\text{Ambiente} \left\{ \begin{array}{l} \text{Discreto } (N \times N); \\ \text{Dinâmico (robôs mudam de posição);} \\ \text{Mapa conhecido.} \end{array} \right.$$

- Os robôs são holonômicos com velocidade linear constante e  $\omega = 0$ . Além disso a sua ação está sujeita erros e a observação é também imperfeita. Os robôs tem total conhecimento de suas posições iniciais no ambiente.

$$\text{Robôs} \left\{ \begin{array}{l} \text{Robôs holonômicos;} \\ \text{Posição inicial conhecida;} \\ \text{Incerteza de atuação;} \\ \text{Incerteza de observação;} \\ \text{Velocidade linear } v \text{ constante e } \omega = 0. \end{array} \right.$$

- Os nós sensores possuem posição incerta no ambiente. Todos os robôs, inicialmente a mesma estimativa sobre a posição desses sensores. Além disso, os nós sensores são previamente alocados para os robôs.

$$\text{Nós Sensores} \left\{ \begin{array}{l} \text{Número de sensores conhecido;} \\ \text{Posição incerta;} \\ \text{Taxa de crescimento dos dados fixa;} \\ \text{Previamente alocados para cada robô.} \end{array} \right.$$

- Comunicação entre os robôs e entre os robôs e os nós sensores é perfeita, porém é limitada por um raio de alcance. Técnicas como TDoA [Savvides et al., 2001]<sup>2</sup> ou RSSI [Whitehouse & Culler, 2002] em conjunto com trilateração<sup>3</sup> são abordagens para implementar essa estimação. O procedimento de trilateração está descrito na Figura 1.2. Nesta tese, assume-se que a estimação da distância entre um robô e um nó sensor é perfeita. Entretanto, a precisão da estimativa sobre o posicionamento do nó sensor dependerá da localização do robô (que é incerta). O procedimento de trilateração é simulado por meio da geração dessa estimativa usando uma distribuição de probabilidades com média na posição real do sensor e desvio padrão igual à distância entre o nó sensor e o robô.

$$\text{Comunicação} \left\{ \begin{array}{l} \text{Comunicação perfeita, porém limitada;} \\ \text{Estimação de posição dos nós sensores.} \end{array} \right.$$

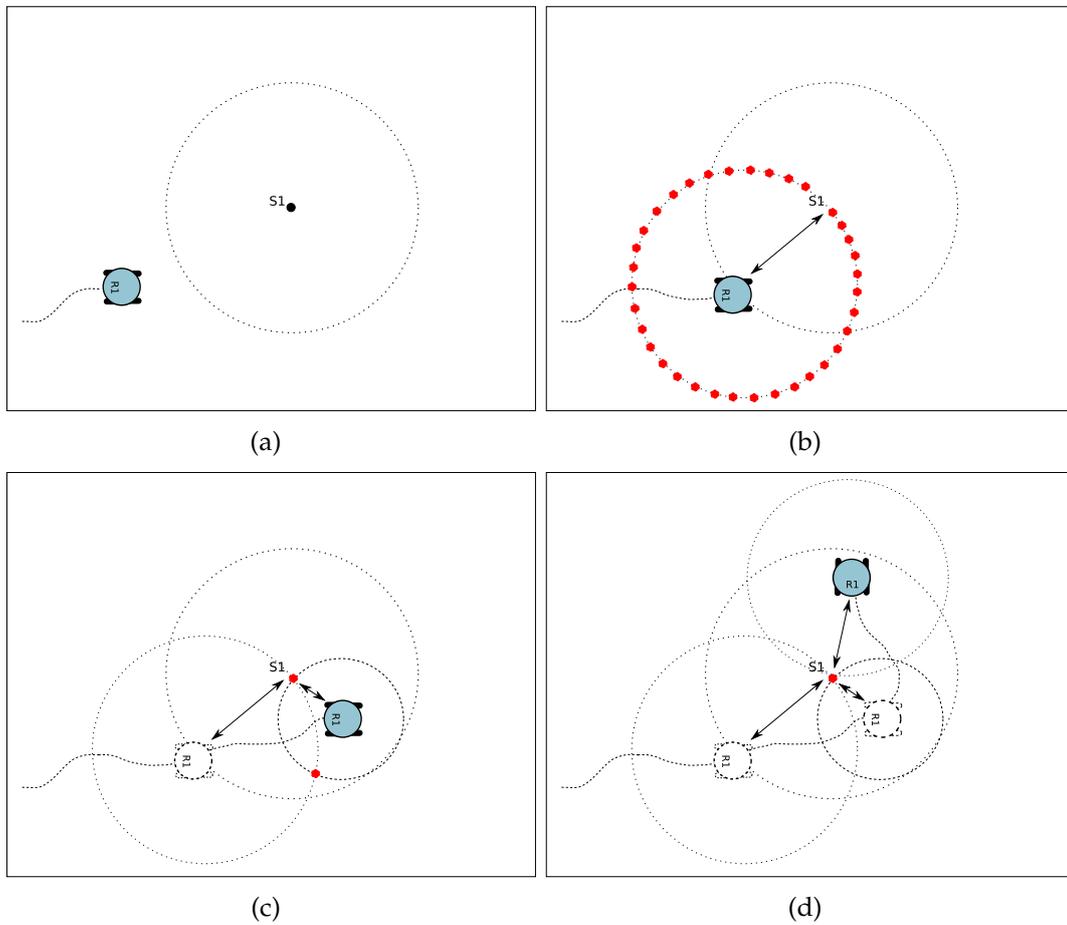
O problema a ser tratado nesta tese pode ser formalmente definido da seguinte maneira:

**Definição 1** *Determinar uma estratégia para maximizar a coleta de dados em um ambiente composto por um conjunto  $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$  de  $n$  nós sensores com posição incerta previamente alocados para um conjunto  $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$  de  $m$  robôs móveis com posição inicial conhecida, atuação e observação estocásticas e comunicação limitada.*

A Definição 1 diz que o ambiente no qual as tarefas serão realizadas é estocástico, uma vez que a atuação e a observação dos robôs é imperfeita. Assim, a realização de tarefas exigirá a estimação de variáveis de estado dos robôs, tanto

<sup>2</sup>No TDoA, a diferença entre o tempo de emissão e de recepção de um pacote fornece uma estimativa da distância entre o emissor e o receptor por meio da equação:  $d = c(t_f - t_i)$ , onde  $d$  é a distância,  $c$  é uma constante representando a velocidade da luz e  $(t_f - t_i)$  representa a diferença entre os tempos de partida e chegada de um pacote.

<sup>3</sup>A distância estimada é utilizada como entrada para um procedimento de trilateração que permite obter a posição do nó sensor (com incerteza). Essa distância representa o raio de uma circunferência com centro na posição do robô. Com duas circunferências, é possível computar dois possíveis pontos como estimativa para a posição do sensor. Com uma terceira circunferência, um desses pontos é selecionado e a estimativa pode ser melhorada daí em diante



**Figura 1.2.** Procedimento de Trilateração. (a) Robô  $R_1$  navega em ambiente com um nó sensor  $s_1$ . (b)  $R_1$  entra no raio de comunicação de  $s_1$ . O nó sensor  $s_1$  pode estar em qualquer posição do raio de comunicação. (c) Com uma segunda leitura a posição do nó sensor é reduzida para duas possibilidades, dadas pela intersecção de duas circunferências. (d) Finalmente, a uma terceira leitura estabelece a posição do nó sensor, dada pela intersecção de três circunferências.

localmente quanto em relação ao grupo. De maneira semelhante, é necessário especificar algum procedimento para reduzir a incerteza de posicionamento dos nós sensores, uma vez as tarefas a serem realizadas necessitam que os robôs alcancem suas posições.

Outro aspecto importante a ser comentado, que está fora do escopo desta tese, diz respeito à alocação dos  $n$  sensores para os  $m$  robôs. Diversas abordagens de alocação de tarefas, que podem ser utilizadas no processo de alocação dos nós sensores, tais como técnicas de leilão, atribuição dinâmica de tarefas, etc, são revisadas e comparadas em [Gerkey & Mataric, 2003]. Nesta tese não será tratado o processo de descoberta e alocação dos nós sensores aos robôs do time. Parte-se do princípio

que os nós sensores foram alocados entre os Robôs Tarefa que, desta forma, têm a função de visitá-los.

### 1.2.1 Mobilidade no ambiente

A capacidade de percepção dos robôs e dos nós sensores, bem como a troca de informações entre os elementos do sistema são fundamentais para o sucesso de uma missão. Um aspecto importante para que esses processos cooperativos sejam eficientemente executados é o grau de mobilidade empregado nesses ambientes. De um modo geral, pode-se definir uma taxonomia com base nessa mobilidade:

- **RSSF sem mobilidade:** os nós sensores não possuem capacidade de locomoção. A transferência dos dados coletados pelos nós sensores é realizada por meio de roteamento multi-saltos.
- **RSSF com mobilidade parcial:** a rede possui alguns nós sensores estáticos e outros com capacidade de locomoção <sup>4</sup>. O roteamento multi-saltos pode ser reduzido, uma vez que os nós móveis podem assumir a função de nós sorvedouros.
- **RSSF com mobilidade total:** todos os nós da rede têm capacidade de locomoção.

O grau de mobilidade a ser disponibilizado na rede envolve vários fatores, muitas vezes conflitantes entre si. Alguns desses fatores são:

1. **Consumo de energia das RSSF:** a utilização de nós móveis em RSSF é vantajosa, conforme apresentado no trabalho de Somasundara et al. [2006], onde robôs móveis foram empregados como nós sorvedouros. O roteamento estático foi mantido, porém sua redução, devido ao emprego de nós sensores móveis, propiciou um aumento significativo no tempo de vida da rede. Esses resultados apontam para a utilização de nós sensores móveis como um método para diminuição dos gastos em energia dos nós sensores.
2. **Custos envolvidos:** nós sensores são dispositivos relativamente baratos. A sua utilização em larga escala é, portanto, uma alternativa viável para diversas aplicações. Porém, como são dispositivos com baixa capacidade de processamento e armazenamento, precisam transferir seus dados com muita

---

<sup>4</sup>Os nós sensores com capacidade de locomoção aqui citados podem ser considerados, basicamente, nós sensores tradicionais conectados a algum robô móvel.

freqüência, fazendo com que seu “tempo de vida” diminua rapidamente. Dessa forma, a substituição de nós sensores inativos ou defeituosos torna-se um problema logístico de grande magnitude. Conforme discutido anteriormente, a mobilidade minimiza esse problema uma vez que o tempo de vida útil dos nós sensores estáticos tende a aumentar com a utilização de sorvedouros móveis. Em contrapartida, sorvedouros móveis geralmente são instalados sobre robôs, de modo que questões associadas ao custo desses equipamentos e custos de manutenção não podem ser desprezadas. Logo, a análise desses custos também é importante para determinação da razão entre o número de sensores móveis e estáticos a serem utilizados. Em uma situação ideal, essa razão deverá ser determinada de forma a garantir uma razão custo/benefício adequada ao propósito da aplicação.

3. **Aplicação alvo:** os nós sensores móveis podem ser empregados com o objetivo de solucionar outros problemas não diretamente relacionados ao consumo de energia. Problemas de cobertura e conectividade podem ser atenuados de acordo com a mobilidade disponível na rede [Wang et al., 2006, 2003; Howard et al., 2003; Butler & Rus, 2003]. Em ambientes nos quais eventos a serem monitorados acontecem de forma esporádica, a utilização de muitos nós sensores espalhados no ambiente pode não ser eficiente. Nesse caso, a melhor opção poderia ser a utilização de uma RSSF com mobilidade total ou parcial. Deve-se considerar, porém, que o custo de implantação de uma rede com mobilidade total pode ser proibitivo, uma vez que envolve o custo e manutenção de veículos ou robôs, o que pode tornar a mobilidade parcial a melhor alternativa. Complementarmente, a mobilidade parcial pode ser mais interessante em sistemas nos quais exista a necessidade de onipresença nos pontos críticos de um ambiente. Esse é o caso, por exemplo, de sistemas de rastreamento de intrusos. Se um robô móvel não estiver presente em algum ponto crítico no momento em que ocorrer alguma invasão, esse fato poderá ser percebido por nós sensores estáticos ali posicionados. Redes desconectadas podem também utilizar robôs móveis com a função de criar uma “ponte” para os dados, conforme mostrado no trabalho de Wang et al. [2005a], ou para deposição de sensores estáticos que restabeleçam a conectividade da rede. A razão entre a quantidade de nós sensores móveis e estáticos, nesse caso, tende a se manter pequena, uma vez que os robôs móveis são utilizados basicamente em tarefas de manutenção da rede original.

A definição de métricas para análise e decisão de qual modelo de mobilidade adotar está fora do escopo desta tese. O modelo de mobilidade dos robôs está definido abaixo:

**Definição 2** *O conjunto  $\mathcal{S}$  é constituído de  $n$  nós sensores estáticos. O conjunto  $\mathcal{R}$  possui  $m$  robôs que se movimentam com velocidade linear constante dentro do intervalo  $[v_{min}, v_{max}]$  e  $\omega = 0$ .*

### 1.2.2 Comunicação

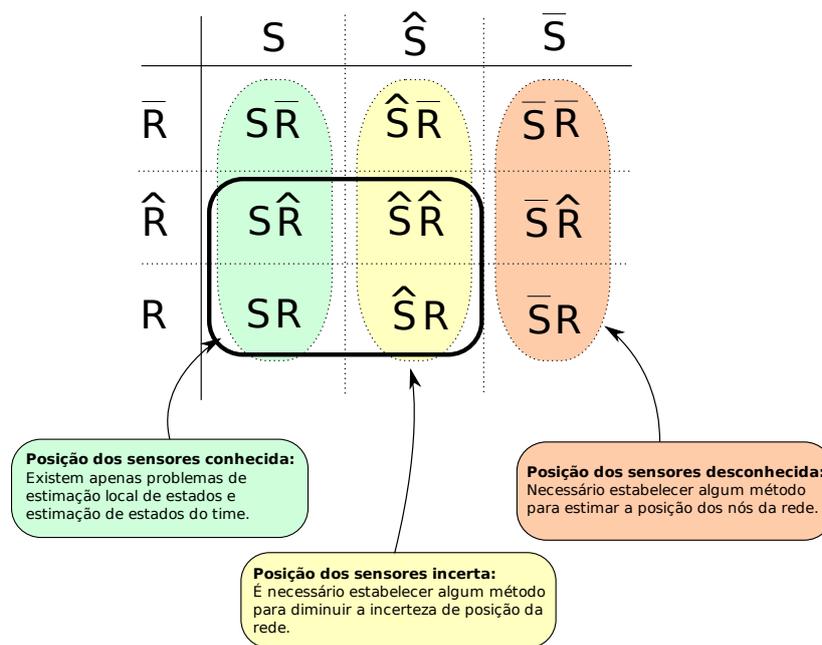
A comunicação entre robôs é muitas vezes aplicada nesses cenários sob a forma de um mecanismo para coordenação. Quando a incerteza em relação às observações e atuações realizadas é muito freqüente, a queda de eficiência na execução das tarefas pode ser significativa. Dessa forma, a comunicação é utilizada para transmissão de variáveis de estados e de compartilhamento de informação. Porém, dois aspectos precisam ser considerados: i) existe um custo associado à comunicação que pode inviabilizar a coordenação e ii) o processo de comunicação também não está isento de falhas, que podem ocorrer devido a desconexões e recebimento de dados incorretos. Entretanto, nesta tese, será considerado que:

**Definição 3** *A comunicação é perfeita, porém limitada por um raio de alcance constante dentro do intervalo  $[0, r_{max}]$ . Esse raio não é necessariamente o mesmo para todos os robôs e nós sensores.*

## 1.3 Escopo da abordagem

A Figura 1.3 apresenta esquema com os possíveis escopos criados a partir da variação da incerteza em duas variáveis: i) posição do time de robôs e ii) posição dos nós sensores. Por exemplo,  $S$ ,  $\hat{S}$ ,  $\bar{S}$  representam, respectivamente, que a posição dos nós sensores é conhecida, incerta ou desconhecida (nesse último caso, não é conhecido nem mesmo o número de sensores do ambiente). Similarmente, a posição do time é representada pela variação em  $R$ . É importante salientar que  $R$ ,  $\hat{R}$  e  $\bar{R}$  dizem respeito à incerteza em relação à posição de todos os robôs dado o ponto de vista de um determinado robô.

Os cenários a serem considerados nesta tese estão destacados na Figura 1.3. Eles constituem quatro cenários nos quais as posições dos sensores e do time de robôs é conhecida ou incerta. Nos casos onde a posição dos nós sensores é conhecida, as tarefas de atendimento aos nós sensores podem ser distribuídas entres os



**Figura 1.3.** Eixos de incerteza em relação aos nós sensores  $S$ ,  $\hat{S}$ ,  $\bar{S}$  e em relação ao robô  $R$ ,  $\hat{R}$ ,  $\bar{R}$ .

robôs do time e executadas de acordo com alguma metodologia cooperativa em ambientes estocásticos. Quando existe incerteza associada à posição dos nós sensores, é necessário desenvolver metodologias que diminuam essa incerteza, de forma a possibilitar a execução das tarefas pelo time de robôs.

Os cenários nos quais tem-se  $\bar{S}$  representam uma condição inicial para a metodologia que será apresentada nesta tese. Ou seja, para que nós sensores sejam atendidos pelo time de robôs é necessário que as suas posições sejam primeiramente descobertas antes de que as tarefas (por exemplo, coleta de dados) possam ser executadas. A exploração do ambiente para descobrir a posição dos sensores, bem como a melhor alocação desses sensores entre os robôs estão fora do escopo deste trabalho. Da mesma forma, os cenários nos quais não existe qualquer estimativa sobre a posição dos robôs não serão tratados nesta tese, uma vez que aqui assume-se que os robôs sempre possuirão alguma estimativa sobre a posição dos outros robôs do time.

## 1.4 Objetivos

Os objetivos desse trabalho podem ser sumarizados por meio dos seguintes itens:

- Propor um arcabouço cooperativo para coleta de dados por meio robôs mó-

veis autônomos em uma rede de sensores sem fio com comunicação limitada, incerteza de posição dos nós sensores e incerteza de atuação e observação dos robôs;

## 1.5 Contribuições

As contribuições estão listadas abaixo:

1. Desenvolvimento de uma abordagem para tratamento de incertezas referentes ao posicionamento dos nós sensores em uma RSSF;
2. A utilização de uma metodologia cooperativa para tratamento de incertezas na RSSF e no time de robôs móveis. Para tanto, o time de robôs é dividido em dois tipos: i) Robôs Tarefa, responsáveis pela visita aos nós sensores e ii) Robôs Exploradores, que navegam pelo ambiente com a função de coletar informações sobre os sensores (posição). Para transferência eficiente de dados, foi desenvolvida uma técnica para tratamento do problema *Rendezvous*. Essa técnica considera a característica assíncrona de execução das tarefas pelos robôs.

## 1.6 Organização do texto

Este texto está organizado da seguinte maneira: o Capítulo 2 apresenta a revisão bibliográfica dos trabalhos que se relacionam ao problema aqui considerado. O capítulo 3 mostra os principais conceitos e modelos utilizados para o desenvolvimento desta tese. O Capítulo 4 apresenta a metodologia desenvolvida para criação do arcabouço proposto. O Capítulo 5 apresenta e discute os resultados obtidos e finalmente o texto é concluído e direções futuras são apresentadas no Capítulo 6.

# Capítulo 2

## Trabalhos Relacionados

Cooperação em sistemas multi-robô é um tema vastamente estudado sob vários aspectos, como controle, planejamento, alocação de tarefas em ambientes determinísticos ou ruidosos [Chaimowicz, 2002; Tavares, 2004; Parker, 1998; Botelho, 2000; Botelho & Alami, 2000]. Nesses sistemas, a tomada de decisão descentralizada é um aspecto chave para o sucesso de missões cooperativas e esse procedimento precisa ser robusto o suficiente para permitir que o robô se recupere de possíveis falhas de atuação e observação.

Em redes de nós sensores tradicionais (estáticas), a cooperação entre os nós é utilizada para tarefas de processamento ou transferência dos dados coletados, por exemplo, fusão de dados. O emprego de robôs móveis em redes de nós sensores sem fio é recente e abre um novo leque de oportunidades em relação às arquiteturas cooperativas que podem ser empregadas nesses ambientes. Um dos problemas que podem ser tratados é a coleta de dados em redes de nós sensores sem fio. A execução dessa tarefa tem como objetivo aumentar o tempo de vida da rede, à medida que explicitamente requer que o nó móvel navegue pela rede em busca da informação contida em cada um dos nós sensores do ambiente.

Neste capítulo serão discutidos alguns trabalhos representativos que mostram os esforços realizados no sentido de criar aplicações que integram os componentes móveis e estáticos para coleta de dados em redes de nós sensores com incerteza.

### 2.1 Mobilidade em RSSF

A mobilidade em RSSF é utilizada por Huang et al. [2002]; Small & Haas [2003], onde o objetivo das entidades móveis era coletar os dados dos nós sensores. Nesses trabalhos foram utilizados, respectivamente, zebras e baleias que se moviam aleato-

riamente pelo ambiente, para estabelecer comunicação entre dispositivos anexados a eles e os nós sensores presentes no ambiente. Shah et al. [2003] utilizam o termo “data mules” para descrição de agentes que coletam dados da rede enquanto aleatoriamente movem-se pelo ambiente. Um dos problemas dessa abordagem reside no fato de que a tarefa de coleta de dados é dependente do movimento aleatório de entidades móveis. No contexto dos métodos propostos, alguns nós podem ser atendidos tardiamente ou eventualmente nunca serem atendidos.

Uma metodologia baseada no movimento de um veículo de transporte foi empregada por Chakrabarti et al. [2003]. Os nós sensores despertam quando os veículos estão no seu raio de comunicação por meio de um sinal para despertar (*wake signal*) e, nesse momento, começam a transferência dos dados. Nesse método, o movimento da entidade móvel está restrita a uma trajetória fixa. Nós sensores que estejam longe dessa trajetória precisam transferir dados por grandes distâncias ou então a rede deve prover facilidade de roteamento: em ambos os casos é uma operação que consome muita energia.

Kansal et al. [2004] apresentam vantagens da utilização de entidades em relação ao roteamento estático: aumento do tempo de vida da rede, manutenção da conectividade, diminuição de erros de recepção, etc. Um algoritmo de controle de velocidade para um robô móvel é utilizado na coleta de dados em uma trajetória retilínea. Esse trabalho é expandido por Somasundara et al. [2006] e modelos para consumo de energia em redes estáticas e em redes com entidades móveis são desenvolvidos e comparados, com significativas vantagens para o segundo caso.

Conforme é possível perceber nos trabalhos discutidos aqui, os nós móveis precisam navegar de maneira eficiente pelo ambiente. Segundo Kansal et al. [2004], o tipo de movimentação pela rede é classificada como: i) *randômica*, onde os nós móveis se movimentam de maneira aleatória, ii) *previsível*, onde a movimentação do nó móvel segue padrão fixo e portanto pode ser predita e iii) *controlável*, onde a movimentação do nó móvel é controlada por algum procedimento de alto nível. A mobilidade controlável é o esquema de movimentação mais flexível, pois diferentemente dos esquemas de mobilidade randômico e previsível, permite que os nós móveis se locomovam a partir de parâmetros dinâmicos.

### 2.1.1 Relação entre os trabalhos e esta tese

Os trabalhos que serão discutidos neste capítulo empregam soluções para problemas de otimização (Seção 2.1.2) como forma de determinar o melhor modelo e estratégia de navegação das entidades móveis pela rede de sensores. Dessa forma,

preocupam-se em maximizar ou minimizar diversas variáveis de interesse, por exemplo, quantidade de dados coletados, tempo de vida da rede, latência dos dados, etc. Outro aspecto investigado nesses trabalhos é a integração entre os robôs móveis e a RSSF (Seção 2.1.3). Para que isso seja realizado, problemas relacionados à comunicação precisam ser tratados, tendo-se em vista que, a despeito das vantagens que ela pode propiciar, seu emprego é custoso. Nesses trabalhos, a comunicação é empregada como um meio para que aplicações que necessitam de integração (por exemplo, rastreamento de objetos, deposição, etc) sejam implementadas. Porém, o custo associado ao processo de comunicação não é modelado. Nesta tese, o custo de comunicação é um fator que influencia a tomada de decisão dos robôs, pois dependendo de seu valor, o robô poderá optar entre comunicar imediatamente ou se aproximar do sensor para coletar os dados.

Parte do problema definido no Capítulo 1 é tratado como um problema de otimização. Robôs responsáveis pela realização das tarefas navegam pelo ambiente de maneira previsível, mas geram essa trajetória a partir da solução de um problema de otimização. A comunicação entre os robôs é modelada e a decisão do momento no qual ela deve ser realizada ocasiona ganhos locais que podem reduzir o custo total da tarefa. Os trabalhos discutidos na Seção 2.1.2 se enquadram no escopo de trabalhos da Figura 1.3, onde não existem incertezas no ambiente (caso *SR*).

Alguns trabalhos consideram a incerteza presente nesses sistemas, por exemplo modelando a navegação dos robôs (Batalin & Sukhatme [2004]) de forma a minimizar os erros de navegação ou fazendo uma análise do número de transmissões feitas em função das probabilidades de erro de comunicação (Tekdas et al. [2009]) (representando os casos  $S\hat{R}$  e  $\hat{S}R$ , respectivamente). Porém, até onde foi investigado, os trabalhos não modelam o processo cooperativo, levando em conta não apenas os problemas locais incerteza (atuação e observação), como também incertezas em relação ao time de robôs (ações realizadas e os efeitos delas no ambiente).

Nesta tese, o processo de decisão de uma parcela do grupo de robôs é modelada por meio de um arcabouço que considera incertezas locais de atuação e observação, uma vez que as suas ações independem das ações do time. Existem robôs nos quais o processo de decisão é modelado de forma a considerar também incertezas em relação ao time. Já a incerteza em relação à posição dos nós sensores no ambiente é tratada na literatura por diversos métodos ([de Oliveira, 2008]), tais como *Ad Hoc Positioning System (APS)* [Niculescu & Nath, 2001], *Recursive Position Estimation (RPE)* [Alvin, 2001], ou localização utilizando *beacons* móveis [Sichitiu &

Ramadurai, 2003]. Esses *beacons* são chamados de *âncoras*, pois conhecem as suas posições com perfeição em qualquer instante de tempo, em contrapartida aos nós sem essa característica, chamados de “nós ordinários”.

Nessa linha, Shigeng et al. [2008] tratam o problema da estimação da posição dos nós sensores em uma rede com mobilidade total. Nesse trabalho os “nós ordinários” estimam suas posições a partir de um método *bounding-box* que mostra-se superior aos métodos cuja localização é baseada em SMC (*Sequential Monte Carlo*), tanto em acurácia quanto em tempo computacional. Lui et al. [2009] também tratam o mesmo problema, porém é considerada a existência de incerteza na posição dos nós sensores âncora e incerteza na velocidade de propagação do sinal. Nesta tese, considera-se que a posição dos robôs móveis é incerta, logo a estimação da posição dos nós sensores é realizada tendo-se em vista essa posição incerta. A incerteza na distância inferida por meio de RSSI é estudada por Nguyen et al. [2011] que propõe uma abordagem para estimar a posição dos nós sensores a partir dos nós âncora. Para tanto, a estimação da posição dos nós sensores e a estimação dos parâmetros do modelo não-linear de perda de sinal do RSSI em função da distância são realizadas em tempo real. No trabalho de Higuchi et al. [2011], todos os nós sensores possuem mobilidade, porém a estimação de suas posições é realizada considerando os sensores vizinhos que estejam parados em determinado instante. Esses sensores são utilizados então como sensores âncora. Diversas outras abordagens podem ser vistas em trabalhos recentes [Shi et al., 2010; Ahn et al., 2010; Kadkhoda et al., 2010; Leng & Wu, 2010].

Enquanto nos trabalhos discutidos, a estimação da posição dos sensores é o foco principal, nesta tese, a incerteza de posição é tratada como um problema que impacta a eficiência da execução das tarefas pelos robôs em um cenário cooperativo. Como os robôs móveis não tem conhecimento global do sistema, apenas parcial, são propostos métodos para que o compartilhamento dessa informação.

### 2.1.2 Solução de problemas de otimização para controle de entidades móveis

Alguns trabalhos demonstram o esforço na construção de estratégias de navegação do tipo controlável, onde o nó móvel tem um determinado grau de autonomia para decidir a trajetória que irá executar, com base nos parâmetros de entrada. Nesse sentido, soluções heurísticas foram propostas por Somasundara et al. [2004] para *Mobile Element Scheduling Problem* (MES). Nesse problema, um conjunto de nós sensores precisa ser atendido antes que suas memórias de dados sejam totalmente

preenchidas. O conceito de *Deadlines Dinâmicos* é usado para denotar o tempo restante para que a quantidade de memória utilizada atinja um valor máximo. Esse problema é inspirado no *Problema do Roteamento de Veículos ou PRV (VRP - Vehicle Routing Problem)* [Dantzig & Ramser, 1959], mas com diferenças significativas: por exemplo, é permitido que um nó seja visitado mais de uma vez. Esse trabalho é estendido para o caso de múltiplas entidades móveis por Jea et al. [2005]. Gu et al. [2005] tratam o MES por meio de um algoritmo que divide o problema em dois subproblemas: a divisão da rede de sensores de acordo com a taxa de aquisição de dados dos mesmos e o escalonamento da entidade móvel entre o nós levando em conta a frequência com a qual os nós coletam dados. Esse algoritmo obteve melhor desempenho quando comparado com a melhor heurística apresentada por Somasundara et al. [2004].

Somasundara et al. [2007] estendem o trabalho proposto para controle de entidades móveis desenvolvido por Somasundara et al. [2004] por meio de uma comparação entre a Heurística *Minimum Weighted Sum* e a tradicional *Heurística de Inserção* para o Problema do Roteamento de Veículos com Janelas de Tempo [Cordeau et al., 2001]. A Heurística de Inserção foi modificada para solucionar o MES. Resultados referentes ao *overflow* da rede mostram que o método baseado na Heurística de Inserção apresentou melhor desempenho. Isso reforça a abordagem adotada neste trabalho que baseia-se em um planejador otimizado para navegação eficiente pela rede de sensores.

A estratégia de navegação também mostrou-se fundamental para tratar o problema da coleta de dados [Pereira et al., 2004]. Nesse trabalho, a “urgência” com a qual um nó sensor precisa ser atendido foi modelada por meio de uma função de potencial cuja amplitude é diretamente relacionada à quantidade de dados presente no nó sensor. O robô utiliza então um controlador que segue o campo de potencial criado. O método apresentado no trabalho pode ser considerado uma solução heurística para o Problema do Caixeiro Viajante com Coleta de Prêmios [Balas, 1989]. Soluções para o Problema do Caixeiro Viajante ou PCV (*Travelling Salesman Problem*) serviram de inspiração para alguns trabalhos de navegação e coleta de dados. Soares et al. [2007b] estende o trabalho de Pereira et al. [2004] inserindo o conceito de *gradiente preferencial* que pode ser: i) a direção do próximo nó sensor presente na rota do PRV (*Método do Gradiente do PRV*) ou ii) a direção do nó com maior urgência de atendimento (*Método do Maior Gradiente*). Os resultados comparativos entre as duas abordagens demonstraram que o Método do Gradiente do PRV apresentava resultados superiores, no que diz respeito à otimização das rotas e quantidade de dados perdidos na rede.

Similarmente, soluções para o PCV e PRV fazem parte dos métodos propostos por Aioffi [2007] e Valle et al. [2008]. Aioffi [2007] propõe duas abordagens: i) SHS (*Single Hop Strategy*) e ii) *Multi-Hop Strategy*, onde a mobilidade do sorvedouro móvel e técnicas de agrupamento dos nós são integradas com roteamento de um salto e multi-saltos, respectivamente, com o objetivo de melhorar métricas como o tempo de vida da rede, cobertura e conectividade. Porém, os resultados mostraram também que as estratégias multi-saltos (roteamento tradicional) são melhores que os métodos que empregam mobilidade no que diz respeito ao atraso na entrega dos dados. Recentemente, o trabalho de Wu et al. [2009] apresentou uma abordagem para minimização da latência e consumo de energia em ambientes similares aos cenários descritos em [Aioffi, 2007], ou seja, tem-se que a rede é composta por agrupamentos (que podem ser desconectados e esparsos) e, assume-se que existe roteamento entre os nós sensores em cada agrupamento. O método de navegação pela rede é, mais uma vez, baseado em soluções para o PCV.

Huang & Ozdalar [2006] desenvolvem uma estratégia para particionamento da rede e controle da potência de transmissão visando a maximização do tempo de vida da rede. Cada partição emprega um agente móvel que coleta os dados dos nós sensores estáticos. Em cada partição os agentes móveis percorrem rotas obtidas por meio de alguma heurística para o PCV. O trabalho objetiva encontrar as partições e a força de transmissão ótimas de forma a maximizar o tempo de vida da rede e garantir a estabilidade do sistema (o sistema é estável se nenhum nó sensor precisa esperar um tempo infinito para rotear seus dados).

Da mesma forma, Wang et al. [2005b] tratam o problema de maximização do tempo de vida da rede utilizando um sorvedouro móvel. O algoritmo desenvolvido utiliza programação linear para encontrar um escalonamento ótimo que determina a ordem na qual o sorvedouro deve visitar os nós sensores e o tempo da duração atribuído a cada visita. Nesse trabalho, os nós sensores são organizados em uma topologia de *grid* e apenas uma entidade móvel é empregada. Enquanto o sorvedouro permanece na região de um determinado nó, os outros nós transmitem seus dados por meio de roteamento *multi-hop*.

Wang et al. [2005a] utilizam nós móveis para comparar três métodos de coleta de dados: i) coleta em uma rede estática, ii) coleta utilizando uma entidade móvel chamada de *mobile relay* e iii) coleta utilizando um sorvedouro móvel. A utilização de um sorvedouro móvel apresenta melhores resultados no que diz respeito ao tempo de vida da rede. Porém, conforme os autores, nem sempre é possível a utilização de sorvedouros móveis em ambientes inóspitos. Dessa forma, a utilização de *mobile relays* é sugerida e mostra-se que o aumento no tempo de vida da rede

em relação à coleta de dados em redes estáticas pode ser no máximo 4 vezes maior. Um algoritmo de roteamento similar ao desenvolvido em Wang et al. [2005b] é construído com o objetivo de atingir esse limite superior. Também é mostrado que em um ambiente de raio  $R$  são necessárias  $O(R)$  entidades móveis para se obter um tempo de vida igual ao obtido pela utilização de um sorvedouro móvel.

Kinalis & Nicoletseas [2007] propõem um método simples e escalável com múltiplos sorvedouros móveis. Com o objetivo de minimizar o consumo da rede e o atraso na recepção dos dados (*delivery delay*), três principais aspectos são abordados: i) um estudo sobre o impacto da utilização de múltiplos sorvedouros móveis, ii) coleta de dados utilizando nós móveis e nós estáticos (roteamento entre nós estáticos é empregado) e iii) o desenvolvimento de um método que permita a distribuição dos sorvedouros móveis pelo ambiente.

Outra variável interessante a ser minimizada é o tempo de coleta dos dados armazenados nos nós sensores. Bhadauria & Isler [2009] propõem um novo problema chamado Problema de Coleta de Dados (DGP - *Data Gathering Problem*) no qual o objetivo é gerar uma rota para que cada robô visite os nós sensores de forma a minimizar essa variável. Nesse problema, o tempo total de coleta é a soma de duas componentes: i) tempo para navegar até o ponto de coleta e ii) tempo para realizar a transferência dos dados. São apresentadas duas heurísticas para variações dos PCV que foram alteradas e combinadas para gerar uma solução para o DGP.

### 2.1.3 Integração e comunicação entre entidades móveis e nós sensores estáticos

Para que as aplicações alvo de RSSF sejam implementadas eficientemente, é necessário que a comunicação entre os elementos da rede seja realizada de maneira satisfatória. Dentre os trabalhos mostrados a seguir, alguns apresentam um grau de cooperação, na medida que permitem que a informação entre os nós seja compartilhada durante a realização das tarefas.

O robôs móveis podem ser utilizados para deposição de nós sensores à medida que a exploração do ambiente é realizada. O trabalho de Batalin & Sukhatme [2004] trata dos seguintes problemas: i) deposição de nós sensores e ii) exploração do ambiente por um robô móvel. Nesse tipo de aplicação, a interação da rede torna-se clara à medida que o robô realiza a exploração sugerida pelos nós sensores da rede que foram depositados pelos próprios robôs. Os nós recomendam direções para as quais o robô pode navegar. Quando o robô entra no raio de alcance de um nó sensor, esse nó sensor indica primeiramente as direções não visitadas. Caso

todas direções já tenham sido visitadas, o nó sensor indicará a direção com menos ocorrências de visitas. Assim, é possível ao mesmo tempo, navegar por todo o ambiente e efetuar a deposição de nós em locais que o robô estiver explorando pela primeira vez. O ambiente em que o robô atua é modelado como um grafo no qual a transição para um nó é realizada com base em uma função de probabilidade  $p(x'|x, u)$  (probabilidade de chegar ao vértice  $x'$  estando no vértice  $x$  e executando a ação  $u$ ). A escolha de uma abordagem probabilística deve-se ao desconhecimento do mapa do ambiente. Ainda que não tenha sido proposto com a função de extração de dados na rede, o algoritmo se mostra interessante para realizar tal ação, em cenários nos quais nenhum mapa é fornecido e o robô não possua dispositivos globais de localização (por exemplo, GPS).

Li et al. [2003] apresentam um algoritmo distribuído para guiar agentes autônomos (que podem ser pessoas) em uma rede de sensores sem fio, por regiões consideradas seguras. Por exemplo, os nós podem estar espalhados em uma área florestal para monitoramento de incêndios, e as regiões que detectassem eventos do tipo “temperatura alta” seriam consideradas inseguras e estariam excluídas do caminho gerado pelo algoritmo. A técnica utilizada consiste em modelar as regiões inseguras por meio de *campos de potencial* repulsivos. Dessa forma, o algoritmo precisa planejar rotas por meio da rede de nós sensores que levem o agente da posição atual até uma posição alvo, desviando de regiões críticas. A sua utilização é particularmente interessante para redes de nós sensores voltadas a *eventos*, onde espera-se que exista solicitação de atendimento por parte dos nós sensores que detectam eventos. Entretanto, isso requer constante comunicação entre o agente móvel e a rede de nós sensores, o que pode trazer uma série de problemas, tais como perda de dados, congestionamento da rede (já que *broadcast* é constante) e conexão assimétrica entre os nós. O algoritmo empregado é *guloso* e consegue obter a rota ótima entre dois pontos, visto que os nós sensores da rede estão uniformemente dispostos no ambiente.

Corke et al. [2003] estendem o trabalho de Li et al. [2003] para o caso de um robô autônomo aéreo (helicóptero). O algoritmo tem as seguintes características: i) cada nó da rede precisa saber sua localização (não necessariamente exata) para prosseguir com a geração do caminho que o robô móvel seguirá; ii) cada nó da rede tem que ter a capacidade de computar, alterar e armazenar o caminho que o robô móvel seguirá (quando o nó fizer parte do caminho); iii) o robô móvel tem que ter a capacidade de interagir com os nós da rede de forma a tomar conhecimento do caminho gerado e de eventuais alterações. Como esse trabalho é apenas uma extensão de Li et al. [2003], os mesmos problemas podem ser aqui encontrados.

Corke et al. [2004] estendem o trabalho anterior [Corke et al., 2003] para a deposição de nós sensores e a manutenção da conectividade em uma rede de nós sensores sem fio usando um robô autônomo aéreo. A etapa de deposição é feita da seguinte maneira: dada uma *topologia* específica de rede e uma *escala de deposição*, constrói-se um plano que os robôs devem seguir para depositar os nós. Cada nó sensor torna-se um vértice de um grafo na topologia pré-determinada: *estrela*, *grade* ou *aleatória*. A escala na qual os nós são depositados depende do alcance dos mesmos. A segunda fase consiste em testar a conectividade da rede. Um nó especial é configurado com a função de enviar pacotes a todos os nós. Esse nó aguarda a resposta de todos os outros, de forma a certificar-se de que a rede esteja totalmente conectada. Se for detectada alguma desconexão na rede, alguns pontos de deposição são gerados de forma a restabelecer a conectividade. A maneira como o robô irá alcançar as *posições-alvo* para depositar os nós é feita da seguinte forma: i) para cada ponto  $(x_i, y_i)$  dos nós sensores da rede, primeiro ordenam-se os pontos pela coordenada  $x$  e depois pela coordenada  $y$ ; ii) depositam-se os nós na ordem estabelecida. Com esse método, a rota gerada pode estar muito aquém de uma rota ótima que contivesse todos os pontos, acarretando um custo maior (energia, combustível, etc.) ao robô aéreo que realizasse a deposição.

Peterson & Rus [2004] apresentam um dispositivo de *hardware* chamado de *Flashlight* que interage com a rede de nós sensores exatamente como foi descrito por Li et al. [2003]. As funções realizadas pelo dispositivo são: i) ativar ou desativar os nós sensores de uma área específica da rede (pois pode ser interessante manter certos nós *adormecidos* para diminuir o gasto em energia); ii) detectar a ocorrência de eventos na rede, tais como aumento de temperatura; iii) guiar o agente móvel ou robô para locais seguros em atividades de resgate de vítimas de incêndios florestais, terremotos, etc. Ainda utilizando a mesma técnica de campos de potencial para geração de caminhos, Kantor et al. [2003] usam *Flashlight* para gerar rotas através da rede de nós sensores em um cenário de procura e resgate de vítimas de incêndios. Nesse trabalho e nos trabalhos citados anteriormente a rede de sensores pode ser vista como um repositório distribuído de informação. Cada nó sensor consegue estimar sua distância até o alvo e áreas de risco, após receber estimativas dos nós sensores vizinhos. Dessa forma, a comunicação entre os nós sensores é indispensável.

Tang et al. [2004] descrevem um algoritmo para deposição de nós sensores em ambientes fechados. A metodologia empregada nesse trabalho utiliza um planejador centralizado para geração das posições dos nós que irão compor a rede. Primeiramente, é gerado um mapa de ocupação [Elfes, 1987; Moravec, 1988] em

que estará representado o ambiente no qual o robô atuará, incluindo corredores, portas e obstáculos. Em um segundo momento, são geradas as posições nas quais os nós serão depositados, que levam em conta vários aspectos como: manter a comunicação entre os nós, evitar que as posições geradas sejam próximas demais a ponto de dificultar a deposição, evitar posições nas quais existam portas ou obstáculos, etc. Essas posições são geradas em grupos de  $n$  nós sensores que, em um último momento, são atribuídos a robôs líderes responsáveis por guiar os nós (já que se tratam de nós sensores móveis) até a suas posições de destino. Esse método, implementado em um grupo de 70 nós sensores móveis, mostrou-se interessante para ambientes fechados, nos quais as restrições devido à portas e obstáculos produz uma cobertura uniforme sobre todo o ambiente.

Wu et al. [2004] introduzem o termo *redes distribuídas de nós sensores baseadas em agentes móveis (mobile agent-based distributed sensor networks)* em uma tarefa de detecção e rastreamento de alvos móveis. O problema é traçar a rota de um robô móvel na rede de nós sensores para realizar a fusão das informações coletadas pelos nós, à medida que esses detectam um alvo de interesse dentro do seu alcance. Para tanto, é definida uma função diretamente proporcional à energia do sinal (*signal strength*) recebida pelos nós quando o alvo está sendo detectado e inversamente proporcional as perdas associadas ao sinal transmitido e recebido pelos nós ao longo do caminho (*path loss*) e também ao consumo de energia pela rede (*energy consumption*). O trabalho prova que se trata de um problema *NP-completo* cuja solução proposta pelos autores é obtida por meio de heurísticas baseadas em *Algoritmos Genéticos* [Holland, 1975].

As vantagens da utilização de um robô móvel foi também investigada por Tekdas et al. [2009]. Nesse trabalho, muitos conceitos básicos referentes ao emprego de mobilidade em RSSF são revistos e um cenário, no qual é disposta uma grade de  $3 \times 4$  nós sensores, é proposto e avaliado no que diz respeito ao Número Médio de Transmissões. Nesse contexto, o número de transmissões de um pacote é representado pela quantidade de saltos (*hops*) necessários para que um pacote chegue ao nó sorvedouro, já que admite-se que os nós comunicam-se apenas com os seus vizinhos. Perceba que quando o nó sorvedouro é móvel, para um pacote é necessário apenas uma transmissão, assumindo que a probabilidade de falha na transmissão é zero. O número total de transmissões é investigado sem a utilização de robô móvel, com a utilização de um robô móvel, e com a utilização de *relays* (necessários quando o robô móvel não se encontra no raio de alcance do nó sensor). Essa análise é realizada em função da probabilidade do erro de transmissão dos pacotes.

A decisão sobre o momento ideal para pacotes sejam transmitidos em uma RSSF com mobilidade foi investigada em [Bölöni & Turgut, 2008]. Idealmente a comunicação deve ser estabelecida quando o nó sensor estiver no raio de comunicação da entidade móvel. Porém, em muitos casos, pode-se ter urgência em transmitir dados (por exemplo, se os dados forem críticos ou se a memória estiver acabando), não sendo possível aguardar a chegada da entidade móvel. Nesse contexto, são propostos os seguintes métodos para tomada de decisão: i) um método ótimo baseado em programação dinâmica para o caso no qual a trajetória da entidade móvel é conhecida previamente e ii) um método baseado em *Processos de Decisão de Markov* para quando a trajetória não é conhecida previamente.



# Capítulo 3

## Fundamentos

Nesta tese, o tratamento de incerteza é fundamental para que os robôs consigam desempenhar eficientemente suas tarefas. Logo, a tomada de decisão precisa ser realizada de maneira cautelosa, uma vez que existe a possibilidade de falhas na execução de ações e o estado nos quais os robôs se encontram é desconhecido ou incerto. Nesse capítulo, serão mostradas ferramentas para modelagem de sistemas que tratam incertezas com um ou mais robôs. Primeiramente, será abordado o conceito de *Redes Bayesianas* [Pearl, 1988], arcabouço fundamental para compreensão dos Modelos de Decisão Markovianos que serão apresentados posteriormente para tratamento de incertezas de atuação, observação e de grupo.

### 3.1 Tomada de Decisão em Cenários Dinâmicos e Estocásticos

Para a concepção dos modelos apresentados nesta tese, foram utilizados muitos conceitos da teoria de probabilidades. Está além do escopo deste trabalho, descrevê-los minuciosamente, porém é importante descrever conceitos fundamentais que permeiam toda a teoria de decisão: o *Teorema de Bayes* e as *Redes Bayesianas*. Esse Teorema estabelece a seguinte relação:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)},$$

onde A e B são duas variáveis aleatórias. Essa equação deriva de duas outras equações: i)  $p(A \wedge B) = P(A|B)p(B)$  e ii)  $p(B \wedge A) = P(B|A)p(A)$ . Igualando os dois termos à esquerda dessas equações obtém-se o *Teorema de Bayes*.

O *Teorema de Bayes* diz que a probabilidade de um evento  $A$  ocorrer dado que o evento  $B$  ocorreu é igual à probabilidade de  $B$  ocorrer dado que  $A$  ocorreu multiplicada pela probabilidade de  $A$ . Para entender melhor, considere um problema de navegação no qual um robô precise se locomover por um conjunto de células em um ambiente discretizado. Suponha que o robô possa identificar a célula na qual se encontra, uma vez que junto a ela existe um marco que a distingue de todas as outras. Porém essa identificação é sujeita a erros, uma vez que os sensores retornam leituras ruidosas. Nesse caso, considere que:  $A$  é o evento que caracteriza a presença do robô  $R_1$  em uma célula  $C_1$ ,  $B$  é o evento que caracteriza a identificação da célula na qual o robô se encontra. Essas duas probabilidades podem ser entendidas através da relação de “sintoma e diagnóstico”: enquanto  $B$  representa a possível evidência de que o robô se encontra na célula  $C_1$ , ou seja, o “sintoma” (já que os sensores do robô indicam isso), o evento  $A$  é o “diagnóstico” obtido em função disso, uma vez que computa a probabilidade de o robô estar de fato em  $C_1$  em função desse “sintoma”.

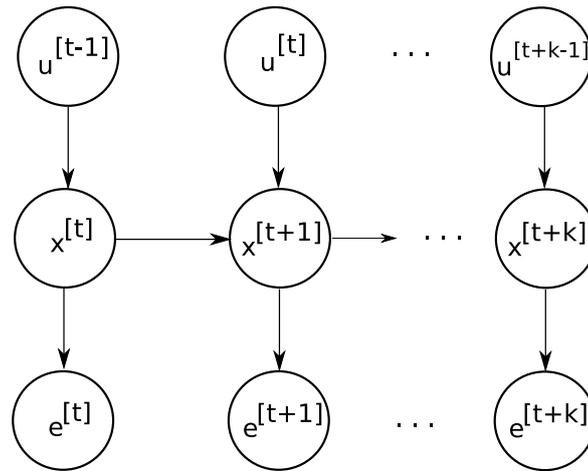
Dessa forma o *Teorema de Bayes* pode ser mapeado para:

$$p(R_1 \text{ está em } C_1 | R_1 \text{ identifica } C_1) = \frac{p(R_1 \text{ identifica } C_1 | R_1 \text{ está em } C_1)p(R_1 \text{ está em } C_1)}{p(R_1 \text{ identifica } C_1)}.$$

Nessa equação,  $p(R_1 \text{ identifica } C_1 | R_1 \text{ está em } C_1)$  é uma probabilidade que descreve o comportamento dos sensores, podendo ser mapeada previamente, de maneira empírica. Já  $p(R_1 \text{ está em } C_1)$  é um valor obtido, obrigatoriamente, a partir do histórico de eventos anteriores. Por exemplo, a partir de um conjunto de amostras que representam as visitas dos robôs nas células do ambiente, pode-se obter a probabilidade de que o robô esteja em cada uma das células dividindo-se o número de vezes que cada célula foi visitada pelo número total de visitas (inicialmente, essa probabilidade é uniformemente distribuída entre todas as células). Por fim,  $p(R_1 \text{ identifica } C_1)$  é uma constante que garante que:

$$\sum_{i=1}^n p(R_1 \text{ está em } C_n | R_1 \text{ identifica } C_1) = 1.$$

onde  $n$  é o número total de células no ambiente. Na verdade, a probabilidade  $p(R_1 \text{ identifica } C_1)$  pode ser calculada por meio do *Teorema da Probabilidade Total*,



**Figura 3.1.** Exemplo de Rede Bayesiana Dinâmica. Por exemplo, a aresta entre  $x^{[t+1]}$  e  $x^{[t]}$  especifica  $p(x^{[t+1]}|x^{[t]})$  no modelo de transição. Já a aresta entre  $x^{[t+1]}$  e  $e^{[t+1]}$  especifica  $p(e^{[t+1]}|x^{[t+1]})$  no modelo de correção.

uma vez que:

$$p(R_1 \text{ identifica } C_1) = \sum_{i=1}^n p(R_1 \text{ identifica } C_1 | R_1 \text{ está em } C_n) p(R_1 \text{ está em } C_n).$$

### 3.1.1 Redes Bayesianas Dinâmicas

Uma *Rede Bayesiana* é um grafo no qual os nós são variáveis aleatórias de estados do sistema e as arestas representam as relações de dependência entre elas. Essas relações são as probabilidades condicionais entre duas variáveis. De maneira geral, essas redes podem ser descritas pelos seguintes itens [Pearl, 1988; Russel & Norvig, 2003]:

1. Um conjunto de variáveis aleatórias discretas ou contínuas;
2. Um conjunto de arestas direcionadas conectando pares de nós. Por exemplo, se existe uma aresta ligando o nó  $X$  ao nó  $Y$ , diz-se que  $X$  é pai de  $Y$ ;
3. Cada nó  $X$  possui uma distribuição de probabilidades condicional  $p(X_i | \text{Pais}(X_i))$  que quantificam a influência dos pais sobre o nó;
4. O grafo não possui ciclos (a permanência em determinado estado é representado por meio da replicação de nós e não por meio de um ciclo).

Por exemplo, considere uma rede com dois nós,  $A$  e  $B$ . No caso da rede possuir uma aresta ligando  $A$  a  $B$ , além das definições das probabilidades a priori

$p(A)$  e  $p(B)$ , define-se também a probabilidade condicional  $p(B|A)$ , uma vez que  $A$  é pai de  $B$ .

Uma rede como essa possui uma limitação: é estática, ou seja, possui um número fixo de nós e as probabilidades são previamente determinadas. Em alguns casos, pode ser necessário construir uma rede que capture a dinâmica do sistema ao longo do tempo, por meio da mudança no valor das variáveis de estado que o constituem. Esse tipo de rede, conhecida como *Rede Bayesiana Dinâmica*, compreende dois tipos básicos de variáveis:

1.  $X_i^{[t]}$ , que descrevem as variáveis não observáveis do sistema;
2.  $O_i^{[t]}$ , que descrevem as variáveis observáveis do sistema.

onde o superscrito  $t$  representa o instante de tempo no qual a inferência é realizada. Como as variáveis são indexadas no tempo e, em princípio, o tempo não é limitado, o número de arestas no grafo entre os estados (e conseqüentemente as tabelas de distribuição de probabilidades condicionais) pertencentes a instantes de tempo diversos tende a crescer infinitamente. Uma rede bayesiana dinâmica é ilustrada na Figura 3.1 para um filtro de *Kalman* [Kalman, 1960]. O filtro de *Kalman* está sendo utilizado na figura para estimar a variável  $X_i^{[t+1]}$  dado o estado anterior  $X_i^{[t]}$ . Além disso, note que a observação  $O$  depende do estado atual no qual o robô se encontra. Nesse tipo de modelagem, o estado no instante de tempo  $t$  depende de um conjunto finito de estados anteriores. Para o caso mais simples, no qual considera-se apenas a dependência com o estado anterior, tem-se um *Processo Markoviano de Primeira Ordem*. Formalmente:

$$p(X^{[t]}|X^{[0:t-1]}) = p(X^{[t]}|X^{[t-1]}), \quad (3.1)$$

que é chamado de *Modelo de Transição do Sistema*.

As variáveis observáveis do sistema aparecem no *Modelo de Observação*, definido formalmente como:

$$p(O^{[t]}|X^{[0:t]}, O^{[0:t-1]}) = p(O^{[t]}|X^{[t]}). \quad (3.2)$$

A definição desses modelos é indispensável para que inferência em sistemas dinâmicos possa ser realizada. Nas situações tratadas neste trabalho, e na grande maioria de aplicações que utilizam esse tipo de abordagem, tem-se interesse em um tipo particular de inferência, conhecida como *predição*, na qual o objetivo é encontrar a distribuição *a posteriori*  $p(X^{[k+t]}|O^{[1:t]})$  para  $k > 0$ . A predição tem o

objetivo de estimar o estado no qual o sistema se encontra. Para isso, pode-se utilizar uma série de estimadores de estados baseados no modelo bayesiano, tais como *Filtros de Kalman*, *Modelos Ocultos de Markov* ou *Filtros de Partículas*. Porém, para este trabalho, mais importante do que essa estimação é a tomada de decisão, assunto que será aprofundado nas seções seguintes.

### 3.1.2 Atuação e observação

Na maioria dos cenários é sabido que as variáveis tratadas no problema são eminentemente estocásticas, de forma que abordagens determinísticas não se apresentam como uma alternativa viável para o problema tratado. Nesse tipo de cenário são comuns: i) incertezas na atuação e ii) incertezas na observação. Antes de prosseguir, a seguinte definição precisa ser estabelecida:

**Definição 4** *A ação efetuada em um estado  $(x_i^{[t]}, y_i^{[t]})$ , onde  $0 \leq i \leq N \times N$  é sujeita a erros. Dessa forma, a transição para um estado  $(x_j^{[t+1]}, y_j^{[t+1]})$ , onde  $0 \leq j \leq N \times N$ , é modelada por uma distribuição de probabilidades.*

Os erros de atuação surgem, como se pode supor, a partir de erros nos atuadores e efetuidores do robô. Quando um robô precisa dirigir-se a determinada localidade para lá executar alguma ação pré-estabelecida, não é possível garantir que os atuadores funcionarão perfeitamente a todo o momento. Dessa forma, é importante definir alternativas para a tomada de decisão diante da ocorrência de erros. Nesse caso, quando a ocorrência de um erro é confirmada, deve-se apontar uma solução que represente a melhor decisão a ser tomada para que a ação possa ser concretizada. A noção do que representa “o melhor” dependerá do problema a ser tratado, mas é comum que isso esteja associado ao custo decorrente de tais decisões. No caso apresentado na figura anterior, definir a direção a ser seguida representaria essa decisão que poderia levar em conta os custos com o consumo de energia, ou a urgência em alcançar o objetivo, entre outras razões.

Os erros de observação são caracterizados, basicamente, pelo fato de não se ter certeza do estado atual do sistema. Dessa forma, pode-se definir:

**Definição 5** *Em um estado  $(x_i^{[t]}, y_i^{[t]})$ , onde  $t \geq 1$  e  $0 \leq i, j \leq N \times N$ , uma observação  $e^{[t]}$  realizada pelos sensores montados no robô é sujeita a erros inerentes aos sensores. Dessa forma, a observação  $o^{[t]}$  em  $(x_i^{[t]}, y_i^{[t]})$  é modelada de acordo com uma distribuição de probabilidades.*

Suponha, por exemplo, que um robô móvel precisa realizar uma missão de coleta de dados em uma RSSF. Nesse caso, os estados poderiam ser representados por meio de tuplas que contêm a posição do robô e a quantidade de dados presentes nos nós sensores como  $(s_i, d_i(t))$ , onde  $0 \leq i \leq n$ . Como os sensores do robô são imperfeitos, erros de leitura ou medição (ou seja, dados ruidosos obtidos por meio de *encoders*, lasers, sonares, etc) podem ocorrer. Considere, por exemplo, que o robô não tenha certeza de estar diante do nó  $s_i$  que apresenta uma quantidade de dados  $d_i(t)$ . O robô não tem como determinar em qual estado se encontra com absoluta certeza, tendo apenas uma crença sobre aquele estado, que é representada por meio de distribuições de probabilidades. Dessa maneira, a modelagem de um problema que leva em conta erros de observação precisa considerar um espaço de estados significativamente maior, uma vez que o robô poderá estar próximo, com determinada probabilidade, a qualquer nó sensor. Devido ao tamanho do espaço de estados, as soluções para esses problemas geralmente são baseadas em heurísticas, por se tratar de um problema com complexidade não-polinomial.

### 3.1.3 Tratamento de Erros de Atuação

O planejamento de uma missão ou de cada uma das sub-tarefas que a constituem pode ter a sua execução dificultada ou até mesmo impossibilitada se eventos externos àqueles considerados inicialmente ocorrerem. Dentre esses eventos, pode-se citar as falhas de atuação do robô responsável pela execução da tarefa. Faz-se necessário, dessa maneira, construir um plano que permita ao robô tomar decisões que minimizem as penalizações advindas da falha cometida, visando a conclusão do plano original, com o menor custo possível.

Um *Processo de Decisão de Markov* ou MDP (*Markov Decision Process*) [Bellman, 1957] é modelado por meio de uma rede bayesiana que incorpora as probabilidades de transição entre os estados da rede quando uma ação é realizada, permitindo, assim, que decisões sejam tomadas em cenários nos quais existam erros de atuação. Um MDP é uma quádrupla  $(X, U, \delta, \gamma)$ , onde:

- $X$  é o conjunto de estados do sistema;
- $U$  é o conjunto de ações que podem ser executadas pelo robô;
- $\delta : X|X \times U \rightarrow [0, 1]$  é uma função de transição probabilística;
- $\gamma : X \times U \rightarrow \mathbb{R}$  é a função de reforço aplicada ao robô.

A função de reforço pode ser representada da seguinte maneira:

$$\gamma(x, u) = \begin{cases} +a & \text{se a execução de } u \text{ atinge um estado objetivo} \\ -b & \text{de outra maneira,} \end{cases} \quad (3.3)$$

onde  $b$  é um valor de penalização e  $a$  é um valor de recompensa.

Essa função recompensa o robô quando esse consegue cumprir o seu objetivo ao alcançar o estado final e penaliza o robô para cada ação que não contribua para levá-lo ao estado final. O objetivo do robô, durante a execução de suas tarefas, é maximizar o reforço obtido ao longo do tempo. Como o sistema é estocástico, utiliza-se o valor esperado dos reforços ao longo do tempo como critério de otimização para resolução o MDP:

$$E\left[\sum_{t=0}^{\infty} \alpha^t \gamma^{[t]}\right], \quad (3.4)$$

onde  $\alpha$  é um fator de desconto.

Como a partir de um estado  $x_j$  é possível executar várias ações  $u_i$ , é possível conceber uma função que maximize os ganhos do robô, conhecida como *Função de Utilidade*:

$$V(x) = \max_u \gamma(x, u), \quad (3.5)$$

ou seja,  $V(x)$  possui o máximo valor de utilidade, obtido a partir da execução de  $u$  no estado  $x$ . A função de utilidade pode ser generalizada para  $k$  passos, gerando uma *Função de Utilidade Acumulada*:

$$V^{[k]}(x) = \max_u \left[ \gamma(x, u) + \alpha \int V^{[k-1]}(x') p(x'|x, u) dx' \right]. \quad (3.6)$$

Finalmente, em  $k$  passos, a melhor política de atuação pode ser definida como:

$$\pi^{[k]}(x) = \arg \max_u [V^{[k]}(x)]. \quad (3.7)$$

ou seja, a ação  $u$  é executada no estado  $x$ , pois gera o maior valor de utilidade, em  $k$  iterações.

**Programação Dinâmica para Resolução de MDP.** O *Princípio de Bellman* [Littman et al., 1995a] diz que a solução ótima para um processo dinâmico de  $k$  etapas pode ser computado a partir da solução ótima na etapa  $k - 1$ . Partindo desse princípio, um Processo de Decisão de Markov pode ser resolvido por meio de um algoritmo de programação dinâmica conhecido como *Algoritmo de Iteração por Valor*.

Se o MDP é modelado com  $n$  estados, então haverá  $n$  incógnitas. A solução para o problema consiste em resolver esse conjunto de equações simultâneas, conhecidas como Equações de Bellman:

$$V^*(x) = \max_u \left[ \gamma(x, u) + \alpha \int V^*(x') p(x'|x, u) dx' \right], \quad (3.8)$$

onde  $V^*(x)$  representa o valor de utilidade ótimo para o estado  $x$ .

Por serem equações não lineares (devido ao operador  $\max$ ) uma solução plausível é resolver o sistema por meio de um procedimento iterativo, onde os valores de  $V$  para cada estado são atualizados de acordo com a equação:

$$\hat{V}^{[k]}(x) = \max_u \left[ \gamma(x, u) + \alpha \int \hat{V}^{[k-1]}(x') p(x'|x, u) dx' \right]. \quad (3.9)$$

onde  $\hat{V}^{[k]}(x)$  representa o valor de utilidade estimado na iteração  $k$  para o estado  $x$ .

O processo iterativo continua até que a diferença entre os valores de utilidade entre as iterações  $k$  e  $k - 1$ , para cada estado, seja menor que algum erro pré-definido. Esse procedimento está descrito no Algoritmo 1. No passo 1, o valor de utilidade  $V(x)$  é igual ao valor máximo de reforço obtido ao se executar alguma ação  $u$ . O procedimento de atualização do valor de utilidade é então atualizado por meio de passo 2, até que o erro mínimo  $\epsilon$  seja alcançado (passo 3).

---

**Algoritmo 1** Algoritmo de Iteração por Valor.

---

1. Para todo o estado  $x$  compute  $\hat{V}^{[k]}(x) = \max_u \gamma(x, u)$
  2. Para todo o estado  $x$  compute  $\hat{V}^{[k]}(x) = \max_u \left[ \gamma(x, u) + \alpha \int \hat{V}^{[k-1]}(x') p(x'|x, u) dx' \right]$
  3. Repita o passo 2 até que  $\left[ \hat{V}^{[k]}(x) - \hat{V}^{[k-1]}(x) \right] \leq \epsilon, \forall x \in X$
- 

### 3.1.4 Tratamento de Erros de Observação

Um *Processo de Decisão de Markov com Observação Parcial* ou POMDP (*Partially Observable Markov Decision Process*) [Kaelbling et al., 1998] baseia-se no fato de que o estado não é totalmente observável. As equações para obtenção dos valores de utilidade e política de atuação são definidas em função de crenças e não mais estados. O robô não tem mais a certeza do estado no qual se encontra, e apenas

acredita estar em algum estado com uma probabilidade associada. A Equação 3.6 apresentada na seção anterior pode ser reescrita como:

$$V^{[k]}(\beta) = \max_u \left[ \gamma(\beta, u) + \alpha \int V^{[k-1]}(\beta') p(\beta' | \beta, u) d\beta' \right], \quad (3.10)$$

onde:

$$p(\beta' | \beta, u) = \int_{x'} \left( p(o|x') \int_x p(x'|x, u) p(x) dx \right) dx'. \quad (3.11)$$

Na Equação 3.11 é possível perceber o modelo de transição  $p(x'|x, u)p(x)$  integrado sobre a variável  $x$  e o modelo de correção  $p(o|x')$  integrado sobre a variável  $x'$ . Nessa equação,  $\beta$  é a distribuição de probabilidade sobre os estados do sistema ou seja  $\sum_{i=1}^n \beta(x_i) = 1$ , onde  $n$  é o número total de estados. Dessa forma, o valor obtido para  $p(\beta' | \beta, u)$  deve ser substituído na Equação 3.10 e integrado sobre todos os valores possíveis de  $\beta$ . Daí a complexidade desse problema, uma vez que o espaço de crenças é infinito.

A probabilidade de um estado  $x$  (de acordo com a distribuição de probabilidade  $\beta'$ ) é dada pela equação:

$$\beta'(x', u) = \frac{p(o|x', u)}{\int_{x'} (p(o|x') \int_x p(x'|x, u) \beta(x) dx) dx'} \int_x p(x'|x, u) \beta(x) dx. \quad (3.12)$$

onde, o denominador da fração é um fator de normalização. Dessa forma, tem-se que:

$$\pi^{[k]}(\beta) = \arg_u [V^{[k]}(\beta)], \quad (3.13)$$

onde pode ser observado que as funções  $V^{[k]}$  e  $\pi^{[k]}$  são calculadas a partir da distribuição de probabilidades  $\beta$  sobre os estados do modelo.

Formalmente um POMDP pode ser definido por meio de uma tupla  $(X, O, U, \delta, \varphi, \gamma)$  onde  $X$ ,  $U$ ,  $\delta$ , e  $\gamma$  já foram definidos anteriormente,  $E$  é o conjunto finito de observações e  $\varphi : O | X \times U \rightarrow [0, 1]$  é a função que especifica a probabilidade de observar  $o$  dado o estado atual  $x$  e ação  $u$ .

Considera-se que uma *crença* sobre os estados do sistema é definida por meio de uma distribuição de probabilidades sobre esses estados. Se considerarmos que o *espaço de estados é discreto e finito* então o *espaço de crenças é contínuo e infinito* uma vez que infinitas distribuições de probabilidade podem ser empregadas na especificação de uma crença.

Por exemplo, considere uma situação na qual um robô possa estar em um

de dois estados,  $x_1$  ou  $x_2$ . Nesse caso tem-se um espaço de estados discreto e finito. Considere, agora, que o robô não tem como saber com exatidão em qual estado se encontra e, portanto, define uma distribuição de probabilidades  $\beta$  (crença) sobre os estados de forma que  $\beta(x_1) = p_1$  e  $\beta(x_2) = p_2$ . Como  $p_1$  e  $p_2$  podem assumir infinitos valores no intervalo  $[0, 1]$ , desde que  $\sum_1^n p_i = 1$ , tem-se que o espaço de crenças (distribuições de probabilidades  $\beta$ ) é contínuo e infinito. Para um POMDP de horizonte finito, o algoritmo para o cálculo da solução exata pertence à classe PSPACE-completo [Papadimitriou & Tsitsiklis, 1987], enquanto que para o horizonte infinito, o problema é indecidível [Madani et al., 1999]. Enquanto no MDP o problema consiste em definir valores de utilidade para cada um dos estados de forma a dar condições ao robô de optar pela ação que o leve ao estado com maior utilidade esperada, no POMDP a escolha dessa ação está associada à busca em um espaço infinito de crenças. Logo, soluções para POMDP são obtidas por meio de heurísticas.

### 3.1.5 Tratamento de Erros para Times de Robôs

Em ambientes com mais de um robô, o POMDP pode ser estendido de forma a levar em conta a descentralização do sistema por meio de Processos de Decisão de Markov Descentralizados com Observação Parcial (DEC-POMDP) [Bernstein et al., 2002]. Por meio desse modelo, os robôs tomam decisões baseadas não apenas nas observações e ações locais, mas também nas observações e ações de outros agentes [Spaan, 2006b,a; Rosemary et al., 2004; Barbosa et al., 2009; Spaan et al., 2010; Messias et al., 2010]. Se a comunicação for perfeita e ilimitada, tem-se que um DEC-POMDP pode ser transformado em um POMDP e resolvido por meio dos métodos e heurísticas para o problema. Porém, esse tipo de consideração sobre comunicação perfeita e ilimitada não é válida na maioria dos casos. Logo, os robôs têm que tomar decisões baseando-se em crenças sobre o estado no qual se encontram os outros robôs.

Formalmente um DEC-POMDP pode ser definido da seguinte maneira:

- $\mathcal{R}$  é um conjunto finito de robôs;
- $X$  é o conjunto de estados do sistema e  $X' = \otimes_{i \in \mathcal{R}} x_i$  representa o estado no qual o time de robôs se encontra;
- $U$  é o conjunto de ações que podem ser executadas pelo robô e  $U' = \otimes_{i \in \mathcal{R}} u_i$  representa a ação conjuntas dos robôs;

- $\delta : X'|X' \times U' \rightarrow [0,1]$  é uma função de transição que representa a probabilidade de transitar para o estado conjunto  $x'_1$  dado que o time de robôs encontrava-se no estado conjunto  $x'_2$  e a ação conjunta  $u'$  foi executada;
- $O$  é o conjunto finito de observações e  $O' = \otimes_{i \in \mathcal{R}} e_i$  representa a observação conjunta;
- $\phi' : O'|X' \times U' \rightarrow [0,1]$  é uma função de observação que representa a probabilidade de observar  $O'$  dado que o time de robôs encontrava-se no estado  $X'$  e executou a ação  $U'$ .
- $\gamma : X' \times U' \rightarrow R$  é a função de reforço aplicada ao time de robô.

Como um DEC-POMDP é uma extensão do POMDP e do MDP, o critério de otimalidade para encontrar uma política de atuação conjunta pode ser definido como:

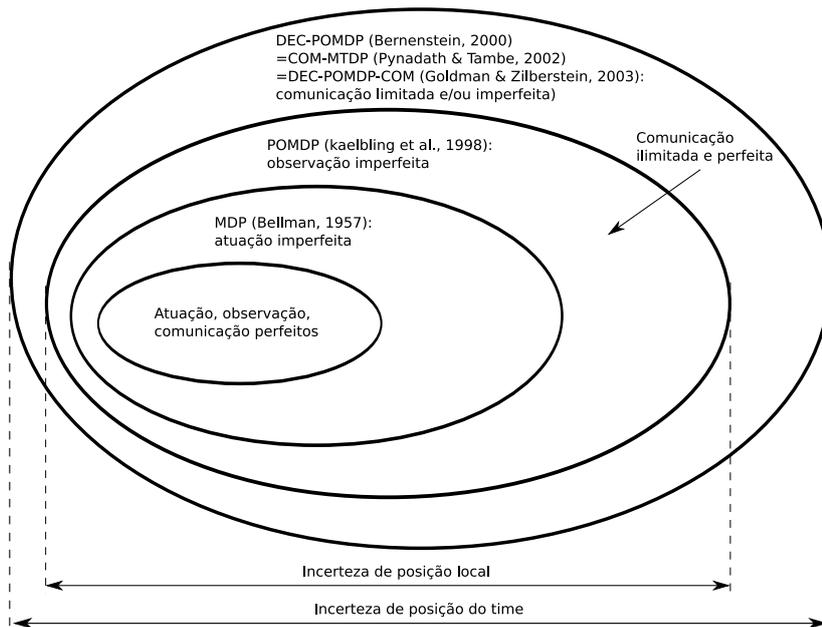
$$E\left[\sum_{t=0}^{\infty} \alpha^t \gamma^{[t]'}\right], \quad (3.14)$$

onde vê-se a utilização da função de reforço conjunta,  $\gamma'$ . Seuken & Zilberstein [2008] apresentam uma série de problemas que envolvem planejamento multi-robôs, tais como DEC-POMDP, COM-MTDP e DEC-POMDP-COM e provam a equivalência entre eles. Além disso são discutidas soluções algorítmicas exatas e aproximadas para esses casos. A discussão de tais soluções está fora do escopo desta tese.

## 3.2 Hierarquia de Problemas de Decisão

A Figura 3.2 mostra a hierarquia de problema de Decisão de Markov. O núcleo central apresenta problemas que não modelam erros de atuação, observação ou comunicação. Os problemas de atuação e observação são modelados por um MDP [Bellman, 1957] e POMDP [Kaelbling et al., 1998], respectivamente. Os problemas descentralizados de *Markov* encapsulam os outros problemas.

Nesse ponto, cabe uma observação: como tratam-se de problemas descentralizados com observação parcial, os estados nos quais os outros agentes se encontram é obtido por meio de uma estimativa de estados ou por meio de comunicação. Assim, a comunicação desempenha um papel importante no processo de cooperação, sendo que a sua modelagem também pode levar em conta incertezas inerentes ao processo. Existem duas modelagens para problemas de decisão com comunicação: i) COM-MTDP, (Communicative Multiagent Team Decision Process) [Pynadath &



**Figura 3.2.** Esquema e hierarquia dos Problemas de Decisão de *Markov*. De dentro para fora, a incerteza aumenta. Problemas de atuação e observação são apresentados como problemas de incerteza local enquanto que problemas de comunicação e coordenação com o time de robôs são apresentados como problemas de incerteza de time.

Tambe, 2002] e ii) DEC-POMDP-COM [Goldman & Zilberstein, 2003]. Esses problemas modelam a comunicação por meio de ações comunicativas. Esse mecanismo mostra-se interessante, pois no modelo tradicional os processos comunicativos são exteriores ao modelo. O trabalho de [Seuken & Zilberstein, 2008] mostrou que ambos modelos são equivalentes ao DEC-POMDP [Bernstein et al., 2002].

Soluções exatas para um DEC-POMDP pertencem à classe NEXP-completo [Bernstein et al., 2002] quando o horizonte é finito. Quando o horizonte é infinito o problema é indecidível [Madani et al., 1999].

# Capítulo 4

## Metodologia

Quanto maior o conhecimento do robô sobre ambiente no qual está atuando, maior será a probabilidade de que a tarefa seja realizada satisfatoriamente. Esse conhecimento pode ser adquirido, em parte, por meio da RSSF que, à medida que obtém dados do ambiente, pode criar um repositório de informações relevantes que podem ser repassadas aos robôs. Assim, a RSSF auxilia os robôs em algumas tarefas, dentre as quais pode-se citar exploração e navegação pelo ambiente [Batalin & Sukhatme, 2004].

Porém, em muitas aplicações, a RSSF precisa ser beneficiada pelo emprego de robôs móveis, por exemplo, no roteamento de pacotes e fusão de dados. Nesse tipo de tarefa, conhecer a localização do nó sensor é importante, uma vez que quanto maior é a proximidade com o nó sensor, menores são as chances de falha de comunicação. Porém, nesta tese, a posição dos nós sensores não é conhecida perfeitamente, de forma que as tarefas podem não ser realizadas de forma eficiente sem essa estimativa.

Neste capítulo, será apresentada uma estratégia cooperativa para que um time de robôs consiga explorar o ambiente e executar suas tarefas em um ambiente com incertezas de atuação, observação e posicionamento dos nós sensores.

### 4.1 Cooperação em RSSF utilizando robôs móveis

As tarefas a serem cumpridas nos cenários estudados nesta tese consistem na navegação de robôs para realizar, por exemplo, a coleta e fusão de dados, etc. As incertezas a serem tratadas dizem respeito a cada robô individualmente e a todo o time. Localmente, cada robô precisa tratar os problemas que surgem a partir da execução das ações (por exemplo, a ação não leva o sistema ao estado esperado)

e a partir da realização de observações (por exemplo, a observação é ruidosa). A incerteza em relação ao time de robôs surge quando é necessário compartilhar alguma informação ou ação com outros robôs. Essa troca de informação pode ser realizada de maneira mais eficiente se os robôs conseguirem reduzir as incertezas em relação às ações e estados dos outros componentes do time. Para tanto, estimar o estado de outros robôs torna-se um fator fundamental.

Considere, por exemplo, um cenário no qual a posição dos nós sensores é totalmente desconhecida. Como as tarefas de visitação obviamente dependem do conhecimento dessas posições, uma possível estratégia do robô seria navegar aleatoriamente coletando informação de localização dos nós sensores. Com base nesses dados, seria possível gerar um plano para executar as tarefas. Considere, no entanto, que existam vários robôs navegando no ambiente e que, em alguns casos, mais de um robô possa descobrir a posição de um nó sensor e alocar para si a tarefa de visitá-lo. Essa situação pode ser inadequada, uma vez que mais de um robô poderá ser responsável por um mesmo nó sensor, desperdiçando recursos. Esse problema pode ser resolvido se algum procedimento de negociação for adotado pelo time de robôs.

Dada a localização dos nós sensores (é importante lembrar que à estimativa da posição de um nó está associada uma incerteza), os robôs podem finalmente executar as tarefas às quais foram designados. Entretanto, o cenário pode ser dinâmico e não controlado: os sensores podem ter sua posição alterada em função de algum evento no ambiente. Nesse caso, a informação de posicionamento desses nós será diferente da armazenada nos robôs. Chavear novamente para uma ação de navegação aleatória poderia ser uma solução, mas com um alto custo, pois como a posição dos sensores é novamente desconhecida, o robô é obrigado a desviar de sua rota planejada em busca desse nó que, eventualmente, poderá nunca ser encontrado.

## 4.2 Modelo

A solução para o problema definido nesta tese foi modelada da seguinte maneira:

- $\mathcal{R}$  é um conjunto finito de robôs;
- $\mathbf{X} = \{\otimes_{i \in \mathcal{R}} x_i, \otimes_{j \in \mathcal{D}} d_j\}$  é o estado conjunto do time, representado pelos estados individuais  $x_i \in X$  de cada robô e pela quantidade de dados  $d_j \in \mathcal{D}$  presente em cada nó sensor,  $0 \leq i \leq m$  e  $0 \leq j \leq n$ ;

- $\mathbf{U} = \bigotimes_{i \in \mathcal{R}} u_i$  é a ação conjuntas dos robôs, composta pelas ações  $u_i \in U$  de cada robô. Uma instância de  $U$  é  $(u_1, \dots, u_m)$ , onde tem-se as ações individuais dos  $m$  robôs. As ações  $u$  são comandos de navegação que o robô executa para mover-se do estado atual para algum estado vizinho: move-up, move-down, move-right, move-left, move-up-right, move-up-left, move-down-right, move-down-left;
- $\delta : \mathbf{X} | \mathbf{X} \times \mathbf{U} \rightarrow [0, 1]$  é uma função de transição que representa a probabilidade de transitar para o estado conjunto  $x^{[t+1]}$  dado que o time de robôs encontrava-se no estado conjunto  $x^{[t]}$  e a ação conjunta  $u^{[t]}$  foi executada;

– Considera-se que:

$$p(\mathbf{x}^{[t+1]} | \mathbf{x}^{[t]}, \mathbf{u}^{[t]}) = p(x_1^{[t+1]} | x_1^{[t]}, u_1^{[t]}) \dots p(x_m^{[t+1]} | x_m^{[t]}, u_m^{[t]}) \\ p(d_1^{[t+1]} | d_1^{[t]}, x_1^{[t]}, u_1^{[t]}, \dots, x_m^{[t]}, u_m^{[t]}) \\ p(d_n^{[t+1]} | d_n^{[t]}, x_1^{[t]}, u_1^{[t]}, \dots, x_m^{[t]}, u_m^{[t]}),$$

ou seja, a probabilidade de transição entre estados conjuntos é dada pela multiplicação das probabilidades de transição dos estados individuais dos robôs. Assume-se, portanto, que a ação executada por um robô não influencia as ações tomadas por outros robôs e que os eventos de transição de estado são independentes;

- Assume-se que  $p(d_j^{[t+1]} > 0 | d_j^{[t]} = 0, x_1^{[t]}, u_1^{[t]}, \dots, x_m^{[t]}, u_m^{[t]}) > 0$  e está relacionada à taxa de aquisição de dados  $g_j$  do nó sensor  $s_j$ ;
- Assume-se que  $p(d_j^{[t+1]} = 0 | d_j^{[t]} > 0, x_1^{[t]}, u_1^{[t]}, \dots, x_m^{[t]}, u_m^{[t]}) > 0$  e está relacionada à probabilidade de o robô encontrar-se no raio de comunicação do nó sensor  $s_j$ ;
- $O = \bigotimes_{i \in \mathcal{R}} o_i$  é a observação conjunta, composta pelas  $o'_i \in O'$  observações individuais de cada robô. Uma instância de  $O$  é  $(o'_1, \dots, o'_m)$ , onde tem-se as observações individuais dos  $m$  robôs;
- $\varphi' : \mathbf{O} | \mathbf{X} \times \mathbf{U} \rightarrow [0, 1]$  é uma função de observação que representa a probabilidade de observar  $e$  dado que o time de robôs encontrava-se no estado  $x$  e executou a ação  $u$ .
- $\gamma : \mathbf{X} \times \mathbf{U} \rightarrow \mathbb{R}$  é a função de reforço aplicada ao time de robô. A função de

reforço é definida como:

$$\gamma(\mathbf{x}, \mathbf{u}) = \sum_1^n f(d_j^{[t]}), \forall d_j \in \mathcal{D}, \quad (4.1)$$

onde:

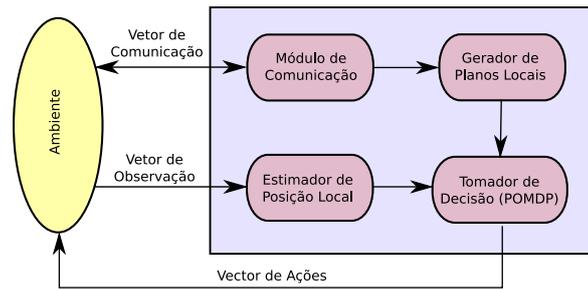
$$f(d_j^{[t]}) = \begin{cases} 0, & \text{se } d_j^{[t]} = 0, t = 0, \text{ ou seja, o nó sensor } s_j \text{ ainda não} \\ & \text{coletou nenhum dado} \\ 1, & \text{se } d_j^{[t]} = 0, t > 0, \text{ ou seja, o nó sensor } s_j \text{ já foi visitado} \\ 0, & \text{se } d_j^{[t]} > 0, t > 0, \text{ ou seja, o nó sensor } s_j \text{ não foi visitado.} \end{cases} \quad (4.2)$$

### 4.3 Abordagem Heurística Adotada

A abordagem utilizada nesta tese foi a divisão do time de robôs em equipes com características diferenciadas, que permitissem que tarefas específicas e não conflitantes pudessem ser executadas cooperativamente. Nesta tese, são utilizados dois tipos de robôs, descritos a seguir:

- **Robô Tarefa:** Com grande capacidade de armazenamento, são os robôs que realizarão as tarefas de atendimento à rede de sensores. Os principais módulos desse tipo de robô são i) *Alocador de Tarefas* e ii) *Tomador de Decisão*;
- **Robô Explorador:** Mais rápidos e com menor capacidade de armazenamento, têm a função de auxiliar os robôs tarefa, fornecendo informações sobre a posição dos nós sensores aos Robôs Tarefa. Os principais módulos desse tipo de robô são: i) *Estimador de posição do time* e ii) *Tomador de Decisão*.

A principal consequência da utilização desses dois tipos de robôs aparece nas etapas responsáveis pela tomada de decisão. Enquanto os Robôs Tarefa são responsáveis pela visitação previamente planejada aos nós sensores espalhados no ambiente, os Robôs Exploradores navegam pelo ambiente coletando a informação de posicionamento dos mesmos. Na abordagem adotada nesta tese, os Robôs Tarefa planejam suas rotas a partir da estimativa local de posicionamento dos nós sensores. A solução para esse problema, portanto, pode ser local, independentemente dos outros robôs. Em contrapartida, os robôs exploradores precisam transferir as informações coletadas na rede, uma vez o cumprimento da missão pelo time depende dessas informações.

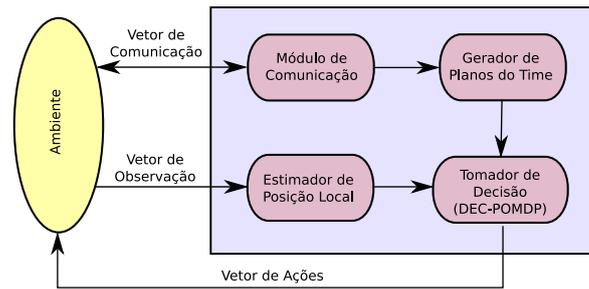


**Figura 4.1.** Esquema de um Robô Tarefa. O *Módulo de Comunicação* recebe o vetor de comunicações do ambiente, composto de dados provenientes dos sensores ou mensagem enviadas por outros robôs. As mensagens referentes às posições dos sensores servem de entrada para o *Gerador de Planos Locais* responsável pela criação de rotas que especificam a ordem de visitação dos sensores. O vetor de observações é entrada para o *Estimador de Posição Local*. Por fim o *Tomador de Decisão* é o responsável por implementar a tomada de decisão com base em Processos Markovianos. O Tomador de Decisão gera uma ação que atua diretamente no ambiente.

### 4.3.1 Robôs Tarefa

Os módulos presentes nos Robôs Tarefa, mostrados na Figura 4.1, são descritos a seguir:

- **Módulo Comunicação:** Por meio desse módulo, as informações relevantes para cada um dos componentes do sistema podem ser trocadas. Essa camada implementa os protocolos de comunicação empregados entre as entidades (robôs e nós sensores) que compõem o ambiente. A sua funcionalidade é dependente da existência de dispositivos de radiocomunicação.
- **Estimador de Posição Local:** Módulo que recebe medidas dos sensores embarcados para que a estimação do estado local do robô, ou seja, sua posição seja realizada. A funcionalidade desse módulo depende da existência de dispositivos de sensoriamento, como sonares ou dispositivos *laser*.
- **Gerador de Planos Locais:** Os planos locais de cada robô especificam a ordem na qual os nós sensores alocados aos robôs devem ser visitados. Nesta tese, o plano é gerado por meio de algum algoritmo de otimização para o PRV.
- **Tomador de Decisão (POMDP):** Esse módulo recebe informações enviadas pelos módulos Estimador de Posição Local e Gerador de Planos Locais e especifica a seqüência de ações a ser executada. Como o plano precisa ser robusto para tratar as incertezas do ambiente, é nesse módulo que se implementa a



**Figura 4.2.** Esquema de um Robô Explorador. Diferencia-se do esquema do Robô Tarefa ao introduzir os módulos *Gerador de Planos do Time* e *Tomador de Decisão*. O estimador de posições do time é um módulo que permite obter a estimativa sobre os planos gerados pelos outros robôs no ambiente. O conhecimento dessa informação é indispensável para o *Tomador de Decisão*, que implementa uma solução para um DEC-POMDP.

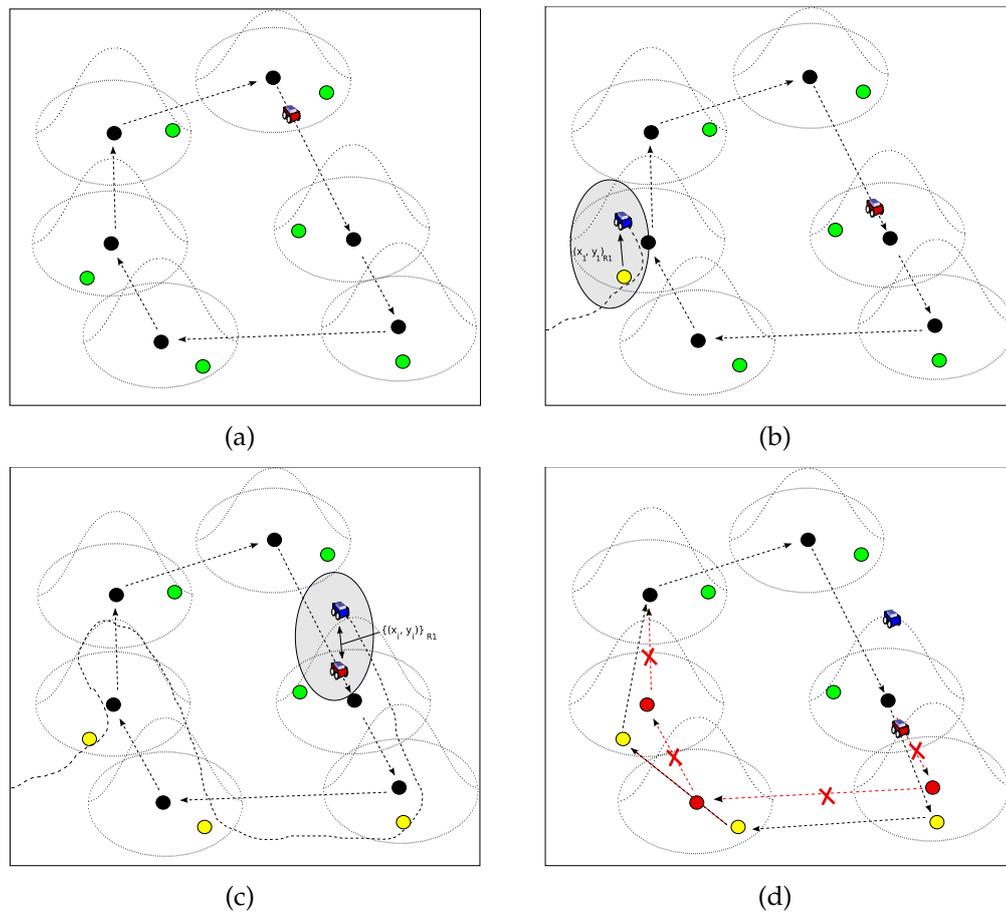
tomada de decisão dos robôs, utilizando para tal, ferramentas como *Processos de Decisão de Markov com Observação Parcial*, conforme descrito na Seção 3.1.4.

### 4.3.2 Robôs Exploradores

O Esquema dos Robôs Exploradores, mostrado na Figura 4.2, apresenta os mesmos módulos dos Robôs Tarefa, com exceção de:

- **Gerador de Planos do Time:** Este módulo computa o plano local de cada Robô Tarefa que está sob a responsabilidade do Robô Explorador. Os nós sensores são alocados aos Robôs Tarefa pelos Robôs Exploradores, que conhecem, portanto, a posição estimada desses nós sensores no momento da alocação. Com isso, os Robôs Exploradores podem computar o plano para visitaç o da mesma forma que os Robôs Tarefa.
- **Tomador de Decisão (DEC-POMDP):** Em um sistema cooperativo, no qual robôs tarefa dependem n o s o das informa es coletadas por eles mesmos, mas tamb m das informa es repassadas por outros rob s tarefa e exploradores, a estimac o da posic o dos outros rob s do time torna-se necess ria sempre que alguma informa o nova precisar ser comunicada com o m nimo custo, em uma situa o similar ao problema do *rendezvous* (Bernstein et al. [2002]). Com base nas informa es de posicionamento dos componentes do time, os rob s exploradores podem definir a melhor pol tica de atua o, visando reduzir os custos associados   execu o de tarefas.

Esses dois m dulos s o respons veis por estimular a coordena o e coopera o dos rob s. O primeiro passo nesse sentido   a estimac o dos planos locais



**Figura 4.3.** Troca de informações entre os robôs. (a) Robô Tarefa realizando uma tarefa de visitação em uma rede de nós sensores. (b) Robô Explorador estima a posição  $(x_1, y_1)$  do nó sensor  $s_1$  ao atingir seu raio de comunicação. (c) Robô Explorador repassa a informação sobre a posição dos nós sensores para o Robô Tarefa. (d) Robô Tarefa atualiza a rota com novas informações obtidas.

de cada Robô Tarefa por parte do Robô Explorador. Esses planos irão capturar, na visão do Robô Explorador, as incertezas locais de cada Robô Tarefa. São dois os principais problemas decorrentes dessa abordagem: i) os robôs não executam suas ações de maneira sincronizada, dessa forma a estimação sobre a posição dos robôs pode estar atrasada ou adiantada e ii) os robôs tem conhecimento local, ou seja, se a posição de um nó sensor for atualizada, não é garantido que os outros robôs possuirão esse conhecimento.

## 4.4 Comunicação

Neste trabalho, assume-se que a comunicação é perfeita, porém com alcance limitado. Dessa forma, os robôs e os nós sensores não terão omnisciência do estado no qual o sistema se encontra. As informações de todos os robôs serão locais. Assim, o conhecimento sobre a rede pode ser incrementado por meio de processos comunicativos que permitam reduzir a incerteza sobre o posicionamento do time de robôs e dos nós sensores.

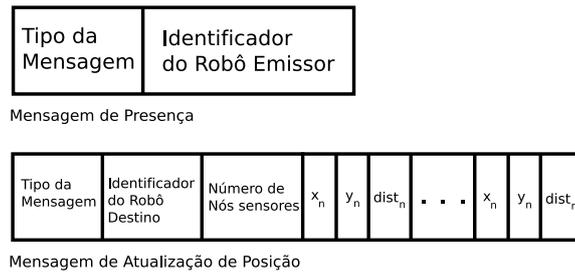
A Figura 4.3(a) apresenta uma rede composta de seis nós sensores com posições estimadas representadas por círculos pretos e posições reais representadas por círculos verdes. Um robô tarefa realiza um procedimento de coleta de dados por uma rota previamente planejada. Na figura, considera-se que o robô coletará a informação de um nó sensor quando passar sobre a sua posição. Nesse caso, nota-se que essa tarefa não será realizada eficientemente, uma vez que o robô não alcançará todos os nós sensores.

Na Figura 4.3(b), um Robô Explorador, representado por meio de robô azul, navega aleatoriamente no ambiente e atravessa o raio de comunicação do nó sensor  $s_1$ . Essa é uma oportunidade para o robô explorador estimar a posição do nó sensor uma vez que a comunicação é estabelecida. Uma possível abordagem para efetuar essa estimativa é utilizar TDoA e trilateração conforme descrito no Capítulo 1.

Nesta tese, assume-se que a estimativa da distância entre um robô e um nó sensor é perfeita. Entretanto, a precisão da estimativa sobre o posicionamento do nó sensor dependerá da localização do robô (que é incerta). O procedimento de trilateração é simulado por meio da geração dessa estimativa usando uma distribuição de probabilidades com média na posição real do sensor e desvio padrão igual à distância entre o nó sensor e o robô.

A troca de informações entre os robôs é mostrada na Figura 4.3(c), onde vê-se que o Robô Explorador comunica a posição estimada para o Robô Tarefa. Junto à posição estimada é associado um valor de “distância estimada”, obtido, por exemplo, por RSSI. Com base nessa informação de distância, o Robô Tarefa tem condições de avaliar se deve manter o seu conhecimento prévio sobre a posição do sensor ou modificá-lo com a informação recebida. Essa decisão depende do valor absoluto da distância estimada: o robô sempre modificará seu conhecimento quando a distância estimada for menor que a distância atual associada àquele nó sensor.

Os tipos de mensagens trocadas entre os robôs são classificadas nas seguintes categorias:

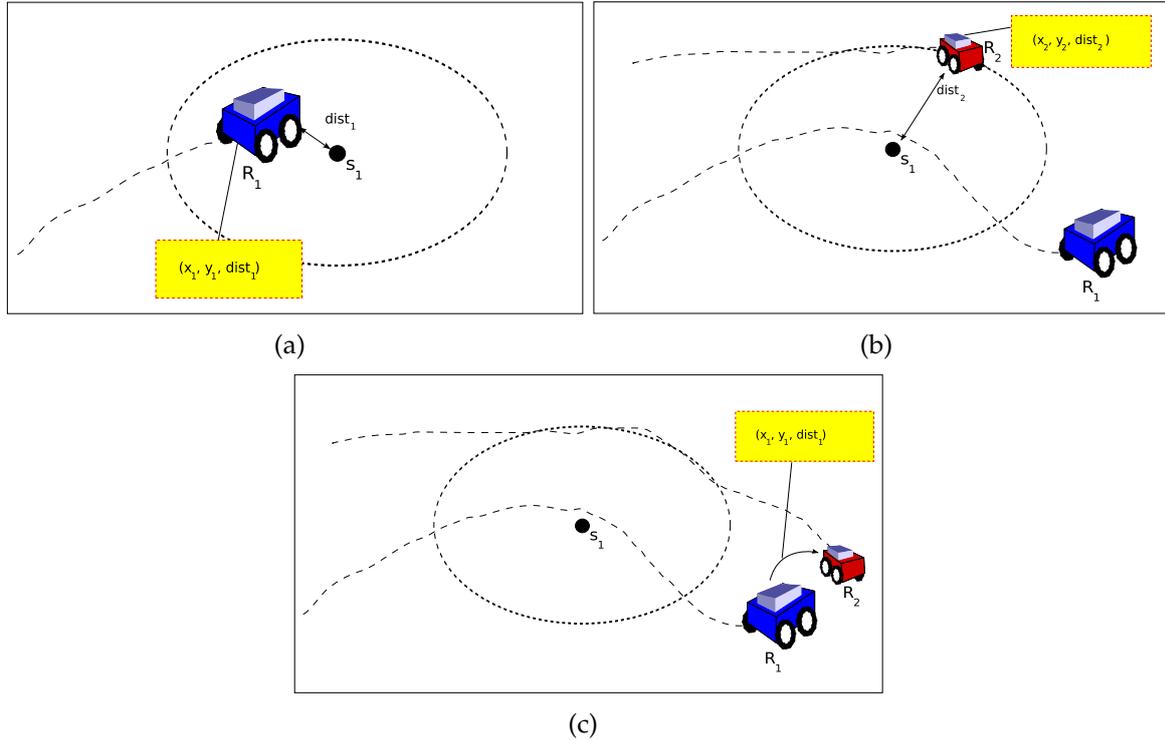


**Figura 4.4.** Campos das mensagens de “Presença” e “Atualização de Posição”. A mensagem de “Presença” contém um campo com o identificador do robô emissor para ser reconhecido pelos outros robôs. A mensagem de “Atualização de posição” é enviada em resposta ao recebimento da mensagem de “Presença” com a informação de localização dos nós sensores. Essa mensagem contém o identificador do robô destino que enviou a mensagem de “Presença”.

- **Mensagem de Presença:** Esse tipo de mensagem é constantemente emitido pelos robôs para indicar a sua presença no ambiente. Assim que outro robô recebe essa mensagem, o processo de transferência de dados pode ser iniciado. As mensagens desse tipo têm em seu conteúdo a identificação do robô emissor (Figura 4.4);
- **Mensagem de Atualização de Posição:** Quando um robô recebe uma mensagem de presença, tem início a transferência dos dados referentes à posição dos sensores. Esse processo é mostrado na Figura 4.5. Em (a), vê-se o robô  $R_1$  estimando a posição  $(x_1, y_1)$  enquanto que em (b) o robô  $R_2$  obtém a estimativa  $(x_2, y_2)$ . Quando os robôs se comunicam em (c) a troca de dados é feita por meio de uma mensagem de atualização de posição. Ambos robôs transmitem as informações referentes à posição dos nós. O robô receptor atualiza suas tabelas de posição sempre que a “distância estimada” associada à posição do nó sensor presente na mensagem for menor que a “distância estimada” atual de posição. O conteúdo da mensagem é composto do identificador do robô remetente, identificador do robô destinatário (essa informação é obtida analisando a mensagem de presença enviada), número de sensores sob sua responsabilidade e as posições estimadas de cada nó sensor (Figura 4.4).

## 4.5 Gerador de Planos Locais

Esse módulo gera um plano que estabelece a ordem de visitação dos nós sensores. A alternativa adotada nesta tese foi gerar esse plano de acordo com uma solução para o PRV [Soares et al., 2007a]. No PRV, tem-se um conjunto de clientes e um



**Figura 4.5.** Estimação da posição dos nós sensores. (a) Robô  $R_1$  estima a posição do nó sensor  $s_1$  com  $dist_1$ . (b) Robô  $R_2$  estima a posição do nó sensor  $s_1$  com  $dist_2$ . (c) Como a “distância estimada” associada ao robô  $R_1$  é menor que a “distância estimada” associada ao robô  $R_2$ , o robô  $R_2$ , atualiza sua estimativa sobre  $s_1$ .

conjunto de veículos para atender esses clientes.

Considere a instância  $I = (G, W, Q, D)$  onde:

1.  $G = (\mathcal{S}', \mathcal{A})$ , onde  $G$  é um grafo,  $\mathcal{S}' = \{s_0\} \cup \mathcal{S}$ ,  $\mathcal{A}$  é o conjunto de arestas ligando os elementos de  $\mathcal{S}'$ , sendo que  $|\mathcal{S}'| = (n + 1)$  e  $s_0$  é a unidade central;

2.  $W = \begin{bmatrix} w_{00} & \cdots & w_{0n} \\ & & \vdots \\ w_{n0} & \cdots & w_{nn} \end{bmatrix}$  é uma matriz de custos  $(n + 1) \times (n + 1)$  onde cada elemento  $w_{ij}$  representa o custo (energia, distância, tempo, etc) associado à aresta que liga o nó  $i$  ao nó  $j$ .

3.  $Q = [Q_1, \cdots, Q_m]$  é um vetor de  $m$  elementos representando as capacidades associadas aos robôs.

4.  $D = [C_1, \cdots, C_n]$  é um vetor de  $n$  elementos representando as capacidades máximas dos nós sensores.

A solução para o problema do PRV consiste em achar uma matriz  $F$ , de dimensão  $(n + 1) \times (n + 1) \times m$ , onde cada elemento  $f_{ijk}$  tem o seguinte significado:

$$f_{ijk} = \begin{cases} 1, & \text{se o robô } R_k \text{ visita o nó } s_j \\ & \text{imediatamente após ter visitado} \\ & \text{nó } s_i \\ 0, & \text{de outra maneira} \end{cases}, \quad (4.3)$$

de forma que o somatório

$$\sum_{i,j=0}^n w_{ij} \sum_{k=1}^m f_{ijk} \quad (4.4)$$

seja minimizado. O número de robôs pode variar, mas assume-se que  $m > 0$  e  $m \in \mathbb{I}$ . As seguintes restrições devem ser consideradas para a solução do problema:

$$\sum_{k=1}^m \sum_{j=1}^n f_{ijk} = \begin{cases} 1, & 1 \leq i \leq n \\ m, & i = 0 \end{cases}, \quad (4.5)$$

$$\sum_{k=1}^m \sum_{i=1}^n f_{ijk} = \begin{cases} 1, & 1 \leq j \leq n \\ m, & j = 0 \end{cases} \text{ e} \quad (4.6)$$

$$\sum_{i,j=1}^n D_i f_{ijk} \leq Q_k, \quad 1 \leq k \leq m. \quad (4.7)$$

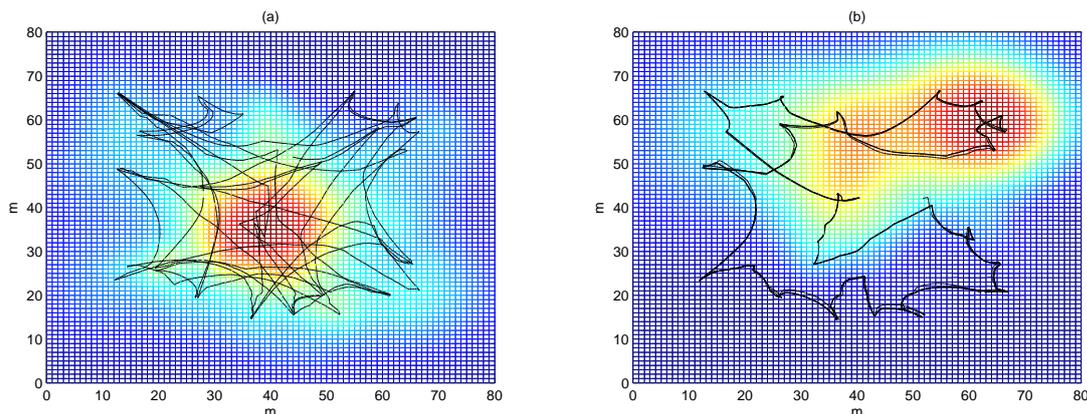
As restrições representadas pelas Equações 4.5 e 4.6 especificam que todos os nós sensores devem ser visitados por apenas um robô, com exceção do ponto de onde os robôs partem. A restrição representada pela Equação 4.7 especifica que a soma das capacidades  $D_i$  dos nós sensores visitados por um robô não deve ultrapassar a capacidade  $Q_k$  desse robô, pois de outra forma ocorrerá perda de dados.

Finalmente, o conjunto  $\mathcal{P}$  estabelece uma rota:

$$\mathcal{P} = \{(s_i \prec s_j) \mid s_i \in \mathcal{S}' \wedge s_j \in \mathcal{S}'\}, \quad (4.8)$$

onde  $i \neq j$ ,  $0 \leq i, j \leq n$  e  $\prec$  indica precedência.

A Figura 4.6 mostra duas rotas: em (a) a rota foi obtida sem otimização enquanto que em (b) foi utilizada uma solução para o PRV. A trajetória mostrada sugere que os nós são visitados de maneira mais eficiente.



**Figura 4.6.** Trajetórias em um ambiente com 60 nós sensores. (a) Rota obtida sem planejamento do PRV. (b) Rota obtida por meio de planejamento do PRV. As áreas mais claras indicam as regiões com maior urgência de atendimento.

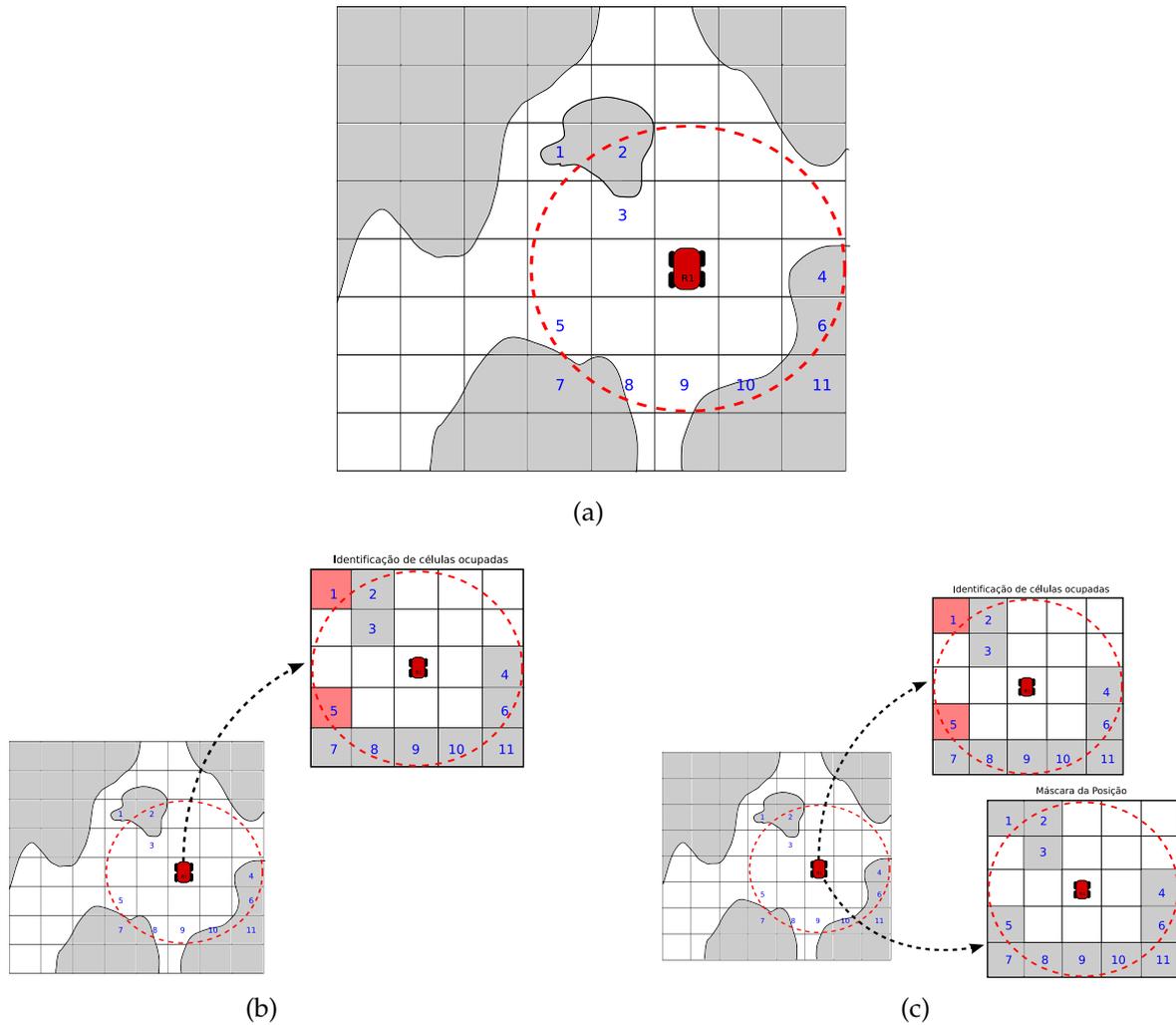
## 4.6 Estimador de Posição Local

Os robôs possuem uma crença sobre o estado do sistema que é representada por uma distribuição de probabilidades. Considere que  $\beta$  é a distribuição de probabilidade sobre os estados do sistema. A probabilidade de estar no estado  $x$  é representada por  $\beta(x)$ . Ao executar a ação  $u$ , o robô modifica essa distribuição de probabilidades da seguinte forma:

$$\beta'(x) = \sum_i p(x | x_i, u) \beta(x_i). \quad (4.9)$$

onde  $1 \leq i \leq |X|$  e  $X$  é o conjunto de variáveis de estado do sistema e  $|\cdot|$  é a cardinalidade desse conjunto.

O *Estimador de Posição Local* recebe os dados obtidos pelos sensores embarcados no robô. Aqui, essas observações indicam a ocupação de uma célula do mapa. A abordagem adotada para estimar o estado do robô é conhecido como “Localização em Grid” [Simmons & Koenig, 1995] e está ilustrada na Figura 4.7. Na Figura 4.7(a) vê-se que o ambiente é discretizado em células e a ocupação de uma célula é dada pela cor escura. A partir da posição do robô, é possível obter uma matriz de leituras (Figura 4.7(b)) que indica as células que são reconhecidas pelos sensores do robô como ocupadas. Percebe-se que as células 1 e 5 não foram reconhecidas como ocupadas devido à oclusão causada por outras células ou devido à limitação de alcance do dispositivo de sensoriamento. Na Figura 4.7(c), essa matriz de leituras é comparada com uma máscara de mesma dimensão cujo célula central representa a posição onde o robô poderá estar. O estado de todas as células vizinhas à célula



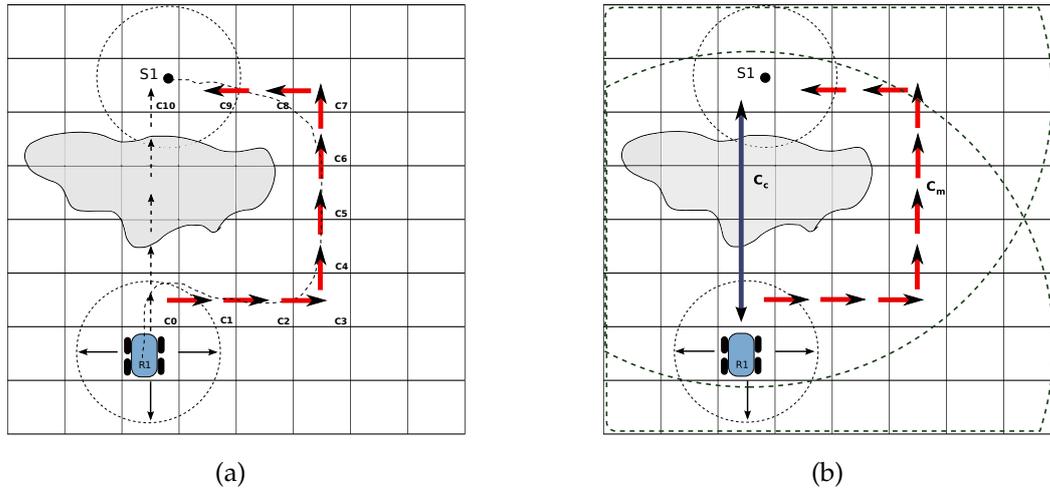
**Figura 4.7.** Estimação de posição local. A partir da posição do robô, é possível avaliar a probabilidade  $\beta'(x)$  de o robô estar no estado  $x$ . Isso é feito por meio de uma máscara, para cada estado  $x$  e da aplicação das Equação 4.11 para cada célula ocupada reconhecida pelos dispositivos de sensoriamento do robô.

central é facilmente obtido uma vez que o mapa é conhecido previamente. Quanto maior for o número de casamento de células (células escuras de uma matriz coincidem com as células escuras de outra matriz) maior será a probabilidade de o robô se encontrar nessa célula central.

A probabilidade sobre os estados é então modificada em função de todas essas observações da seguinte maneira:

$$\beta''(x) = \prod_j p(o_j | x) \beta'(x), \quad (4.10)$$

onde  $1 \leq j \leq \|O\|$  e  $O$  é o conjunto de contém as observações do sistema. Essa



**Figura 4.8.** Ambiente mostrando a política de atuação obtida por um MDP. Em (a), a trajetória indicada pela seta tracejada representa o plano do PRV. Esse plano determina uma política de atuação para tratar as incertezas de atuação e observação. Em (b)  $C_c$  e  $C_m$ , representamos, respectivamente, o custo de comunicação e o custo de movimentação do robô. Se  $C_c < C_m$ , o robô comunicará, ao invés de mover-se até o local de transferência dos dados.

equação é exemplificada da seguinte maneira:

$$\beta''(x) = \prod_j p(\text{observar } x_j \text{ ocupado a partir de } x \mid x_j \text{ é ocupado}) \beta'(x), \quad (4.11)$$

onde  $x_j$  é uma célula vizinha e  $x$  é a célula central da matriz.

## 4.7 Tomador de Decisão

O módulo responsável pela tomada de decisão computa a melhor política de atuação a ser tomada para cada estado do sistema. Esse procedimento precisa levar em consideração os erros de atuação e observação.

### 4.7.1 Tomada de Decisão em Resposta a Erros de Atuação

Considere, como exemplo, o ambiente ilustrado na Figura 4.8(a). Considere, também, que o robô, ao partir da célula  $C_0$  estabelece um plano de ações indicado pela seta tracejada. Essa navegação é constituída por uma série de deslocamentos entre as células como: `move_up`, `move_down`, `move_left`, `move_right`. Porém, ao executar o plano, o robô depara-se com um obstáculo imprevisto, que o impossibilita de seguir para a célula seguinte do plano. Diante dessa situação, o robô não terá outra

alternativa a não ser utilizar navegação reativa com base na informação de sonares, *lasers*, etc, com o objetivo de evitar a colisão com os obstáculos e chegar à célula C10. Nesta tese, o desvio de obstáculos é feito por meio do *Algoritmo Vortex Field* [de Luca, 1994], que gera um vetor resultante que depende do vetor componente de repulsão (devido ao obstáculo) e ao vetor gerado em função do MDP.

Neste trabalho, a abordagem adotada para navegação eficiente é a utilização de soluções para um MDP que, conforme apresentado no Capítulo 3, têm a função de estabelecer essas políticas de atuação, usando para isso, probabilidades de transições entre as células e custos associados a cada ação. Nesse tipo de planejamento, o robô tem observação total, o que significa que ele conhece com absoluta certeza o estado no qual se encontra. Os casos tratados nesta tese têm observação parcial, mas a solução adotada nesta tese é baseada em uma heurística que resolve inicialmente um MDP.

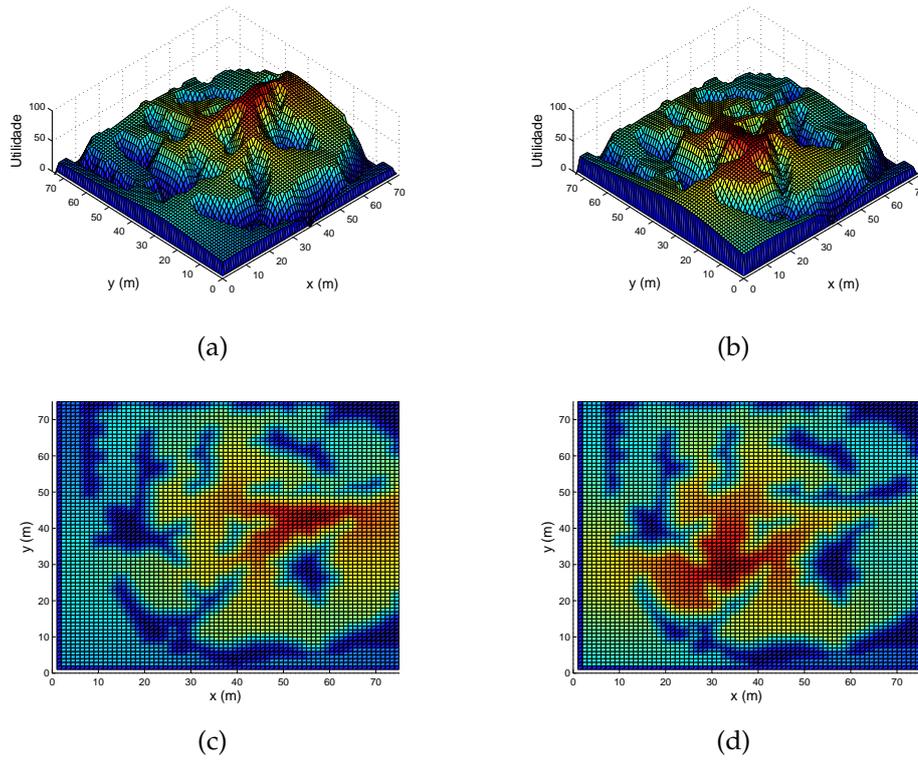
A Figura 4.9 apresenta a solução de um MDP por meio do Algoritmo de Iteração por Valor, para um ambiente com obstáculos. O ponto com maior utilidade representa o ponto alvo, ao qual o robô deve chegar. Percebe-se que a melhor política de atuação para cada célula é uma ação que direciona o robô diretamente para o alvo.

**Método de Comunicação.** Na Figura 4.8(b) é exibido o mesmo ambiente com dois raios de comunicação:  $r_{min}$  e  $r_{max}$ . Nesta tese considera-se que o robô consegue comunicar com qualquer entidade que esteja no raio de comunicação  $r_{max}$ , porém quanto maior a distância em relação à posição do robô, maior é o custo de comunicação. Além disso a transferência de dados estará mais sujeita a interferências e perturbações quanto mais distante estiver da fonte. Dessa forma o valor do reforço da Equação 3.3 é atualizado para:

$$\gamma(x, u) = \begin{cases} +a - C_c(R_i, s_i) & \text{se a execução de } u \text{ atinge um estado final} \\ & \text{e comunicação com } s_i \text{ é estabelecida} \\ +a - C_c(R_i, R_j) & \text{se a execução de } u \text{ atinge um estado final} \\ & \text{e comunicação com } R_j \text{ é estabelecida} \\ -b & \text{de outra maneira,} \end{cases} \quad (4.12)$$

onde  $b$  é um valor de penalização,  $a$  é um valor de recompensa,  $C_c(R_i, s_i)$  é o custo de comunicação entre o robô  $R_i$  e o nó sensor  $s_i$  e  $C_c(R_i, R_j)$  é o custo de comunicação entre o robô  $R_i$  e o robô  $R_j$ .

O método de comunicação trabalha sobre o planejamento prévio obtido por



**Figura 4.9.** Política de atuação para POMDP. (a)-(b) Mostram a política de atuação para dois pontos do ambiente (c)-(d) Visão de topo de (a) e (b), respectivamente

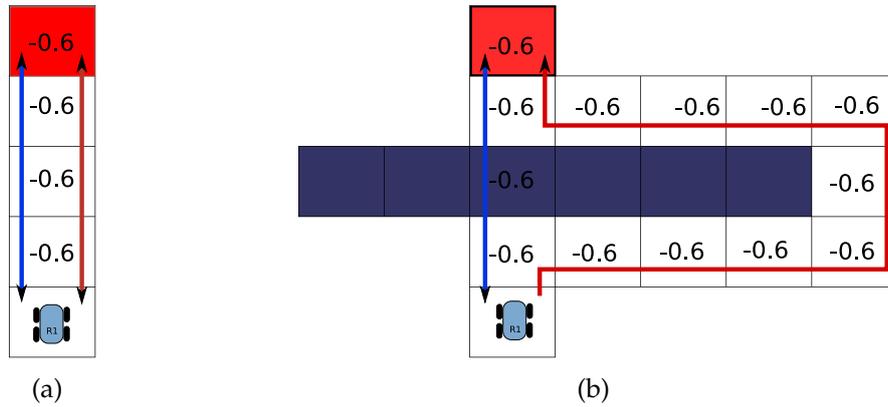
meio da solução para o MDP. Enquanto na Figura 4.8(b) vê-se uma trajetória marcada por setas de custo total  $C_m$ , o robô têm a opção de comunicar diretamente com alguma entidade (sensor, robô), se  $C_c < C_m$ , sendo que:

$$C_c = c(\|q_R - q_i\|), \quad (4.13)$$

onde  $c$  é um valor que representa o custo (sempre maior ou igual a zero) para comunicar na mesma célula,  $q_R$  é posição do robô e  $q_i$  é a posição do  $i$ -ésimo nó sensor.

Nas Figuras 4.10(a)-(b) são mostrados dois exemplos de funcionamento da método. Em (a) tem-se uma trajetória retilínea. Nesse caso, o funcionamento da método pode ser resumido da seguinte maneira:

- $c = b$ : nesse caso a comunicação não é a melhor ação a ser tomada, pois, embora o custo  $C_m$  de movimentação seja igual, a comunicação será menos sujeita a interferência se for realizada com maior proximidade.
- $c < b$ : a comunicação sempre será mais vantajosa, pois  $C_c < C_m$ ;



**Figura 4.10.** (a) O robô precisa navegar até a célula vermelha. Cada ação é penalizada com  $b = 0.6$ . Nesse caso, o robô preferirá comunicar (seta azul) somente se  $c < 0.6$ . (b) Quando a trajetória não é representada por um linha reta (como na maioria dos casos) o parâmetro de comunicação  $c$  pode ser grande suficiente a ponto de compensar o custo total de navegação  $C_m$ .

- $c > b$ : a movimentação sempre será mais vantajosa, pois  $C_m < C_c$

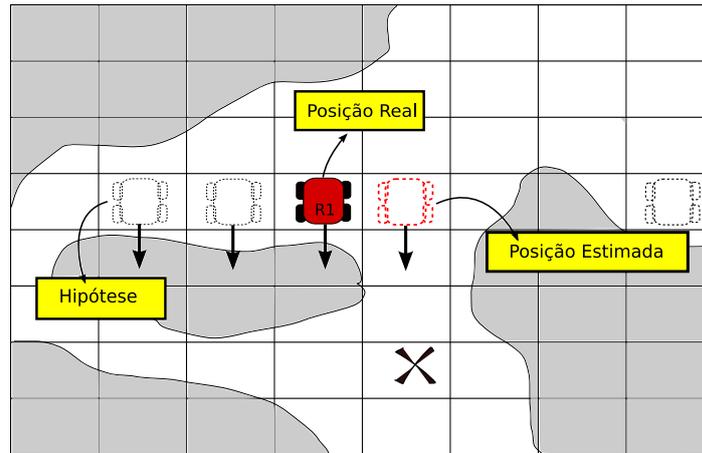
Já na Figura 4.10(b), considere que  $n_m$  é o número de células necessárias para o robô navegar de sua posição até o alvo e  $n_c$  o número de células entre a posição do robô e o alvo (considerando-se uma linha reta). Então, tem-se que  $C_m = (b)n_m$ . Dessa forma, sempre será vantajoso comunicar quando:

$$C_c < C_m = (c)n_c < (b)n_m = c < \frac{(b)n_m}{n_c}. \quad (4.14)$$

#### 4.7.2 Tomada de Decisão em Resposta a Erros de Observação

A tarefa de um tomador de decisão que trata erros de observação (POMDP) é similar à tarefa de um módulo que trata apenas erros de atuação: encontrar um plano que represente a melhor alternativa para cumprimento da tarefa em um cenário ruidoso. A diferença fundamental reside no fato de que o robô não tem mais certeza do estado no qual se encontra.

A Figura 4.11 ilustra um cenário no qual o robô precisa tomar uma decisão para alcançar o alvo marcado com "X". Porém, a estimativa de sua posição difere da posição real. A Figura 4.11 mostra ainda a distribuição de probabilidades sobre os estados. Caso o robô confiasse plenamente em sua estimativa de estados a melhor ação a ser tomada o levaria a colidir diretamente com o obstáculo mostrado. Isso aconteceria porque a posição estimada do robô indica que o caminho a sua frente está livre de obstáculos, sugerindo dessa forma uma ação do tipo `move_up`.



**Figura 4.11.** Observação parcial do ambiente. Robôs com tracejado escuro representam a distribuição de probabilidades sobre a posição do robô, o robô com tracejado vermelho representa o estado estimado e o robô rotulado como  $R_1$  representa a posição real. As setas indicam uma ação possível a ser executada em cada estado. A ação indicada não é ideal, uma vez que para três estados da distribuição de probabilidades, o robô colidiria com um obstáculo.

Na prática, o robô poderia não colidir, se o desvio de obstáculos tiver sido implementado satisfatoriamente, porém uma ação inadequada teria sido tomada e esse processo poderia se repetir indefinidamente.

Porém, se a distribuição de probabilidades fosse considerada, a ação de `move_up` poderia não ser a ação escolhida, uma vez que para três estados presentes na distribuição de probabilidades, o robô colidiria diretamente com a parede.

A solução para um POMDP não pode ser encontrada em tempo polinomial. A solução adotada nesta tese é a utilização da Heurística QMDP [Littman et al., 1995b], apresentada no Algoritmo 2.

No passo 1, um valor de utilidade é computado para cada estado do ambiente, por meio da solução para um MDP apresentada na última seção. No passo

---

#### **Algoritmo 2** Heurística para solução do POMDP

---

1. Calcular o valor da utilidade  $\hat{V}$  para cada estado  $x_i$  do ambiente, por meio do algoritmo para resolução do MDP.

2. Para cada ação  $u$  calcule  $M(x_i, u) = \gamma(x_i, u) + \sum_{j=1}^N \hat{V}(x_j) p(x_j | u, x_i)$ .

3. Retorne  $\arg \max_u \sum_{i=1}^N p_i M(x_i, u)$

---

2, vê-se que o valor armazenado em  $M(x_i, u)$  representa o valor da utilidade obtido no estado  $x_i$  ao executar a ação  $u$ , considerando-se a transição para todos os estados vizinhos, uma vez que  $\sum_{j=1}^N \hat{V}(x_j)p(x_j | u, x_i)$  representa a soma sobre todos os estados vizinhos de  $x_i$ . Quanto maior o valor de  $\hat{V}(x_j)$ , maior o valor de  $M(x_i, u)$ . O passo 3 retorna a ação que maximiza o valor esperado de  $M(x_i, u)$ , uma vez que o somatório considera a distribuição de probabilidades sobre os estados ( $p_i = \beta''(x_i)$ ).

Dessa forma, na Figura 4.11 a ação indicada por setas pretas não deverá ser selecionada, uma vez que a o valor da utilidade  $\hat{V}(x_j)$  é menor para células ocupadas, o que levaria a um valor menor de  $M(x_i, u)$  ao executar a ação  $u$  (no caso, `move_up`) no estado  $x$ .

### 4.7.3 Tomada de Decisão Descentralizada

O processo de decisão nos Robôs Exploradores é diferente do processo de decisão nos Robôs Tarefa. Conforme dito anteriormente, os Robôs Tarefas executam suas tarefas independentemente dos outros robôs. Portanto o processo descrito nas duas últimas seções é aplicado. O mecanismo de cooperação empregado nos Robôs Exploradores é baseado no fato de que é possível computar uma estimativa da posição na qual se encontra o time de robôs. Esse é um fator importante porque os robôs precisam transferir as informações sobre o posicionamento dos nós sensores coletadas na rede. Quando essa estimativa é conhecida, os robôs podem navegar até uma posição de encontro (esse problema é conhecido como *Rendezvous*).

Com esse intuito, Robôs Exploradores modelam o processo de cooperação como um DEC-POMDP, similar ao processo descrito na última seção. Porém, nesse caso, a função de reforço apresentada em 4.12 precisa ser modificada:

$$\gamma(x, u) = \begin{cases} +a - C_c(R_i, R_j), & \text{se a ação } u \text{ leva ao estado final} \\ & x = (x_{R_T}^{[t_i]}, x_{R_S}^{[t_j]}), \\ & \text{e comunicação com } R_j \text{ é} \\ & \text{estabelecida,} \\ -b, & \text{de outra forma,} \end{cases} \quad (4.15)$$

onde  $x_{R_T}^{[t_i]}$  e  $x_{R_S}^{[t_j]}$  representa o estado final dos Robôs Tarefa e Explorador, respectivamente, e  $x_{R_T}^{[t_i]} = x_{R_S}^{[t_j]}$ .

No ponto de vista dos Robôs Exploradores, a rota realizada pelo Robô Tarefa

pode ser definida como:

$$\mathcal{P}_T = L_1 \otimes L_2 \otimes \dots \otimes L_n, \quad (4.16)$$

onde

$$L_k = \langle x_{i,k}^{[t_i]} \prec x_{i+1,k}^{[t_{i+1}]} \prec \dots \prec x_{j,k}^{[t_j]} \rangle, \quad (4.17)$$

o que significa que a seqüência  $L_k$  iniciada no estado  $x_i$  no tempo  $t_i$ , precede a visita do estado  $x_j$  no tempo  $t_j$ . Observe que  $s_i \in x_{i,k}$  e  $s_j \in x_{j,k}$  representam os nós sensores inicial e final na seqüência  $L_k$  dos nós a serem visitados. Os estados intermediários entre  $x_i$  e  $x_j$  representam os estados presentes no plano MDP. A combinação das seqüências  $L_k$  resulta no plano MDP completo empregado pelo Robô Tarefa. Além disso, as seguintes restrições precisam ser consideradas:

$$[t_{j,k}] \leq [t_{i,k+1}], \quad (4.18)$$

$$V(x_{i,k}^{[t_i]}) \leq V(x_{i+1,k}^{[t_{i+1}]}). \quad (4.19)$$

A primeira sentença especifica que a visita ao último estado na seqüência  $k$  precisa preceder a visita ao primeiro estado na seqüência  $k + 1$ . A segunda sentença especifica que o valor de utilidade no estado  $x_{i,k}^{[t_i]}$  é menor ou igual ao valor no estado  $x_{i+1,k}^{[t_{i+1}]}$  porque o robô navega em direção à célula com maior utilidade.

A rota de um Robô Explorador é definida da seguinte maneira:

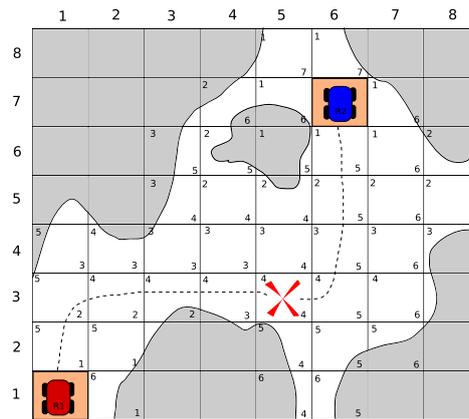
$$\mathcal{P}_S = \{L_k\}, \quad (4.20)$$

porém, nesse caso, a equação não é sujeita à Restrição 4.18 uma vez que as seqüências  $L_k$  são independentes umas das outras. O primeiro estado visitado na seqüência  $L_k$  tem  $t = 1$ , o segundo tem  $t = 2$  e assim por diante, até que algum estado limite seja alcançado (os limites horizontais e verticais da grade). Esse processo produz um mapa radial crescente.

Em síntese, com esse método, dada a seqüência  $\mathcal{P}_T$  representando a rota do Robô Tarefa, o conjunto  $\mathcal{P}_S$ , representando o conjunto de rotas possíveis do Robô Explorador, o estado final do sistema composto de  $x_{R_S}^{[t_j]}$  e  $x_{R_T}^{[t_i]}$  será selecionado considerando-se a simples restrição:

$$\min(t_j - t_i). \quad (4.21)$$

A Figura 4.12 mostra o procedimento: no canto inferior direito de cada célula, os tempos de ciclo do Robô Tarefa,  $R_1$ , são mostrados. De forma similar, no canto



**Figura 4.12.** Estimação da célula de encontro de um Robô Tarefa feito por um Robô Explorador. Robô  $R_1$  tem seus tempos de ciclo indicados no canto inferior direito de cada célula. Similarmente,  $R_2$  tem seus tempos de ciclo indicados no canto superior esquerdo de cada célula. Dessa forma, o robô  $R_2$  pode estar na célula (6,6) no tempo 2, na célula (5,6) no tempo 3, etc. O Robô Explorador estima a célula (3,5) como o local de encontro, pois essa célula possui a menor diferença entre os tempos de ciclo.

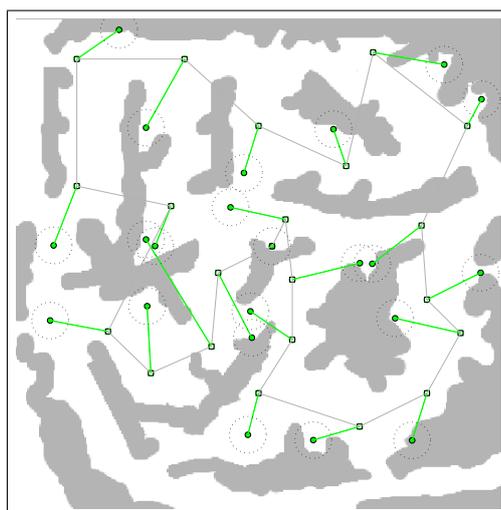
superior esquerdo, os tempos de ciclo do Robô Explorador,  $R_2$ , são mostrados. O robô  $R_1$  encontra-se inicialmente na célula (1,1) ao passo que o robô  $R_2$  encontra-se na célula (7,6). Perceba que, de acordo com o mapa,  $R_2$  pode se mover para todas as células de sua vizinhança e isso é indicado pelo tempo de ciclo 1 no canto superior esquerdo. A partir dessas células rotuladas com tempo de ciclo 1, o robô  $R_2$  pode se mover para todas as células rotuladas com 2 e assim por diante. Daí a denominação de mapa radial crescente. O mesmo esquema é aplicado para o robô  $R_1$ . A diferença entre os rótulos na célula (3,5) é igual a zero e, portanto, o Robô Explorador especifica esse local como a célula de encontro. Por simplicidade, nessa figura, para ambos os robôs, os tempos para cada célula foram estimados usando um mapa radial crescente.



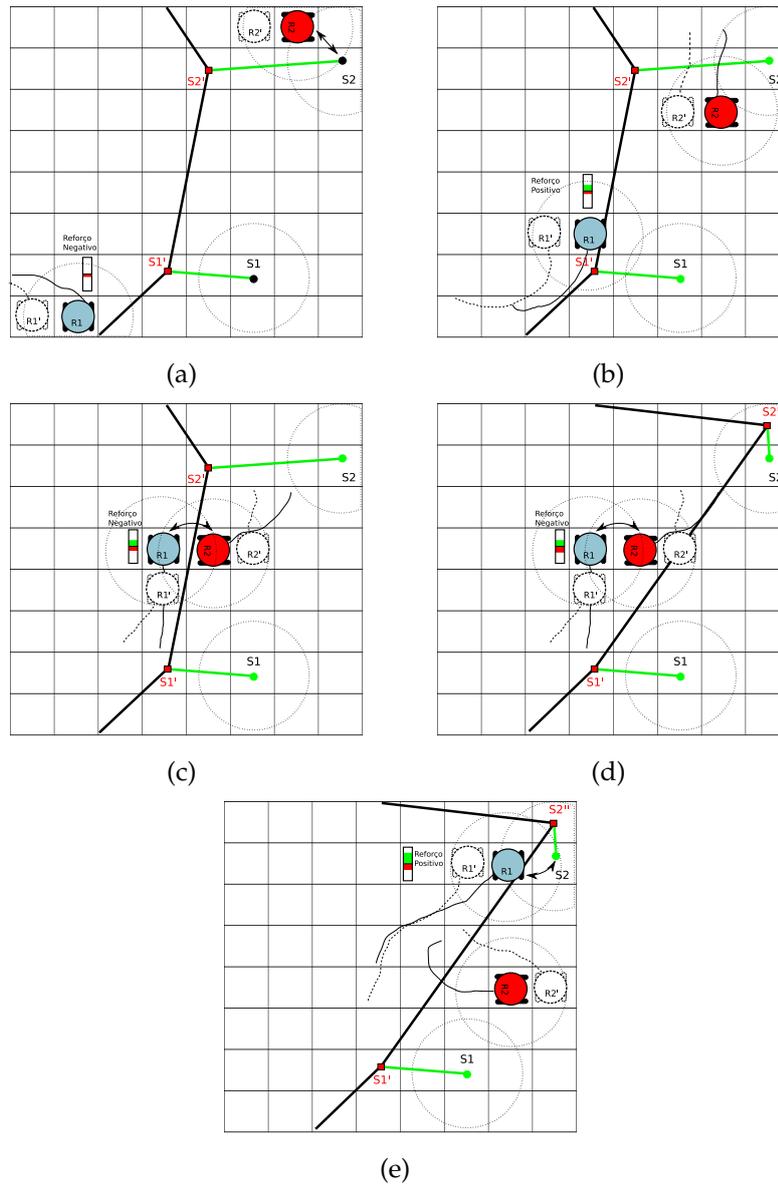
## Capítulo 5

# Resultados Experimentais

Com o objetivo de avaliar a metodologia proposta, foram realizadas simulações utilizando o *Player/Stage* [Gerkey et al., 2001] no ambiente mostrado na Figura 5.1(a). Nessa figura, tem-se 22 nós sensores, onde a posição real desses nós é mostrada em verde e a posição estimada é representada como um retângulo (as duas posições estão ligadas por meio de uma linha verde). Dada a posição estimada (incerta), os Robôs Tarefa geram uma trajetória por meio de solução para o PRV. No caso desta tese, as soluções para o PRV geram uma trajetória que passa pela posição de cada sensor no ambiente. Os Robôs Tarefa precisam visitar cada sensor, conforme especificado por sua rota. Nas simulações, foram utilizados apenas robôs holonômicos,



**Figura 5.1.** Ambiente de testes. O ambiente consiste em uma área de  $75\text{m} \times 75\text{m}$  com obstáculos. As posições reais dos sensores mostradas em verde estão conectadas as posições estimadas dos sensores (retângulos) por meio das quais a rota do PRV é gerada.



**Figura 5.2.** Ambiente de simulação. (a) um Robô Tarefa (vermelho) navega seguindo a rota do PRV sobre as posições estimadas dos nós sensores (quadrados vermelhos) e um Robô Explorador (azul) navega pelo ambiente. O Robô Tarefa é penalizado pelas ações de navegação que não o levem ao estado final (marcador vermelho na barra de reforço), representado pela posição estimada do nó sensor  $s_1$ . (b) O Robô Tarefa atinge posição final estimada  $s'_1$  e recebe um reforço positivo (marcador verde na barra de reforço). O Robô Explorador estima a posição  $s''_2$  ao comunicar com o nó sensor  $s''_2$ . (c) O Robô Explorador transfere a informação sobre a nova estimativa de  $s_2$  para o Robô Tarefa que (d) ajusta a rota do PRV. (e) O robô Tarefa atinge o estado final representado pela posição estimada  $s''_2$  e comunica com o nó sensor  $s_2$ , recebendo um reforço positivo.

devido a sua simplicidade e para evitar problemas relacionados ao planejamento de trajetórias. A velocidade empregada nos robôs Tarefa e Explorador foi de  $1m/s$  e  $3m/s$ , respectivamente.

A simulações realizadas nesta tese seguem o esquema apresentado na Figura 5.2. Nessas simulações foram empregados um Robô Tarefa e  $n$  Robôs Exploradores. O Robô Tarefa executa um rota do PRV sobre as posições estimadas dos nós sensores, enquanto que os Robôs Exploradores coletam informações do meio e as transmitem para o Robô Tarefa. Na Figura 5.2(a), vê-se um Robô Tarefa  $R_1$  e um robô Explorador  $R_2$  com suas posições reais e estimadas. O Robô Tarefa navega seguindo a rota mostrada e a navegação é penalizada por meio de um reforço negativo (representado por meio de um marcador vermelho). Na Figura 5.2(b), quando o Robô Tarefa atinge a posição estimada  $s'_1$  do nó sensor  $s_1$ , recebe um reforço positivo (representado por meio de um marcador verde). Embora a comunicação não seja estabelecida, o reforço é positivo porque o robô conseguiu alcançar a posição estimada. Ao mesmo tempo, o Robô Explorador  $R_2$  comunica com o nó sensor  $s_2$  e estima novamente sua posição. Na Figura 5.2(c), os robôs se encontram e a informação obtida pelo Robô Explorador é transferida para o Robô Tarefa que pode então reajustar sua rota para compreender a nova posição estimada  $s''_2$  do nó sensor  $s_2$  (Figura 5.2(d)). Finalmente, na Figura 5.2(e) o Robô Tarefa atinge a posição do nó sensor  $s_2$  e consegue estabelecer comunicação, recebendo um reforço positivo por essa ação executada. Cada trecho da rota do PRV executado pelo Robô Tarefa é considerado com uma subtarefa, ou seja, é definido um estado objetivo que é representado pelo estado referente à posição estimada do nó sensor  $s_i$ . Dessa forma, o reforço do time fica representado pelo reforço do Robô Tarefa, que não é de todo individual: quanto mais informações receber dos Robôs Exploradores, maior será o seu valor de reforço.

## 5.1 Metodologia de Testes

Duas métricas foram utilizadas para avaliar o sistema proposto: i) Reforço Médio (RM) e ii) Erro Médio de Posicionamento dos Sensores (EMPS). O RM é a média dos reforços obtidos em cada simulação. Esse reforço, por sua vez é a soma dos reforços obtidos a cada trecho da rota (aresta que liga cada par de nós sensores) executado pelo Robô Tarefa. O reforço computado em cada trecho considera os custos de navegação e comunicação (penalização) e a recompensa por alcançar o estado objetivo (comunicar com o nó sensor ou então atingir o estado estimado

desse nó sensor). O Erro Médio de Posicionamento dos Sensores (EMPS) é computado como a soma da diferença entre a posição real do nó sensor e a posição estimada dividida pelo número total de nós sensores.

A escolha do RM como métrica principal de avaliação deve-se principalmente a dois fatores: i) é a métrica mais utilizada em avaliação de sistemas multi-agentes que consideram incertezas e ii) é genérica a ponto de se relacionar indiretamente com outras métricas relevantes. No caso desta tese, o RM considera principalmente custos de navegação e comunicação. Esses custos indiretamente relacionam-se com outras grandezas como energia consumida pelos robôs, perda de dados na rede, quantidade de dados coletados e tempo de execução. Cabe ressaltar, porém, que a relação indireta entre o RM e essas grandezas pode incluir deficiências na modelagem, não sendo a métrica ideal de análise dos dados, especialmente quando é necessário fazer comparação entre diversos métodos encontrados na literatura. Entretanto, para o caso desta tese, onde a análise visa principalmente analisar o desempenho diante da incerteza do ambiente, o RM é satisfatório.

Em outras palavras, quanto menor o custo de comunicação, mais rapidamente os robôs conseguirão executar suas tarefas e os custos relacionados a outras grandezas será menor.

Assim sendo, os resultados foram definidos de forma a avaliar:

- Reforço Médio (RM) em função da:
  - probabilidade de erro de observação;
  - probabilidade de erro de atuação;
- Erro Médio de Posicionamento dos Sensores (EMPS) em função da:
  - probabilidade de erro de observação;
  - probabilidade de erro de atuação;
- Reforço Médio (RM) em função da:
  - probabilidade de erro de observação  $\times$  custo de comunicação;
  - probabilidade de erro de atuação  $\times$  custo de comunicação;
  - parâmetro de comunicação  $c$ ;
- Reforço Médio em cenários diversos.

Os métodos comparados basearam-se na utilização de um Robô Tarefa, que executa um ciclo do VRP completo (conforme Figura 5.1 e um ou mais Robôs

Exploradores com a função de auxiliar os Robôs Tarefa, fornecendo informações de posicionamento dos nós sensores para esses. Esses métodos são sumarizados abaixo:

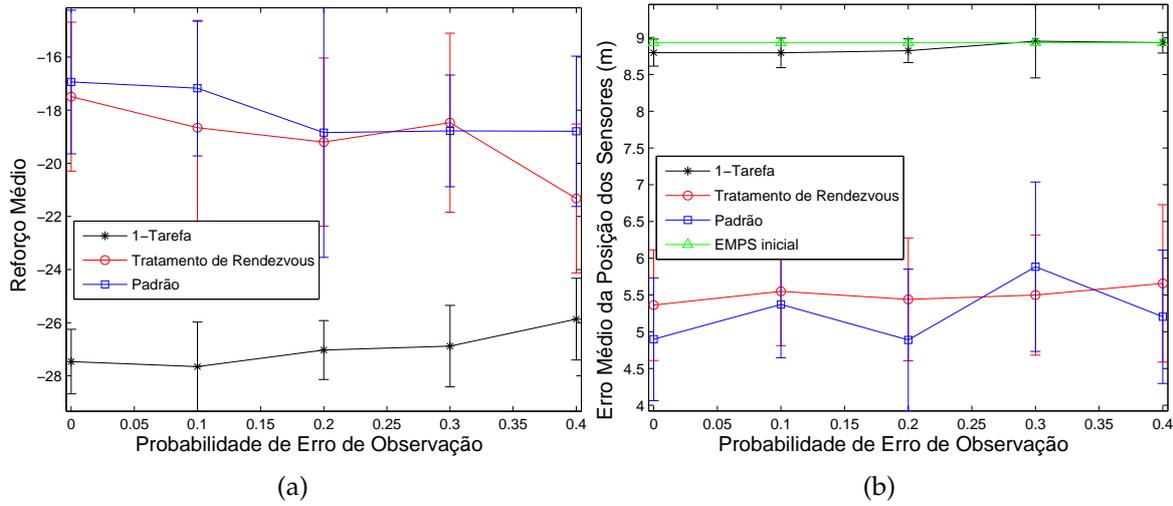
- 1-Tarefa: apenas um Robô Tarefa navega no ambiente seguindo a rota do PRV. Nenhum robô explorador é utilizado;
- Padrão: são empregados 1 robô tarefa e 1 explorador que navega pelo ambiente de maneira randômica;
- Tratamento de *Rendezvous*: são empregados 1 robô tarefa e 1 explorador que implementa um método para rastrear a posição do Robô Tarefa quando necessita transferir informações coletadas;

## 5.2 Comportamento do Sistema com Erros de Observação e Atuação

Os primeiros testes realizados visam demonstrar a viabilidade e utilidade do emprego de Robôs Exploradores em tarefas de coleta de dados em redes de nós sensores. Para tanto, o cenário consistiu na utilização de um Robô Tarefa realizando um ciclo do PRV gerado a partir das  $n$  posições estimadas para os nós sensores do ambiente. A posição real dos nós sensores foi configurada de forma a evitar o máximo possível a coleta eventual de dados pelos robôs, ou seja, sem o conhecimento dessas posições reais, o Robô Tarefa não consegue coletar os dados. Nestas simulações, os resultados representam a média de 20 amostras nas quais foi utilizado apenas 1 Robô Explorador, pois assim pode-se atingir um limite inferior para os resultados obtidos com a metodologia. Essa quantidade de amostras foi escolhida tendo-se em vista o grande número de variáveis envolvidas (5 variações de erros de observação e 5 variações de erros de atuação), do número de métodos testados (3) e do tempo necessário para os Robôs Tarefa realizarem uma rota do PRV completa (aproximadamente 10 minutos em tempo real de simulação).

A utilização de Robôs Exploradores foi comparada de duas formas: i) com o *Método de Tratamento Rendezvous* e ii) com o *Método Padrão*. Além disso, o sistema foi avaliado sem a presença de Robôs Exploradores, por meio do *Método 1-Tarefa*. Na Figura 5.3(a), é possível notar que o reforço médio, nas abordagens que utilizam robôs exploradores, não é significativamente distinto.

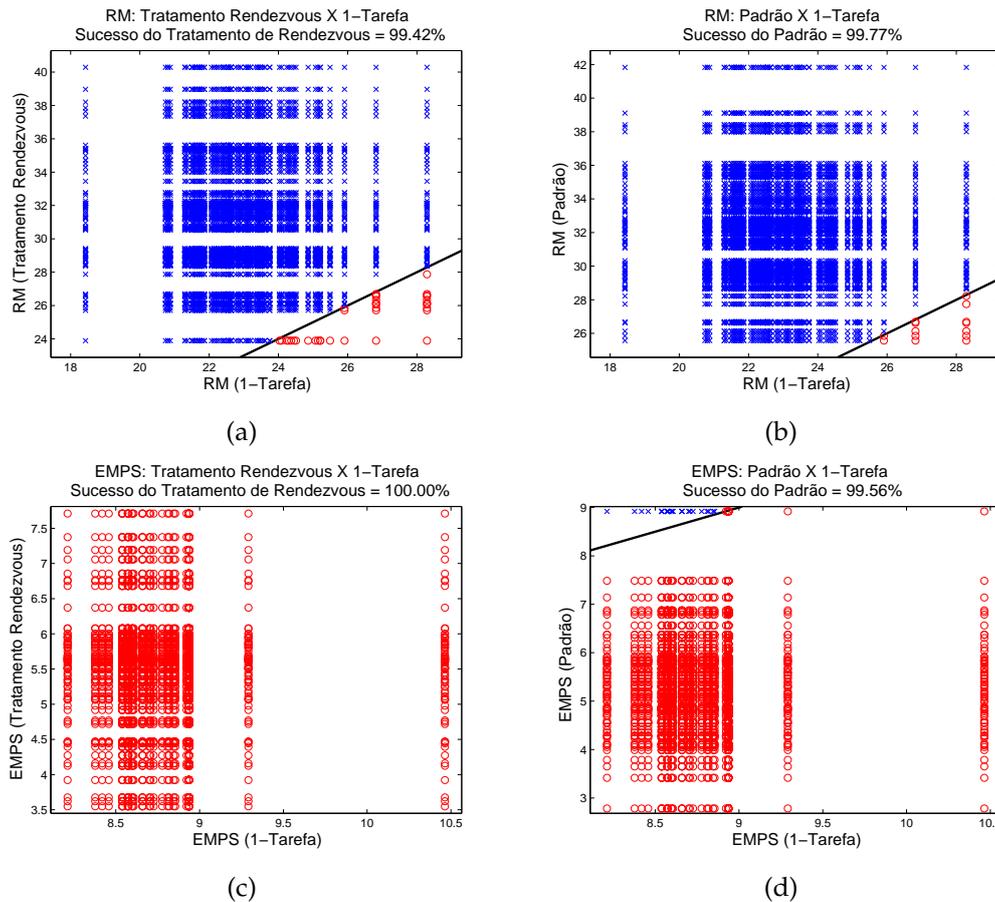
Perceba a melhora no RM ocasionada pelo emprego de Robôs Exploradores, comparado com o RM obtido empregando-se apenas um Robô Tarefa. Outro as-



**Figura 5.3.** (a) Reforço médio e (b) Erro médio de Posicionamento dos Sensores em função da probabilidade de erro de observação. Em (a), os resultados mostrados indicam, em média, uma melhora no sistema por meio da utilização de Robôs Exploradores. A utilização do *Método Rendezvous* nos Robôs Exploradores mostrou resultados similares a sua não utilização. Em (b) o erro médio da posição dos sensores é, em média, muito similar entre os métodos, com exceção do *Método 1-Tarefa* que é claramente inferior. Nesses gráficos, usou-se  $p(x|x, u) = 0.8$ , ou seja o robô tem 80% de chance de executar a ação com sucesso.

pecto que deve ser observado é o desvio padrão. Observe que para a curva obtida com um Robô Tarefa o desvio padrão não é alto, pois os Robôs Tarefa não possuem em sua trajetória nenhum ponto que intercepte o raio de comunicação de algum sensor. Porém, os outros métodos apresentam desvios significativamente maiores, uma vez que a troca de informação entre os robôs para obter conhecimento da rede ocorrerá quando os robôs exploradores possuírem informações para repassar, e isso, evidentemente, varia de execução para execução. O resultado apresentado em 5.3(b) mostra que o EMPS é maior empregando-se apenas 1 Robô Tarefa, sem a presença de Robôs Exploradores. Porém, para os outros dois métodos, o erro mostrou-se similar. O EMPS inicial do sistema é mostrado por meio da reta em verde no gráfico.

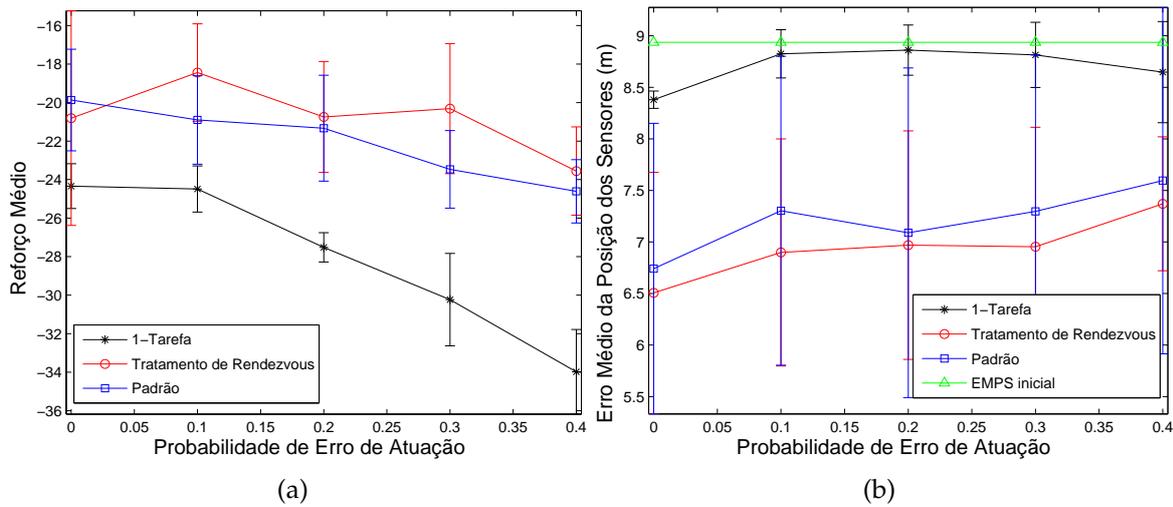
Visando investigar o comportamento de dispersão das curvas apresentadas nos gráficos anteriores, a Figura 5.4(a-b), mostra a comparação entre os valores de RM da seguinte maneira: a) *Método de Tratamento de Rendezvous*  $\times$  *Método 1-Tarefa* e ii) o *Método Padrão*  $\times$  *Método 1-Tarefa*. O *Método de Tratamento do Rendezvous* é representado por marcadores “x” em azul, ao passo que o *Método 1-Tarefa* é representado por marcadores “o” vermelhos. A reta mostrada no gráfico é formada por



**Figura 5.4.** Comparação entre o RM e o EMPS obtidos com a variação do erro de observação. (a-b) Comparação entre o Reforço Médio obtido com o Método 1-Tarefa e com os métodos (a) *Tratamento de Rendezvous* e (b) *Padrão*. (c-d) Comparação entre o Erro Médio da Posição dos Sensores obtido com o Método 1-Tarefa e com os métodos (c) *Tratamento de Rendezvous* e (d) *Padrão*. Comparativamente, o Método *Rendezvous* mostra-se superior ao Método 1-Tarefa (99,42% e 100%), assim como o Método *Padrão* mostra-se superior ao Método 1-Tarefa (99,77% e 99,56%).

pontos com valores iguais de RM para os dois métodos. Percebe-se que o *Método de Tratamento de Rendezvous* apresenta resultados melhores que o *Método 1-Tarefa* na maioria dos casos (99,42%). Similarmente, o *Método Padrão* apresenta resultados melhores que o *Método 1-Tarefa* na maioria dos casos (99,77%).

A Figura 5.4(c-d) mostra a comparação entre os valores de EMPS da seguinte maneira: a) *Método de Tratamento de Rendezvous* × *Método 1-Tarefa* e ii) *Método Padrão* × *Método 1-Tarefa*. Contrariamente ao caso dos gráficos de RM, o *Método 1-Tarefa* apresenta a maioria dos casos com valores de EMPS superiores aos valores dos outros métodos, indicando portanto, que o erro é maior. Dessa maneira, o



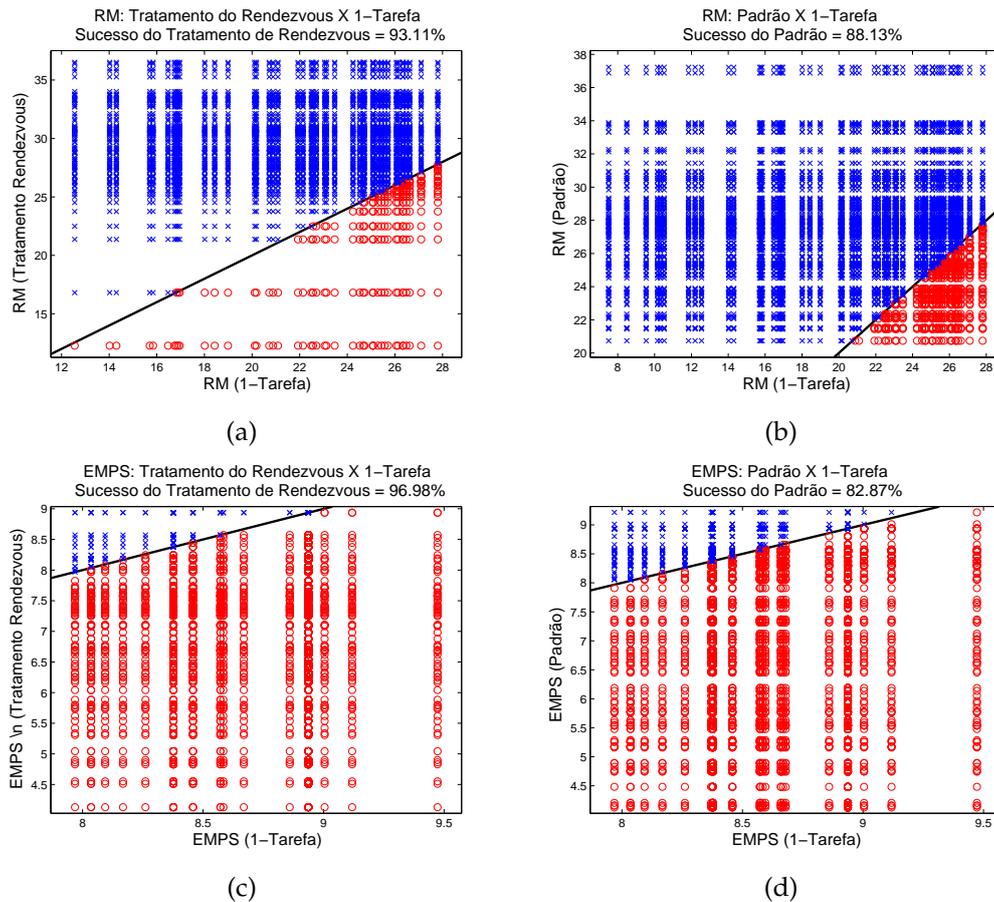
**Figura 5.5.** (a) Reforço médio e (b) Erro Médio de Posicionamento dos Sensores em função da probabilidade de erro de atuação. Os resultados obtidos com a variação da probabilidade do erro de atuação apresentam comportamento similar ao gráfico apresentado na Figura 5.3. Nesses gráficos, usou-se  $p(o|e) = 1$ , ou seja o robô não apresenta erros de observação.

*Método de Tratamento de Rendezvous* apresenta resultados melhores que o *Método 1-Tarefa*(100%). Similarmente, o *Método Padrão* apresenta resultados melhores que o *Método 1-Tarefa* na maioria dos casos (99,56%).

As Figuras 5.5 e 5.6, com a variação da probabilidade do erro de atuação, apresentam resultados similares aos das Figuras 5.3 e 5.4. Enquanto o gráfico de EMPS na Figura 5.5(b) é similar ao gráfico apresentado em 5.3(b), o gráfico referente ao RM apresenta curvas monotonicamente decrescentes, diferenciando-se ligeiramente em relação ao gráfico com variação da probabilidade do erro de observação. O comportamento do RM na Figura 5.3(a) é não monotônico porque o modelo de observação estabelece uma “linha de corte para convergência”, abaixo da qual o robô não consegue estimar sua posição e, portanto, suas trajetórias real e estimada divergem.

### 5.3 Comunicação para Cooperação

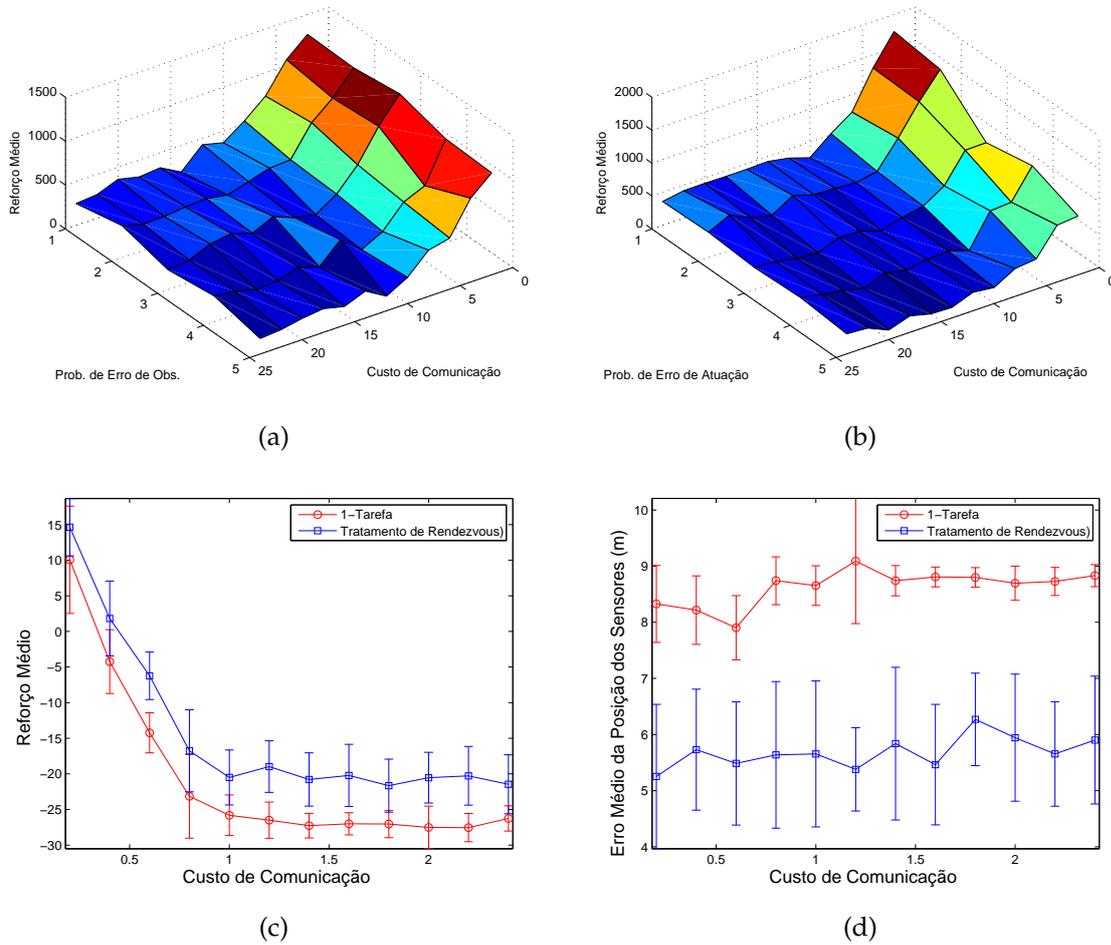
Os resultados dessa seção mostrarão o comportamento do sistema com o uso do Método de Comunicação. A Figura 5.7(a-b) mostra a variação do RM em função da variação do custo de comunicação e da probabilidades de erro de (a) observação (o erro de atuação foi de 20%, ou seja  $p(x|x, u) = 0.8$ ) e (b) atuação (sem erro de observação, ou seja  $p(o|e) = 1$ ). O valor do reforço médio na Figura 5.7 exhibe um



**Figura 5.6.** Comparação entre o RM e o EMPS obtidos com a variação do erro de atuação. (a-b) Comparação entre o Reforço Médio obtido com o *Método 1-Tarefa* e com os métodos (a) *Tratamento de Rendezvous* e (b) *Padrão*. (c-d) Comparação entre o Erro Médio da Posição dos Sensores obtido com o *Método 1-Tarefa* e com os métodos (c) *Tratamento de Rendezvous* e (d) *Padrão*. Comparativamente, o *Método de Tratamento de Rendezvous* mostra-se superior ao *Método 1-Tarefa* (93,11% e 96,98%), assim como o *Método Padrão* mostra-se superior ao *Método 1-Tarefa* (88,13% e 88,87%).

comportamento aparentemente semelhante para  $0 \leq \text{prob}(e|o) \leq 5$ , reforçando o que já ocorria no gráfico da Figura 5.3(a), onde os pontos da curva, a despeito do desvio apresentado, mantinham um valor aproximado para probabilidades de erro menores. Isso estabelece a robustez do sistema, já que variações de probabilidades de erro interferem na observação do robô, mas não diretamente na sua ação.

Diferentemente, na Figura 5.7(b), vê-se que, à medida que a probabilidade de erro de atuação aumenta, o RM diminui, gradativamente. Nesse caso, a ação já foi tomada e caso ela não leve a um estado final, uma penalização será aplicada. Como a aplicação dessas penalizações é proporcional à probabilidade do erro de



**Figura 5.7.** Variação do Reforço Médio em função do Custo de Comunicação em função de (a) probabilidade de erro de observação (o erro de atuação foi de 20%, ou seja  $p(x|x, u) = 0.8$ ) e (b) probabilidade de erro de atuação (sem erro de observação, ou seja  $p(o|e) = 1$ ). Ambos os resultados mostram um crescimento do Reforço Médio com o decréscimo da probabilidade dos erros de atuação e observação, bem como um decréscimo no custo de comunicação. Em (c), vê-se a variação do RM em função do custo de comunicação e em (d) vê-se a variação do EMPS para os métodos *Rendezvous* e *1-Tarefa*.

atuação, a curva de variação do RM tende a ser monotonicamente decrescente com o aumento do erro.

Na Figura 5.7(c), o RM decresce à medida que o custo de comunicação aumenta. Quando custo de comunicação se torna maior ou igual ao valor da penalização devido a uma penalização inadequada, ou seja  $\|c\| \geq \|b\|$ , as curvas se estabilizam, pois o robô passa a não mais utilizar a comunicação a longas distâncias. Na figura, vê-se que a utilização de um Robô Explorador aumentou o RM, comparado com o valor do reforço utilizando-se apenas um Robô Tarefa. Comple-

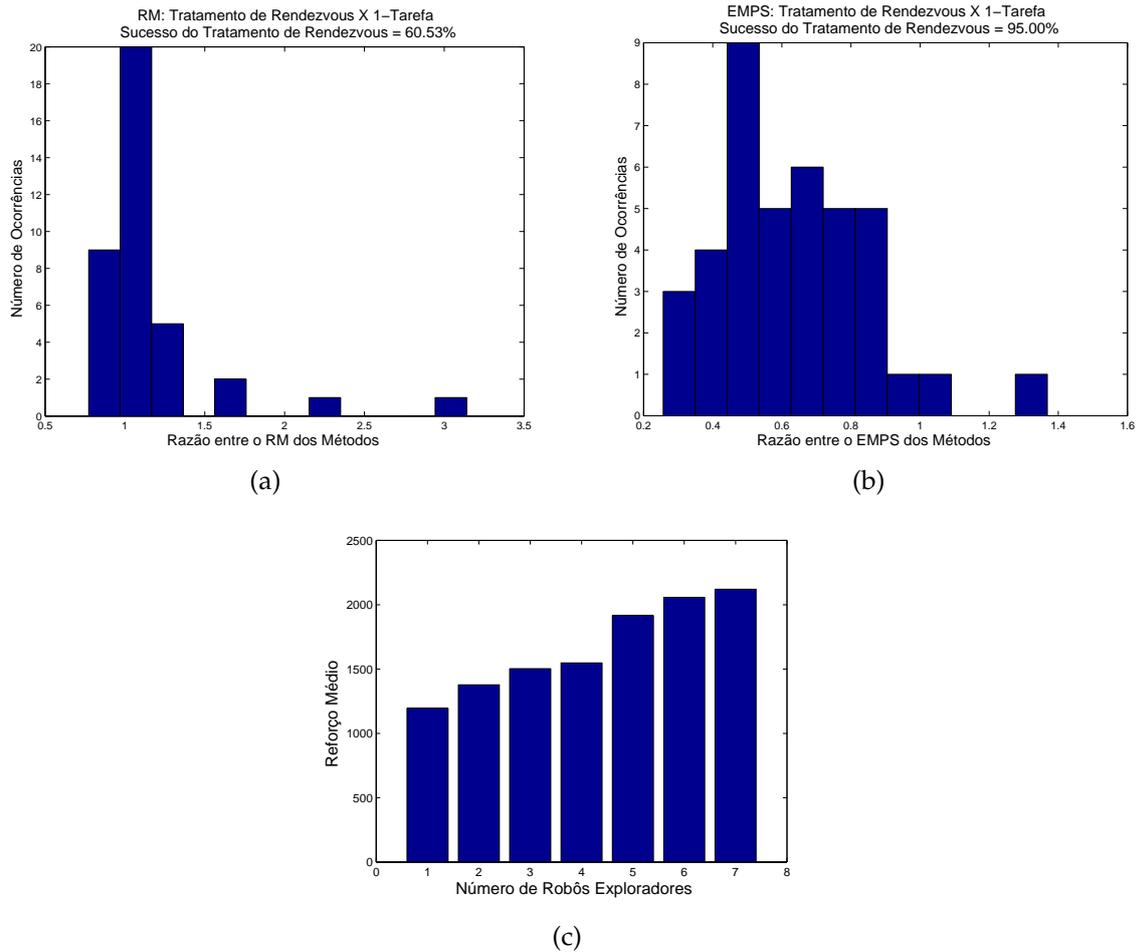
mentarmente, na Figura 5.7(d), nota-se que o EMPS obtido por meio do *1-Tarefa* é maior que o EMPS obtido por meio do *Método de Tratamento do Rendezvous*.

## 5.4 Avaliação em Cenários Diversos

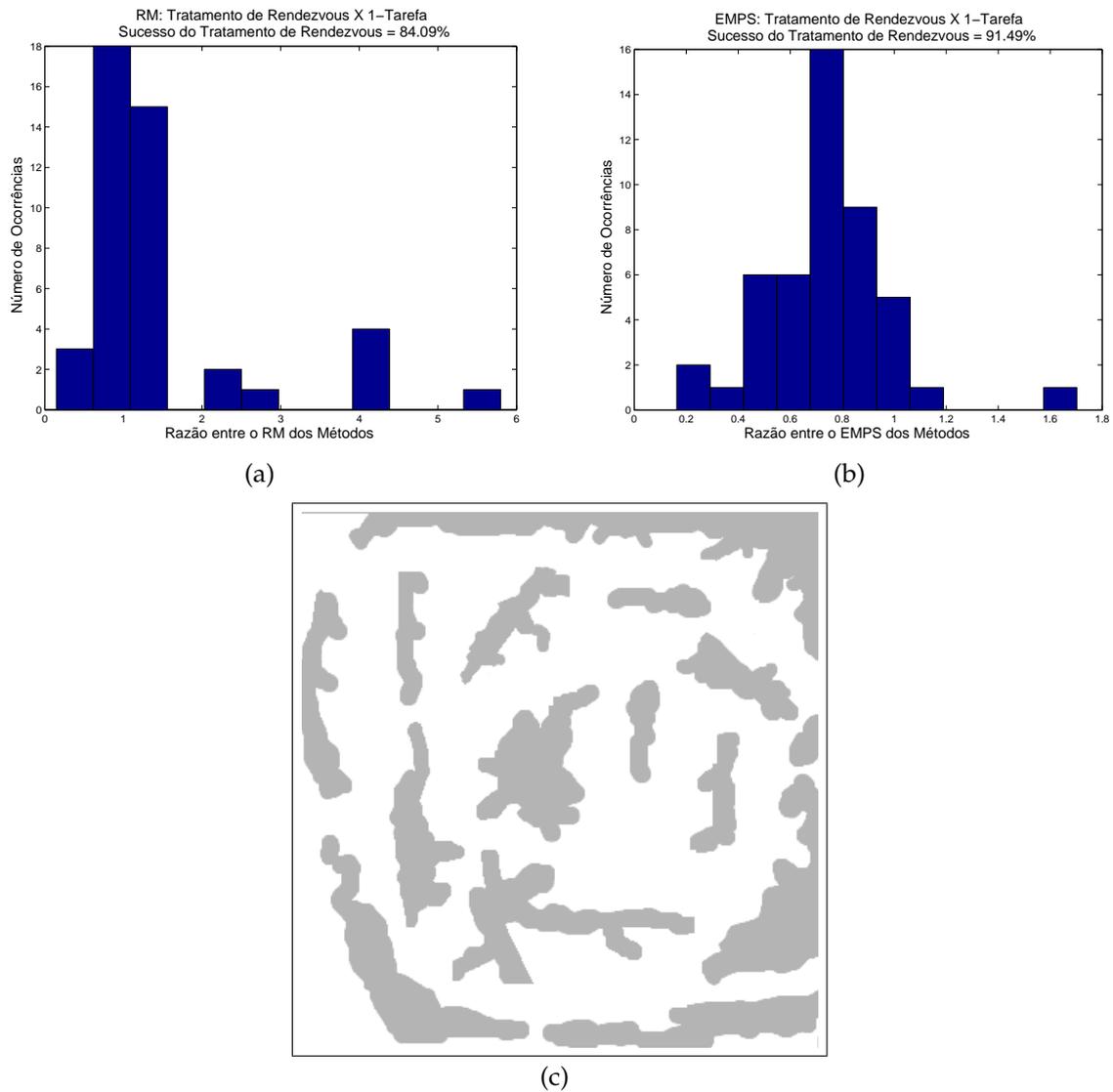
A primeira parte desta seção se dedica a avaliar os resultados da execução de tarefas em 38 cenários gerados randomicamente (mas com o mesmo mapa do ambiente). Posições reais e estimadas foram geradas para simular o conhecimento inicial dos Robôs Tarefa, quando é iniciada a visitação aos nós que pertencem a sua rota. Os Robôs Tarefa criam uma rota a partir de soluções para o PRV com base nas posições estimadas, conforme discutido nesta tese. O objetivo da análise é mostrar a validade da metodologia em ambientes diversos. Alguns desses cenários são mostrados na Figura 5.11.

A Figura 5.8 mostra a razão entre o RM e EMPS obtido com o *Método de Tratamento do Rendezvous* e *1-Tarefa*. A mesma rota foi computada em ambos os casos. O resultados na Figura 5.8(a) mostram que o *Método Rendezvous* possui desempenho superior ao *Método 1-Tarefa*, já que na maioria dos casos a razão entre o RM nos dois métodos é maior que 1. Complementarmente, para o EMPS, o *Método Rendezvous* possui desempenho superior ao método *1-Tarefa*, uma vez que existem poucos casos nos quais a razão entre os valores obtidos pelos dois métodos é maior que 1. Finalmente, a Figura 5.8(c) mostra o RM em função do número de Robôs Exploradores. Conforme esperado, a presença de mais Robôs Exploradores resulta em melhor desempenho.

Foram realizadas também, simulações em 44 cenários gerados randomicamente, porém com a utilização de outro mapa do ambiente. Os resultados, exibidos na Figura 5.9(a-b) confirmam a melhora obtida pelo *Método de Tratamento do Rendezvous* sobre o *Método 1-Tarefa*. Na Figura 5.9(c) é exibido o mapa do ambiente sobre o qual as simulações foram realizadas.



**Figura 5.8.** (a-b) Resultados mostrando o desempenho do *Método Rendezvous* comparado ao *Método 1-Tarefa* para Reforço Médio and Erro Médio da Posição dos Sensores. (c) Reforço Médio em função do Número de Robôs Exploradores.



**Figura 5.9.** (a-b) Resultados mostrando o desempenho do *Método Rendezvous* comparado ao *Método 1-Tarefa* para Reforço Médio and Erro Médio da Posição dos Sensores. (c) Mapa do ambiente onde as simulações foram realizadas.

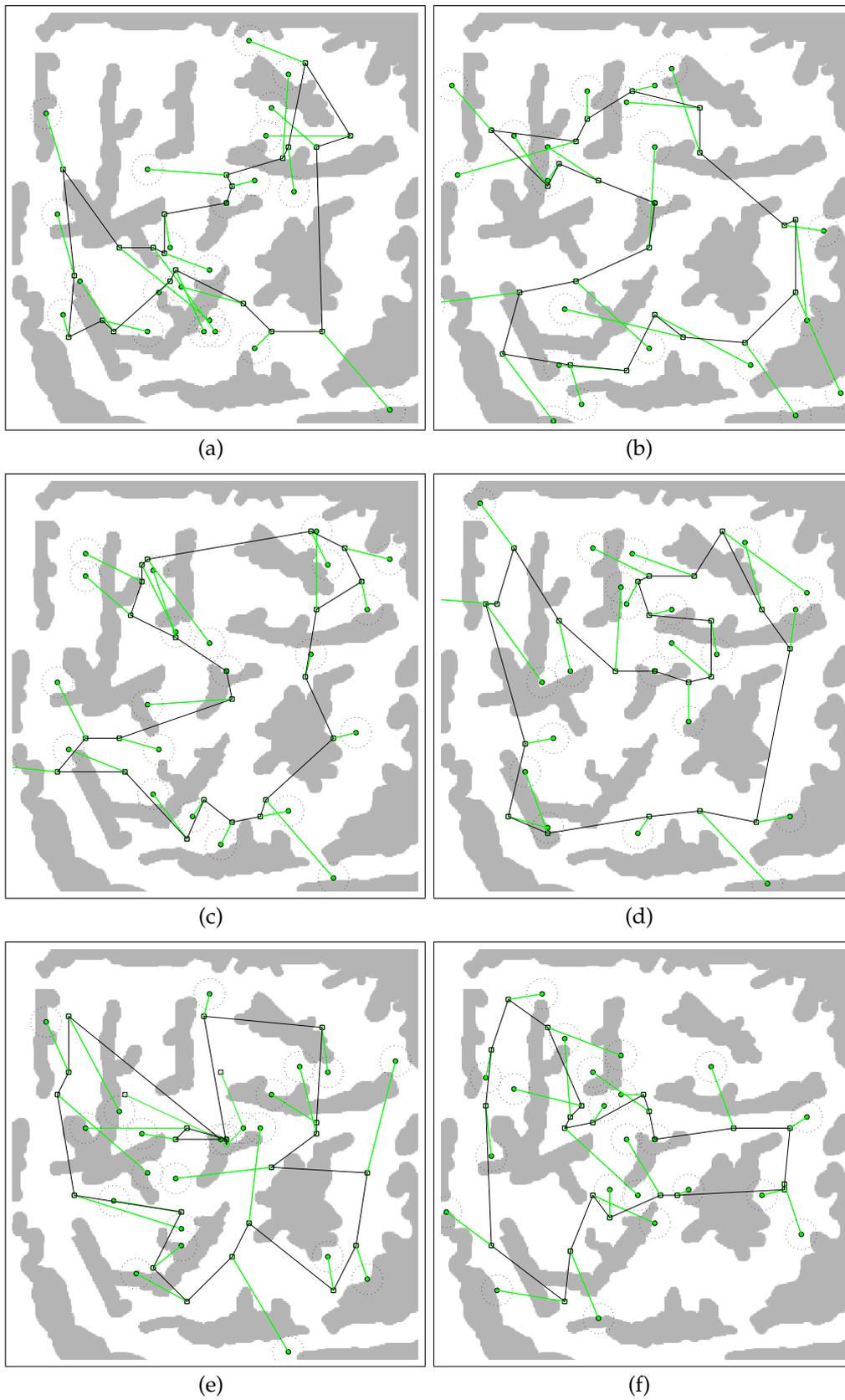


Figura 5.10. Cenários 1-6 utilizados na Seção 5.4.

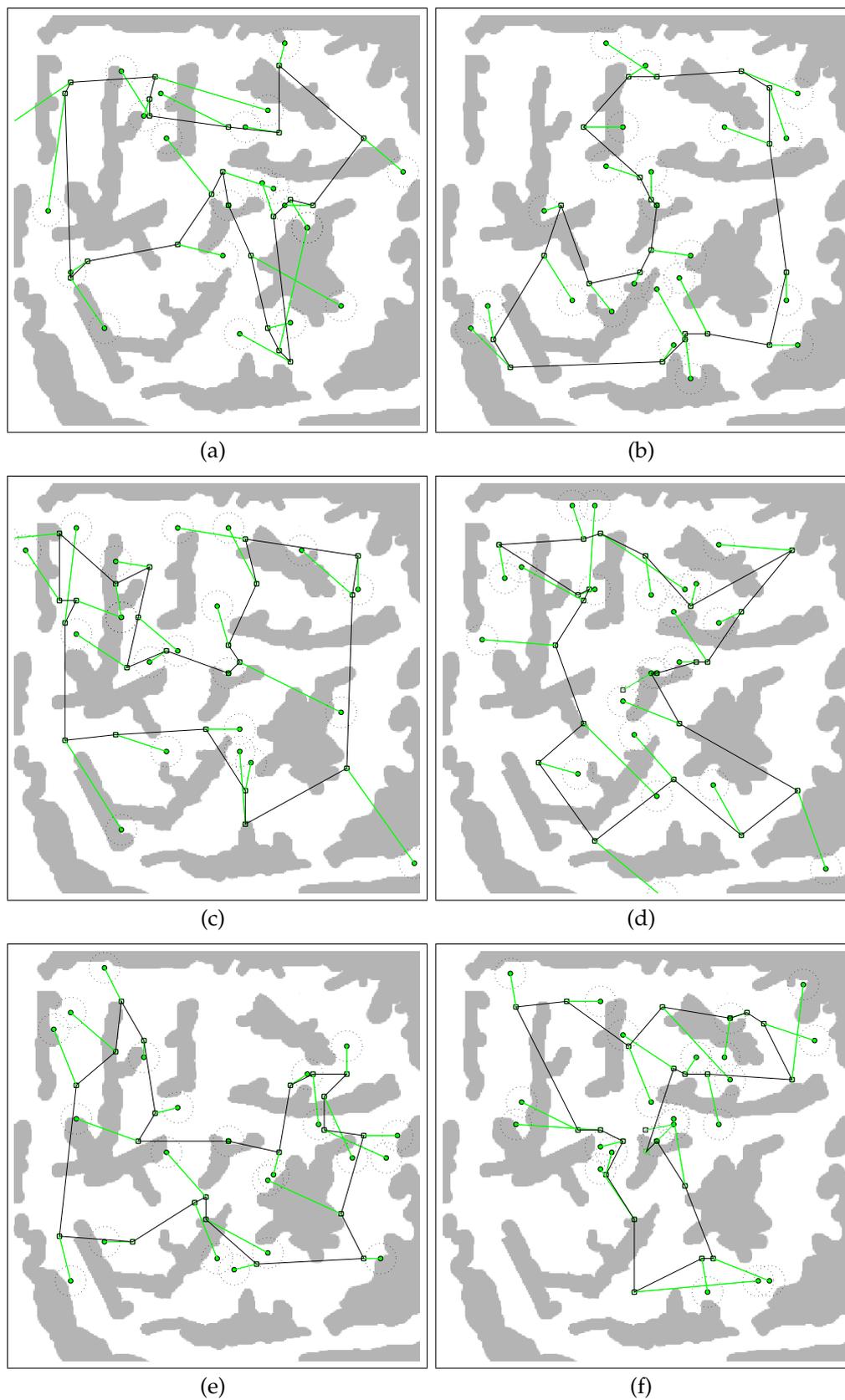


Figura 5.11. Cenários 7-12 utilizados na Seção 5.4.



# Capítulo 6

## Conclusões e Trabalhos Futuros

### 6.1 Conclusões

Esta tese apresentou uma abordagem para coleta de dados em redes de nós sensores sem fio para um time de robôs cooperativos, considerando incertezas de atuação e observação dos robôs e incerteza de posicionamento dos nós sensores. Para que a coleta de dados seja efetuada eficientemente, os robôs precisam comunicar constantemente, tanto com outros robôs como com os nós sensores da rede, que armazenam a informação. Embora a comunicação, nesta tese, seja perfeita, é também limitada e onerosa.

A abordagem utiliza dois tipos de robôs: i) Robôs Tarefa, responsáveis por visitar os nós sensores e coletar os dados e ii) Robôs Exploradores responsáveis por auxiliar os Robôs Tarefa à medida que exploram o ambiente e descobrem a posição “real” dos nós sensores. O ambiente considerado é composto de uma rede de nós sensores espalhados randomicamente pelo ambiente. Os Robôs Tarefa possuem uma estimativa da posição desses nós, o que lhes permite gerar uma rota com base nessas posições. Porém, a posição “real” do nó sensor é desconhecida (o nó sensor pode ter mudado de posição em função de algum evento natural, por exemplo).

Os Robôs Tarefa e os Robôs Exploradores são modelados, respectivamente, por meio de POMDP e DEC-POMDP (extensão para casos descentralizados). Robôs Tarefa realizam as suas tarefas visitando cada nó sensor presente em sua rota (computada a partir de soluções para o VRP). Para tanto, os robôs precisam tratar suas incertezas de atuação e observação. Robôs Exploradores precisam tratar, além disso, incertezas em relação ao time (posicionamento dos membros do time), porque quando novos dados referentes à posição dos nós sensores são descobertos, é necessário que o robô navegue até uma posição de encontro para transferir os

dados.

A utilização de Robôs Exploradores foi avaliada em função da variação de erros de atuação e observação. A comparação entre quatro métodos foi mostrada: i) *Método 1-Tarefa*, onde emprega-se apenas 1 Robô Tarefa, ii) *Método Padrão*, no qual os robôs exploradores sempre navegam aleatoriamente pelo ambiente, transferindo suas informações ao encontrar os outros robôs e iii) *Método de Tratamento de Rendezvous* no qual os Robôs Exploradores rastreiam a posição de outros robôs do time sempre que precisam transferir alguma informação sobre posição dos nós. Para variação do erro de observação, os resultados mostram que o *Método de Tratamento de Rendezvous* apresentou melhores resultados comparado ao *Método 1-Tarefa* (99,42% dos casos para RM e 100% dos casos para EMPS). O *Método Padrão* também mostrou-se superior ao *Método 1-Tarefa* (99,57% dos casos para RM e 99,56% dos casos para EMPS). Já com a variação do erro de atuação, os resultados também mostram que o *Método de Tratamento de Rendezvous* apresentou melhores resultados comparado ao *Método 1-Tarefa* (93,11% dos casos para RM e 96,98% dos casos para EMPS). O *Método Padrão* também mostrou-se superior ao *Método 1-Tarefa* (88,13% dos casos para RM e 82,87% dos casos para EMPS).

O impacto da comunicação foi também avaliado. Os resultados mostram que quanto menores forem os erros de observação e atuação e menor for o custo em comunicação, maior é o reforço médio obtido. Além disso, dependendo do custo de comunicação, os robôs podem decidir entre duas ações: i) navegar até o ponto para efetuar a comunicação ou ii) comunicar a longas distâncias. Esse método é interessante porque os robôs podem evitar a navegação para regiões inóspitas se for possível comunicar. Da mesma maneira, os robôs podem preferir navegar até o alvo, se a comunicação for de baixa qualidade.

Finalmente o sistema foi avaliado para um conjunto de cenários randomicamente gerados. Os resultados mostraram que a utilização de Robôs Exploradores com o *Método de Tratamento de Rendezvous* é vantajosa, comparando-se com a não utilização de Robôs Exploradores. Em relação ao reforço médio, a utilização de Robôs Exploradores pelo *Método de Tratamento de Rendezvous* é melhor do que a não utilização de Robôs Exploradores na maioria dos casos (60,53% e 84,03% para dois mapas distintos do ambiente). Da mesma forma, em relação ao EMPS, o uso de Robôs Exploradores diminui a incerteza do posicionamento dos nós sensores (95% e 94,49% para dois mapas distintos do ambiente).

## 6.2 Trabalhos Futuros

**Implementação em robôs reais.** O principal aspecto a ser considerado em trabalhos futuros é a implementação em robôs reais, em ambientes internos e externos. Em ambientes que possuem grande área, a discretização em células nas quais os robôs navegarão pode trazer uma série de problemas, dentre os quais:

- Conhecimento do mapa: ambientes externos podem ser mais complicados de modelar, uma vez que o conhecimento da topografia do ambiente pode não ser facilmente discretizável;
- Necessidade de resolver o MDP continuamente: se o ambiente possui grande área, esse pode ser um procedimento custoso. Uma solução seria resolver o MDP em uma região vasta o suficiente a ponto de incluir os pontos de navegação inicial e final.

**Modelagem dos processos comunicativos.** A comunicação é um aspecto que difere significativamente em cenários reais e a especificação do custo de comunicação requer um estudo menos empírico. A determinação de um modelo para consumo de energia devido à comunicação seria importante para definição do custo. Além disso, modelos como o COM-MTDP [Pynadath & Tambe, 2002] e DEC-POMDP-COM [Goldman & Zilberstein, 2003] servem de inspiração para que ações comunicativas sejam inseridas na modelagem do processo de decisão. Pelo fato de a comunicação ser um processo ruidoso, esses modelos permitem a análise probabilística dos efeitos dessas ações, ao mesmo tempo que regulam a transição entre estados. Nesse caso, os estados, bem como a transição entre eles, seriam dependentes tanto da localização do robô (conforme é modelado atualmente), como também do “estado comunicativo” no qual eles se encontram.

**Processo de atribuição dos nós sensores a um grupo de Robôs Tarefa.** Nesta tese, considerou-se que os nós sensores da RSSF são conhecidos, ainda que exista incerteza de posicionamento. O processo de descoberta dos nós sensores no ambiente e posterior atribuição para os Robôs Tarefa não foi estudado, mas levanta questões interessantes:

- Como os nós sensores serão atribuídos aos Robôs Tarefa? Essa atribuição, depende de vários fatores, a saber: número de sensores que podem ser atribuídos aos robôs, proximidade com os outros sensores previamente alocados, etc. Além disso, idealmente, o processo deveria ser mediado por algum Robô

Explorador, pois esse tipo de robô possui conhecimento sobre os outros Robôs Tarefa do ambiente.

- Como redistribuir os sensores em caso de falha nos robôs? quando um robô não consegue cumprir suas tarefas, devido a algum defeito, os nós que lhe foram atribuídos precisam ser distribuídos entre outros robôs, de acordo com alguma métrica. Por exemplo, os nós poderiam ser atribuídos aos Robôs Tarefa de acordo com a proximidade do nó sensor ao centro geométrico da rota gerada para esse robô. Porém, como o sistema é descentralizado, essa atribuição precisaria ser mediada por um Robô Explorador, que seria responsável por sincronizar essa informação entre os outros robôs da rede.

**Estratégias de navegação do Robô Explorador.** Atualmente, a navegação do Robô Explorador segue duas estratégias: i) navegação aleatória e ii) navegação em direção ao ponto de encontro com outro robô. A investigação de outras alternativas que permitam varrer a área da WSN podem ser interessantes, uma vez que a definição de número de robôs pode ser calculada em função da área a ser coberta, do tempo mínimo para executar essa tarefa de cobertura, etc.

**Combinação dos modelos para resolução do VRP e do MDP.** Os robôs Tarefa solucionam o VRP considerando um grafo no qual os nós sensores se ligam por meio de uma linha reta. Claramente, essa não é a situação real, pois o ambiente possui obstáculos, e o robôs erram ao executar as ações e possuem observação parcial. A inclusão do modelo MDP (e suas extensões) no modelo de resolução do VRP, poderia sugerir interessantes alternativas para resolver ambos os problemas de maneira unificada.

# Referências Bibliográficas

- Ahn, S.; Kim, S.; Yoon, J.; Na, K. & An, S. (2010). Hybrid position-detection algorithms for sensor nodes using received signal strength indication and control transmission. In *Proceedings of the 2010 Fourth International Conference on Sensor Technologies and Applications, SENSORCOMM '10*, pp. 127--132, Washington, DC, USA. IEEE Computer Society.
- Aioffi, W. M. (2007). Métodos integrados para organização de redes de sensores sem fio com sorvedouro móvel e controle de densidade. Master's thesis, Programa de Pós-graduação em Ciência da Computação (PPGCC). Universidade Federal de Minas Gerais (UFMG).
- Alvin, J. A. (2001). Recursive position estimation in sensor networks.
- Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, 19(6):621–636.
- Barbosa, M.; Bernardino, A.; Figueira, D.; Gaspar, J.; Gonçalves, N.; Lima, P. U.; Moreno, P.; Pahliani, A.; Santos-Victor, J.; Spaan, M. T. J. & Sequeira, J. (2009). ISRobotNet: A testbed for sensor and robot network systems. In *Proc. of International Conference on Intelligent Robots and Systems*, pp. 2827--2833.
- Batalin, M. & Sukhatme, G. S. (2004). Coverage, exploration and deployment by a mobile robot and communication network. *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, 26(2):181–196.
- Batalin, M. A.; Sukhatme, G. & Hattig, M. (2004). Mobile robot navigation using a sensor network. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'04)*, volume 1, pp. 636– 641, Barcelona, Spain.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

- Bernstein, D. S.; Givan, R.; Immerman, N. & Zilberstein, S. (2002). The complexity of decentralized control of Markov Decision Process. *Mathematics of Operations Research*, 27(4):819–840.
- Bhadauria, D. & Isler, V. (2009). Data gathering tours for mobile robots. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'09)*, pp. 3868–3873, St. Louis, MO, USA.
- Bölöni, L. & Turgut, D. (2008). Should i send now or send later? a decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks. *Wireless Communications and Mobile Computing*, 8(3):385–403.
- Botelho, S. (2000). *Une Architecture Décisionnelle Pour La Coopération Multi-Robots*. PhD thesis, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS.
- Botelho, S. & Alami, R. (2000). A multi-robot cooperative task achievement system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2000)*, pp. 2716–2721.
- Butler, Z. & Rus, D. (2003). Event-based motion control for mobile sensor networks. *IEEE Pervasive Computing*, 2(4):34–42.
- Chaimowicz, L. (2002). *Coordenação Dinâmica de Robôs Cooperativos: Uma abordagem Utilizando Sistemas Híbridos*. PhD thesis, Programa de Pós-graduação em Ciência da Computação (PPGCC). Universidade Federal de Minas Gerais (UFMG).
- Chakrabarti, A.; Sabharwal, A. & Aazhang, B. (2003). Using predictable observer mobility for power efficient design of sensor networks. In *Proceedings of Second International Workshop Information Processing in Sensor Networks, (IPSN)*, pp. 129–145, Palo Alto, CA, USA.
- Cordeau, J.-F.; Desaulniers, G.; Desrosiers, J.; Solomon, M. M. & Soumis, F. (2001). *VRP with Time Windows*, pp. 157–193. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Corke, P.; Hrabar, S.; Peterson, R.; Rus, D.; Saripalli, S. & Sukhatme, G. (2004). Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'04)*, volume 4, pp. 3602–3608, New Orleans, USA.

- Corke, P.; Peterson, R. & Rus, D. (2003). Networked Robots: Flying Robot Navigation Using a Sensor Net. In *Proceedings of International Symposium of Robotics Research (ISRR'03)*, pp. 234–243, Siena, Italy.
- Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Mgmt Sci*, 6(1):80–91.
- de Luca, A. (1994). Local incremental planning for nonholonomic mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation, (ICRA'1994)*, pp. 104–110, San Diego, CA, USA.
- de Oliveira, H. A. B. (2008). *Localization for Time and Space for Wireless Sensor Networks*. PhD thesis, Federal Universidade of Minas Gerais.
- Elfes, A. (1987). Sonar-based real world mapping and navigation. *IEEE Transactions on Robotics and Automation*, 3(3):249–265.
- Gerkey, B. P. & Mataric, M. J. (2003). Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In *Proceedings of IEEE International Conference on Robotics and Automation, (ICRA'03)*, pp. 3862–3868, Taipei, Taiwan.
- Gerkey, B. P.; Vaughan, R. T.; Stoy, K.; Howard, A.; Sukhatme, G. S. & Mataric, M. J. (2001). Most valuable player: A robot device server for distributed control. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'01)*, pp. 1226–1231, Wailea, Hawaii.
- Goldman, C. V. & Zilberstein, S. (2003). Optimizing information exchange in cooperative multi-agent systems. In *Proceeding of International Conference on Autonomous Agents and Multiagent Systems*, pp. 137–144, Melbourne, Australia.
- Gu, Y.; Bozdog, D.; Ekici, E.; Özgüner, F. & Lee, C.-G. (2005). Partitioning based mobile element scheduling in wireless sensor networks. In *Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Network*, pp. 386–395, Santa Clara, CA, USA.
- Higuchi, T.; Fujii, S.; Yamaguchi, H. & Higashino, T. (2011). An efficient localization algorithm focusing on stop-and-go behavior of mobile nodes. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pp. 205–212.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial System*. The University of Michigan Press.

- Howard, A.; Matarić, M. J. & Sukhatme, G. (2003). Mobile sensor networks deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of 6th International Symposium on Distributed Autonomous Systems*, pp. 299–308, Fukuoka, Japan.
- Huang, P.; Oki, H.; Wang, Y.; Martonosi, M.; Peh, L. & Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences on zebranet. In *Proceedings of Architectural Support for Programming Languages and Operating System, (ASPLOS)*, pp. 96–107, San Jose, CA, USA.
- Huang, X. & Ozdalar, A. (2006). Power control and network design in mobile sensor networks. In *Proceedings of 4th International Symposium on Modeling and Optimization in Mobile Ad-hoc and Wireless Sensor Networks*, pp. 197–206, Boston, Massachusetts, USA.
- Jea, D.; Somasundara, A. A. & Shrivastava, M. B. (2005). Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems*, pp. 244–257, Marina del Rey, CA, USA.
- Kadkhoda, M.; Totouchi, M.-R.; Yaghmaee, M. & Davarzani, Z. (2010). A probabilistic fuzzy approach for sensor location estimation in wireless sensor networks. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pp. 1–5.
- Kaelbling, L. P.; Littman, M. L. & Cassandra, A. R. (1998). Planning and action in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.
- Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–46.
- Kansal, A.; Somasundara, A.; Jea, D.; Srivastava, M. & Estrin, D. (2004). Intelligent fluid infrastructure for embedded networks. In *Proceedings of Second International Conf. Mobile System, Applications and Services, (MobiSys)*, pp. 111–124, Boston, MA, USA.
- Kantor, G.; Singh, S.; Peterson, R.; Rus, D.; Das, A.; Kumar, V.; Pereira, G. & Spletzer, J. (2003). Distributed Search and Rescue with Robot and Sensor Team. In *Proceedings of 4th International Conference on Field and Service Robotics*, pp. 327–332.

- Kinalis, A. & Nicoletseas, S. (2007). Scalable data collection protocols for wireless sensor networks with multiple mobile sinks. In *Proceedings of 40th International Simulation Symposium*, pp. 60–72, Norfolk, Virginia, USA.
- Leng, M. & Wu, Y.-C. (2010). Localization of wireless sensor nodes with erroneous anchors via em algorithm. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pp. 1–5.
- Li, Q.; de Rosa, M. & Rus, D. (2003). Distributed Algorithms for Guiding Navigation across a Sensor Net. In *Proceedings of 11th International Symposium of Robotics Research*, pp. 313–325, Siena, Italy.
- Littman, M.; Dean, T. & Kaelbling, L. (1995a). On the complexity of solving markov decision problems. In *Proceedings of the Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pp. 394–40, San Francisco, CA. Morgan Kaufmann.
- Littman, M. L.; Cassandra, A. R. & Kaelbling, L. P. (1995b). Learning policies for partially observable environments: Scaling up. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 362–370. Morgan Kaufmann.
- Lui, K.; Ma, W.-K.; So, H. & Chan, F. (2009). Semi-definite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed. *Signal Processing, IEEE Transactions on*, 57(2):752–763.
- Madani, O.; Hanks, S. & Condon, A. (1999). On the undecidability of probabilistic planning and infinity-horizon partially markov decision problems. *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence*, 147(1-2):5–34.
- Messias, J. V.; Spaan, M. T. J. & Lima, P. U. (2010). Multi-robot planning under uncertainty with communication: a case study. In *Multi-agent Sequential Decision Making in Uncertain Domains*. Workshop at AAMAS10.
- Moravec, H. P. (1988). Sensor fusion in certainty grids fo mobile robots. *AI magazine*, 9(2):61–74.
- NASA (2011). Mars Exploration Rover Mission Overview, <http://marsrovers.jpl.nasa.gov/overview/>. Acessado em 2011.

- Nguyen, H. A.; Guo, H. & Low, K.-S. (2011). Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint. *Instrumentation and Measurement, IEEE Transactions on*, PP(99):1–10.
- Niculescu, D. & Nath, B. (2001). Ad hoc positioning system (aps). In *IN GLOBE-COM*, pp. 2926–2931.
- Papadimitriou, C. H. & Tsitsiklis, J. N. (1987). The complexity of Markov Decision Process. *Mathematics of Operations Research*, 12(3):441–450.
- Parker, L. (1998). Alliance: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14:504–516.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Pereira, G. A. S.; Soares, M. B. & Campos, M. F. M. (2004). A potential field approach to collect data from sensor networks using mobile robots. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'04)*, volume 4, pp. 3469–3474, Sendai, Japan.
- Peterson, R. & Rus, D. (2004). Interacting with sensor networks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'04)*, volume 1, pp. 180–186, Barcelona, Spain.
- Pynadath, D. V. & Tambe, M. (2002). The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423.
- Rosemary, C. P.; Emery-montemerlo, R.; Gordon, G. & Schneider, J. (2004). Approximate solutions for partially observable stochastic games with. In *In Proc. of Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, pp. 136–143.
- Russel, S. J. & Norvig, P. (2003). *Artificial Intelligence, A Modern Approach*. Prentice Hall.
- Savvides, A.; Han, C.-C. & Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01*, pp. 166–179, New York, NY, USA. ACM.

- Seuken, S. & Zilberstein, S. (2008). Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 17(2):190--250.
- Shah, R. C.; Roy, S.; Jain, S. & Brunette, W. (2003). Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Proceedings of IEEE Workshop Sensor Network Protocols and Applications (SNPA'03)*, pp. 30–41, Anchorage, AK, USA.
- Shi, Q.; He, C.; Chen, H. & Jiang, L. (2010). Distributed wireless sensor network localization via sequential greedy optimization algorithm. *Signal Processing, IEEE Transactions on*, 58(6):3328 –3340.
- Shigeng, Z.; Jiannong, C.; Lijun, C. & Daoxu, C. (2008). Locating nodes in mobile sensor networks more accurately and faster. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pp. 37 –45.
- Sichitiu, M. L. & Ramadurai, V. (2003). Localization of wireless sensor networks with a mobile beacon.
- Simmons, R. & Koenig, S. (1995). Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '95)*, pp. 1080 – 1087.
- Small, T. & Haas, Z. (2003). The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale there is a way). In *Proceedings of ACM MobiHoc*, pp. 233–244, Annapolis, Maryland, USA.
- Soares, M. B.; Campos, M. F. M.; Dutra, D. A.; da S. Campos, V. C. & Pereira, G. A. S. (2007a). Hybrid mobile robot navigational strategy for efficient data collection in sparsely deployed sensor networks. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'07)*, pp. 2833–2838, San Diego, CA, USA.
- Soares, M. B.; Neto, V. F. C.; Dutra, D. A.; da S. Campos, V. C. & Campos, M. F. M. (2007b). Um arcabouço para utilização de robôs móveis na coleta eficiente de dados em redes de nos sensores sem fio. In *Anais do Simpósio Brasileiro de Automação Inteligente, SBAI*, p. 6 páginas, Florianópolis, SC, Brazil.
- Somasundara, A.; Kansal, A.; Jea, D. & Srivastava, D. E. M. (2006). Controllably mobile infrastructure for low embedded sensor networks. *IEEE Transactions on Mobile Computing*, 5(8):958–973.

- Somasundara, A.; Ramammorthy, A. & Srivastava, M. B. (2007). Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410.
- Somasundara, A. A.; Ramamoothy, A. & Shrivastava, M. B. (2004). Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Proceedings of IEEE International Real-Time Systems Symposium*, pp. 296– 305, Lisboa, Portugal.
- Spaan, M. T. J. (2006a). *Approximate planning under uncertainty in partially observable environments*. PhD thesis, Universiteit van Amsterdam.
- Spaan, M. T. J. (2006b). Decentralized planning under uncertainty for teams of communicating agents. In *Proceedings Autonomous Agents and Multi-agents Systems (AAMAS'06)*, pp. 249--256.
- Spaan, M. T. J.; Gonçalves, N. & Sequeira, J. (2010). Multirobot coordination by auctioning POMDPs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1446--1451.
- Tang, Y.; Birch, B. & Parker, L. E. (2004). Planning mobile sensor net deployment for navigationally-challenged sensor nodes. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'04)*, volume 1, pp. 172–179, New Orleans, LA, USA.
- Tavares, A. I. (2004). *Tomada de Decisão Descentralizada com Observação Parcial e Comunicação Limitada em Processos Estocásticos Cooperativos*. PhD thesis, Programa de Pós-graduação em Ciência da Computação (PPGCC). Universidade Federal de Minas Gerais (UFMG).
- Tekdas, O.; Lim, J. H.; Terzis, A. & Isler, V. (2009). Using mobile robots to harvest data from sensor fields. *IEEE Wireless Communications*, 16(1):22--28.
- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
- Valle, C. A.; da Cunha, A. S.; Aioffi, W. M. & Mateus, G. R. (2008). Algorithms for improving the quality of service in wireless sensor networks with multiple mobile sinks. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 239–243, Vancouver, British Columbia, Canada.

- Wang, G.; Cao, G. & Porta, T. F. L. (2003). A bidding protocol for deploying mobile sensors. In *Proceedings of 11th International Conference on Network Protocols*, pp. 315–324, Atlanta, GA, USA.
- Wang, G.; Cao, G. & Porta, T. F. L. (2006). Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652.
- Wang, W.; Srinivasan, V. & Chua, K.-C. (2005a). Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proceedings of International Conference on Mobile Computing and Networking*, pp. 270–283, Cologne, Germany.
- Wang, Z. M.; Basagni, S.; Melachrinoudis, E. & Petrioli, C. (2005b). Exploiting sink mobility for maximizing sensor networks lifetime. In *Proceedings of 38th Hawaii International Conference on System Sciences*, p. 287a, Hawaii, USA.
- Whitehouse, K. & Culler, D. (2002). Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02*, pp. 59–67, New York, NY, USA. ACM.
- Wu, F.-J.; Huang, C.-F. & Tseng, Y.-C. (2009). Data gathering by mobile mules in a spatially separated wireless sensor network. In *Proceedings Tenth International Conference on Mobile Data Management: Systems, Services and Middleware (MDM '09)*, pp. 293–298, Washington, DC, USA. IEEE Computer Society.
- Wu, Q.; Rao, N. S. V.; Barhen, J.; Sitharama, S.; Vaishnavi, V.; Qi, H. & Chakrabarty, K. (2004). On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):740–753.

