

**UMA ABORDAGEM PARA MINIMIZAÇÃO  
DE CONSUMO DE ENERGIA EM  
REDES DE SENSORES SEM FIO COM  
SORVEDOUROS MÓVEIS**



CARLA OLIVEIRA BECHELANE

UMA ABORDAGEM PARA MINIMIZAÇÃO  
DE CONSUMO DE ENERGIA EM  
REDES DE SENSORES SEM FIO COM  
SORVEDOUROS MÓVEIS

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

ORIENTADOR: ALEXANDRE SALLES DA CUNHA

Belo Horizonte, MG

Maio de 2009

© 2009, Carla Oliveira Bechelane.  
Todos os direitos reservados.

Oliveira Bechelane, Carla  
D1234p Uma Abordagem Para Minimização De Consumo  
De Energia Em Redes De Sensores Sem Fio Com  
Sorvedouros Móveis / Carla Oliveira Bechelane. —  
Belo Horizonte, MG, 2009  
xx, 94 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais

Orientador: Alexandre Salles da Cunha

1. Redes de Sensores Sem Fio. 2. Programação  
Inteira. 3. Heurísticas. I. Título.

CDU 519.6\*82.10



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

## FOLHA DE APROVAÇÃO

Uma abordagem para minimização de consumo de energia em redes de sensores  
sem fio com sorvedouros móveis

**CARLA OLIVEIRA BECHELANE**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ALEXANDRE SALLES DA CUNHA - Orientador  
Departamento de Ciência da Computação - UFMG

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

PROF. JOSÉ MARCOS SILVA NOGUEIRA  
Departamento de Ciência da Computação - UFMG

PROF. MAURÍCIO CARDOSO DE SOUZA  
Departamento de Engenharia de Produção - EE- UFMG

Belo Horizonte, 16 de outubro de 2009.



*Dedico este trabalho aos meus queridos pais, grandes responsáveis por esta conquista.*





# Agradecimentos

Agradeço primeiramente aos meus pais, Efigênia e Ricardo, por serem um verdadeiro exemplo de dedicação e amor incondicional. Durante toda a minha vida, me incentivaram, apoiaram e fizeram de tudo para me proporcionar uma educação de qualidade. Sou muito grata e me sinto privilegiada por ter pais como eles, que não mediram esforços para que eu chegasse até aqui.

Agradeço a todos aqueles que torceram pelo meu sucesso, especialmente à minha irmã Marcela. Obrigada pelo seu companheirismo e amizade, pelo seu apoio sincero, por querer ajudar. Agradeço também ao Paul pelo apoio e compreensão ao longo destes últimos meses, e principalmente por torná-los mais prazerosos.

Aos meus amigos, agradeço por todos os momentos de alegria, descontração, e principalmente, por estarem lá quando eu precisei. Aos colegas e amigos do LaPO e do Synergia, por tornarem o ambiente de trabalho incrivelmente agradável e descontraído, e por contribuírem com o meu crescimento profissional e acadêmico.

Ao Prof. Geraldo Robson Mateus, meu co-orientador neste trabalho e meu orientador durante a graduação, agradeço por toda o acompanhamento e orientação, pela disposição para ajudar, ensinar e indicar o caminho, e, principalmente, por contribuir enormemente para minha formação acadêmica e profissional.

Um agradecimento especial ao Prof. Alexandre Salles da Cunha, meu orientador. Sua dedicação e profissionalismo como orientador e professor são exemplos a serem seguidos por todos. Com ele pude aprender lições valiosíssimas que levarei para sempre comigo (isso inclui a escrita de um bom texto!). Ele é diretamente responsável pela qualidade deste trabalho. Além disso, foi um grande companheiro nos bons momentos e nas horas difíceis.

Finalmente, sou eternamente grata a Deus, por todas as bênçãos, oportunidades e pessoas fantásticas presentes ao longo de minha vida. Sem essas pessoas, eu não teria alcançado tantas conquistas.



# Resumo

Nessa dissertação, introduzimos um novo Problema de Otimização Combinatória aplicado ao contexto de Redes de Sensores Sem Fio (RSSFs) com sorvedouros móveis. Uma formulação de Programação Inteira Mista é apresentada, cujo objetivo é minimizar o custo das árvores de comunicação, as quais definem a organização hierárquica dos nós sensores em agrupamentos. Como a energia gasta com comunicação é a principal responsável pelo consumo de energia em RSSFs, o custo das árvores representa a quantidade de energia necessária para transmitir dados em cada enlace de comunicação. Os problemas de agrupamento e roteamento são tratados de forma integrada, respeitando as restrições de número máximo de saltos e de comprimento máximo da rota do nó sorvedouro. A resolução exata do modelo através de um algoritmo *Branch-and-bound* (*BB*) é computacionalmente custosa, apresentando-se impraticável para aplicações em RSSFs, uma vez que a organização da rede tem que ser feita de forma rápida para não comprometer seu funcionamento. Dessa forma, apresentamos uma heurística que encontra, em tempo razoável para aplicações em RSSFs, soluções viáveis para o problema proposto. Testes computacionais extensivos foram realizados para avaliar o impacto da abordagem proposta, através da simulação de RSSFs. Os resultados de simulação indicam que nossa abordagem leva a melhoras significativas no consumo de energia, no tempo de vida operacional e em alguns parâmetros de Qualidade de Serviço (QoS) para RSSFs.

**Palavras-chave:** Redes de Sensores Sem Fio, Otimização Combinatória, Heurísticas, Sorvedouros móveis.



# Abstract

In this work, we introduce a new network topology model and algorithms to minimize the energy consumption in Wireless Sensor Networks (WSNs) with mobile sinks. On the one hand, the proposed optimization problem explicitly minimizes the energy consumption rates due to message forwarding. On the other hand, it also attempts to keep message delay rates at low levels by imposing a maximum distance to be traveled by each mobile sink. In addition to all that, our approach also differs from most studies in the literature since the clustering and routing problems are dealt jointly. To the best of our knowledge, the Combinatorial Optimization Problem that arises when all these features are considered together has never been studied before. We present a Mixed Integer Programming formulation for the problem based on network flows and various heuristics to quickly find feasible integer solutions. Our computational experiments indicated that the proposed heuristics works pretty well in practice. We also implemented a simulation framework which embeds the proposed heuristics to validate the network topology introduced here. The simulation results show that our approach leads to significant improvements for all Quality of Service parameters in WSN like energy consumption, network lifetime, among others.

**Keywords:** Wireless Sensor Networks, Combinatorial optimization, Heuristics, Mobile sinks.



# Lista de Figuras

2.1	Possível disposição de uma RSSF para a agricultura. Sensores detectam temperatura, luz e umidade do solo em centenas de pontos distribuídos em uma plantação e se comunicam através de uma rede multi-saltos. . . . .	6
2.2	Exemplos de nós sorvedouros móveis terrestres e aéreos. . . . .	7
2.3	Grafo não direcionado que representa uma rede com comunicação multi-saltos e nó sorvedouro fixo. O sorvedouro é representado pelo vértice mais à esquerda. . . . .	9
2.4	Topologia de uma rede SHS. . . . .	13
2.5	Topologia de uma rede MHS-3, com comunicação limitada a 3 saltos. . . . .	14
2.6	RSSF com $K = 2$ sorvedouros móveis e restrição de saltos $H = 3$ . . . . .	16
3.1	Topologia proposta para o PFMR com $H = 2$ . . . . .	23
4.1	Ilustração da construção de uma árvore de acordo com uma atribuição de rótulos dos vértices alcançados por sua raiz. . . . .	32
4.2	Exemplo de uma <i>troca de 2 elementos</i> na Busca 2-opt. . . . .	35
4.3	Árvore do PAGMS "equivalente" a uma floresta do PFMR com $K = 1$ . . . . .	38
5.1	MICA2 Motes. . . . .	51
5.2	Estados operacionais em uma simulação de RSSF com 250 nós sensores, com restrição de saltos $H = 4$ , utilizando o HRFI-1. . . . .	60
5.3	Fluxograma da simulação utilizando o HRFI-K. . . . .	61
5.4	Cenário da simulação <b>SimInvert_MHS-I</b> para uma RSSF com 150 nós sensores, com restrição de saltos $H = 4$ . . . . .	64
5.5	<b>SimInvert_MHS-I</b> : Atraso na entrega de mensagens em função do número de sensores. . . . .	69
5.6	<b>SimInvert_MHS-I</b> : Energia residual total dos nós sensores em função do tempo de simulação. . . . .	70

5.7	SimInvert_MHS-I: Cobertura em função do tempo de simulação. . . . .	72
5.8	SimInvert_MHS-I: Taxa de mensagens entregues em função do número de nós sensores. . . . .	72
5.9	As duas abordagens para o MHS-I: Cobertura em função do tempo de simulação. . . . .	73
5.10	MHS-I: Atraso em função do número de nós sensores. . . . .	74
5.11	MHS-I: Taxa de mensagens entregues em função do número de nós sensores.	75
5.12	Aumentando o $D_{max}$ : Energia residual total dos nós sensores. . . . .	76
5.13	Aumentando o $D_{max}$ : Cobertura em função do tempo de simulação. . . . .	77
5.14	Aumentando o $D_{max}$ : Atraso na entrega de mensagens. . . . .	78
5.15	Aumentando o $D_{max}$ : Taxa de mensagens entregues. . . . .	78
5.16	Múltiplos sorvedouros: Atraso na entrega de mensagens. . . . .	80
5.17	Múltiplos sorvedouros: Cobertura em função do tempo de simulação. . . . .	81
5.18	Múltiplos sorvedouros: Taxa de mensagens entregues em função do número de nós sensores. . . . .	82



# Lista de Tabelas

4.1	Consumo de corrente do nó sensor MICA2 com transmissão de mensagens.	41
4.2	Resultados computacionais associados ao HRFI-1 e ao BB. . . . .	44
4.3	Resultados computacionais associados ao HRFI-1 e ao BB com solução inicial provida por HRFI-1. . . . .	46
5.1	Parâmetros de simulação. . . . .	65
5.2	Abordagens a serem comparadas. . . . .	68



# Sumário

Agradecimentos	ix
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
<b>1 Introdução</b>	<b>1</b>
1.1 Principais contribuições e organização do texto . . . . .	2
<b>2 Redes De Sensores Sem Fio</b>	<b>5</b>
2.1 Problemas em Redes de Sensores Sem Fios . . . . .	7
2.1.1 Controle de densidade . . . . .	8
2.1.2 Mobilidade em RSSFs . . . . .	9
2.1.3 Agrupamento e roteamento . . . . .	12
2.2 A nossa contribuição . . . . .	15
<b>3 O Problema da Floresta Mínima Com Distância Restrita entre as Raízes</b>	<b>19</b>
3.1 O PFMR como um Problema de Otimização em Grafos . . . . .	19
3.2 Uma formulação de Programação Inteira Mista para o PFMR . . . . .	21
3.3 Comentários . . . . .	27
<b>4 Abordagem heurística e avaliação experimental dos algoritmos de otimização</b>	<b>29</b>
4.1 HRFI-K . . . . .	29
4.1.1 Representação da floresta restrita a $H$ saltos . . . . .	30

4.1.2	Fase construtiva . . . . .	32
4.1.3	Buscas Locais . . . . .	35
4.2	Experimentos computacionais . . . . .	39
4.2.1	Instâncias testes . . . . .	39
4.2.2	Comparação entre os resultados do HRFI-1 e de um algoritmo Branch-and-bound . . . . .	42
4.2.3	Fornecendo ao BB uma solução obtida pelo HRFI-1 . . . . .	45
4.3	Considerações finais . . . . .	47
<b>5</b>	<b>Simulação de RSSFs</b>	<b>49</b>
5.1	Arcabouço de simulação . . . . .	49
5.2	Cenário de simulação . . . . .	50
5.2.1	Modelo de energia para Redes de Sensores sem Fio . . . . .	50
5.2.2	Carga de trabalho . . . . .	55
5.2.3	Camada de enlace . . . . .	55
5.2.4	Modelo de propagação de sinal . . . . .	55
5.2.5	Outras definições para o cenário de simulação . . . . .	56
5.3	Simulação de RSSFs utilizando o HRFI-K . . . . .	57
5.4	Outra abordagem para simulação de RSSFs . . . . .	62
5.4.1	MHS-I . . . . .	62
5.4.2	Simulação de RSSFs utilizando o MHS-I . . . . .	63
5.5	Parâmetros de simulação . . . . .	64
5.6	Medidas de QoS . . . . .	66
5.7	Experimentos computacionais . . . . .	67
5.7.1	Aspectos gerais dos testes . . . . .	68
5.7.2	MHS-I . . . . .	68
5.7.3	HRFI-1 . . . . .	75
5.7.4	Múltiplos sorvedouros . . . . .	79
<b>6</b>	<b>Considerações finais</b>	<b>83</b>
6.1	Conclusões . . . . .	83
6.2	Trabalhos futuros . . . . .	84
	<b>Referências Bibliográficas</b>	<b>87</b>

# Capítulo 1

## Introdução

Os recentes avanços tecnológicos nas áreas de eletrônica digital e telecomunicações possibilitaram o aparecimento de um novo tipo de rede sem fio *ad hoc*, as Redes de Sensores Sem Fio (RSSFs). As RSSFs são compostas por pequenos dispositivos chamados nós sensores, os quais têm capacidade de comunicação sem fio e monitoramento do mundo físico através de dispositivos de sensoriamento. Os nós sensores realizam as tarefas da RSSF de forma colaborativa, disseminando as informações coletadas para outros nós e, eventualmente, para um observador final através de pontos de acesso. Esses pontos de acesso podem ser estações rádio-base, ou até mesmo os nós sensores, chamados nesse caso de nós sorvedouros.

Devido às suas dimensões reduzidas, os nós sensores possuem recursos limitados de processamento computacional, capacidade da bateria e quantidade de memória para armazenamento de dados. Tais limitações, aliadas ao fato de que RSSFs são um novo tipo de rede com particularidades próprias, introduzem novos desafios para o projeto e operação de RSSFs. Dadas suas especificidades, protocolos e algoritmos desenvolvidos para outros tipos de redes são difíceis de serem adaptados às RSSFs.

Por esses motivos, as RSSFs têm sido objeto de estudo de diversas linhas de pesquisa, como por exemplo: algoritmos de roteamento ([Rajaraman \[2002\]](#)), projeto de hardware com baixo consumo de energia, controle de topologia e otimização ([Cardei et al. \[2002a\]](#); [Ye et al. \[2002\]](#); [Zhang & Hou \[2005\]](#)).

A economia de energia costuma ser um dos principais focos de pesquisas em aplicações de RSSFs que necessitam operar por um longo período de tempo, uma vez que, em geral, os nós sensores são lançados em áreas inóspitas e de difícil acesso, impossibilitando a troca ou recarga de suas baterias. Diversos trabalhos, os quais serão citados

ao longo deste texto, consideram a mobilidade de um ou mais nós sorvedouros móveis como forma de reduzir o gasto de energia e, conseqüentemente, estender o tempo de vida da RSSF. O trabalho de [Aioffi et al. \[2007b\]](#), por exemplo, apresenta abordagens centralizadas com sorvedouro móvel e número limitado de saltos. Seus resultados computacionais indicam que RSSFs que operam com sorvedouros móveis possuem tempo de vida muito maior do que redes com sorvedouro estático. Naquele estudo também foi observado que, entretanto, a introdução da mobilidade do nó sorvedouro levou a um grande atraso na entrega de mensagens.

Na abordagem MHS de [Aioffi et al. \[2007b\]](#), primeiramente o problema de agrupamento de nós sensores é resolvido. Em seguida, uma rota do nó sorvedouro (de comprimento idealmente curto), é calculada de forma que ele visite tais agrupamentos. Após a resolução desses problemas, um mecanismo de controle de densidade é empregado, de forma a decidir quais nós sensores devem monitorar a rede em determinado tempo. Um dos objetivos do MHS é reduzir o atraso na entrega de mensagens, através da minimização da rota do nó sorvedouro. Já a economia de energia é tratada apenas através da introdução da mobilidade do nó sorvedouro e do controle de densidade.

O objetivo do nosso trabalho é diminuir o gasto de energia dos elementos da RSSF, tratando os problemas de agrupamento e roteamento de forma integrada, utilizando um ou mais nós sorvedouros móveis. Adicionalmente, pretendemos explorar a flexibilidade que essa abordagem integrada oferece para controlar o atraso na entrega de mensagens. Estudamos também o impacto do controle de densidade sobre o gasto de energia das funções de comunicação dos nós sensores.

## 1.1 Principais contribuições e organização do texto

A principais contribuições deste trabalho são:

- Identificamos um novo Problema de Otimização Combinatória, chamado PFMR, que modela os problemas de agrupamento e roteamento de forma integrada, e considera o gasto de energia na sua função de minimização. Apresentamos também uma formulação de Programação Inteira Mista para o problema.
- Desenvolvemos Heurísticas para obter soluções viáveis para o PFMR. Ao assim proceder, o nosso objetivo é obter um método escalável que possa ser usado no contexto de RSSFs, para a coordenação centralizada do(s) nó(s) sorvedouro(s) da topologia e operação dos elementos da RSSF.

- Apresentamos resultados computacionais avaliando o método proposto, estudando o impacto do número limitado de saltos, do comprimento da rota e da presença de mais sorvedouros móveis no desempenho da rede.
- Apresentamos também uma outra forma de empregar o mecanismo de controle de densidade durante a operação da RSSF. Mostramos que essa outra abordagem permite diminuir ainda mais o gasto de energia em RSSFs com sorvedouros móveis.

O restante desse texto é organizado da seguinte forma. No Capítulo 2, apresentamos os principais problemas em RSSFs tratados em nosso trabalho, bem como uma revisão bibliográfica dos mesmos. No Capítulo 3, introduzimos o PFMR, apresentando uma formulação para o mesmo como um Problema de Otimização em Grafos e um Modelo de Programação Inteira Mista. No Capítulo 4, apresentamos uma heurística para resolver o PFMR. No Capítulo 5, introduzimos um arcabouço de simulação que permite avaliar se o modelo de arquitetura de rede aqui proposto alcança os objetivos para os quais foi concebido. Encerramos a dissertação no Capítulo 6, apresentando as conclusões do trabalho e apontando direções de pesquisa futura.





## Capítulo 2

# Redes De Sensores Sem Fio

Uma Rede de Sensores Sem Fio (RSSF) é um tipo especial de rede móvel *ad hoc* (MANET - Mobile Ad Hoc Networks), composta por centenas, ou mesmo milhares de dispositivos autônomos, chamados nós sensores. Cada nó sensor é um pequeno dispositivo geralmente composto por: um processador com poder de processamento limitado, uma quantidade restrita de memória, uma bateria com energia limitada, placa de sensoriamento utilizada para adquirir dados, e um rádio para a realização da comunicação sem fio. Com esses dispositivos, um nó sensor pode realizar atividades de sensoriamento, respondendo a sinais ou estímulos, processamento e comunicação.

Tipicamente, os nós sensores são distribuídos em uma área com o objetivo de executar tarefas colaborativas, tais como detecção de eventos, monitoração de dados do ambiente, classificação e rastreamento de objetos. Esses dados são monitorados, processados e enviados a outros nós da rede e, eventualmente, a um observador externo (ou usuário final). Esses dados chegam ao observador através de um ou mais pontos de acesso da rede, os quais podem ser estações rádio base ou nós sensores, que nesse caso passam a ser denominados nós sorvedouros. Uma RSSF pode ter um ou mais nós sorvedouros, fixos ou móveis.

As RSSFs podem ser empregadas em diversos contextos, como por exemplo: aplicações industriais (têxtil, espacial), suporte para engenharia de tráfego, monitoramento de animais ou de certas características de um ambiente de difícil acesso (como temperatura, condições atmosféricas, luminosidade), aplicações militares e de segurança (detecção de presença, rastreamento e monitoração de tropas inimigas, detecção de radioatividade e gases tóxicos, captação de comunicação sem fio de tropas inimigas), dentre várias outras aplicações. A Figura 2.1 ilustra uma representação de uma RSSF aplicada à agricultura.



Figura 2.1: Possível disposição de uma RSSF para a agricultura. Sensores detectam temperatura, luz e umidade do solo em centenas de pontos distribuídos em uma plantação e se comunicam através de uma rede multi-saltos.

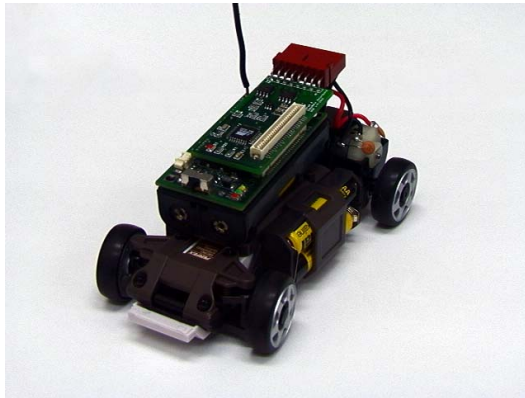
Os nós sensores são os principais componentes de uma Rede de Sensores Sem Fio. Cada nó é responsável por realizar, de forma colaborativa, as tarefas de uma RSSF.

Devido às suas dimensões reduzidas, o nó sensor tem baixo poder de processamento, memória limitada e quantidade de energia bastante restrita. Dessa forma, estender o tempo de vida da rede costuma ser o principal foco das pesquisas em RSSFs, já que, nas principais aplicações, a rede é projetada para operar em um ambiente no qual os nós sensores são distribuídos em áreas de difícil acesso, impossibilitando a troca ou recarga de suas baterias. Essa restrição de energia levou à intensificação dos esforços para o desenvolvimento de um nó sensor de tamanho reduzido com baixo consumo de energia.

Na arquitetura de uma RSSF existe um nó especial, chamado sorvedouro, cuja função é receber os dados coletados e processados pela RSSF, e enviá-los para o usuário final, fora da rede, funcionando como uma ponte (*gateway*) entre a RSSF e o mundo exterior. Uma RSSF pode ter um ou mais nós sorvedouros, fixos ou móveis. Como nó especial, o sorvedouro não possui as restrições de energia do nó sensor: é considerado um dispositivo com energia infinita.

Neste trabalho são tratadas Redes de Sensores Sem Fio nas quais o sorvedouro é um dispositivo móvel. Assume-se que o sorvedouro móvel possua a capacidade de se movimentar em linha reta, a uma velocidade constante. Projetos de robótica podem prover sorvedouros móveis terrestres compatíveis com os considerados nesta dissertação, como nos projetos *CotsBots* ([CotsBots \[2006\]](#)) e *K-Team* ([Khepera-III \[2009\]](#)), mostrados nas Figuras 2.2a e 2.2b, nos quais um nó sensor é acoplado a um veículo

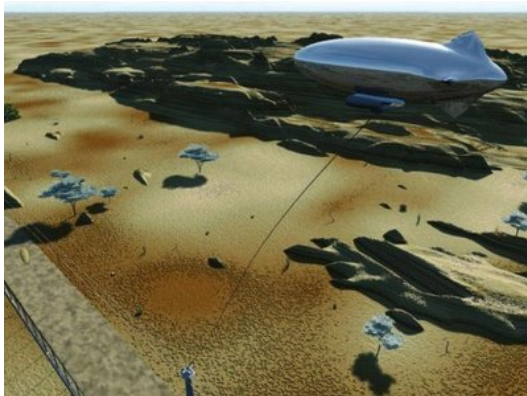
terrestre de pequenas dimensões. Um nó sorvedouro móvel pode também ser aéreo, como por exemplo um pequeno zepelim (Figura 2.2c), o qual poderia sobrevoar a área a uma baixa altitude com um nó sensor acoplado a ele. Outra alternativa seria utilizar um avião controlado remotamente como um nó sorvedouro, como no projeto Wifly (Wifly [2007]).



(a) CotsBots.



(b) Khepera III.



(c) Zepelim sobrevoando um campo.



(d) Avião R/C do projeto Wifly.

Figura 2.2: Exemplos de nós sorvedouros móveis terrestres e aéreos.

## 2.1 Problemas em Redes de Sensores Sem Fios

A disseminação do uso das RSSFs impõe novos desafios para prover algoritmos e protocolos de rede adequados às especificidades de tais redes. Os principais aspectos de interesse em qualidade de serviço em RSSFs são manutenção da cobertura da área monitorada, garantia da conectividade entre os nós sensores ativos e o roteamento dos dados coletados. As pesquisas tornam-se ainda mais desafiadoras visto que todas as soluções propostas devem levar em consideração os limites de energia da rede. Por isso,

a economia de energia é um dos principais interesses em aplicações de RSSFs que necessitam operar por um longo período de tempo. Dessa forma, apresentamos nessa seção os principais problemas em RSSFs tratados em nosso trabalho, os quais consideramos relevantes para contribuir com a diminuição do gasto de energia pelos nós sensores.

### 2.1.1 Controle de densidade

Pesquisas recentes têm mostrado que economias significativas de energia podem ser alcançadas com a adoção de mecanismos de controle de densidade em RSSFs densas. Nessas redes, podem existir vários nós sensores monitorando uma mesma área, resultando em gasto desnecessário de energia, redundância de dados e, principalmente, aumento do tráfego da rede (Cardei et al. [2002b]).

Em uma RSSF na qual implementa-se o controle de densidade, alguns nós sensores são agendados para dormir ou para entrar no modo de economia de energia, enquanto outros nós continuam os serviços de sensoriamento, coleta, processamento e disseminação de dados. O objetivo é geralmente minimizar a área de redundância e manter a cobertura da rede. A (re)ativação de nós que estão *dormindo* pode ser feita através de métodos que deixam o rádio em baixa energia até receber um estímulo externo para ligar o rádio (Polastre et al. [2004]; Correia et al. [2005]). Como com o controle de densidade os nós não ficam ativos o tempo todo, o tempo de vida da RSSF é estendido.

Nessa linha de trabalhos, destaca-se o trabalho de Meguerdichian & Potkonjak [2003], que apresenta várias formulações de Programação Linear Inteira (PLI) para tratar o problema de cobertura em RSSFs, através de métodos que controlam a densidade de nós ativos na rede. Em Nakamura et al. [2005], um PLI também é proposto para agendar tarefas e resolver o problema de cobertura em RSSFs. O mecanismo de controle de densidade em ambos é feito de forma centralizada, ou seja, é controlado pelo nó sorvedouro.

O problema de cobertura também é tratado por Quintão et al. [2004] através da utilização de algoritmos genéticos híbridos. Esse trabalho também trata do problema de conectividade em RSSFs, ou seja, além de garantirem a cobertura da área, o conjunto de nós escolhidos para permanecerem ativos também garante que os dados serão roteados ao ponto de acesso da rede.

Em Siqueira et al. [2006], são apresentadas duas abordagens que combinam o algoritmo de controle de densidade OGDC (Zhang & Hou [2005]) com um algoritmo clássico de roteamento em árvores para RSSFs denominado EFTREE (Figueiredo et al.

[2004]). Os resultados obtidos demonstram a importância da integração desses dois problemas para obter benefícios para a RSSF, como a extensão do tempo de vida da rede e melhor taxa de entrega de mensagens.

Na literatura, além do controle de densidade, muitos trabalhos consideram também a mobilidade do nó sorvedouro para diminuir a energia gasta com roteamento de mensagens e, assim, prolongar a vida útil da rede. Dessa forma, apresentamos a seguir as principais questões relacionadas à mobilidade em RSSFs.

### 2.1.2 Mobilidade em RSSFs

Para permitir a entrega dos dados coletados pelos nós de uma RSSF, vários trabalhos apresentaram redes com nó sorvedouro fixo, nós sensores estáticos e comunicação multi-saltos (Heinzelman et al. [2002]; Figueiredo et al. [2004]; Siqueira et al. [2006]; Ye et al. [2002]; Zhang & Hou [2005]). Nesse contexto, os próprios nós sensores retransmitem a mensagem pela rede até que a mesma alcance o nó sorvedouro (Figura 2.3).

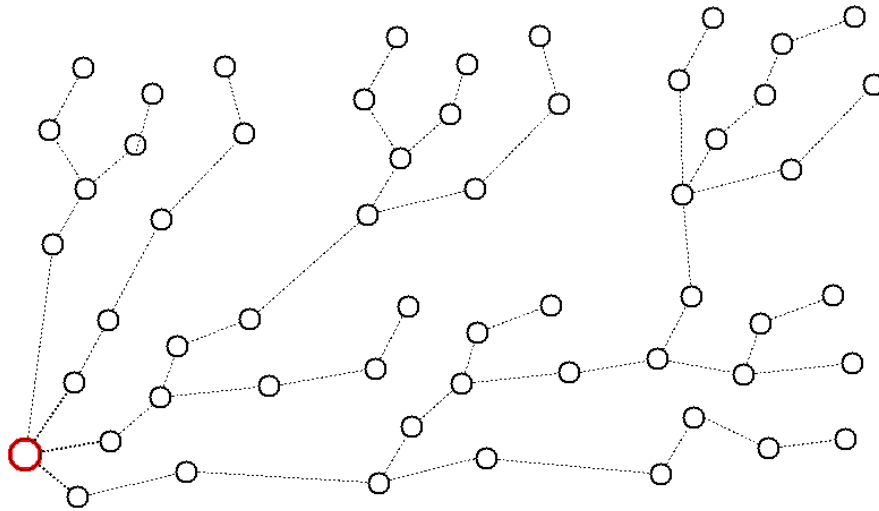


Figura 2.3: Grafo não direcionado que representa uma rede com comunicação multi-saltos e nó sorvedouro fixo. O sorvedouro é representado pelo vértice mais à esquerda.

Entretanto, em Redes de Sensores Sem Fios, a comunicação de dados multi-saltos é a principal responsável pelo consumo de energia (Kim et al. [2003]). Isso inclui o custo de manter o rádio ligado e o custo de transmissão e recepção de dados, o qual inclui a realização do roteamento de mensagens de outros nós sensores até o nó sorvedouro.

Devido a esse alto consumo de energia, vários trabalhos na literatura propõem uma abordagem de comunicação utilizando um número controlado de saltos, sendo en-

tão necessária a utilização de dispositivos móveis para coletar os dados dos nós sensores. Esse esquema de comunicação introduz importantes melhoras para a RSSF, especialmente no aumento do seu tempo de vida. Ocorre a redução do tráfego de mensagens e, conseqüentemente, a redução da energia gasta com roteamento de mensagens pelos nós sensores (Gandham et al. [2003]; Luo & Hubaux [2005]; Papadimitriou & Georgiadis [2005]; Wang et al. [2005b]). Outra vantagem do emprego de nós sorvedouros móveis é a possibilidade de comunicação entre redes que seriam desconexas caso sorvedouros fixos fossem empregados.

Estas alternativas que visam reduzir o consumo de energia acarretam, por outro lado, o aumento do atraso na entrega de mensagens (Wang et al. [2005a]). O atraso é definido como o tempo transcorrido entre a geração de uma mensagem em um nó sensor e a sua recepção pelo nó sorvedouro. Como a velocidade de um dispositivo móvel é geralmente muito menor do que a velocidade com que uma mensagem trafegaria pela rede, via comunicação sem fio, o atraso é fortemente influenciado pela espera da chegada do sorvedouro móvel para transmissão dos dados coletados.

### 2.1.2.1 Mobilidade dos sorvedouros na literatura

Os trabalhos da literatura podem ser divididos de acordo com o tipo de mobilidade do sorvedouro, que pode ser controlável e não controlável. Nas abordagens em que o dispositivo tem mobilidade não controlável, sua trajetória é classificada em previsível e não previsível.

No trabalho de Kim et al. [2003], a trajetória dos dispositivos móveis é não controlável e não previsível. Os sorvedouros se movimentam aleatoriamente pela rede, se comunicando através de sensores específicos chamados *pontos de acesso*. É utilizado um protocolo de disseminação com uma estrutura baseada em árvore, a qual é reconfigurada de tempos em tempos para que os *pontos de acesso* fiquem mais próximos de um sorvedouro móvel.

Outros autores, como Small & Haas [2003] e Jain et al. [2006], também utilizam movimentação não controlável e não previsível: elementos móveis presentes no ambiente de sensoriamento, como animais e carros, são utilizados para movimentar o sorvedouro. Já Chakrabarti et al. [2003] propôs um método em que a movimentação dos sorvedouros pode ser prevista, uma vez que eles são montados sob uma rede de transportes pública na qual seus elementos se movem pela mesma trajetória repetidamente.

O uso de sorvedouros com mobilidade previsível foi explorado nos trabalhos de Mhatre et al. [2005], Tong et al. [2003] e Venkitasubramaniam et al. [2004]. Nesses

trabalhos, os sorvedouros são pequenos aviões, os quais sobrevoam a RSSF e coletam os dados dos nós sensores periodicamente. Apesar de nesses trabalhos o movimento do sorvedouro ser totalmente controlável, ele é externo à infra estrutura da rede, ou seja, sua trajetória não é determinada pela atividade e características dos componentes da rede.

A utilização de veículos não tripulados foi bastante investigada por [Tirta et al. \[2004\]](#). Os nós sensores enviam seus dados para os líderes de grupo (*cluster heads*), através de um esquema de comunicação multi-saltos. Os veículos então passam pelos líderes de grupo e coletam os dados. Três soluções diferentes são avaliadas, as quais definem diferentes abordagens de escalonamento dos veículos que visitam os líderes de grupo. No trabalho de Tirta et al., os sorvedouros não são móveis, e os veículos têm que retornar à estação base periodicamente para entregar os dados da rede.

A idéia de utilizar um pequeno número de elementos da rede como nós móveis foi introduzida por Shah et al. nos seus trabalhos sobre *data MULES* ([Shah et al. \[2003\]](#); [Jain et al. \[2006\]](#)). Nós móveis, chamados *MULES*, coletam os dados sensoreados através de comunicação um-salto (*single hop*), à medida que se movimentam pela rede e ficam dentro do raio de comunicação dos vários nós sensores. Eventualmente o *MULE* entrega os dados ao sorvedouro. A arquitetura do *data MULE* é eficiente em conservação de energia no contexto de RSSFs tolerantes ao atraso ([Small & Haas \[2005\]](#)). Em tal arquitetura, admite-se um crescimento do atraso em favor de uma redução do consumo de energia, ou seja, a energia que seria utilizada para rotear mensagens pela rede é trocada pelo custo de esperar pelo *MULE* para enviar os dados, mais o custo do tempo que o *MULE* leva para chegar ao sorvedouro. No trabalho de [Ekici et al. \[2006\]](#), é explorada a movimentação controlada dos *MULES* para evitar perda de dados dos sensores, o que acontece quando há um overflow do buffer de memória enquanto um nó sensor espera por um *MULE*.

Nos trabalhos de Wang et al. ([Wang et al. \[2005a, 2008\]](#)), é explorado o uso de nós móveis como *relays*, dispositivos com baterias com maior carga do que as tipicamente empregadas nos nós estáticos. Tais dispositivos podem se movimentar pela rede, se posicionando ao lado de um nó sensor e o substituindo nas suas tarefas de roteamento de mensagens e sensoreamento, durante um certo tempo. Os resultados obtidos por meio de simulação naqueles estudos mostram que o tempo de vida da rede dobra em comparação com redes puramente estáticas.

Trabalhos sobre a performance das RSSFs com *relays* mostram que utilizar sorvedouros móveis é mais promissor para se obter um melhor equilíbrio entre o consumo de energia e o atraso na entrega de mensagens. Nesse sentido, [Gandham et al. \[2003\]](#), [Luo](#)

& Hubaux [2005], Papadimitriou & Georgiadis [2005] e Wang et al. [2005b] utilizam algoritmos centralizados, nos quais o sorvedouro planeja a rota e o esquema de comunicação de dados baseado em parâmetros globais da rede. Dessa forma, o sorvedouro se move pela rede com nós sensores estáticos e coleta os dados que são enviados a ele através de comunicação multi-saltos. Outros trabalhos que exploram a mobilidade do nó sorvedouro com controle centralizado são os de Aioffi et al. (Aioffi et al. [2007a,b]), nos quais abordagens um-salto e multi-saltos são testadas e avaliadas segundo várias métricas de desempenho da rede.

### 2.1.3 Agrupamento e roteamento

Em vários trabalhos, o agrupamento de nós sensores (*clustering*) é um método muito utilizado para organizar hierarquicamente os elementos constituintes das RSSFs. O agrupamento de nós sensores proporciona melhorar diversos parâmetros de qualidade da rede, pois permite melhor alocação de recursos, reuso de largura de banda, maior eficiência do controle de potência, aumento da escalabilidade, dentre outros aspectos (Kwon & Gerla [1999]; Heinzelman et al. [2002]).

O agrupamento de nós sensores é utilizado na abordagem SHS (*Single Hop Strategy*), proposta por Aioffi et al. [2007a], para facilitar a coleta de dados pelo sorvedouro móvel. No SHS, os nós sensores que fazem parte de um agrupamento (*cluster*) enviam seus dados ao nó sorvedouro, o qual fica encarregado de encaminhar a mensagem ao usuário final. Assim, é utilizado o esquema de comunicação um-salto, no qual o sorvedouro se comunica diretamente com cada nó da rede. A Figura 2.4 ilustra um cenário em que o SHS foi aplicado a uma RSSF.

A idéia do SHS é definir um conjunto mínimo de agrupamentos, para que assim haja um menor número de pontos de coleta para o nó sorvedouro visitar. Dessa forma, o comprimento da trajetória do sorvedouro será possivelmente menor e, conseqüentemente, o atraso na entrega de mensagens também será menor.

Para que a comunicação ocorra, os nós sensores são organizados em agrupamentos (*clusters*) circulares de raio de comunicação  $R_C$  entre os nós sensores e o sorvedouro. O sorvedouro se posiciona no centro do agrupamento, sendo assim capaz de se comunicar com os sensores com apenas um-salto.

No SHS, a topologia da rede é determinada através de um método de duas fases. A primeira fase consiste na definição dos agrupamentos de nós sensores. Como a intenção é obter o menor número possível de agrupamentos, o problema de agrupamento foi modelado como o *min-size k-clustering problem* (Bilò et al. [2005]) e resolvido utilizando



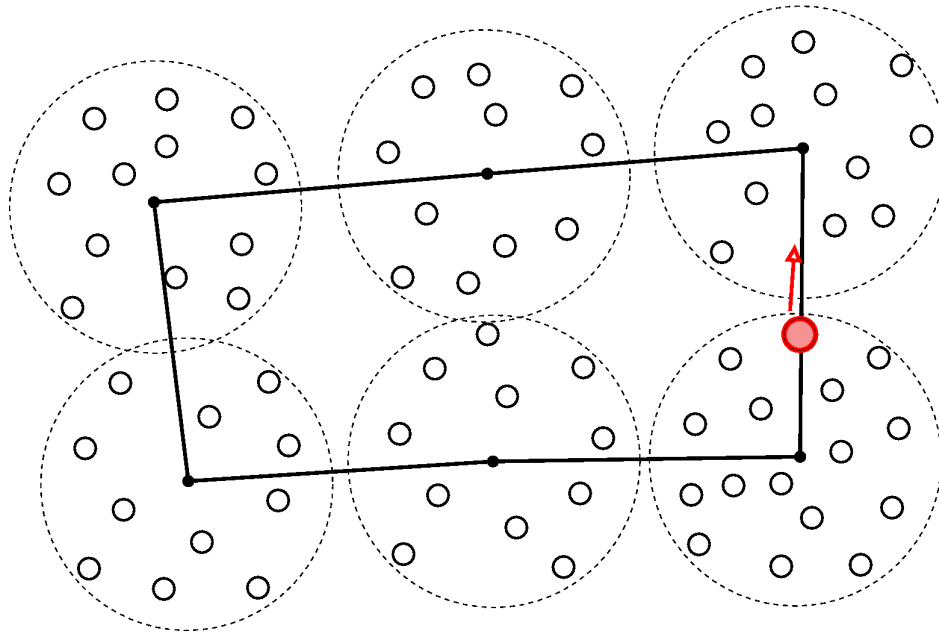


Figura 2.4: Topologia de uma rede SHS.

a heurística gulosa apresentada em [Jain \[1991\]](#).

A segunda fase consiste na resolução do problema de *roteamento*, isto é, no planejamento da trajetória do sorvedouro, que deve visitar o centro de cada agrupamento. Tal trajetória é modelada como um Problema do Caixeiro Viajante (PCV) ([Dantzig et al. \[1954\]](#)), onde cada centro geométrico de um agrupamento é tratado como um vértice a ser visitado. O planejamento da rota, portanto, consiste na definição da ordem em que cada agrupamento será visitado pelo sorvedouro.

A estratégia do SHS mostrou-se eficiente em termos de energia. Entretanto, o atraso na entrega de mensagens ao sorvedouro foi muito alto, uma vez que os nós sensores têm que esperar pela aproximação do nó sorvedouro para darem início à transmissão dos dados.

Para obter uma solução intermediária entre o SHS e uma abordagem com sorvedouro fixo e multi-saltos, [Aioffi et al. \[2007b\]](#) propôs o método MHS- $\lambda$  (*Multi-Hop Strategy*), o qual utiliza comunicação multi-saltos com um número de saltos limitado  $\lambda$ . Devido à comunicação multi-saltos, o MHS permite a criação de agrupamentos maiores e, portanto, o número de pontos de parada do sorvedouro é menor, uma vez que menos agrupamentos são necessários para cobrir a área de sensoriamento. Dessa forma, o atraso na entrega de mensagens tende a ser menor devido à trajetória possivelmente mais curta, uma vez que menos agrupamentos serão visitados. Esse método

multi-saltos, entretanto, possui maior gasto de energia do que o SHS.

Similarmente ao SHS, no MHS a topologia da rede é determinada através de um método de duas fases, no qual primeiramente o problema de agrupamento é tratado, e em seguida, de forma independente, o problema de roteamento é resolvido.

Cada agrupamento do MHS corresponde a uma árvore na qual o número máximo de saltos entre um nó sensor do agrupamento e a raiz do mesmo é no máximo  $\lambda$  saltos. O objetivo é definir uma floresta com o menor número de árvores possível. Para isso, o problema do agrupamento no MHS foi modelado como o Problema p-Centro Invertido (*Inverse p-Center*) (Mirchandani & Francis [1990]). A resolução de tal problema fornece todas as árvores de coleta de dados, assim como suas respectivas raízes. A trajetória do sorvedouro é calculada de modo similar ao SHS, sendo que a raiz de cada árvore é tratada como um vértice a ser visitado no PCV. A Figura 2.5 mostra um cenário em que o MHS-3 foi aplicado a uma RSSF.

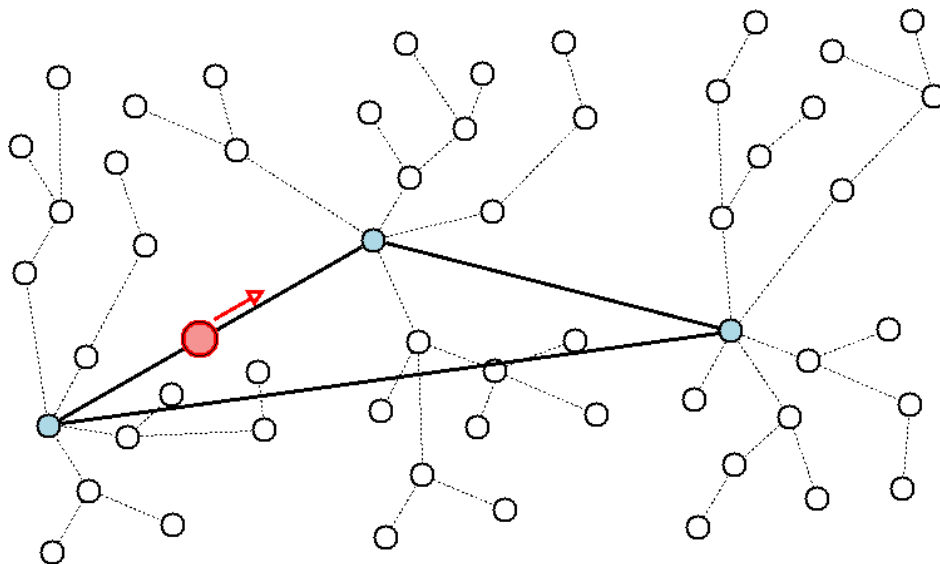


Figura 2.5: Topologia de uma rede MHS-3, com comunicação limitada a 3 saltos.

Tanto na abordagem SHS quanto na abordagem MHS, o controle de densidade é coordenado pelo nó sorvedouro. O problema do controle de densidade foi modelado como o Problema de Cobertura de Conjuntos (*Set Covering Problem*) (Garey & Johnson [1990]). Para modelar o problema, a área a ser monitorada é discretizada em regiões, cada qual com necessidades de sensoriamento homogêneas. Assim sendo, dadas as energias residuais dos nós sensores, deve-se escolher um conjunto mínimo deles que seja capaz de atender a todas as demandas de sensoriamento. Tanto no SHS quanto no MHS, o modelo de Programação Inteira do Problema de Cobertura de Conjuntos resul-

tante é resolvido na otimalidade. O emprego dos mecanismos de controle de densidade permitiu aprimorar tanto a cobertura quanto a vida útil da rede, quando comparados aos métodos SHS e MHS desprovidos de tal mecanismo.

Assim como o SHS, o MHS não necessariamente gera rotas curtas, uma vez que o problema do agrupamento é tratado de modo totalmente independente do problema de roteamento. Agrupamentos com raízes muito distantes umas das outras podem ser gerados, resultando em rotas desnecessariamente grandes, com alto atraso na entrega de mensagens e possível perda de dados.

Além disso, no MHS a distância entre dois nós sensores que se comunicam pode ser muito grande, uma vez que, na solução do Problema p-Centro Invertido, a energia não é considerada para determinação das árvores de comunicação. A resolução do problema do agrupamento consiste em minimizar o número de agrupamentos com no máximo  $\lambda$  saltos, não tratando, portanto, o comprimento dos enlaces entre os nós sensores. Isto é, na resolução do Problema do Agrupamento, não é considerado o gasto energético decorrente da comunicação entre os elementos da rede. A consequência disso, no contexto de RSSFs, é o aumento desnecessário do gasto de energia com transmissão de mensagens.

## 2.2 A nossa contribuição

A arquitetura de rede que propomos neste trabalho é, em alguns aspectos, similar à arquitetura MHS apresentada em [Aioffi et al. \[2007b\]](#). Assim como naquele trabalho, os nós sensores são aqui organizados hierarquicamente por meio de um conjunto de árvores de coleta de dados onde cada aresta representa um enlace de comunicação entre sensores. Nestas árvores, o número de saltos entre qualquer sensor e a raiz da árvore à qual pertence não deve exceder um valor de projeto  $H$ .

Da mesma forma como no MHS, na nossa abordagem os dados da rede são coletados através da comunicação direta entre um nó sorvedouro móvel e as raízes das árvores, aqui denominadas de líderes de grupo (do Inglês *cluster heads*). Dessa forma, cada nó sensor tem que enviar seus dados à raiz de sua árvore, sendo essa, por sua vez, a responsável por enviar esses dados ao sorvedouro que os solicitou.

As diferenças entre a abordagem aqui proposta e o MHS são muitas. Por um lado, consideramos a energia gasta na comunicação entre nós sensores no problema de otimização que formulamos para projetar a rede. A cada possível enlace de comunicação entre dois nós sensores, associamos um peso ou custo que representa uma estimativa

da energia gasta quando os sensores se comunicam. O modelo aqui proposto visa então minimizar o consumo de energia na rede, isto é, a soma dos pesos dos arcos empregados nas árvores de comunicação de dados.

Por outro lado, em nossa abordagem tratamos RSSFs com exatamente  $K$  sorvedouros móveis. Apenas um deles é considerado o *sorvedouro principal*, o qual é responsável por tomar todas as decisões em relação às rotas e à topologia da rede, e repassá-las aos demais sorvedouros, chamados de *sorvedouros auxiliares*. Cada um dos  $K$  sorvedouros considerados no nosso trabalho percorre uma rota que corresponde a uma trajetória que envolve apenas líderes de grupo, tal que cada líder de grupo é um ponto de parada para coleta de dados dos nós sensores. Todas as  $K$  rotas têm um mesmo líder de grupo como ponto de partida/chegada, chamado aqui de *ponto de encontro*. Excetuando-se esse líder de grupo em particular, os conjuntos de líderes de grupo visitados por cada nó sorvedouro são distintos. A Figura 2.6 ilustra uma RSSF com dois nós sorvedouros e restrição de saltos  $H = 3$ .

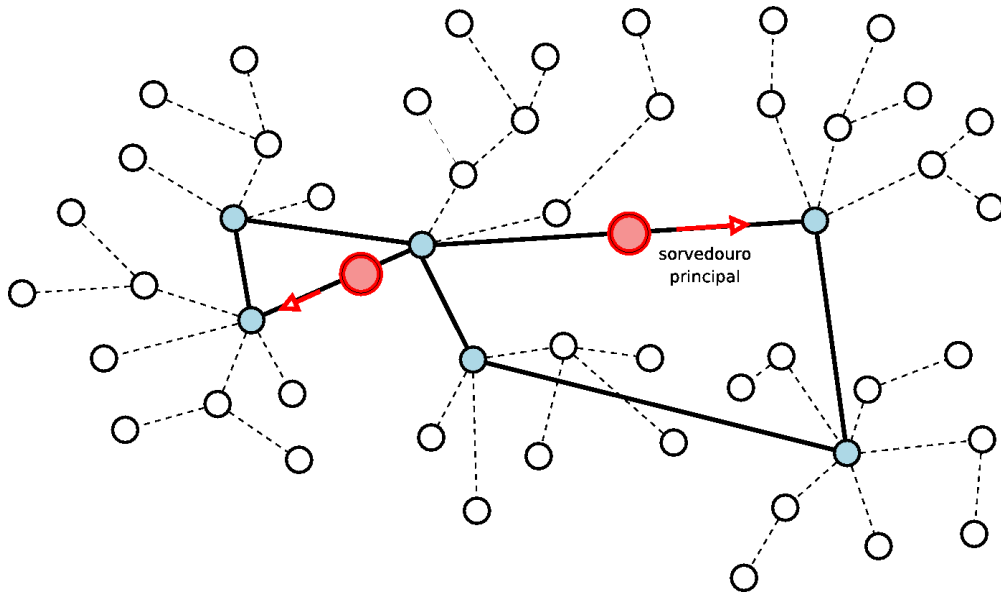


Figura 2.6: RSSF com  $K = 2$  sorvedouros móveis e restrição de saltos  $H = 3$ .

O atraso médio na entrega de mensagens em RSSFs com sorvedouros móveis é aproximadamente proporcional ao tempo médio em que os nós sensores esperam pela aproximação de um sorvedouro para enviar seus dados. Dessa forma, decidimos controlar o atraso na entrega de mensagens por meio da introdução de uma restrição que limita superiormente o comprimento da trajetória dos sorvedouros a um parâmetro de projeto  $D_{max}$ . Assim, o atraso na entrega de mensagens pode ser, de certa forma, controlado pelo projetista da RSSF: de posse do máximo atraso na entrega de men-

sagens "desejado", e da velocidade média dos sorvedouros, é possível definir o valor do parâmetro  $D_{max}$  que atenda aos requisitos de projeto.

Além de controlar o atraso na entrega de mensagens, a estratégia de limitar o comprimento da rota dos sorvedouros também permite, indiretamente, controlar o gasto de energia. Isto é verdade porque quanto menor forem as rotas dos sorvedouros, mais vezes eles percorrerão a área monitorada em um mesmo intervalo de tempo e, conseqüentemente, mais frequentemente eles se comunicarão com os nós sensores, gerando maior gasto de energia com transmissão de dados. Assim, poderia ser do interesse do usuário definir um parâmetro  $D_{max}$  um pouco maior para que haja maior economia de energia.

Cabe observar que, para  $K = 1$ , a arquitetura da rede proposta torna-se bastante parecida com a do MHS apresentada em [Aioffi et al. \[2007b\]](#). Entretanto, mesmo nesse caso, por realizarmos o planejamento da rede tratando o problema de agrupamento (definição das árvores de coleta de dados e suas raízes) e de roteamento de forma integrada, nos distanciamos significativamente da abordagem MHS. Isto porque no MHS tais problemas são tratados de forma independente.

O modelo de arquitetura de rede que propomos deu origem a um novo Problema de Otimização Combinatória que, a julgar pela nossa revisão bibliográfica, é inédito na literatura. Trata-se de um Problema de Otimização NP-difícil, similar ao PAGMS - Problema da Árvore Geradora Mínima Restrita por Saltos (Hop-Constrained Minimum Spanning Tree Problem, [Gouveia \[1996\]](#)). O problema aqui estudado, denominado Problema da Floresta Mínima Com Distância Restrita entre as Raízes (PFMR), introduz uma restrição adicional ao PAGMS, tornando-o ainda mais difícil de ser resolvido. Essa restrição adicional refere-se à limitação do comprimento das  $K$  rotas que envolvem os líderes de grupo.

No próximo Capítulo, introduzimos o PFMR, apresentando uma formulação para o mesmo como um Problema de Otimização em Grafos e um modelo de Programação Inteira Mista.



## Capítulo 3

# O Problema da Floresta Mínima Com Distância Restrita entre as Raízes

Neste capítulo formalizamos o Problema da Floresta Mínima Com Distância Restrita entre as Raízes (PFMR). Apresentamos uma formulação em Grafos, bem como uma formulação de Programação Inteira Mista, tratando os problemas de agrupamento e roteamento de forma integrada, minimizando o custo da floresta de coleta de dados e restringindo o tamanho máximo das rotas dos nós sorvedouros.

### 3.1 O PFMR como um Problema de Otimização em Grafos

Para apresentar o problema objeto de estudo desta dissertação como um Problema de Otimização em Grafos, empregaremos um conjunto de vértices  $V = \{1, \dots, n\}$  que denota o conjunto de nós sensores da RSSF.

O modelo que pretendemos apresentar deve ser capaz de capturar os aspectos essenciais do PFMR: o consumo de energia e o atraso na entrega de mensagens em RSSFs. Por esse motivo, utilizaremos o grafo  $D = (V, A)$ . O conjunto de arcos  $A$  será empregado para modelar as trajetórias dos  $K$  sorvedouros. Para modelar as árvores de comunicação, apenas um subconjunto  $\hat{A} \subseteq A$  será empregado.

Vamos primeiramente apresentar a modelagem das trajetórias dos  $K \in \mathbb{N}_+$  nós

sorvedouros. Para tanto, utilizamos o conjunto de arcos  $A$ . Assumimos que  $A$  denota um conjunto completo de arcos em  $V = \{1, \dots, n\}$ . A todos os arcos de  $A$ , associamos distâncias Euclidianas  $\{d_{ij} : i, j \in V, i \neq j\}$ . Definimos que o vértice  $1 \in V$  é o ponto de encontro (inicial/final) das trajetórias dos nós sorvedouros e, portanto, necessariamente deverá ser um líder de grupo, ou seja, a raiz de alguma árvore de comunicação.

Cada nó sorvedouro  $k \in \mathcal{K} = \{1, \dots, K\}$  visita um número  $t(k)$  de líderes de grupo. Assim, temos que a trajetória do nó sorvedouro  $k \in \mathcal{K}$  consiste na visita aos líderes de grupo em  $R^k = \{r_1^k, \dots, r_{t(k)}^k : k \in \mathcal{K}\} \subseteq V$ , de forma que induzam um circuito Hamiltoniano em que o comprimento total dos arcos  $(r_1^k, r_2^k), \dots, (r_{t(k)-1}^k, r_{t(k)}^k), (r_{t(k)}^k, r_1^k)$  é não superior a  $D_{max}$ .

Nesse contexto,  $\{r_j^k, j = 2, \dots, t(k)\}$  representam líderes de grupo visitados unicamente pelo nó sorvedouro  $k \in \mathcal{K}$ . Além disso, como todas as trajetórias têm o vértice 1 como ponto de partida, temos que  $r_1^1 = r_1^2 = \dots = r_1^K = 1$ . Portanto, o número total  $\bar{t}$  de líderes de grupo em  $R := R^1 \cup R^2 \cup \dots \cup R^K$  é dado por  $\bar{t} = 1 - K + \sum_{k \in \mathcal{K}} t(k)$ .

Vamos discutir agora como definimos o subconjunto de arcos  $\hat{A}$  de  $A$ , que poderão ser empregados para construir as árvores de comunicação. Consideremos as distâncias Euclidianas  $\{d_{ij} : i, j \in V, i \neq j\}$  entre os vértices de  $V$ . Assuma que a cada vértice  $i \in V$  seja associado um número real não negativo  $e_i$  que denota a energia residual do nó sensor representado por  $i$ . Assuma também que o raio máximo de comunicação dos nós sensores é dado por  $R_C \in \mathbb{R}_+$ . Aos arcos de  $\hat{A}$  são associados os custos  $\{c_{ij} \geq 0 : [i, j] \in \hat{A}\}$ , que representam a energia necessária para transmitir um pacote, de tamanho fixo, de  $i$  para  $j$ . Se  $i \in V$  possuir energia  $e_i$  suficiente para comunicação com  $j \in V$ , e se  $d_{ij}$  for menor ou igual a  $R_C$ , o arco  $[j, i]$  é incluído em  $\hat{A}$ . Adicionamos o arco  $[j, i]$  ao invés de  $[i, j]$  pois o sentido de transmissão das informações sensoreadas entre os nós sensores é contrário ao do arco na modelagem proposta.

As árvores de comunicação são representadas pela floresta  $F = (V, A_F)$  de  $D$ , que consiste em uma coleção de arborescências  $T_i^k = (V_i^k, A_i^k)$ ,  $k \in \mathcal{K}$ ,  $i = 1, \dots, t(k)$ , tal que  $A_F \subseteq \hat{A}$  e  $A_F = \bigcup_{k \in \mathcal{K}} \bigcup_{i=1}^{t(k)} A_i^k$ . Para garantir a cobertura da rede, os conjuntos de vértices  $V_i^k$  são tais que  $\bigcup_{k \in \mathcal{K}} \bigcup_{i=1}^{t(k)} V_i^k = V$ . Dado o valor de projeto  $H \in \mathbb{N}_+$ , dizemos que  $F$  é uma floresta restrita por  $H$  saltos de  $D$  se, para cada arborescência  $T_i^k$ , o número máximo de arcos existentes no caminho (único) de qualquer vértice  $j \in V_i^k \setminus \{r_i^k\}$  até a raiz  $r_i^k$  da arborescência não excede  $H$ . Dizemos então que  $F$  é uma floresta restrita a  $H$  saltos. Nesse contexto, o vértice  $r_i^k$  representa o líder de grupo da arborescência  $T_i^k$ , o qual é visitado pelo nó sorvedouro  $k \in \mathcal{K}$ .

Vale mencionar que, quando  $D_{max} = 0$ , o PFMR reduz ao PAGMS (Problema da



Árvore Geradora Mínima com Restrição de Saltos, apresentado em Gouveia [1996]), um Problema de Otimização Combinatória NP-difícil.

Formalmente, no PFMR, o objetivo é encontrar uma floresta restrita a  $H$  saltos, minimizando a função de custo:

$$\min \sum_{[i,j] \in A_F} c_{ij} \quad (3.1)$$

Sujeita a:

- $\forall k \in \mathcal{K}, \forall i = 1, \dots, t(k), T_i^k = (V_i^k, A_i^k)$  induz uma arborescência, orientada a partir de  $r_i^k$ , com no máximo  $H$  saltos no caminho de qualquer  $j \in V_i \setminus \{r_i^k\}$  até  $r_i^k$ .
- $\forall k \in \mathcal{K}$ , existe um circuito Hamiltoniano que visita os vértices em  $\{r_1^k, \dots, r_{t(k)}^k\}$ , cujo comprimento seja não superior a  $D_{max}$ .

Cabe destacar que o número de arborescências  $\bar{t}$  e suas raízes  $r_i^k$ , assim como o número  $t(k)$  de raízes que cada nó sorvedouro  $k \in \mathcal{K}$  tem que visitar, não são pré-definidos. Na verdade, encontrar tais raízes faz parte do problema que pretendemos resolver.

Deste ponto em diante, passaremos a denominar as arborescências  $T_i^k$  como árvores, a fim de dar mais simplicidade e leveza ao texto.

## 3.2 Uma formulação de Programação Inteira Mista para o PFMR

Para formular o PFMR como um Programa Inteiro Misto utilizaremos duas redes: a *rede de energia*, que modela as árvores de comunicação, e a *rede de translação*, que modela as trajetórias dos nós sorvedouros.

Uma vez que os modelos de Programação Matemática que apresentamos para o PFMR são modelos de fluxos em redes, faremos algumas modificações no grafo de definição do problema, que contemplam a inclusão de vértices e arcos artificiais. Estas modificações nos permitirão representar os problemas de comunicação entre os nós sensores e o problema de roteamentos dos sorvedouros como Problemas que envolvem determinar caminhos sujeitos a restrições adicionais, e a restrições de acoplamento entre os dois supracitados problemas.

Vamos definir primeiramente a *rede de translação*. Sabemos que os nós sorvedouros iniciam suas rotas no vértice  $1 \in V$ , e visitam um (desconhecido) conjunto de outros nós sensores. Como as trajetórias dos nós sorvedouros consistem em um circuito Hamiltoniano, utilizaremos nesta formulação o nó artificial  $n^*$ , o qual representa uma cópia do vértice inicial 1 e deve ser o último a ser "visitado". Dessa forma, podemos modelar cada trajetória como um caminho simples, cujos vértices de origem e destino são, respectivamente, 1 e  $n^*$ . Para isso, as distâncias  $d_{in^*}$  entre os vértices  $i \in V \setminus \{1\}$  e  $n^*$  são definidas como  $d_{in^*} = d_{i1} = d_{1i}$ .

Observe que o subgrafo de  $(V, A)$  que define as rotas para as quais um conjunto de vértices  $R^K$  é visitado pode ser mapeado em um conjunto simples de  $D'$ , cujo vértice inicial é  $r_1^k = 1$  e destino é  $n^*$ , e cujos vértices internos são  $R^k \setminus \{r_1^k\}$ .

Assim, para modelar as rotas dos sorvedouros, utilizamos o digrafo  $D' = (V', A')$ , no qual temos o conjunto de vértices  $V' = V \cup \{n^*\}$ , e  $A'$  é o conjunto completo de arcos definido sobre  $V'$ . O digrafo  $D' = (V', A')$ , juntamente com as distâncias  $\{d_{ij} : i \neq j, i, j \in V'\}$ , define a *rede de translação*.

Definimos agora a *rede de energia* da nossa formulação proposta para o PFMR. Para este propósito, considere o grafo direcionado  $\bar{D} = (\bar{V}, \bar{A})$ , onde  $\bar{V} = \{0\} \cup V'$  e  $\bar{A} = \hat{A} \cup \{[0, i] : i \in V'\}$ . Observe que  $\bar{D}$  contempla os vértices  $V'$  e um novo vértice artificial 0, bem como arcos artificiais  $\{[0, i] : i \in V'\}$ . A estes arcos artificiais atribuímos custos  $\{c_{0i} = 0, i \in V'\}$ . O digrafo  $\bar{D}$  é utilizado para modelar a comunicação e a energia da rede da nossa aplicação.

É possível notar que uma floresta restrita a  $H$  saltos em  $\bar{D}$  pode ser mapeada em uma árvore restrita a  $H + 1$  saltos em  $\bar{D}$  enraizada em 0, e vice-versa. Para isso, considere uma árvore restrita a  $H + 1$  saltos em  $\bar{D}$ , cuja raiz é 0. Por definição, o caminho entre qualquer vértice  $i \in V \cup \{n^*\}$  e 0 não pode possuir mais do que  $H + 1$  arcos. Por construção, esses caminhos consistem em arcos artificiais de 0 a cada um dos vértices em  $\{1, n^*\} \cup \{r_i^k : k \in \mathcal{K}, i = 2, \dots, t(k)\}$ . Assuma que  $\bar{t} \geq 1$  desses arcos artificiais existam na árvore. Se removermos o vértice 0 e os  $\bar{t}$  arcos  $\{[0, 1] \cup [0, n^*]\} \cup \{[0, r_i^k] : k \in \mathcal{K}, i = 2, \dots, t(k)\}$ , o grafo resultante tem que necessariamente induzir uma floresta restrita a  $H$  saltos em  $\bar{D}$ . Por esse motivo, a *rede de energia* da nossa formulação, representada pelo digrafo  $\bar{D}$ , consiste em uma árvore restrita a  $H + 1$  saltos enraizada em 0.

Para melhor visualizar a topologia das redes propostas na nossa Formulação de Fluxos, a Figura 3.1 ilustra a modelagem de uma RSSF com  $K = 2$  nós sorvedouros, e parâmetro de restrição de saltos  $H = 2$ . As linhas contínuas modelam, na rede de

translação, as trajetórias dos sorvedouros, cujos vértices de origem e destino são 1 e  $n^*$ . Já as linhas pontilhadas modelam, na rede de energia, a comunicação entre os nós sensores. As setas indicam o sentido do fluxo das mercadorias pelas duas redes. No caso da rede de energia, esse sentido é contrário ao da transmissão de informações sensoreadas entre os nós sensores na RSSF. Note que  $n^*$  não tem filhos, uma vez que ele é um vértice artificial, sem função na RSSF, que representa uma cópia do vértice 1.

Para que as restrições do PFMR sejam satisfeitas, cada nó sensor  $r_i^k : i = 1, \dots, t(k)$  visitado pelo sorvedouro  $k \in \mathcal{K}$  deve ser um nó filho do vértice artificial 0, na árvore restrita a  $H + 1$  saltos em  $\bar{D}$ .

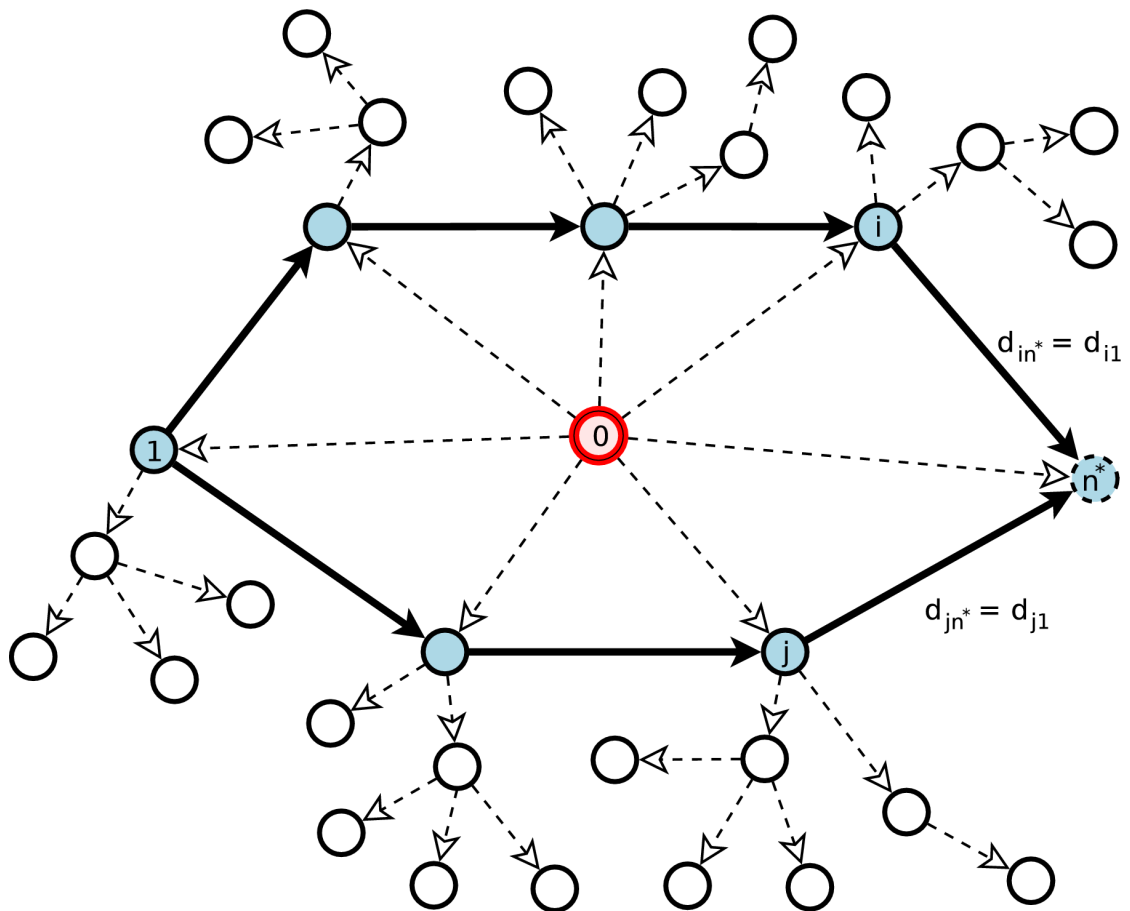


Figura 3.1: Topologia proposta para o PFMR com  $H = 2$ .

Uma vez introduzidos os grafos  $D'$  e  $\bar{D}$ , podemos agora discutir como formulamos o PFMR como um Problema de Fluxos em Redes. As variáveis de decisão que empregamos foram as seguintes:

- Para modelar a árvore restrita com  $H + 1$  saltos (*rede de energia*):
  - (i) variáveis binárias  $z_{ij} \in \{0, 1\}$ ,  $\forall [i, j] \in \bar{A}$ , assumindo valor 1 se arco  $[i, j] \in \bar{A}$  pertence à árvore (0 caso contrário);
  - (ii) variáveis binárias  $y_i^k \in \{0, 1\}$  :  $\forall i \in V'$ , assumindo valor 1 se  $i$  é um líder de grupo visitado pelo sorvedouro  $k \in \mathcal{K}$  e, portanto, o mesmo se conecta diretamente ao vértice artificial 0 (valor 0, caso contrário);
  - (iii) variáveis reais não negativas  $f_{ij}^q, \forall q = 1, \dots, n, \forall [i, j] \in \bar{A}$ , denotando o fluxo de uma mercadoria  $q$  na *rede de energia*, através do arco  $[i, j]$ .
- Para modelar as rotas dos sorvedouros (*rede de translação*):
  - (i) variáveis binárias  $y_i^k \in \{0, 1\}$ ,  $\forall i \in V'$ , indicam se o vértice  $i$  é visitado pelo sorvedouro  $k \in \mathcal{K}$ ;
  - (ii)  $w_{ij}^k \in \{0, 1\}$ ,  $\forall [i, j] \in A'$ , indicando se o líder de grupo  $j$  é visitado logo após o líder de grupo  $i$ , na rota do sorvedouro  $k \in \mathcal{K}$ ;
  - (iii) variáveis reais não negativas  $x_{ij}^v \in \mathbb{R}_+$ , representado o fluxo de uma mercadoria  $v \in V'$  pela *rede de translação* através do arco  $[i, j] \in A'$ .

A Formulação de Fluxos proposta para o PFMR é então dada por:

$$\min l = \sum_{[i,j] \in \bar{A}} c_{ij} z_{ij} \quad (3.2)$$

$$\sum_{i \in V \setminus \{1\}} x_{1i}^v = \sum_{k \in \mathcal{K}} y_i^k, \quad \forall v \in V' \setminus \{1\}, \quad (3.3)$$

$$\sum_{[i,j] \in A'} x_{ij}^i - \sum_{[j,i] \in A'} x_{ji}^i = - \sum_{k \in \mathcal{K}} y_i^k, \quad \forall i \in V' \setminus \{1\}, \quad (3.4)$$

$$\begin{aligned} \sum_{[i,j] \in A'} x_{ij}^v - \sum_{[j,i] \in A'} x_{ji}^v &= 0, \\ \forall v, i \in V' \setminus \{1\} : v &\neq i, \end{aligned} \quad (3.5)$$

$$\begin{aligned} x_{ij}^v &\leq w_{ij}^k \\ w_{ij}^k &\leq y_i^k, \quad \forall k \in \mathcal{K}, \quad \forall v \in V' \setminus \{1\}, \quad \forall [i, j] \in A', \\ w_{ij}^k &\leq y_j^k \end{aligned} \quad (3.6)$$

$$\sum_{[i,j] \in A'} w_{ij}^k d_{ij} \leq D_{max}, \quad \forall k \in \mathcal{K}, \quad (3.7)$$

$$\sum_{j \in V' \setminus \{1\}} w_{1j}^k = 1, \forall k \in \mathcal{K}, \quad (3.8)$$

$$\sum_{j \in V} w_{n^*j}^k = 0, \forall k \in \mathcal{K}, \quad (3.9)$$

$$\sum_{k \in \mathcal{K}} y_i^k \leq 1, \forall i \in V' \setminus \{1, n^*\}, \quad (3.10)$$

$$\sum_{j \in V', j \neq i} w_{ij}^k \leq y_i^k, \forall k \in \mathcal{K}, \forall i \in V' \setminus \{1\}, \quad (3.11)$$

$$\sum_{j \in V', j \neq i} w_{ji}^k = y_i^k, \forall k \in \mathcal{K}, \forall i \in V' \setminus \{1\}, \quad (3.12)$$

$$\sum_{[i,j] \in \bar{A}} z_{ij} = n + 1, \quad (3.13)$$

$$\sum_{j \in \bar{V}} f_{0j}^q = 1, \forall q \in V', \quad (3.14)$$

$$\sum_{[i,j] \in \bar{A}} f_{ij}^q - \sum_{[j,i] \in \bar{A}} f_{ji}^q = 0, \forall i, q \in V', i \neq q, \quad (3.15)$$

$$\sum_{[i,j] \in \bar{A}} f_{ij}^i - \sum_{[j,i] \in \bar{A}} f_{ji}^i = -1, \forall i \in V', \quad (3.16)$$

$$f_{ij}^q \leq z_{ij}, \forall q \in V', \forall [i,j] \in \bar{A}, \quad (3.17)$$

$$\sum_{[i,j] \in \bar{A}} f_{ij}^q \leq H + 1, \forall q \in V', \quad (3.18)$$

$$\sum_{[i,j] \in \bar{A} \setminus \{[0,j]\}} z_{ij} + \sum_{k \in \mathcal{K}} y_j^k = 1, \forall j \in V \setminus \{1, n^*\}, \quad (3.19)$$

$$\sum_{[i,j] \in \bar{A} \setminus \{[0,j]\}} z_{ij} = 0, \forall j \in \{1, n^*\}, \quad (3.20)$$

$$z_{0i} = y_i^k, \forall k \in \mathcal{K}, \forall i \in V', \quad (3.21)$$

$$y_1^k = 1, \quad y_{n^*}^k = 1, \quad \forall k \in \mathcal{K}, \quad (3.22)$$

$$\begin{aligned} x_a^v \in \mathbb{R}_+, \forall a \in A', \forall v \in V'; \quad w_{ij}^k \in \mathbb{B}, \forall k \in \mathcal{K}, \forall [i,j] \in A'; \\ y_i^k \in \mathbb{B}, \forall i \in V', \forall k \in \mathcal{K}; \quad z_{ij} \in \mathbb{B}, \forall [i,j] \in \bar{A}; \quad f_a^q \in \mathbb{R}_+, \forall a \in \bar{A}, \forall q \in \bar{V}. \end{aligned} \quad (3.23)$$

A idéia central da formulação (3.2)-(3.23) consiste em criar demandas e ofertas de mercadorias artificiais nos vértices  $V \cup \{n^*, 0\}$  e tentar distribuí-las utilizando as redes de translação e de energia que foram apresentadas. Na formulação acima, o problema de

definir a trajetória dos sorvedouros foi representado pelas restrições (3.3)-(3.12). Trata-se de um Problema de Fluxo de custo mínimo definido sobre a rede  $D' = (V', A')$ , o parâmetro  $D_{max}$ , e distâncias  $\{d_{ij} : [i, j] \in A'\}$ . Neste PFCM, empregamos uma mercadoria para cada vértice de  $V' \setminus \{1\}$ , disponibilizada em 1. A demanda de cada uma delas encontra-se distribuída em cada um dos nós sensores que fazem parte de uma das  $K$  rotas (vértices que são líderes de grupo). Assim, cada mercadoria na rede de translação deve ser entregue do ponto inicial da rota (vértice 1) ao seu destino usando um caminho simples. As variáveis  $w_{ij}^k$  é que são responsáveis por definir de qual rota (dentre as  $K$  rotas) faz parte cada arco utilizado na rede de translação.

O conjunto de restrições (3.3) garante que o fluxo de cada mercadoria  $v \in V' \setminus \{1\}$ , tal que  $v$  é um líder de grupo, tenha início no vértice 1. Se o vértice  $v$  não for um líder de grupo, ou seja,  $y_v^k = 0, \forall k \in \mathcal{K}$ , não haverá a oferta da mercadoria correspondente em 1 e, conseqüentemente, não haverá o fluxo da mercadoria  $v$  pela rede de translação. As restrições (3.4) e (3.5) modelam o balanço do fluxo de cada mercadoria  $v \in V' \setminus \{1\}$  pela rede de translação, nos vértices de transbordo e de destino.

As restrições (3.6) fazem o acoplamento lógico entre as variáveis  $y$ ,  $w$  e  $x$  que definem as trajetórias dos sorvedouros pelos líderes de grupo. Já a restrição (3.7) limita superiormente o comprimento total dessas trajetórias.

As restrições (3.8) e (3.9) definem os vértices inicial e final das trajetórias dos sorvedouros no modelo, sendo eles, respectivamente, 0 e  $n^*$ . Já o conjunto de restrições (3.10) garante que cada líder de grupo seja visitado por apenas um dos sorvedouros  $k \in \mathcal{K}$ , à exceção dos vértices 1 e  $n^*$  que consistem nos pontos de partida e chegada de todos os sorvedouros.

As restrições (3.11) e (3.12) evitam que as trajetórias dos sorvedouros contenham arcos que incidam a vértices que não sejam líderes de grupo (aqueles cujo valor de  $y_i^k = 0$ ).

Também para modelar a árvore de comunicação restrita a  $H + 1$  saltos, empregamos argumentos de fluxos em redes, dados pelas restrições (3.13)-(3.20). Neste modelo, entretanto, empregamos uma mercadoria para cada vértice de  $\bar{V} \setminus \{0\}$  disponibilizada em 0. A demanda de cada uma delas no PFCM associado encontra-se, obviamente, distribuída em cada um dos nós sensores. Neste caso, cada mercadoria na rede de energia deve ser entregue da origem (vértice 0) ao seu destino usando um caminho simples que não emprega mais de  $H + 1$  arcos.

A restrição (3.13) define o número total de arcos utilizados na rede de energia. Uma vez que  $n = |V|$  o número total de arcos utilizados na rede de energia é, então,

igual a  $n + 1$ , já que as variáveis  $z$  induzem uma árvore em  $\bar{D} = (\bar{V}, \bar{A})$ , cujo conjunto de vértices  $\bar{V} = V \cup \{0, n^*\}$  possui  $n + 2$  elementos.

O conjunto de restrições (3.14) faz com que o nó artificial 0 seja a origem do fluxo de todas as mercadorias pela rede de energia, uma vez que a árvore de comunicação consiste em uma árvore restrita a  $H + 1$  saltos enraizada em 0. Já as restrições (3.15) e (3.16) garantem o balanço do fluxo dessas mercadorias pela árvore.

O acoplamento entre as variáveis  $z$  e  $f$  é assegurado através das restrições (3.17). Já as restrições (3.18) garantem que não haja mais do que  $H + 1$  saltos entre qualquer vértice  $i \in V \cup \{n^*\}$  e o vértice 0 na árvore de comunicação.

As equações (3.19) e (3.20) garantem que, para cada nó  $j$  que é um líder de grupo, nenhum arco em  $\bar{A} \setminus \{[0, i] : i \in V'\}$  pode ser incidente a ele na *rede de energia*. As restrições (3.21) impõem que sempre que o vértice  $i \in V'$  é um líder de grupo, o arco artificial correspondente  $[0, i]$  deve ser incidente a ele.

Finalmente, as restrições (3.22) definem os valores fixos das variáveis  $y_1^k$  e  $y_{n^*}^k$  necessários para modelar os pontos inicial e final das trajetórias dos sorvedouros  $k \in \mathcal{K}$ , e as restrições (3.23) definem os valores válidos para as variáveis de decisão.

### 3.3 Comentários

Nesse capítulo foram apresentadas uma formulação em grafos e uma formulação de Programação Inteira Mista para o PFMR.

Neste trabalho, empregamos a formulação de Programação Inteira Mista proposta em um algoritmo *Branch-and-bound*, utilizando o pacote de otimização comercial **ILOG CPLEX Solver** [2009]. Entretanto, o tempo computacional necessário para resolver o PFMR dessa forma é grande mesmo para um pequeno número de vértices. Nesse sentido, no próximo Capítulo é apresentada uma abordagem heurística, bem como os resultados dos experimentos computacionais das duas abordagens.





## Capítulo 4

# Abordagem heurística e avaliação experimental dos algoritmos de otimização

Neste trabalho, propomos uma arquitetura de rede em que todas as decisões referentes à topologia são tomadas pelo sorvedouro principal. Isso implica que, dentre outras atribuições, o sorvedouro principal é o responsável por resolver o PFMR. Como para um melhor desempenho da RSSF, todas as decisões têm que ser tomadas de forma rápida, apresentamos neste capítulo uma heurística, denominada HRFI-K (Heurística para Construção e Otimização da Rota e Floresta de forma Integrada) para o PFMR, em que  $K$  denota o número de nós sorvedouros presentes na RSSF considerada. Ao final do Capítulo, apresentamos resultados computacionais obtidos comparando os algoritmos estudados.

### 4.1 HRFI-K

A heurística HRFI-K possui três componentes principais. O primeiro é um algoritmo guloso que procura construir, de forma integrada, as rotas candidatas para os nós sorvedouros e a floresta de coleta de dados restrita a  $H$  saltos. O segundo componente é uma busca local baseada em VND (Variable Neighborhood Descent) ([Mladenović & Hansen \[1997\]](#)), que tem como finalidade diminuir o comprimento das rotas encontradas caso exceda  $D_{max}$ . Finalmente, o terceiro componente é uma busca local cujo objetivo é diminuir o custo da floresta de coleta de dados.

Cabe recordar que para modelar o PFMR, empregamos o grafo completo  $D = (V, A)$ , no qual  $V$  representa o conjunto de nós sensores da RSFF,  $A$  representa os arcos que podem ser empregados nas trajetórias dos nós sorvedouros, e o subconjunto  $\hat{A} \subseteq A$  define o conjunto de arestas que podem ser utilizadas para construir a floresta de coleta de dados.

Antes de apresentar os algoritmos que compõem o HRFI-K, explicamos a seguir como a floresta restrita a  $H$  saltos é construída e representada. Essa informação é importante pois, sendo o HRFI-K uma heurística que trata as rotas e a floresta de forma integrada, a estrutura da floresta frequentemente estará presente nos vários procedimentos do HRFI-K.

#### 4.1.1 Representação da floresta restrita a $H$ saltos

Neste trabalho, utilizamos a representação de níveis de vértices proposta por [Gouveia et al. \[2007\]](#), a qual é explicada detalhadamente nesta seção.

Uma floresta restrita a  $H$  saltos pode ser representada pela atribuição de um número entre  $0, 1, \dots, H$  a cada nó em  $V$  da seguinte forma. O número atribuído a cada vértice  $j \in V_i \setminus \{r_i^k\}$  é chamado de *nível* do nó, e indica o número de arcos no caminho entre  $j$  e  $r_i^k$ . Por definição, o nível dos líderes de grupo  $\{r_1^k, \dots, r_{i(k)}^k, k \in \mathcal{K}\}$  é sempre 0. Dessa forma, uma determinada atribuição de níveis implica em uma floresta restrita a  $H$  saltos de acordo com a seguinte observação ([Gouveia et al. \[2007\]](#)):

**Observação 1** *Se, para cada nó em  $V$ , sabemos o seu nível, então a floresta mais barata restrita a  $H$  saltos que satisfaz tal atribuição de níveis é unicamente determinada pela ligação de menor custo entre cada nó de nível  $l$  e um dos nós com nível igual a  $l - 1$ .*

A representação de uma floresta restrita a  $H$  saltos dada pela Observação 1 sugere uma vizinhança para procedimentos de busca local na qual os níveis dos vértices de  $V$  são perturbados. Para exemplificar, seja  $B$  um subconjunto de  $V$ . Em seguida, mude os níveis dos nós em  $B$ . Como observado anteriormente, uma nova solução pode ser obtida facilmente dessa nova atribuição de níveis. Entretanto, não é claro como se pode escolher o subconjunto  $B$  e os novos níveis de seus nós, de forma a satisfazer a restrição de saltos.

Além do mais, essa atribuição de níveis é muito restritiva. Por exemplo, considere a situação em que se escolhe apenas um nó  $i \in V$  para fazer a mudança de níveis. O que pode acontecer é acharmos soluções inviáveis e de baixa qualidade explorando tal

vizinhança, uma vez que seria natural que esse movimento alterasse os níveis dos nós que fazem parte da sub-árvore enraizada em  $i$ , por exemplo.

Para tal definição de níveis, os problemas apontados podem ser parcialmente superados através de uma definição relaxada dos níveis dos nós. Cada nó em  $V$  recebe um rótulo, o qual define o nível máximo que um nó pode estar, na árvore à qual pertence. A observação a seguir descreve como uma floresta restrita a  $H$  saltos pode ser obtida por meio de um conjunto de rótulos que satisfaz esta definição relaxada (Gouveia et al. [2007]).

**Observação 2** *Dados os rótulos que definem o nível máximo de cada vértice em  $V$ , a floresta mais barata restrita a  $H$  saltos que satisfaz tal atribuição de níveis é unicamente determinada pela ligação de menor custo entre cada nó com rótulo igual a  $l$  e um dos nós com rótulos iguais a  $0, 1, \dots, l - 1$ ,  $1 \leq l \leq H$ .*

Essa definição relaxada de níveis permite mais movimentos na vizinhança considerada. A mudança do rótulo de um vértice não-folha  $j$ , por exemplo, implicitamente rearranja a sub-árvore enraizada em  $j$  sem necessariamente alterar todos os valores dos rótulos dos vértices enraizados em  $j$ , se a floresta for construída da forma descrita na Observação 2. Dada a nova atribuição de rótulos, caso seja possível conectar todos os vértices a uma das árvores, a nova solução é viável. Note que, nesses casos, a restrição de saltos sempre é satisfeita.

A Figura 4.1a contém um exemplo de uma árvore restrita a  $H = 3$  saltos construída conforme a atribuição relaxada de rótulos dos vértices pertencentes a ela. Já a Figura 4.1b mostra um exemplo de ligação não permitida entre os vértices, pois viola a restrição de saltos. Entretanto, note que se um nó  $j$  de rótulo 3 se ligasse a um nó de nível 1, outro nó de rótulo 3 poderia se ligar a  $j$ , caso esse arco fosse de menor custo, e a restrição de saltos ainda seria satisfeita. Por esse motivo, a Observação 2 não necessariamente gera a floresta restrita a  $H$  saltos mais barata. Entretanto, construir a floresta segundo a Observação 2 facilita a implementação das buscas locais.

No HRFI-K, a floresta restrita a  $H$  saltos, construída segundo a Observação 2, é representada através de um vetor de números inteiros de tamanho  $n = |V|$ , no qual o valor do número em cada posição  $i \in V$  corresponde ao vértice pai de  $i$ , ou seja, aquele para o qual  $i$  enviará seus dados sensorizados durante a coleta de dados pelo nó sorvedouro. No caso dos líderes de grupo  $\{r_1^k, \dots, r_t^k\}$  esse valor é nulo, pois tais vértices não enviam dados a nenhum outro nó sensor. Como é necessário verificar, para cada vértice  $v \in V \setminus R$ , qual das suas arestas de saída tem menor custo, dada

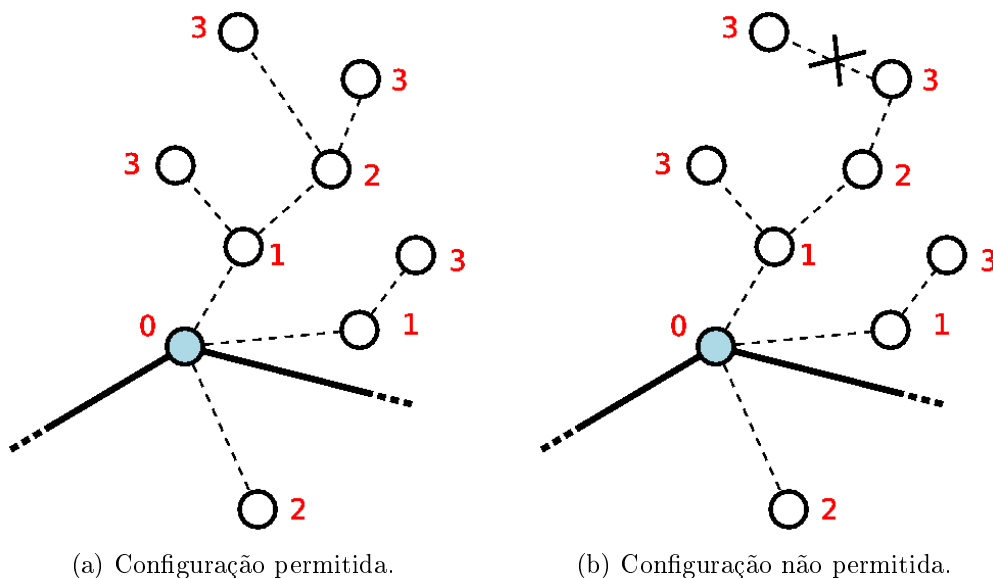


Figura 4.1: Ilustração da construção de uma árvore de acordo com uma atribuição de rótulos dos vértices alcançados por sua raiz.

uma atribuição de rótulos, o procedimento de construção de uma floresta tem ordem de complexidade  $O(n^2)$ .

Conhecidas a construção e a representação da floresta restrita a  $H$  saltos, explicamos a seguir os algoritmos que compõem o HRFI-K.

#### 4.1.2 Fase construtiva

O objetivo da fase construtiva é encontrar um conjunto  $R^k = \{r_1^k, \dots, r_t^k\}$  de líderes de grupo para ser visitado por cada nó sorvedouro  $k \in \mathcal{K}$ . A ordem dos vértices listados em  $R^k$  define, em  $A$ , a trajetória  $\{(r_1^k, r_2^k), \dots, (r_{t(k)-1}^k, r_{t(k)}^k), (r_{t(k)}, r_1^k)\}$  do nó sorvedouro  $k$ . Lembre-se que  $D = (V, A)$ ,  $V = \{1, \dots, n\}$  é um grafo completo. Lembre-se também que todos os vértices  $\{r_1^k = 1 : k \in \mathcal{K}\}$  representam o vértice de partida/chegada das trajetórias dos nós sorvedouros.

No início da fase construtiva, fazemos  $R^k = \{r_1^k\}$ ,  $\forall k \in \mathcal{K}$ . Em seguida, outros vértices são iterativamente acrescentados a esses conjuntos, de forma gulosa, segundo uma política de inserção que será explicada mais adiante. Cada vértice  $v \in V$  inserido na rota do sorvedouro  $k$  é um líder de grupo, ou seja, é a raiz  $r_i^k$  de uma árvore  $T_i^k$ , restrita a  $H$  saltos. A inserção de vértices nas rotas prossegue até que todos os nós de  $V$  sejam alcançáveis a partir de algum líder de grupo usando no máximo  $H$  arcos de  $A$ . Uma característica importante do algoritmo construtivo que elaboramos é que,

na medida em que novos líderes de grupo são inseridos na trajetória, uma floresta contendo árvores restritas a  $H$  saltos é implicitamente construída.

O procedimento de adição de nós ao conjunto de líderes de grupo é baseado no AVMP - Algoritmo de Inserção do Vizinheiro Mais Próximo (Bryant A. Julstrom [1999]), originalmente proposto para o Problema do Caixeiro Viajante. Para o caso do PCV, a idéia principal do AVMP é, dado um conjunto de vértices  $V_C$ , construir iterativamente uma rota envolvendo  $q$  vértices a partir de uma rota anterior que continha  $q-1$  vértices, até que todos os  $|V_C|$  vértices tenham sido adicionados à rota. Para o PCV, o AVMP funciona da seguinte forma. Sejam  $S$  e  $\bar{S}$  o conjunto de nós contidos na rota atual e em seu complemento, respectivamente. Assuma que  $p$  é visitado logo após  $i$  na rota atual. Para todo  $j \in \bar{S}$ , seja  $\Delta_{ip}^j := d_{ij} + d_{jp} - d_{ip}$  o custo para inserir  $j \in \bar{S}$  entre  $i$  e  $p$ . Da mesma forma, seja  $\Delta^j := \min\{\Delta_{ip}^j : i, p \in S, p \text{ é visitado logo após } i\}$  o menor incremento do custo ao incluir  $j$  em  $S$ , considerando todas as posições possíveis para sua inserção. Suponha que o vértice a ser inserido na rota seja  $z \in \arg \min\{\Delta^j : j \in \bar{S}\}$ . O algoritmo então insere  $z$  na posição  $ip$  para a qual estava associada o mínimo  $\Delta^z$ , e remove  $z$  de  $\bar{S}$ . Esse procedimento continua até que  $\bar{S} = \emptyset$ .

O AVMP foi adaptado para encontrar rotas candidatas para os  $K$  sorvedouros, para o uso do problema aqui estudado. Para entender como isso é feito, seja  $\omega(j)$  o conjunto de nós que podem ser alcançados a partir do vértice  $j$ , utilizando no máximo  $H$  saltos. Redefinimos  $S$  como sendo o conjunto de vértices que pertencem a alguma árvore  $T_i^k$  determinada até aquele momento, ou seja,  $S = \bigcup_{k \in \mathcal{K}} (R^k \cup \bigcup_{i \in R^k} \omega(i))$ . O conjunto  $\bar{S}$  é o complemento de  $S$  em  $V$ , isto é,  $\bar{S} = V \setminus S$ . O vértice de  $\bar{S}$  que será inserido na rota  $R^k$  (assim como a posição da inserção) é escolhido através de uma política de inserção gulosa, adaptada a partir daquela empregada pelo AVMP para o PCV. Uma vez que um nó  $j \in \bar{S}$  é selecionado para fazer parte da rota, o nó  $j$  e os nós  $v$  em  $\omega(j) \cap \bar{S}$  são adicionados ao conjunto  $S$  e removidos do conjunto  $\bar{S}$ . Adicionalmente, os nós  $v$  em  $\{\omega(j) \cap \bar{S}\}$  recebem um rótulo que define seu *nível* máximo na árvore enraizada em  $j$ . A inserção de vértices nas rotas dos sorvedouros acontece de forma alternada, ou seja, primeiramente é inserido um vértice em  $R^1$ , em seguida em  $R^2$  (caso  $K > 1$ ), e assim sucessivamente. Após a inserção de um vértice em cada um dos conjuntos  $R^k$ ,  $k \in \mathcal{K}$ , o processo descrito reinicia em  $R^1$ . Não nos preocupamos em fazer inserção de vértices na rota de menor comprimento pois verificamos que, para valores muito baixos de  $D_{max}$  (ou seja, valores para os quais é difícil de achar uma rota de comprimento viável), a fase construtiva raramente consegue obter soluções inicialmente viáveis.

É importante observar, portanto, que uma floresta  $F$  restrita a  $H$  saltos é obtida,

durante a fase construtiva, de forma progressiva através da inserção de um líder de grupo  $r_i^k$  na rota de algum dos sorvedouros em  $\mathcal{K}$ . Quando tal líder de grupo é inserido na rota de  $k$ , cada vértice  $v$  em  $\omega(r_i^k) \cap \bar{S}$  recebe um rótulo, que corresponde, nesse momento, ao número de arcos em  $A$  existente entre  $v$  e  $r_i^k$ . Note que os vértices em  $\omega(r_i^k) \cap S$  não recebem rótulos. Como os vértices em  $S$ , nesse ponto do algoritmo, já fazem parte de alguma árvore restrita a  $H$  saltos, tais vértices já receberam seus respectivos rótulos em uma iteração anterior.

Para o problema tratado, a regra de inserção do vizinho mais próximo do AVMP pode não ser a melhor política de inserção. No momento de decidir qual vértice incluir na rota do sorvedouro  $k$ , dois fatores devem ser balanceados: o custo da expansão da rota e o número de vértices que continuarão descobertos após a expansão. Portanto, a política de inserção utilizada na fase construtiva do HRFI-K não pode ser a regra do menor incremento. Ao invés disso, na nossa abordagem, o vértice  $p$  que expande a rota é dado por  $p \in \arg \min \{ \Delta^j + \lambda |\omega(j)| : j \in \bar{S} \}$ , onde  $\lambda$  é um parâmetro de implementação. Verificamos que o fator  $\lambda$  depende das propriedades geométricas da área que contém os nós sensores. Nas nossas instâncias de teste, um determinado número de nós sensores são aleatoriamente distribuídos em uma área quadrada de lado  $L$ . Diferentes valores de  $\lambda$  foram testados e, após alguns experimentos, definimos que  $\lambda = -0.075L^2$ . A fase construtiva termina quando todos os nós  $j \in V$  pertencem a  $S$ , i.e., todos os nós estão cobertos.

O pseudo-código da fase construtiva encontra-se no Algoritmo 4.1.1.

---

**Algoritmo 4.1.1** Heurística construtiva do HRFI-K( $H, V$ )

---

**Entrada:** número máximo de saltos  $H$ , conjunto de nós sensores  $V$

**Saída:** solução inicial com  $K$  rotas

- 1:  $S \leftarrow \{1\} \cup \omega(1)$
  - 2:  $\bar{S} \leftarrow V \setminus S$
  - 3: **para todo**  $1 \leq k \leq K$  **faça**
  - 4:    $R^k[1] \leftarrow 1$    /\*  $R^k[]$  representa o vetor associado ao conjunto  $R^k$ . \*/
  - 5: **fim para**
  - 6:  $k \leftarrow 1$
  - 7: **enquanto**  $\bar{S} \neq \emptyset$  **faça**
  - 8:   calcule  $\omega(i)$ ,  $\forall i \in \bar{S}$
  - 9:   encontre  $j$  tal que  $j \in \arg \min \{ \Delta_{ab}^q + \lambda |\omega(q)| : q \in \bar{S}, [a, b] \in A, \forall a, b \in R^k \}$
  - 10:    $R^k[a] \leftarrow j$    /\* insere  $j$  na rota do sorvedouro  $k$ , na posição  $a$  \*/
  - 11:    $S \leftarrow \{j\} \cup \omega(j)$
  - 12:    $\bar{S} \leftarrow \bar{S} \setminus (\{j\} \cup \omega(j))$
  - 13:    $k \leftarrow (k \bmod(K)) + 1$
  - 14: **fim enquanto**
-

Ao final da fase construtiva, temos uma solução candidata para o PFMR que atende à restrição de saltos, e cobre todos os vértices. Entretanto, como pode ser observado, a floresta encontrada pode não ser viável, uma vez que uma ou mais trajetórias definidas para os nós sorvedouros podem ter comprimento maior do que  $D_{max}$ . Assim sendo, torna-se necessário um procedimento que visa diminuir o comprimento das rotas dos sorvedouros, apresentado na seção a seguir. Também nessa sessão, apresentamos a Busca Local utilizada para diminuir o custo da floresta restrita a  $H$  saltos.

### 4.1.3 Buscas Locais

#### 4.1.3.1 Buscas Locais para reduzir o comprimento das rotas dos sorvedouros

Ao final da fase construtiva, uma Busca Local baseada em 2-opt é executada. A Busca 2-opt foi proposta por Croes [1958] para resolver o PCV, e funciona da seguinte maneira: duas soluções  $s$  e  $s'$  são vizinhas se, e somente se,  $s'$  pode ser obtida de  $s$  ao se remover 2 arcos a reconectá-los de modo a obter um outro caminho completo. Esse movimento é chamado de *troca de 2 elementos*, o qual é ilustrado na Figura 4.2. Repare que, para uma *troca de 2 elementos*, um dos dois caminhos parciais é invertido após o religamento dos arcos selecionados. O movimento tem complexidade  $O(n)$ .

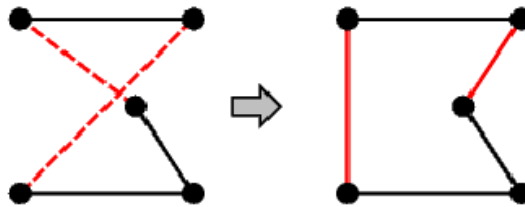


Figura 4.2: Exemplo de uma *troca de 2 elementos* na Busca 2-opt.

A Busca 2-opt pode ser facilmente adaptada ao nosso problema. Considerando cada conjunto de vértices  $R^k$ ,  $k \in \mathcal{K}$  de forma isolada, e tratando cada líder de grupo  $r_i^k \in R^k$ ,  $i = 1, \dots, t(k)$  como um vértice a ser visitado no PCV, a busca 2-opt pode ser aplicada ao conjunto  $R^k$  da mesma forma como é feito no PCV. Note que ao aplicarmos essa busca, preserva-se o conjunto de líderes de grupo da solução atual. Por essa razão, não é necessário verificar a cobertura da rede. Como esse movimento não insere ou remove nenhum vértice dos conjuntos  $R^k$ , a floresta de coleta de dados é mantida intacta e, portanto, a cobertura é mantida.

Após a execução da Busca 2-opt, é verificado, para a trajetória de cada sorvedouro  $k \in \mathcal{K}$ , se o comprimento do circuito Hamiltoniano envolvendo os vértices em

$R^k$  ultrapassa  $D_{max}$ . Caso não ultrapasse, o algoritmo AVMP puro, descrito anteriormente, é aplicado para adicionar mais vértices à rota do sorvedouro  $k$ , até que o caminho resultante não exceda  $D_{max}$ . Quanto mais líderes de grupo houver na RSSF, mais barata ficará a árvore.

Caso o comprimento de pelo menos uma das rota dos sorvedouros exceda  $D_{max}$ , um procedimento baseado em VND é executado. O objetivo de tal procedimento é viabilizar a solução através da tentativa de diminuir os comprimentos das rotas até o valor  $D_{max}$ . São utilizadas as cinco vizinhanças  $N_q$ ,  $q = \{1, 2, 3, 4, 5\}$  enumeradas a seguir. Com exceção das buscas sobre  $N_1$  e  $N_5$ , as demais Buscas são realizadas sobre a rota de maior comprimento.

$q=1$ : **2-opt** - executada como explicado anteriormente, exceto na primeira iteração, uma vez que ela já foi efetuada antes do início do VND.

$q=2$ : **DROP** - é verificado se é possível que algum líder de grupo  $j = r_i^k \neq 1$  seja removido da rota do sorvedouro  $k \in \mathcal{K}$ , o qual possui a rota de maior comprimento. A solução resultante deve manter a cobertura da rede. A verificação da cobertura é feita em conjunto com a reconstrução da árvore, assim como a atribuição do novo rótulo para o vértice  $j$ . Considerando todos os vértices que  $j$  alcança, ele recebe o menor rótulo possível, obtido de forma gulosa. O mesmo acontece com os vértices que antes pertenciam à sua árvore. Esse procedimento sempre começa a partir dos vértices que alcançam algum líder de grupo, e possui ordem de complexidade  $O(n^2)$ . Os nós candidatos à remoção são ordenados por ordem decrescente de decremento do comprimento da rota de  $k$ , caso sejam removidos da mesma. O vértice 1 não pode ser removido, uma vez que é, por definição, o vértice inicial de todas as rotas dos sorvedouros. Note que, para instâncias Euclidianas, a remoção de um líder de grupo de  $R^k$  sempre resulta em uma rota de comprimento igual ou menor do que o da rota anterior, uma vez que haverá menos líderes de grupo para visitar. A vizinhança DROP tem complexidade  $O(n)$ .

$q=3$ : **SWAP** - consiste em aleatoriamente trocar um líder de grupo que faz parte da rota do sorvedouro  $k$ , que possui maior comprimento, por um de seus descendentes na árvore. O nó descendente é colocado na mesma posição em que antes estava o líder de grupo, enquanto que esse último recebe, de forma gulosa, um rótulo de valor entre  $1, \dots, H$ . Essa atribuição é feita em conjunto com a verificação de cobertura, ou seja, tem ordem  $O(n^2)$ . Tal movimento só é considerado válido se todos os vértices forem cobertos, e o comprimento da rota resultante for menor do que o comprimento da solução anterior, **ou** menor ou igual a  $D_{max}$ . Em uma



execução da busca **SWAP** são feitas  $10 * |R^k|$  tentativas de troca. Essa vizinhança tem complexidade  $O(n^3)$ .

$q=4$ : **ADD** - essa vizinhança consiste em tentativas de adicionar um nó aleatório da floresta de coleta de dados à rota de maior comprimento, desde que o novo comprimento dessa rota seja menor ou igual a  $D_{max}$ . É necessário reconstruir a árvore caso essa condição seja satisfeita. São feitas  $\hat{n}/2$  tentativas, em que  $\hat{n}$  é o número total de nós da rota de maior comprimento. Logo, essa vizinhança tem ordem de complexidade  $O(n^2\hat{n})$ .

$q=5$ : **ROUTE-SWAP** - a busca sobre essa vizinhança é baseada em um operador utilizado em Problemas de Roteamento de Veículos (PRV) (de Oliveira et al. [2007]). Esse operador é aplicado às diversas rotas dos veículos, permitindo fugir de alguns mínimos locais através de perturbações. Portanto, como esse operador permite a realização de buscas em mais de uma rota, ele é aplicado apenas quando  $K > 1$ . O **ROUTE-SWAP** funciona da seguinte maneira. Um número aleatório (entre 1 e  $|R^1|/2$ ) de líderes de grupo é removido da rota de número 1, em que o número  $|R^1|/2$  é o resultado da divisão arredondado para baixo. O mesmo é feito para as demais  $K - 1$  rotas do problema. Em seguida, um procedimento similar à heurística construtiva apresentada na seção 4.1.2 é realizado, o qual reconstrói todas as rotas de forma gulosa, com o objetivo de readquirir cobertura, e obter rotas com comprimento possivelmente não superior a  $D_{max}$ . Lembre-se que o nó inicial 1 não pode ser removido de nenhuma das rotas. O **ROUTE-SWAP** não garante uma solução que obedece à restrição de comprimento imposta por  $D_{max}$ .

O VND é executado três vezes, durante as quais a melhor solução encontrada é guardada. Definimos aqui que a melhor solução é aquela cuja rota de maior comprimento é mais próxima a  $D_{max}$ . Isso é feito de forma a permitir que o procedimento **ROUTE-SWAP** encontre soluções inviáveis, fugindo de mínimos locais ao fazer perturbações na solução corrente. Assim, se ao final de cada execução do VND uma solução viável não foi encontrada, essa melhor solução é utilizada para a execução do próximo VND. Caso contrário, não é necessário executar o VND novamente.

Se o comprimento de alguma das  $K$  rotas ainda exceder  $D_{max}$  após a terceira execução do VND, a heurística termina e o problema é considerado inviável. Caso contrário, é executada uma Busca Local, chamada **MIN-FOREST**, cujo objetivo é reduzir o custo da floresta de comunicação.

#### 4.1.3.2 Busca Local para reduzir o custo da floresta de comunicação

Uma floresta do PFMR restrita a  $H$  saltos pode ser vista como uma árvore do PAGMS restrita a  $H + 1$  saltos, bastando para isso conectar os vértices líderes de grupo a um vértice artificial 0, conforme ilustrado na Figura 4.3. Dessa forma, apresentamos aqui o procedimento MIN-FOREST, que consiste em uma busca sobre a vizinhança SHIFT-SWAP (proposta por Gouveia et al. [2007] para o PAGMS) adaptada para o PFMR.

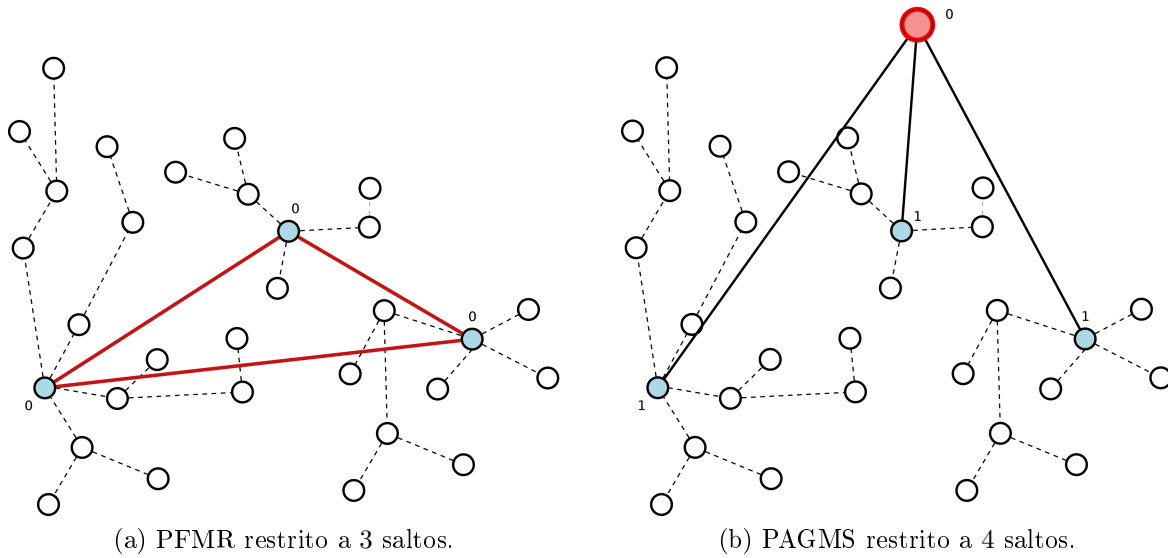


Figura 4.3: Árvore do PAGMS "equivalente" a uma floresta do PFMR com  $K = 1$ .

A vizinhança SHIFT-SWAP de Gouveia et al. [2007] consiste em investigar todas as soluções que se diferem da solução atual através de um movimento SHIFT ou de um movimento SWAP. Na nossa abordagem, um movimento SHIFT consiste na troca do valor do rótulo de um vértice  $j \in V \setminus \bigcup_{k \in \mathcal{K}} R^k$ , escolhido aleatoriamente. O novo valor do rótulo de  $j$  também é escolhido aleatoriamente, e deve ser diferente de 0. Já um movimento SWAP consiste em escolher aleatoriamente dois vértices  $i, j \in V$ , e trocar o valores dos seus respectivos rótulos:  $i$  recebe o valor do rótulo de  $j$ , e similarmente  $j$  recebe o valor do rótulo de  $i$ . A vizinhança SHIFT contém  $O(nH)$  vizinhos, enquanto que a vizinhança SWAP contém  $O(n^2)$  vizinhos.

Se o movimento SWAP envolver um líder de grupo  $i \in \bigcup_{k \in \mathcal{K}} R^k$  e um nó  $j$  que não é líder de grupo, a troca ocorre da seguinte maneira. O vértice  $i$  passará a ter o valor do rótulo de  $j$ , passando então a fazer parte dos níveis inferiores de uma das árvores do PFMR. Já o vértice  $j$  passa a ser um líder de grupo, inserido na mesma rota e na mesma posição que antes eram ocupadas pelo vértice  $i$ .

Os movimentos SHIFT e SWAP só são permitidos se a nova solução for viável, e

se o custo total da nova floresta restrita a  $H$  saltos for menor do que o custo total da floresta anterior. A cada troca de nós, é necessária a reconstrução da árvore.

Como uma busca na vizinhança *SHIFT* é mais barata, primeiramente são feitos vários movimentos *SHIFT* até que não se consiga mais melhorar a solução após  $n$  trocas. Então, tenta-se melhorar a solução através de um movimento *SWAP*, para o qual são feitas 100 tentativas. Se for possível melhorar a solução, a sequência de movimentos *SHIFT* seguidos de um *SWAP* reinicia. Senão, o procedimento *MIN-FOREST* termina. No trabalho de [Gouveia et al. \[2007\]](#) é mostrado que essa troca sistemática na busca das vizinhanças *SHIFT* e *SWAP* é bastante eficaz para fugir de mínimos locais, quando comparada com as buscas feitas apenas na vizinhança *SHIFT* ou apenas na vizinhança *SWAP*. O *HRFI-K* termina com o fim da execução do procedimento *MIN-FOREST*.

O Algoritmo 4.1.2 apresenta o pseudo-código da heurística *HRFI-K*, contemplando todas as suas fases.

## 4.2 Experimentos computacionais

Nessa seção, apresentamos a comparação dos resultados computacionais obtidos com um algoritmo *Branch-and-bound* (BB) baseado na formulação (3.2)-(3.22) e com o *HRFI-K*. A formulação de Programação Inteira Mista aqui proposta mostrou-se muito pesada mesmo para  $K = 1$  nós sorvedouros. Para números maiores de sorvedouros na rede, a formulação fica ainda mais pesada. Por esse motivo, apresentamos aqui resultados considerando um nó sorvedouro apenas.

### 4.2.1 Instâncias testes

Para a realização dos nossos experimentos computacionais, foram utilizadas instâncias Euclidianas, nas quais os vértices em  $V$  foram aleatoriamente distribuídos em uma área quadrada plana de lado  $L = 100m$ . O valor de  $D_{max}$  foi fixado como sendo 2,5 vezes o valor de  $L$ .

Utilizamos duas classes de instâncias testes, EB e EC, as quais foram definidas de modo similar às instâncias utilizados na literatura do *PAGMS*. Nas instâncias do tipo EB, o vértice 1 corresponde àquele mais próximo a uma das bordas do quadrado que define a área da *RSSF*. Já para as instâncias do tipo EC, o vértice 1 é aquele mais próximo ao centro do quadrado. Para cada classe EB e EC, foram definidas 5 instâncias com 20, 40, 60 e 80 nós, totalizando 40 instâncias diferentes.

---

**Algoritmo 4.1.2** HRFI-K( $H, V$ )

---

**Entrada:** número máximo de saltos  $H$ , conjunto de nós sensores  $V$ 

```

1: solucao ← heurísticaConstrutiva( $H, V$ )
2: solucao ← busca2opt(solucao)
3: para todo  $k \in \mathcal{K}$  tal que comprimentoRota(solucao,  $k$ ) <  $D_{max}$  faça
4:   AVMP(solucao,  $k$ )
5: fim para
6: se comprimentoAlgumaRota(solucao) >  $D_{max}$  então
7:   /* Inicia o VND. */
8:   numVND ← 1
9:   enquanto numVND ≤ 3 faça
10:     $q \leftarrow 1$ 
11:    enquanto  $q \leq 5$  faça
12:      $s \leftarrow$  buscaLocal(solucao,  $N_q$ )
13:     se  $q = 1$  ou  $s =$  solucao então
14:        $q \leftarrow q + 1$ 
15:     senão
16:        $q \leftarrow 1$ 
17:       solucao ←  $s$ 
18:     fim se
19:   fim enquanto
20: fim enquanto /* Fim do VND. */
21: se comprimentoAlgumaRota(solucao) >  $D_{max}$  então
22:   solucao ← melhorSolucao
23: senão
24:   numVND ← 4
25: fim se
26: numVND ← numVND + 1
27: fim se
28: se comprimentoAlgumaRota(solucao) >  $D_{max}$  então
29:   retorne inviável
30: senão
31:   solucao ← min-forest(solucao)
32: fim se
33: retorne solucao

```

---

Os custos dos arcos  $c_{ij}$ ,  $[i, j] \in A$ , que representam uma estimativa da energia necessária para a transmissão de um pacote, de tamanho fixo, de  $i$  para  $j$ , são definidos em função da distância entre os vértices. Quanto mais distantes estão transmissor e receptor, maior será a potência necessária para que o sinal chegue ao seu destino. Para cada valor de potência de transmissão está associado um consumo de corrente elétrica, cujo valor varia de acordo com o *hardware* do nó sensor considerado. Esse consumo de corrente elétrica e seu tempo de duração definem a energia gasta com transmissão, dada por  $I\Delta t$ . Nessa fórmula,  $I$  corresponde à corrente elétrica consumida, e  $\Delta t$  ao intervalo de tempo durante o qual a transmissão ocorreu. Como estamos considerando o custo para transmitir um pacote de tamanho fixo, temos que o valor de  $\Delta t$  é o mesmo para todos os arcos em  $A$ . Assim, podemos definir os custos  $c_{ij}$  simplesmente como o valor da corrente elétrica consumida.

Dessa forma, a Tabela 4.1 apresenta os valores desses custos, presentes na coluna *Corrente*. Para um determinado enlace  $[i, j]$ , o custo  $c_{ij}$  corresponde à corrente associada à potência cujo alcance seja imediatamente superior ao da distância  $d_{ij}$ , quando essa não for encontrada na Tabela. Esses valores são baseados no hardware do nó sensor MICA2 [2009], e foram calculados usando um modelo de propagação em espaço livre.

Tabela 4.1: Consumo de corrente do nó sensor MICA2 com transmissão de mensagens.

<i>Nível de potência (dBm)</i>	<i>Alcance (m)</i>	<i>Corrente (mA)</i>
-20	8,435	5,3
-19	9,464	6,9
-18	10,619	7,0
-17	11,915	7,1
-16	13,369	7,2
-15	15,000	7,4
-14	16,830	7,4
-13	18,884	7,5
-12	21,188	7,6
-11	23,773	7,7
-10	26,674	7,9
-9	29,999	7,9
-8	33,581	8,2
-7	37,678	8,4
-6	42,276	8,7

## 4.2.2 Comparação entre os resultados do HRFI-1 e de um algoritmo Branch-and-bound

O PFMR foi resolvido através de dois algoritmos, *Branch-and-bound* (BB) e HRFI-1. O algoritmo BB é baseado na formulação (3.2)-(3.22), e foi implementado utilizando o pacote CPLEX versão 10.2 (com configurações padrão) através do [ILOG Concert Technology \[2009\]](#), em C++. A heurística HRFI-1 também foi implementada em C++. O tempo de CPU foi limitado a 4 horas para o algoritmo BB. Todos os experimentos foram realizados em uma máquina Intel 2,5GHz, com 4GB de memória.

O *gap* de dualidade obtido pelo algoritmo BB, para cada instância de teste, é recuperado. Esse *gap* corresponde à diferença percentual entre a melhor solução encontrada (melhor limite superior  $\bar{l}^*$ ) e o melhor limite inferior  $l_{best}$  obtido pelo BB após 4 horas. O *gap* corresponde então a uma *estimativa* de o quão longe o custo da solução encontrada está do custo da solução ótima.

Na Tabela 4.2, são apresentados os resultados computacionais associados ao HRFI-1 e ao algoritmo BB. A primeira coluna apresenta o número máximo  $H$  de saltos permitido, a segunda consiste no número de vértices de cada instância, e a terceira representa o identificador da instância. Da quarta para a sétima coluna da tabela, são apresentados os resultados para as instâncias do tipo EB: o tempo  $t_{BB}$  de execução do algoritmo BB, em segundos; o tempo total de execução  $t_{heur}$  da heurística HRFI, em segundos; o *gap* de dualidade obtido ao término do algoritmo BB,  $\frac{\bar{l}^* - l_{best}}{l_{best}}$ ; e, finalmente, na sexta coluna, a razão entre o valor do melhor limite superior encontrado ( $\bar{l}^*$ ) e o valor encontrado pela heurística ( $\bar{l}$ ). Resultados similares são apresentados para as instâncias do tipo EC, nas demais colunas da tabela.

Nessa Tabela, as entradas “-” significam que o algoritmo BB não foi capaz de encontrar uma solução viável para o PFMR, dentro do limite de tempo de 4 horas, para a instância associada àquela linha na Tabela.

Os resultados obtidos indicam que é difícil resolver o PFMR através de um algoritmo BB baseado na nossa formulação de Programação Inteira Mista, principalmente para menores valores de  $H$ . Vemos, por exemplo, que, em geral, quanto maior o valor de  $H$ , menor o tempo necessário  $t_{BB}$  para obter a solução do BB, e menor o *gap* de dualidade obtido. Esse fato sugere que o problema é mais fácil para valores maiores de  $H$ .

Os resultados mostram também que as instâncias do tipo EC são mais fáceis de se resolver, tanto para o BB quanto para o HRFI-1. A maior prova disso é o fato de que, para instâncias de 60 nós, o BB não conseguiu achar uma solução viável para

nenhuma do tipo EB e  $H = 2$ . Entretanto, o BB foi capaz de achar um maior número de soluções viáveis para valores de  $H = 3, 4$ , principalmente para as instâncias do tipo EC. Repare também que, na média, o HRFI-1 foi capaz de encontrar melhores soluções (inclusive várias delas ótimas) para instâncias do tipo EC, quando comparadas com as instâncias EB.

Tabela 4.2: Resultados computacionais associados ao HRFI-1 e ao BB.

$H$	$n$	inst	EB				EC			
			$t_{BB}(s)$	$t_{heur}(s)$	$\frac{\bar{l}^* - l_{best}}{l_{best}}$	$\frac{\bar{l}^*}{\bar{l}}$	$t_{BB}(s)$	$t_{heur}(s)$	$\frac{\bar{l}^* - l_{best}}{l_{best}}$	$\frac{\bar{l}^*}{\bar{l}}$
2	20	0	590,66	1,18	OPT	1,0000	45,60	1,18	OPT	1,0000
		1	17,42	1,43	OPT	0,9475	16,04	2,17	OPT	1,0000
		2	47,97	1,87	OPT	0,9617	6,62	2,11	OPT	1,0000
		3	187,62	2,61	OPT	0,8917	25,54	1,74	OPT	1,0000
		4	7,19	1,17	OPT	1,0000	39,47	1,29	OPT	1,0000
	40	0	14400,00	4,08	-	-	14400,00	7,27	2,29	1,0000
		1	14400,00	6,98	16,58	1,0619	14400,00	5,35	1,79	0,9282
		2	14400,00	4,71	10,65	0,9469	14400,00	8,95	5,59	0,9957
		3	14400,00	4,26	40,32	1,2693	14400,00	5,38	4,02	0,9469
		4	14400,00	9,31	36,58	1,3040	14400,00	3,89	29,26	1,0198
	60	0	14400,00	7,50	-	-	14400,00	7,76	-	-
		1	14400,00	7,81	-	-	14400,00	8,11	9,44	1,0031
		2	14400,00	6,28	-	-	14400,00	7,29	-	-
		3	14400,00	7,67	-	-	14400,00	8,10	-	-
		4	14400,00	7,43	-	-	14400,00	8,05	7,93	0,9624
	80	0	14400,00	11,10	-	-	14400,00	13,42	-	-
		1	14400,00	11,32	-	-	14400,00	10,66	-	-
		2	14400,00	11,92	-	-	14400,00	10,27	-	-
		3	14400,00	10,99	-	-	14400,00	11,41	-	-
		4	14400,00	11,40	-	-	14400,00	12,21	-	-
3	20	0	151,28	1,10	OPT	0,9468	26,66	3,33	OPT	0,9857
		1	25,30	3,73	OPT	0,9475	15,74	2,17	OPT	1,0000
		2	46,70	1,49	OPT	0,9914	6,78	2,19	OPT	1,0000
		3	157,44	2,43	OPT	0,8859	16,43	4,45	OPT	1,0000
		4	2,67	2,12	OPT	0,8978	19,13	2,04	OPT	1,0000
	40	0	14400,00	4,12	31,63	1,1138	4041,64	5,15	OPT	0,7961
		1	4927,85	4,96	OPT	0,9102	14400,00	8,32	7,07	0,9673
		2	14400,00	4,76	-	-	3972,96	4,63	OPT	0,9656
		3	14400,00	4,90	24,71	1,1091	14400,00	4,95	0,66	0,9186
		4	14400,00	5,44	8,46	0,9720	14400,00	7,47	6,43	0,9947
	60	0	14400,00	9,05	-	-	14400,00	9,01	4,95	0,9555
		1	14400,00	9,25	7,82	0,9168	14400,00	8,95	13,56	1,0633
		2	14400,00	8,49	-	-	14400,00	9,23	-	-
		3	14400,00	7,64	-	-	14400,00	5,24	10,08	0,9802
		4	14400,00	7,95	-	-	14400,00	7,96	5,38	0,9109
	80	0	14400,00	11,33	-	-	14400,00	11,40	-	-
		1	14400,00	11,29	-	-	14400,00	13,48	-	-
		2	14400,00	10,90	-	-	14400,00	11,82	-	-
		3	14400,00	13,42	-	-	14400,00	11,11	-	-
		4	14400,00	12,78	-	-	14400,00	11,48	-	-
4	20	0	63,24	2,76	OPT	1,0000	35,60	1,10	OPT	0,9857
		1	27,13	2,54	OPT	0,9475	15,21	2,17	OPT	1,0000
		2	31,70	1,73	OPT	0,9862	13,14	2,15	OPT	1,0000
		3	124,04	3,00	OPT	0,8826	16,97	2,72	OPT	1,0000
		4	1,53	1,09	OPT	0,9240	16,78	2,03	OPT	1,0000
	40	0	14400,00	3,79	16,05	0,9280	14400,00	5,76	2,20	1,0000
		1	9474,49	5,02	OPT	0,9302	4728,62	6,25	OPT	0,9277
		2	14400,00	4,87	1,31	0,9165	3123,68	5,50	OPT	0,9947
		3	14400,00	4,92	-	-	14400,00	7,06	1,07	0,9725
		4	8916,08	8,39	0,01	0,9667	14400,00	4,63	2,33	0,8826
	60	0	14400,00	8,01	7,17	0,9027	14400,00	10,61	2,82	0,8984
		1	14400,00	9,54	9,02	0,9506	14400,00	8,49	7,74	1,0110
		2	14400,00	7,26	-	-	14400,00	12,29	-	-
		3	14400,00	7,56	3,91	0,8367	14400,00	11,07	-	-
		4	14400,00	7,94	8,38	0,9753	14400,00	10,75	-	-
	80	0	14400,00	11,99	-	-	14400,00	13,82	-	-
		1	14400,00	14,00	-	-	14400,00	17,56	-	-
		2	14400,00	14,18	-	-	14400,00	12,62	-	-
		3	14400,00	14,45	-	-	14400,00	11,93	-	-
		4	14400,00	13,96	-	-	14400,00	14,50	-	-



### 4.2.3 Fornecendo ao BB uma solução obtida pelo HRFI-1

Para os experimentos cujos resultados apresentamos nessa seção, utilizamos a solução viável obtida pelo HRFI-1 (sendo que  $\bar{l}$  é o limite superior que dela implica) como uma solução inicial para o algoritmo BB. Isso faz com que o BB inicie com uma solução "aquecida", ou seja, uma solução viável que provavelmente não obteria, no início de sua execução, sem tal artifício. Com isso, esperamos que seja capaz de resolver mais instâncias do PFMR. Entretanto, mesmo utilizando-se dessa abordagem, não foi possível obter, após 4 horas, a solução ótima para nenhuma instância com 60 ou 80 nós, nem para a maior parte das instâncias com 40 nós.

Na Tabela 4.3, são apresentados os resultados para as mesmas instâncias da seção anterior, para  $H \in \{2, 3, 4\}$ . Da quarta para, através dos seus arcos  $A_1^m$  a sétima coluna da tabela, são apresentados os resultados para as instâncias da classe EB, sendo que elas representam, respectivamente: o tempo de CPU necessário para calcular o limite de Relaxação Linear  $l_{LP}$  da formulação apresentada na seção 3.2; a razão entre  $l_{LP}$  e o melhor limite superior  $\bar{l}^*$  obtido após a execução do BB,  $\frac{l_{LP}}{\bar{l}^*}$ ; o *gap* de dualidade obtido depois que o algoritmo BB terminou,  $\frac{\bar{l}^* - l_{best}}{l_{best}}$  (onde  $l_{best}$  é o melhor limite inferior obtido pelo BB após 4 horas); e, finalmente, a razão entre o melhor limite superior encontrado pelo BB e  $\bar{l}$ ,  $\frac{\bar{l}^*}{\bar{l}}$ . Resultados similares são apresentados para as instâncias do tipo EC, nas demais colunas da tabela.

Resultados da tabela 4.3 corroboram o fato de que o problema se torna mais fácil quanto maior o valor de  $H$ . Isso pode ser visto pelo fato de que o tempo necessário para calcular  $l_{LP}$ , e os *gaps* de dualidade, diminuam com o aumento de  $H$ . Ainda em conformidade com os resultados apresentados na seção 4.2.2, os resultados sugerem que instâncias da classe EC são mais fáceis do que as instâncias de classe EB, tanto para o BB (devido aos menores tempos para obtenção do limite de relaxação), quanto para o HRFI-1. Também podemos verificar que a heurística conseguiu achar soluções ótimas para boa parte das instâncias EC com 20 nós.

Repare que, à medida que  $n$  aumenta, uma fração considerável do limite de tempo imposto ao BB foi gasto para calcular  $l_{LP}$ . Assim, é esperado que o BB não conseguisse nem mesmo melhorar a solução encontrada pela heurística para instâncias maiores. Esse fato é evidente ao se observar as entradas iguais a 1,000 para instâncias com 80 nós, nas colunas com cabeçalho  $\frac{\bar{l}^*}{\bar{l}}$ . Esses valores indicam que o BB não conseguiu melhorar a solução fornecida pelo HRFI-1, para nenhuma instância de 80 nós, tanto do tipo EB quanto do tipo EC. Cabe mencionar que o BB não conseguiu nem mesmo calcular  $l_{LP}$  dentro do limite de 4 horas imposto, para várias instâncias de 80 nós.

Tabela 4.3: Resultados computacionais associados ao HRFI-1 e ao BB com solução inicial provida por HRFI-1.

$H$	$n$	inst	EB				EC			
			$t_{LP}(s)$	$\frac{l_{LP}}{\bar{l}^*}$	$\frac{\bar{l}^* - l_{best}}{l_{best}}$	$\frac{\bar{l}^*}{\bar{l}}$	$t_{LP}(s)$	$\frac{l_{LP}}{\bar{l}^*}$	$\frac{\bar{l}^* - l_{best}}{l_{best}}$	$\frac{\bar{l}^*}{\bar{l}}$
2	20	0	1.67	0.5421	OPT	1.0000	3.69	0.6801	OPT	1.0000
		1	1.27	0.7681	OPT	0.9475	0.97	0.8432	OPT	1.0000
		2	1.05	0.7944	OPT	0.9617	0.94	0.8145	OPT	1.0000
		3	0.82	0.5379	OPT	0.8917	1.21	0.8199	OPT	1.0000
	40	4	1.50	0.9722	OPT	1.0000	1.03	0.7185	OPT	1.0000
		0	640.55	0.6602	29.97	0.9971	443.33	0.9048	1.47	1.0000
		1	210.01	0.8257	2.15	1.0000	220.55	0.8441	OPT	0.9282
		2	58.94	0.7616	11.51	0.9657	79.51	0.8934	0.46	0.9928
	60	3	146.36	0.7340	20.20	0.9963	165.67	0.8422	2.15	0.9469
		4	127.53	0.8076	15.23	0.9987	203.33	0.5928	28.86	1.0000
		0	1920.64	1.0000	14.73	0.9904	2537.15	0.8621	11.68	0.9882
		1	1952.68	0.8053	17.46	0.9978	1714.02	0.8823	9.58	0.9973
	80	2	1420.20	0.6988	27.60	0.9993	2036.14	0.8213	15.88	0.9899
		3	5049.99	0.7975	18.39	1.0000	1895.48	0.8721	10.87	0.9910
		4	2069.23	0.7948	17.87	0.9980	2171.90	0.8697	10.39	0.9889
		0	14396.19	-	-	-	5764.72	0.8226	16.61	1.0000
	3	1	12642.49	0.8062	19.38	1.0000	10204.72	0.8294	16.96	1.0000
		2	14398.29	-	-	-	11610.74	0.8095	18.79	1.0000
		3	14394.99	-	-	-	14394.71	-	-	-
		4	14395.22	-	-	-	9468.80	0.8395	15.05	1.0000
3	20	0	0.84	0.5468	OPT	0.9468	1.63	0.6801	OPT	0.9857
		1	1.29	0.7681	OPT	0.9475	1.01	0.8432	OPT	1.0000
		2	1.05	0.8232	OPT	0.9914	0.95	0.8145	OPT	1.0000
		3	0.83	0.5664	OPT	0.8859	1.25	0.8199	OPT	1.0000
	40	4	1.07	0.8807	OPT	0.8978	0.97	0.7790	OPT	1.0000
		0	204.63	0.6765	15.11	0.9482	199.73	0.7282	OPT	0.8961
		1	183.61	0.8069	0.01	0.9482	182.99	0.8629	0.01	0.9450
		2	106.41	0.7900	5.70	0.9102	223.21	0.9130	OPT	0.9656
	60	3	142.11	0.7881	12.78	0.9362	136.92	0.8213	3.37	0.9186
		4	102.22	0.8496	2.24	1.0000	181.21	0.7468	10.08	0.9962
		0	1926.85	0.8911	8.17	0.9665	1869.61	0.8970	7.72	0.9825
		1	913.93	0.8265	14.82	0.9905	2112.58	0.8930	4.29	0.9602
	80	2	1530.81	0.8010	17.22	0.9979	2219.71	0.7946	16.24	0.9602
		3	1533.58	0.8067	3.32	0.8479	769.51	0.8563	6.97	0.9455
		4	2617.80	0.8331	14.31	0.9904	2679.99	0.8467	5.33	0.9105
		0	10091.50	0.8424	15.76	1.0000	6556.56	0.8392	14.92	1.0000
	4	1	12906.37	0.8256	17.44	1.0000	8465.37	0.7837	21.59	1.0000
		2	13328.98	0.8286	17.14	1.0000	6066.29	0.8229	16.69	1.0000
		3	14394.77	-	-	-	11756.75	0.8403	15.97	1.0000
		4	14395.92	-	-	-	6792.69	0.8748	11.60	1.0000
4	20	0	0.81	0.6412	OPT	1.0000	1.60	0.6801	OPT	0.9857
		1	1.23	0.7681	OPT	0.9475	0.97	0.8432	OPT	1.0000
		2	1.01	0.8232	OPT	0.9862	0.95	0.8145	OPT	1.0000
		3	0.81	0.5673	OPT	0.8826	1.24	0.8199	OPT	1.0000
	40	4	1.02	0.9108	OPT	0.9240	0.96	0.7819	OPT	1.0000
		0	220.45	0.6684	14.11	0.9102	160.67	0.9147	OPT	1.0000
		1	163.41	0.8297	OPT	0.9302	276.07	0.8507	OPT	0.9277
		2	105.31	0.7916	0.60	0.9165	203.87	0.9470	OPT	0.9947
	60	3	171.23	0.7998	8.36	0.9986	115.37	0.8709	0.44	0.9725
		4	85.73	0.8795	OPT	0.9667	176.43	0.6379	1.57	0.8806
		0	1280.84	0.8184	4.40	0.8675	2126.77	0.8641	1.35	0.8961
		1	1926.09	0.8395	7.78	0.9252	2354.45	0.9049	6.77	0.9913
	80	2	2029.90	0.7959	18.47	0.9953	2066.72	0.8294	8.09	0.9222
		3	1532.00	0.7903	6.63	0.8580	666.39	0.8465	3.53	0.9006
		4	2277.80	0.8623	11.09	0.9978	2490.97	0.9052	5.17	0.9697
		0	5771.51	0.8642	13.30	1.0000	10360.30	0.8503	14.97	1.0000
	4	1	14394.77	-	-	-	5495.86	0.9221	6.87	1.0000
		2	13067.02	0.7957	20.43	1.0000	10305.87	0.8404	15.62	1.0000
		3	14395.08	-	-	-	7805.74	0.8898	10.89	1.0000
		4	14396.67	-	-	-	6828.57	0.8184	17.21	1.0000

Finalmente, os resultados da Tabela 4.3 nos permitem concluir que a heurística HRFI-1 tem um bom desempenho, quando seus resultados são comparados àqueles obtidos pelo BB dentro de 4 horas. Os valores da coluna  $\frac{\bar{l}^*}{\bar{l}}$  indicam que os melhores resultados obtidos pelo HRFI-1 são, na média, no máximo 9% maiores do que aqueles obtidos pelo BB.

### 4.3 Considerações finais

Neste Capítulo, apresentamos a abordagem heurística HRFI-K proposta para resolver o PFMR, bem como resultados dos experimentos computacionais associados ao HRFI-1 e à formulação de Programação Inteira Mista apresentada na seção 3.2, avaliada através de um algoritmo *Branch-and-bound*.

Os resultados apresentados sugerem que ainda existem muitas oportunidades de pesquisa. Para a abordagem exata do PFMR, podem ser explorados possíveis algoritmos exatos capazes de melhorar os limites inferiores. Também podem ser investigadas reformulações de Programação Inteira que nos permitiriam obter algoritmos BB mais efetivos. Para a abordagem heurística, podem ser utilizados métodos *multi start*, como GRASP (Feo & Resende [1995]).

Apesar de todas essas observações, o HRFI-K é perfeitamente aplicável ao problema apresentado em RSSFs, uma vez que precisamos de um algoritmo rápido que possa ser utilizado em um contexto de RSSFs. Nesse sentido, apresentamos no próximo capítulo o arcabouço de simulação para RSSFs utilizado nesse trabalho, bem como resultados computacionais associados à simulação utilizando várias abordagens, dentre elas o HRFI-K.



# Capítulo 5

## Simulação de RSSFs

A simulação de eventos discretos é um método amplamente utilizado para a avaliação experimental de projetos de pesquisa na área de RSSFs. Por um lado, quantificar analiticamente o desempenho e o comportamento de um sistema complexo é frequentemente impreciso. Por outro lado, a realização dos experimentos no mundo real é uma tarefa onerosa, pois é necessário adquirir centenas de dispositivos, configurar seu *software*, encontrar um local livre de interferências externas para a realização de tais experimentos, possivelmente mudar os dispositivos de lugar, dentre outras dificuldades.

Para a realização da simulação de uma aplicação em RSSFs, é necessária a definição dos modelos computacionais, os quais imitam parcial ou totalmente as propriedades do sistema a ser estudado. Assim sendo, neste capítulo apresentamos o arcabouço de simulação utilizado para a realização dos experimentos computacionais, bem como o detalhamento do cenário de simulação e das abordagens testadas. Os resultados dos experimentos computacionais são apresentados ao final deste capítulo.

### 5.1 Arcabouço de simulação

O arcabouço de simulação utilizado neste trabalho é o [JiST/SWANS \[2007\]](#), uma ferramenta de alto desempenho desenvolvida pela Universidade de Cornell.

O JiST (*Java in Simulation Time*) é um simulador de eventos discretos escrito em Java, composto por três partes: um compilador, um *modificador de bytecode* e um núcleo de simulação. Os programas de simulação são escritos na linguagem Java, e compilados para *bytecode* utilizando um compilador Java. Essas classes compiladas são então reescritas pelo *modificador de bytecode*, para que suportem a semântica de *tempo*

de simulação e sejam executadas sobre o núcleo de simulação. Os três componentes do JiST foram escritos inteiramente em Java. Portanto, o processo inteiro é executado sobre uma Máquina Virtual Java (JVM).

Sobre o JiST foi implementado o SWANS (*Scalable Wireless Ad hoc Network Simulator*), um simulador Java de redes *ad-hoc*. O SWANS possui uma arquitetura inspirada no modelo de camadas OSI, no qual os elementos de uma determinada camada só têm acesso aos módulos das camadas imediatamente superior e inferior. O SWANS provê componentes que implementam protocolos de roteamento e acesso ao meio, bem como modelos de ruído, de recepção e transmissão via rádio, e de propagação de sinal.

O JiST/SWANS é mais escalável do que o ns-2 [2007] e o GloMoSim (Bajaj et al. [1999]), como mostrado por Barr et al. [2005]. O ns-2 é o simulador de rede mais utilizado em trabalhos acadêmicos, enquanto que o GloMoSim é amplamente utilizado como simulador comercial de Redes Sem Fio.

Por ser mais escalável do que as ferramentas mais utilizadas em simulação, uma adaptação do arcabouço JiST/SWANS para simulação de RSSFs foi proposta e desenvolvida por Aioffi [2007]. Assim sendo, esse arcabouço foi escolhido como a ferramenta de apoio à simulação de RSSFs para o trabalho aqui apresentado.

Como a heurística HRFI-K foi implementada em C++, a integração entre a heurística e o simulador Java JiST/SWANS é feita através do CDT (CDT - Eclipse C/C++ Development Tooling [2009]). O CDT permite que um programa em C++ seja executado dentro de um programa Java, de forma que os objetos C++ que representam a solução possam ser instanciados como objetos Java. Assim, é possível fazer a integração entre a heurística e a ferramenta de simulação de forma natural e transparente, sem precisar alterar o código fonte do HRFI-K para que o mesmo seja utilizado na simulação.

Na próxima sessão é apresentado o cenário de simulação. Isso inclui o modelo de energia, a carga de trabalho, o modelo de utilização da rede pelos usuários, modelo de propagação de sinal, dentre outros.

## 5.2 Cenário de simulação

### 5.2.1 Modelo de energia para Redes de Sensores sem Fio

Neste trabalho, os parâmetros para modelagem e simulação foram escolhidos tendo como base o hardware do MICA2 [2009]. Esse nó sensor é parte do Projeto Motes,

desenvolvido por pesquisadores na Universidade de Berkeley ([Berkeley Sensor and Actuator Center \[2009\]](#)). Atualmente, o fabricante XBOW comercializa a linha do Projeto Mica Motes, do qual destacam-se três famílias de nós sensores: MICA2, [MICA2DOT \[2009\]](#) e [MICAz \[2009\]](#).

A Figura 5.1a apresenta alguns nós sensores do Projeto Mica Motes. Já a Figura 5.1b contém o diagrama dos componentes do nó MICA2.

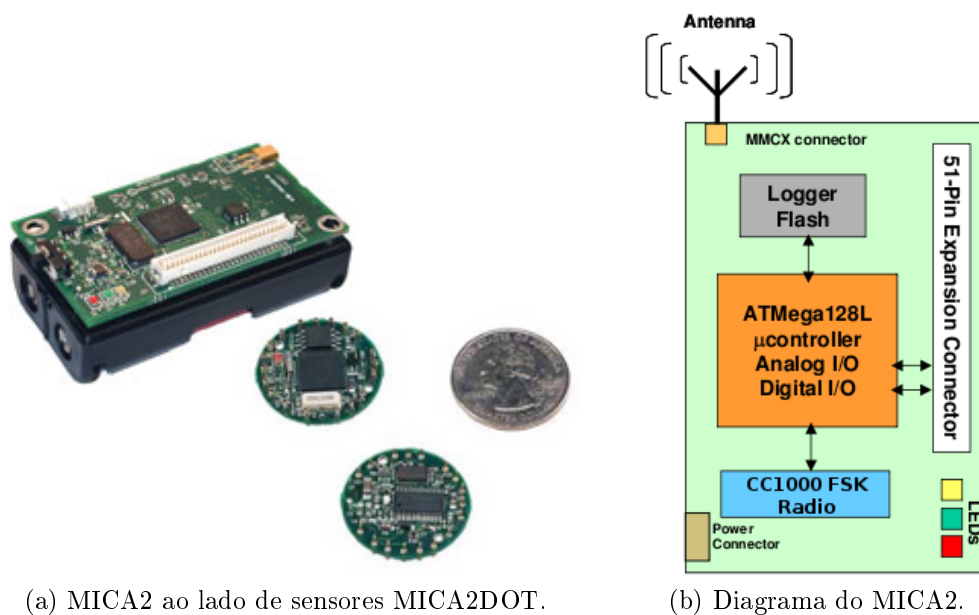


Figura 5.1: MICA2 Motes.

De modo geral, e para os nós sensores do tipo MICA2 em particular, a bateria é a única fonte de energia disponível para os nós sensores. Assim sendo, é de fundamental importância dispormos de um modelo matemático capaz de prever a descarga da mesma. Esse modelo matemático é apresentado na seção a seguir.

### 5.2.1.1 Provedor de energia

Neste trabalho, empregamos o modelo linear de descarga de bateria (veja [Park et al. \[2001\]](#)). Neste modelo, assume-se que a diferença de potencial elétrico entre os pólos da bateria permanecerá constante ao longo de toda a sua vida útil. Dessa forma, temos uma bateria ideal, cuja capacidade máxima é alcançada independentemente da taxa de descarga. Esse modelo simples de bateria permite descobrir a quantidade de carga que é consumida pela aplicação.

Outra hipótese importante no modelo linear é o fato de, durante a execução de um mesmo modo de operação do nó sensor, a corrente elétrica permanecer constante durante o período de tempo gasto para a execução daquele modo de operação.

Denominando por  $E(t_0)$  a carga da bateria no instante  $t_0$ , e assumindo que durante o intervalo de tempo  $[t_0, t_0 + t_d]$  o nó sensor executa o mesmo modo de operação e que  $I$  representa a corrente elétrica associada a este modo, temos que:

$$E(t_0 + t_d) = E(t_0) - \int_{t=t_0}^{t=t_0+t_d} I(t) dt = E(t_0) - I \cdot t_d \quad (5.1)$$

Conhecido o modelo de descarga de bateria utilizado, apresentamos a seguir a modelagem do consumo de energia dos nós sensores.

### 5.2.1.2 Consumidores de energia

No ciclo de vida operacional de uma RSSF, cinco fases podem ser identificadas: configuração, manutenção, sensoriamento, processamento e comunicação. Em um determinado instante, a rede pode executar ações características de mais de uma destas fases, simultaneamente. De forma análoga, durante cada uma destas fases um nó sensor pode assumir um ou mais modos de operação. São eles:

- Transmissão: envio de dados.
- Recepção: recebimento de dados.
- Escuta do canal: monitoramento do canal de comunicação.
- Sensoriamento: monitoramento de um fenômeno ou parâmetro do ambiente.
- Processamento: operações de cooperação entre nós, como auto-teste, tradução de dados, re-roteamento, fusão de dados, descoberta de localização, etc.

Os componentes do nó MICA2 que são responsáveis pela realização dessas operações são o rádio, a placa de sensoriamento e o processador. Esses componentes são os consumidores de energia do nó sensor. Os modelos de consumo de energia de cada uma dessas operações, para um intervalo de tempo, são os seguintes:



**a) Modelo para a energia consumida na transmissão**

$$E_{tx}(t_{tx}) = \alpha_{tx} t_{tx}, \text{ onde} \quad (5.2)$$

- $E_{tx}$  é a energia gasta na transmissão, em mAh.
- $\alpha_{tx}$  é a corrente consumida na transmissão, em mA. O valor dessa corrente depende da potência de transmissão.
- $t_{tx}$  é o intervalo de tempo durante o qual ocorreu transmissão de mensagens, em horas.

**b) Modelo para a energia consumida na recepção**

$$E_{rx}(t_{rx}) = \alpha_{rx} t_{rx}, \text{ onde} \quad (5.3)$$

- $E_{rx}$  é a energia gasta na recepção, em mAh.
- $\alpha_{rx}$  é a corrente consumida na recepção, em mA.
- $t_{rx}$  é o intervalo de tempo durante o qual ocorreu recepção de mensagens, em horas.

**c) Modelo para a energia consumida na escuta de canal**

$$E_e(t_e) = \alpha_e t_e, \text{ onde} \quad (5.4)$$

- $E_e$  é a energia gasta na escuta do canal, em mAh.
- $\alpha_e$  é a corrente consumida na escuta do canal, em mA.
- $t_e$  é o intervalo de tempo durante o qual foi realizada a escuta de canal, em horas.

**d) Modelo para a energia consumida no sensoriamento**

$$E_s(t_s) = \alpha_s t_s, \text{ onde} \quad (5.5)$$

- $E_s$  é a energia gasta no sensoriamento, em mAh.
- $\alpha_s$  é a corrente consumida no sensoriamento, em mA.
- $t_s$  é o intervalo de tempo durante o qual foi realizada a operação de sensoriamento, em horas.

**e) Modelo para a energia consumida no processamento**

$$E_p(t_p) = \alpha_p t_p, \text{ onde} \quad (5.6)$$

- $E_p$  é a energia gasta no processamento, em mAh.
- $\alpha_p$  é a corrente consumida no processamento, em mA.
- $t_p$  é o intervalo de tempo durante o qual foi realizada uma operação de processamento, em horas.

### 5.2.1.3 Consumo de energia de uma aplicação em RSSFs

A determinação do consumo de energia de uma aplicação em RSSFs leva em consideração o modelo de descarga da bateria e os modelos de consumo de energia dos componentes do nó sensor, descritos anteriormente. O consumo de energia de um nó sensor depende do seu estado operacional. Neste trabalho, quatro estados operacionais são empregados para os nós sensores da aplicação:

1. **Ativo** - o nó ativo executa as atividades de transmissão e recepção de dados, escuta do canal, sensoriamento do ambiente e processamento de dados.
2. **Transmitindo** - as atividades de monitoramento do nó no estado *transmitindo* estão desligadas. Entretanto, o seu rádio está ligado, para que este nó possa ser usado para encaminhar mensagens e, assim, conectar trechos da rede. São executadas as atividades de transmissão e recepção de dados, escuta do canal de comunicação e processamento de dados.
3. **Monitorando** - o nó no estado *monitorando* realiza apenas as operações de sensoriamento do ambiente e processamento de dados, ou seja, seu rádio está desligado ou em baixa potência.
4. **Dormindo** - nesse caso o nó encontra-se em estado de baixa potência para economizar energia na rede, pois seu consumo de energia é muito baixo.

Para uma bateria com carga  $E(t_0)$  no instante  $t_0$ , sua nova carga  $E(t_0 + t)$  após o intervalo de tempo  $\Delta t = t - t_0$ , para um sensor no estado *ativo*, é dado por:

$$E(t_0 + t) = E(t_0) - (E_{tx}(\Delta t) + E_{rx}(\Delta t) + E_e(\Delta t) + E_s(\Delta t) + E_p(\Delta t)) \quad (5.7)$$

Similarmente, a nova carga da bateria de um sensor no estado operacional *transmitindo*, após um intervalo de tempo  $\Delta t$ , é dada por:

$$E(t_0 + t) = E(t_0) - (E_{tx}(\Delta t) + E_{rx}(\Delta t) + E_e(\Delta t) + E_p(\Delta t)) \quad (5.8)$$

Para um nó sensor no estado operacional *monitorando*, após o intervalo de tempo  $\Delta t$  a carga de sua bateria é dada por:

$$E(t_0 + t) = E(t_0) - (E_s(\Delta t) + E_p(\Delta t)) \quad (5.9)$$

No estado operacional *dormindo*, consideramos que o consumo de energia é desprezível quando comparado aos demais gastos. No caso em que o sensor foi desligado, não há gasto de energia.

### 5.2.2 Carga de trabalho

Uma aplicação de monitoramento da temperatura ambiente foi escolhida para ser simulada. Nessa aplicação, os nós sensores monitoram periodicamente a temperatura ambiente, em intervalos de 20 segundos. Cada medida de temperatura é codificada por meio de 32 bytes de informação. O tamanho da memória do nó sensor MICA2 é de 4 Kbytes, o que é suficiente armazenar os dados de cerca de 1 hora ininterrupta de monitoramento de temperatura.

### 5.2.3 Camada de enlace

Para a camada de enlace, o JiST/SWANS fornece uma implementação do protocolo IEEE 802.11. Esse protocolo é muito utilizado nas simulações de projetos de pesquisa na área de RSSFs. O processo de transmissão de dados é assíncrono, permitindo colisão de mensagens, e possui um mecanismo para reenvio de mensagens perdidas. Assim, no protocolo IEEE 802.11, algumas mensagens de controle são trocadas entre os elementos da rede. Essas mensagens não são computadas nas medidas de avaliação de desempenho da rede, como taxa de mensagens entregues, mas são consideradas no cálculo do consumo de energia dos nós sensores.

### 5.2.4 Modelo de propagação de sinal

Um modelo de propagação de sinal (*path loss*) tenta, por aproximação, prever o comportamento das ondas eletromagnéticas durante a propagação em um determinado meio. O modelo de propagação de sinal utilizado para a comunicação sem fio, neste trabalho, é o *Free-space path loss* (Balanis [2003]). Esse modelo é utilizado em muitos trabalhos em RSSFs, nos quais estudar a propagação de sinais não é o foco principal. O JiST/SWANS já possui o *Free-space path loss* como recurso disponível para utilização na simulação.

O *Free-space path loss* é utilizado para predição da intensidade com a qual um sinal transmitido chega ao seu receptor, quando as duas pontas do enlace apresentam entre elas uma linha de visada não-obstruída por nenhum obstáculo, o qual pode causar

efeitos de difração, obstrução, refração e reflexão. Neste modelo, a potência que chega ao receptor decai à medida que a distância de separação entre o transmissor e o receptor aumenta, sendo que essa perda é proporcional ao quadrado da distância entre transmissor e receptor.

### 5.2.5 Outras definições para o cenário de simulação

Não é objetivo deste trabalho abordar o problema de localização dos nós sensores. No método aqui descrito assume-se que cada nó sensor sabe sua localização, o que pode ser feito através de técnicas baratas como triangulação (Bulusu et al. [2000]), e outras técnicas presentes na literatura (Ye [2006]; Meguerdichian et al. [2001]; de Oliveira et al. [2005]).

No nosso trabalho, o controle de densidade é realizado de forma centralizada, ou seja, é calculado pelo sorvedouro principal, e controlado por ele e pelos sorvedouros auxiliares. Para que isso seja possível, assume-se também que os sorvedouros conhecem a localização de todos os nós sensores, para que eles possam não apenas coordenar a organização e estrutura da rede, mas também os estados operacionais dos seus elementos. Assim, os nós sorvedouros são capazes de coletar as informações sobre o estado de energia dos nós sensores, necessárias para a resolução do problema de controle de densidade. No início da operação da RSSF todos os nós estão ativos, e à medida que os sorvedouros chegam às raízes das árvores de comunicação, a decisão sobre o controle de densidade é passada aos nós sensores.

Adicionalmente, não são consideradas neste trabalho possíveis restrições à movimentação do nós sorvedouros através das rotas para eles estabelecidas. Dentre essas restrições encontram-se obstáculos e relevo, no caso de sorvedouros terrestres, ou direção e velocidade do vento, no caso de sorvedouros aéreos.

Agora que já definimos e detalhamos o cenário de simulação, descrevemos a seguir o funcionamento da simulação de RSSFs empregada neste trabalho. Apresentamos primeiramente uma abordagem em que o sorvedouro principal utiliza o procedimento HRFI-K para organizar topologicamente a rede e coordenar a operação de seus elementos.

## 5.3 Simulação de RSSFs utilizando o HRFI-K

Em nosso trabalho, a simulação de RSSFs é baseada em períodos. Um período de simulação corresponde à visita do sorvedouro principal a todos os líderes de grupo em  $R^1 = \{r_1^1, \dots, r_{t(1)}^1\}$ , mais a sua espera pela chegada dos sorvedouros auxiliares à base  $1 \in V$ . Convencionamos que  $R^1$  é o conjunto de líderes de grupo contidos na rota de maior comprimento, dentre as  $K$  rotas definidas. A simulação é executada até que o *limite de tempo de simulação*, fornecido como parâmetro de entrada, seja atingido. Dessa forma, o último período da simulação pode acabar prematuramente, caso esse limite de tempo de simulação seja atingido.

É necessário dizer que o *tempo de simulação* é diferente do tempo real de execução do programa. O tempo de simulação é representado pelo relógio da aplicação, o qual não avança para o próximo ponto discreto de tempo enquanto todos os processos para o tempo discreto de simulação atual tenham sido completados. Dessa forma, um experimento que leva alguns minutos para ser executado pode simular a operação de algumas horas de uma RSSF.

A nossa simulação funciona da seguinte maneira. Primeiramente, o cenário de simulação é configurado, o que inclui distribuir os nós sensores pela área a ser monitorada, de acordo com informações passadas ao programa de simulação (número de nós sensores, sua localização geográfica e quantidade de energia inicial). Nesse momento, todos os nós sensores estão no estado operacional *monitorando*.

No início do primeiro período, de posse da localização geográfica e nível de energia de todos os nós sensores presentes na RSSF, o sorvedouro principal resolve o Problema de Controle de Densidade (PCD). Nessa fase, todos os sensores são considerados, uma vez que no início da simulação todos estão ligados. O objetivo é definir quais nós sensores exercerão a função de sensoriamento durante o primeiro período. O PCD foi modelado como o Problema de Cobertura de Conjuntos (*Set Covering Problem*) (Garey & Johnson [1990]), sendo então o mesmo método utilizado pelo MHS (Aioffi et al. [2007b]).

Após a resolução do PCD, o sorvedouro principal calcula as  $K$  rotas definidas pelos líderes de grupo em  $R^1, \dots, R^K$ , e as árvores restritas a  $H$  saltos  $T_i^k = (V_i^k, A_i^k)$ ,  $k \in \mathcal{K}$ ,  $i = 1, \dots, t(k)$  através da abordagem HRFI-K. Esse passo não conta tempo de simulação, pois como a heurística é rápida, consideramos que as decisões podem ser tomadas durante a trajetória do sorvedouro principal pela rede. Os sorvedouros auxiliares percorrem as rotas definidas por  $R^2, \dots, R^K$ . Para o primeiro período, os nós sensores contidos em  $V$  correspondem a todos os nós sensores vivos nesse momento.

Isto é necessário uma vez que, no início da simulação, todos os nós sensores estão vivos e monitorando, e para que sejam desligados precisam ser visitados por algum sorvedouro.

Em seguida, é iniciada a coleta dos dados sensorizados de cada um dos nós sensores em  $V$ . O sorvedouro cuja rota tem o menor comprimento é o responsável por coletar os dados da árvore  $T_1^1$  enraizada em  $r_1^1 = 1$ . Os demais sorvedouros usam o vértice 1 apenas como ponto de partida para suas rotas. Cabe então mencionar que as árvores  $\{T_1^1, \dots, T_1^k\}$ ,  $k \in \mathcal{K}$  são idênticas, assim como são seus líderes de grupo  $\{r_1^1, \dots, r_1^k\}$ .

O processo de coleta de dados funciona da seguinte maneira. O sorvedouro  $m \in \mathcal{K}$  cuja rota tem o menor comprimento é posicionado inicialmente sobre o líder de grupo  $r_1^m = 1$ . O sorvedouro então envia um estímulo aos nós sensores que pertencem a  $V_1^m$ . Ao receber o estímulo, os nós sensores ligam seus rádios, passando nesse momento para o estado operacional *ativo*. O sorvedouro então envia as decisões de topologia e controle de densidade da árvore  $T_1^m = (V_1^m, A_1^m)$ . Após a propagação das decisões de controle de densidade e topologia, os nós sensores podem enviar os seus dados sensorizados, através dos arcos  $A_1^m$ . Os sensores enviam também dados sobre o seu estado atual de energia. Essa informação é utilizada futuramente pelo sorvedouro principal para tomar as decisões sobre agrupamento, roteamento e controle de densidade. Cabe mencionar uma pequena otimização feita aqui. Os nós conectados ao líder de grupo  $r_1^m$  na verdade enviam seus dados diretamente ao sorvedouro. Dessa forma, o líder de grupo não é sobrecarregado com as mensagens de todos os nós de seu agrupamento.

Após a coleta dos dados da árvore  $T_1^m$ , o rádio dos nós sensores em  $V_1^m$  é desligado, e sua função de sensoriamento é ligada/desligada segundo as informações do controle de densidade, resolvido antes da partida do sorvedouro principal. Isso significa que os nós em  $V_1^m$  "ligados" pelo controle de densidade alteram seu estado operacional para *monitorando*, enquanto que os restantes vão para o estado *dormindo*. Em seguida, caso o limite de tempo de simulação não tenha sido atingido, o sorvedouro  $m$  se move para o próximo líder de grupo  $r_2^m$  definido em sua rota, e o mesmo processo se repete para a árvore  $T_2^m = (V_2^m, A_2^m)$ .

Durante a coleta de dados em uma árvore  $T_i^m$ , os nós sensores comunicam-se entre si e com o nó sorvedouro  $m \in \mathcal{K}$  utilizando o protocolo 802.11, que permite colisão de pacotes e contém um mecanismo de reenvio de pacotes perdidos. Assim, é necessário que o sorvedouro mantenha um registro, dizendo se recebeu ou não os dados de cada nó sensor em  $V_i^m$ . Um contador regressivo é mantido para cada sensor em  $V_i^m$ , o qual limita o tempo de espera por mensagens de cada nó sensor. Esse contador é reinicializado toda vez que uma mensagem que não indica o fim da transmissão de

dados de um determinado nó sensor é recebida. Quando os dados de todos os sensores em  $V_i^m$  forem recebidos, ou quando todos os contadores terminarem, a coleta de dados da árvore  $T_i^m$  termina.

O procedimento de coleta de dados realizado pelos demais sorvedouros funciona de modo similar ao explicado para o sorvedouro  $m \in \mathcal{K}$ . A diferença é que tais sorvedouros não coletam os dados da árvore enraizada sobre o seu ponto de partida (vértice 1), como dito anteriormente. No início do período, enquanto o sorvedouro  $m$  inicia a coleta de dados da árvore  $T_1^m$ , cada nó sorvedouro  $k \in \mathcal{K} \setminus \{m\}$  se movimenta em direção à raiz da árvore  $T_2^k$  para coletar os seus dados.

Para que o período de simulação termine é necessário que, após a visita aos líderes de grupo de suas respectivas rotas, todos os sorvedouros estejam posicionados sobre 1. Nesse momento, os sorvedouros auxiliares transmitem ao sorvedouro principal as informações sobre os níveis de energia e as mensagens recebidas dos nós sensores com que se comunicaram. Um novo período de simulação é então iniciado, caso o limite de tempo de simulação não tenha sido atingido.

Os períodos seguintes funcionam de forma similar ao primeiro período. A única diferença é a definição dos nós sensores que farão parte do conjunto  $V$  usado como entrada no procedimento HRFI-K, executado sempre no início de cada período.

Para que o controle de densidade realizado seja bastante efetivo, não é interessante incluir em  $V$  todos os nós vivos. Ao invés disso, para efeito de modelagem,  $V$  representa a união dos conjuntos de nós sensores  $V^d$ ,  $V^a$  e  $V^p$ , explicados a seguir. O conjunto  $V^d$  corresponde aos nós sensores que devem ser "ligados" durante o período atual, de acordo com o último controle de densidade executado. Esses nós precisam fazer parte de  $V$  pois têm que receber a decisão de controle de densidade e, conseqüentemente, devem fazer parte de alguma das árvores  $T_i^k$ . Já o conjunto  $V^a$  contém os nós sensores "ligados" pelo controle de densidade no período anterior. Os dados desses sensores precisam ser coletados uma vez que permaneceram monitorando durante o período anterior. Finalmente, o conjunto  $V^p$  contém os nós sensores  $j$  cujas mensagens foram perdidas, mais aqueles que estavam no caminho entre  $j$  o líder de grupo  $r_i^k$ . A inclusão de tais nós em  $V$  permite que os sorvedouros possam receber mensagens que eventualmente ficaram guardadas na memória de algum nó sensor.

Resumindo, nos períodos de simulação seguintes ao primeiro, o conjunto  $V$  é definido de forma coletar dados sensorizados no período imediatamente anterior, e reconfigurar as árvores de acordo com as novas decisões de controle de densidade, roteamento e agrupamento. Dessa forma, durante a coleta de dados de uma árvore  $T_i^k$ ,

quando o sorvedouro  $k \in \mathcal{K}$  envia o estímulo aos nós sensores em  $V_i^k$ , os nós que estavam no estado operacional *monitorando* vão para o estado *ativo*, e os nós que estavam no estado *dormindo* vão para o estado *transmitindo*.

A Figura 5.2 ilustra os estados operacionais dos nós sensores, para  $K = 1$ , durante a coleta de dados de uma árvore em um período posterior ao primeiro. Nessa Figura, as partes em cinza claro representam a área da rede efetivamente coberta, ou seja, estão dentro do raio de sensoriamento de pelo menos um nó sensor nos estados *ativo* ou *monitorando*. Já o fluxograma apresentado na Figura 5.3 ilustra os passos da simulação acima descrita.

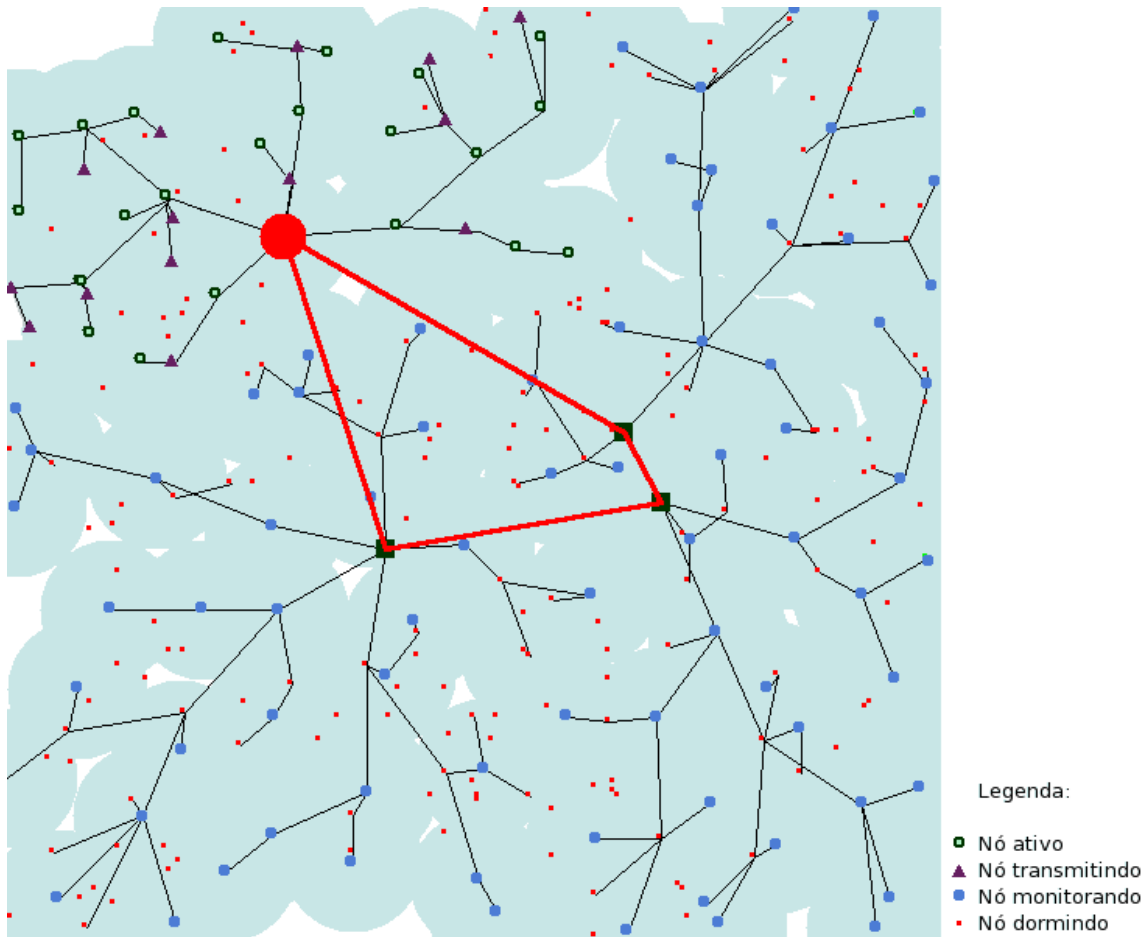


Figura 5.2: Estados operacionais em uma simulação de RSSF com 250 nós sensores, com restrição de saltos  $H = 4$ , utilizando o HRFI-1.



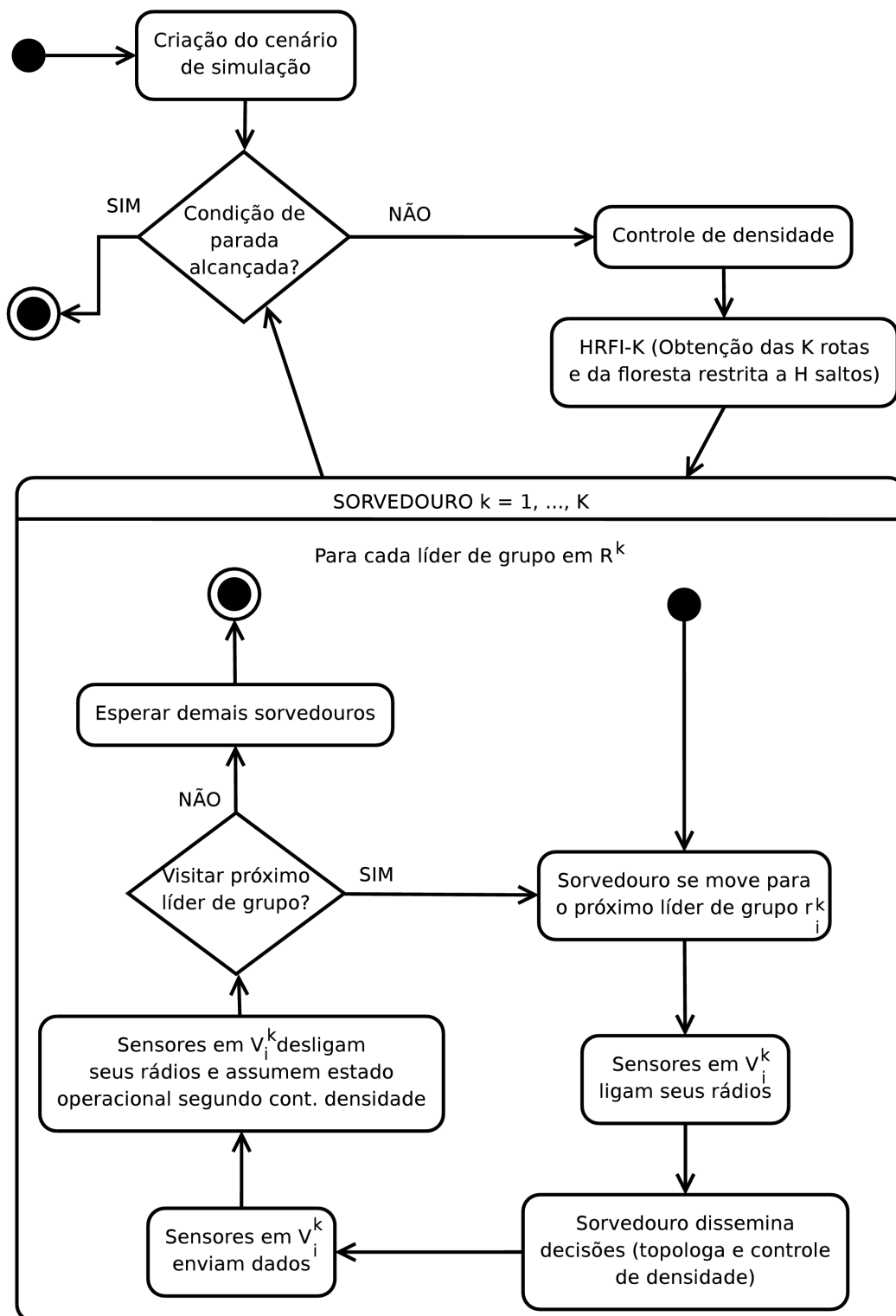


Figura 5.3: Fluxograma da simulação utilizando o HRFI-K.

## 5.4 Outra abordagem para simulação de RSSFs

Apresentamos nessa seção outra heurística que foi utilizada neste trabalho para simulação de RSSFs. Para  $K = 1$ , o PFMR apresenta uma arquitetura de rede similar à do MHS apresentada em Aioffi et al. [2007b]. Assim, propomos aqui uma heurística cujo principal objetivo é equivalente ao objetivo do MHS: encontrar uma rota de menor comprimento possível para o sorvedouro. Dessa forma, teremos como comparar a nossa proposta com aquela descrita em Aioffi et al. [2007b], nos permitindo avaliar o impacto de uma abordagem integrada em fatores como gasto de energia e atraso na entrega de mensagens. Dessa forma, apresentamos a seguir o MHS-I (*MHS Integrado*), cujo objetivo é minimizar o comprimento da rota do sorvedouro, e em seguida, o custo da floresta de coleta de dados para essa rota.

### 5.4.1 MHS-I

Como mencionado na seção 2.2, a nossa arquitetura é, em alguns pontos, similar à apresentada para o MHS (Aioffi et al. [2007b]). Para  $K = 1$ , a única diferença entre as arquiteturas é a limitação do comprimento da rota do nó sorvedouro ao parâmetro  $D_{max}$  presente no nosso problema. Assim, apresentamos aqui uma heurística cujo objetivo é o mesmo do MHS, minimizar a rota do nó sorvedouro. Pretendemos assim comparar quantitativamente a nossa abordagem com a abordagem do trabalho anterior.

O nosso objetivo ao propor o MHS-I é mostrar que a nossa abordagem, a qual resolve de forma integrada os problemas de roteamento do sorvedouro e agrupamento dos nós sensores, é mais flexível do que o MHS. Dessa flexibilidade resultariam rotas menores que aquelas encontradas pelo MHS e, conseqüentemente, menor atraso na entrega de mensagens. Também deseja-se mostrar que nossa abordagem apresenta maior economia na energia utilizada para transmissão de mensagens, ao minimizar, em uma segunda etapa, o custo da floresta de coleta de dados.

A diferença entre as heurísticas HRFI-K e MHS-I, por outro lado, é bastante pequena. Na verdade, são apenas três diferenças. Primeiramente, a vizinhança ADD não é utilizada durante o VND do MHS-I. A segunda diferença reside na obtenção de novas soluções na Busca Local na vizinhança SWAP: apenas movimentos que resultem em rotas com comprimento menor do que o comprimento da rota da solução atual são permitidos. A terceira reside na Busca Local MIN-FOREST. Caso haja mudança do rótulo de algum vértice que é um líder de grupo, apenas movimentos que resultem em uma floresta com custo menor e comprimento de rota menor do que a solução atual

são permitidos.

Vale mencionar que, como o MHS-I considera RSSFs com apenas um nó sorvedouro, a busca sobre a vizinhança *ROUTE-SWAP* não é realizada. É importante observar também que, para o MHS-I, não existem instâncias inviáveis. Cabe destacar que, para o raio de comunicação  $R_C$ , o(s) sorvedouro(s) teriam que visitar todos os nós sensores, tal qual um PCV.

### 5.4.2 Simulação de RSSFs utilizando o MHS-I

No nosso trabalho, o objetivo de simular RSSFs utilizando a abordagem MHS-I é estabelecer uma comparação entre a abordagem anterior MHS e a proposta desse trabalho, o HRFI-K para o PFMR. Para isso, teremos duas versões de simulação de RSSFs utilizando o MHS-I. Para diferenciar essas duas versões e citá-las ao longo desse texto, elas serão chamadas de *Sim\_MHS-I* e *SimInvert\_MHS-I*.

A simulação *Sim\_MHS-I* acontece de forma similar à simulação de RSSFs utilizando o HRFI-K para  $K = 1$ . A única diferença é que a heurística MHS-I é empregada, ao invés da heurística HRFI-K, para o cálculo da rota e a floresta de coleta de dados pelo sorvedouro principal.

Já a simulação *SimInvert\_MHS-I* possui uma diferença significativa em relação à simulação apresentada na seção 5.3: o controle de densidade é feito antes da chamada da heurística que resolve o PFMR, similarmente ao método MHS apresentado em [Aioffi et al. \[2007b\]](#). No MHS, o Problema de Controle de Densidade também é resolvido antes dos problemas de roteamento e agrupamento.

Na simulação *SimInvert\_MHS-I*, o conjunto de nós sensores  $V$  passado à heurística MHS-I corresponde, em todos os períodos de simulação, ao conjunto de nós sensores vivos. Dessa forma, todos eles fazem parte da floresta restrita a  $H$  saltos, ou seja, todos eles são utilizados no processo de coleta dos dados sensorizados pelos elementos da RSSF. O controle de densidade interferirá apenas no conjunto de nós que estão monitorando o ambiente em cada período. Isso é facilmente visualizado na Figura 5.4. Note que todos os nós sensores estão conectados a alguma árvore de coleta de dados.

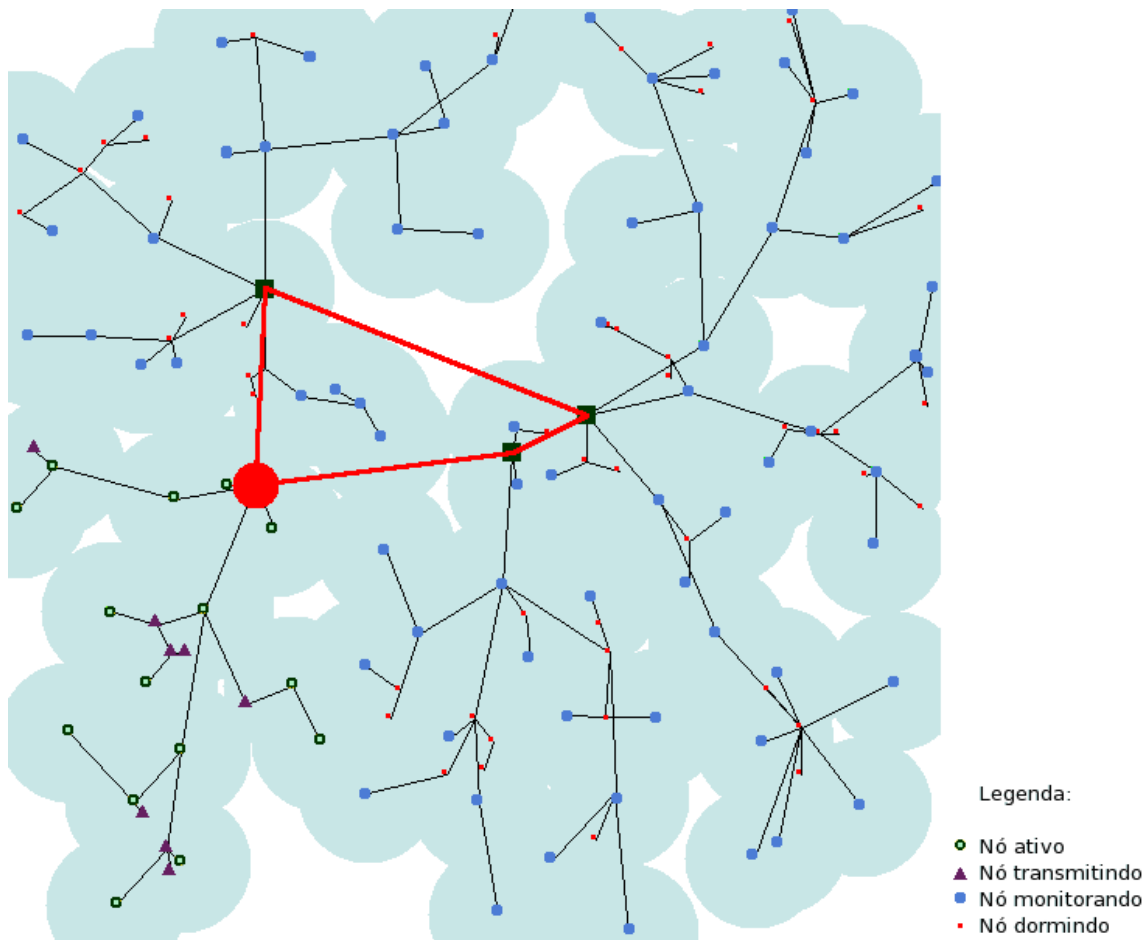


Figura 5.4: Cenário da simulação `SimInvert_MHS-I` para uma RSSF com 150 nós sensores, com restrição de saltos  $H = 4$ .

## 5.5 Parâmetros de simulação

Salvo indicação contrária, os mesmos parâmetros de simulação exibidos na Tabela 5.1 foram utilizados em todos os experimentos computacionais realizados para este trabalho. Tais parâmetros foram definidos tendo como base o *hardware* do nó MICA2.

Como o nosso trabalho possui abordagem centralizada, os nós sensores não realizam nenhuma função referente ao modo de operação de processamento. Portanto, a energia  $E_p$  gasta com processamento é igual a zero.

A energia gasta com transmissão de mensagens, nos nossos experimentos, é variável, segundo um mecanismo de controle de potência simples coordenado pelos nós

Tabela 5.1: Parâmetros de simulação.

<i>Parâmetro</i>	<i>Valor</i>
Área monitorada	40000 $m^2$
Energia inicial do nó sensor	10 $mAh$
Raio de sensoriamento	15 $m$
Raio máximo de comunicação	30 $m$
Corrente consumida na recepção	7 $mA$
Corrente consumida com rádio ativo	9.5 $mA$
Corrente consumida no sensoriamento	1 $mA$
Velocidade do sorvedouro	1 $m/s$
Largura de banda	250 $kbps$

sorvedouros. O emprego desse mecanismo em RSSFs possui várias vantagens, discutidas a seguir.

Diversos trabalhos mostraram que a utilização de um mecanismo de Controle de Potência de Transmissão (CPT) em RSSFs resulta em economia de energia e melhor utilização de largura de banda (Monks [2001]; Jung & Vaidya [2005]). Quando é utilizada uma potência fixa de transmissão, cujo sinal deve ter potência suficiente para alcançar qualquer nó dentro de um raio de comunicação pré-definido, a capacidade da RSSF decresce ao adicionarmos mais nós na rede. Isso acontece devido ao aumento da interferência mútua entre nós ao transmitirem dados (Correia et al. [2005]).

Assim, o uso de protocolos que empregam técnicas de CPT possibilita menor número de colisões e o estabelecimento de enlaces com baixa taxa de erros (Correia et al. [2005]). Assim, a transmissão em potências mais baixas permite aumentar o número de transmissões simultâneas e a vazão da rede (Gomez & Campbell [2004]).

Em todas as simulações do nosso trabalho foi utilizado um mecanismo simples de CPT, baseado puramente na distância entre nós sensores e coordenado pelo sorvedouro. Como este conhece a localização de cada nó sensor e a configuração das árvores de coleta de dados, o sorvedouro envia, juntamente com as informações sobre a topologia de cada árvore, qual deve ser a potência de transmissão de cada nó sensor para se comunicar com seu respectivo nó pai.

As potências de transmissão utilizadas em nosso trabalho são baseadas nos valores da Tabela 4.1, apresentada no Capítulo 4. Para um determinado enlace, a potência empregada corresponde ao valor cujo alcance seja imediatamente superior ao da distância entre o nó transmissor e receptor, quando essa distância não for encontrada na tabela. Esse valor é então acrescido de 2  $dBm$ . O acréscimo foi necessário pois, se utilizarmos exatamente os valores da Tabela 4.1, haverá um número alto de colisão de mensagens

devido à interferência de outros enlaces, já que esses valores correspondem à potência de transmissão mínima necessária para que o rádio alcance determinada distância.

O motivo pelo qual o valor de acréscimo é  $2\text{dBm}$  é permitir igualar a potência máxima da nossa abordagem à potência de transmissão fixa do MHS, utilizada em [Aioffi et al. \[2007b\]](#) (a qual é equivalente a  $-6\text{dBm}$ ). A fim de comparar as abordagens aqui propostas com o MHS exclusivamente sob a ótica da topologia da rede (rota do sorvedouro e árvore de comunicação), os resultados referentes ao MHS exibidos neste trabalho também empregam o CPT anteriormente descrito.

## 5.6 Medidas de QoS

Várias medidas de Qualidade de Serviço (QoS) para RSSFs podem ser utilizadas para avaliar seu desempenho e comparar diferentes abordagens. A seguir apresentamos as medidas que consideramos mais importantes para a avaliação dos resultados do nosso trabalho.

- **Cobertura:** Corresponde à porcentagem da área de monitoramento que é efetivamente coberta por pelo menos um nó sensor. Uma porção da área é considerada coberta se está dentro do raio de sensoriamento de algum nó sensor nos estados *monitorando* ou *ativo*. Esse valor é calculado a cada 10 minutos, e corresponde à média das medidas feitas dentro desse período a cada 20 segundos.
- **Energia residual:** É a porcentagem da energia total que os nós sensores ainda dispõem, em relação à energia inicial total de todos os nós sensores, a cada intervalo de 10 minutos. É esperado que o comportamento dos resultados dessa medida sejam similares aos resultados da medida de cobertura, já que essas medidas são intimamente ligadas: com o passar do tempo, a energia residual dos nós sensores diminui, o que resulta na diminuição da área efetivamente coberta da rede.
- **Atraso na entrega de mensagens:** Tempo médio gasto pelas mensagens de dados para chegar ao nó sorvedouro, durante toda a operação da RSSF. Esse tempo corresponde ao intervalo entre a geração da mensagem e a recepção da mesma pelo nó sorvedouro.
- **Taxa de mensagens entregues:** Relação entre o número de mensagens de dados geradas pelos nós sensores, e o número de mensagens efetivamente entregues

ao nó sorvedouro. Essa medida indica a quantidade de informação que foi perdida durante a operação da RSSF.

Na próxima seção, apresentamos os resultados associados aos experimentos computacionais realizados neste trabalho. Todas as medidas supracitadas foram utilizadas para avaliar as abordagens para simulação de RSSFs apresentadas ao longo deste capítulo.

## 5.7 Experimentos computacionais

Nesta seção, apresentamos os resultados computacionais referentes à simulação de RSSFs utilizando as abordagens HRFI-K, MHS-I e o MHS com CPT. Primeiramente, apresentamos os resultados relacionados à abordagem `SimInvert_MHS-I`. Nessa abordagem, o procedimento de simulação é análogo ao MHS de [Aioffi et al. \[2007b\]](#), principalmente pelo fato dos problemas de agrupamento e roteamento serem resolvidos antes do Problema de Controle de Densidade. Portanto, podemos utilizar a simulação `SimInvert_MHS-I` para avaliar uma das principais vantagens da nossa estratégia (tratar os problemas de agrupamento e roteamento de forma integrada), comparando-a com uma versão do MHS que emprega o mecanismo de CPT anteriormente descrito.

Em seguida, avaliamos a importância do mecanismo de controle de densidade em RSSFs através da comparação dos resultados das abordagens `Sim_MHS-I` e `SimInvert_MHS-I`. A diferença fundamental entre essas duas abordagens é o momento em que é resolvido o Problema de Controle de Densidade. Esperamos assim justificar o motivo pelo qual a simulação utilizando o HRFI-K, apresentada na seção 5.3, resolve o Problema de Controle de Densidade *antes* da resolução dos problemas de roteamento e agrupamento, ao contrário do que é feito no MHS.

Os resultados associados à simulação de RSSFs utilizando o HRFI-K para  $K = 1$  são então apresentados. O HRFI-1 pode ser visto como um MHS-I em que pode-se controlar o tamanho da rota do nó sorvedouro. Dessa forma, podemos avaliar quantitativamente qual o impacto do tamanho da rota do nó sorvedouro no desempenho da rede. Finalmente, comparamos os resultados obtidos para simulações de RSSFs com mais de um nó sorvedouro móvel, utilizando o HRFI-K para  $K > 1$ .

A Tabela 5.2 apresenta um resumo das principais abordagens estudadas ao longo deste trabalho. Adicionalmente, contém a notação utilizada nos gráficos dos resultados apresentados a seguir.

Tabela 5.2: Abordagens a serem comparadas.

<i>Nome</i>	<i>Contém CPT?</i>	<i>Agrup. e rot. integrados?</i>	<i>Min. gasto de energia?</i>	<i>PCD primeiro?</i>	<i>Tem <math>D_{max}</math>?</i>	<i>&gt; 1 sorvedouro?</i>
<a href="#">Aioffi et al. [2007b]</a>	-	-	-	-	-	-
Mod. MHS	SIM	-	-	-	-	-
MHS-I	SIM	SIM	SIM	-	-	-
MHS-I Inv	SIM	SIM	SIM	SIM	-	-
HRFI	SIM	SIM	SIM	SIM	SIM	-
HRFI-K	SIM	SIM	SIM	SIM	SIM	SIM

### 5.7.1 Aspectos gerais dos testes

Nossos experimentos computacionais foram conduzidos utilizando instâncias Euclidianas, nas quais o número de nós sensores varia de 50 a 600 nós. Os nós sensores são aleatoriamente distribuídos em uma área quadrada plana de lado  $L = 200m$ . Para cada abordagem apresentada, foram testados três diferentes valores para o número máximo de saltos  $H = 2, 3, 4$ . Cada resultado apresentado corresponde à média da execução de 33 experimentos computacionais. Vale mencionar que estas instâncias são diferentes daquelas apresentadas no capítulo 4, e são exatamente iguais àquelas utilizadas em [Aioffi et al. \[2007b\]](#).

Para as simulações de RSSFs utilizando o HRFI-1 e o MHS-I, o vértice inicial 1 é definido como o nó sensor em  $V$  mais próximo do centro do primeiro quadrante. Já para as simulações com mais de um nó sorvedouro, esse vértice inicial é o nó sensor em  $V$  mais próximo do centro da área quadrada.

Para todos os resultados aqui apresentados, o atraso na entrega de mensagens e a taxa de mensagens entregues referem-se à simulações de 10 horas de operação das RSSFs. Já os resultados referentes à cobertura da rede e à energia residual foram obtidos através de simulações de 25 horas de operação das RSSFs, apenas para instâncias de 400 nós sensores. Para essas duas últimas medidas, foi necessário aumentar o tempo de simulação para que o gasto de energia e suas consequências ficassem mais evidentes.

### 5.7.2 MHS-I

Nesta seção, apresentamos os resultados comparativos entre a simulação `SimInvert_MHS-I` e o método MHS. A Figura 5.5 apresenta os resultados referentes ao atraso na entrega de mensagens em função do número de sensores presentes na RSSF.



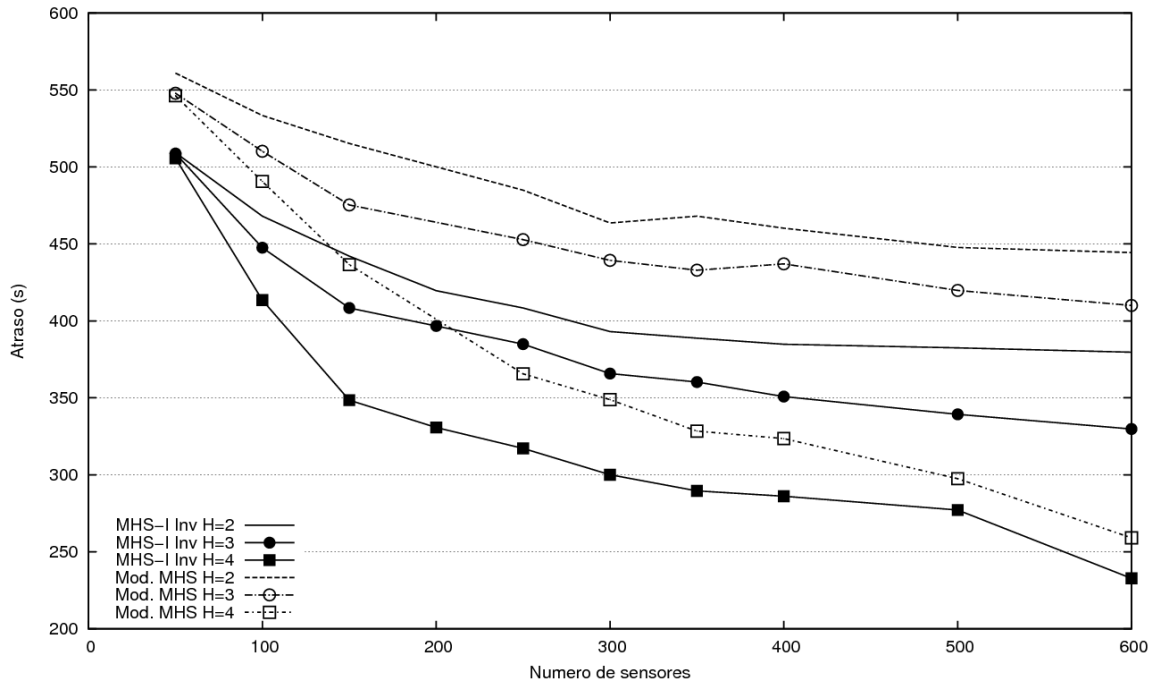


Figura 5.5: SimInvert\_MHS-I: Atraso na entrega de mensagens em função do número de sensores.

Avaliando os resultados apresentados nessa Figura, verificamos que o MHS-I realmente apresentou uma melhora no atraso na entrega de mensagens quando comparado ao MHS. Isso era esperado pois, na nossa abordagem, tratamos os problemas de roteamento e agrupamento de forma integrada.

Para redes mais esparsas, com 50 nós, o MHS apresentou atraso na entrega de mensagens cerca de 10 % maior do que o atraso do MHS-I. Já para redes mais densas, com 600 nós, o MHS apresentou atraso na entrega de mensagens 20% maior do que o MHS-I para  $H = 2$ , e 25% maior para  $H = 3$ . Entretanto, para  $H = 4$  a diferença foi apenas cerca de 7%.

Percebemos também que, para uma mesma abordagem, não há diferença evidente no atraso quando a rede tem apenas 50 nós. Isto se deve ao fato de que, mesmo para valores maiores de  $H$ , a profundidade das árvores não é explorada devido à natureza esparsa da rede. Entretanto, na medida em que a densidade de nós sensores aumenta, fica mais evidente o impacto que diferentes valores de  $H$  têm sobre o atraso na entrega de mensagens.

O aumento da densidade da rede levou também à diminuição do atraso na entrega de mensagens. Esse resultado se deve ao fato de que as redes mais densas oferecem mais opções de conectividade entre nós sensores, permitindo menores rotas para o nó

sorvedouro.

Os resultados sobre o atraso na entrega de mensagens apresentados na Figura 5.5 sugerem que as rotas encontradas pelo MHS-I são, na média, menores do que as rotas encontradas pelo MHS. Isso é exatamente o que esperávamos de uma abordagem integrada.

Para as abordagens de RSSFs com sorvedouro móvel tratadas nesse trabalho, quanto menor a rota do nó sorvedouro, mais frequentemente se dará sua comunicação com as árvores, resultando em maior gasto de energia. Assim, era de se esperar que a simulação `SimInvert_MHS-I` apresentasse maior gasto de energia do que o MHS. Entretanto, como podemos ver pela Figura 5.6, não é exatamente isso que acontece.

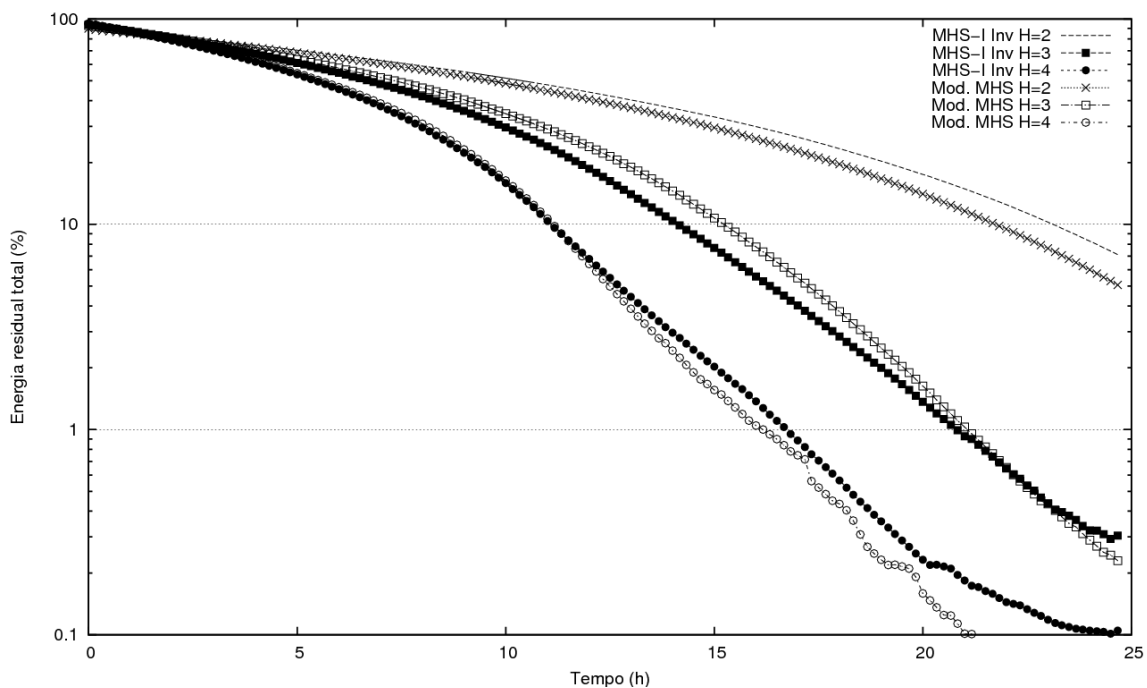


Figura 5.6: `SimInvert_MHS-I`: Energia residual total dos nós sensores em função do tempo de simulação.

A Figura 5.6 apresenta a porcentagem da energia residual total dos nós sensores em cada momento da simulação. O resultado mais evidente mostrado nesse gráfico é que, quanto maior o limite de número de saltos  $H$ , maior é o gasto de energia. Isso acontece pois, quanto maior  $H$ , mais profundas são as árvores de comunicação, aumentando o número de mensagens retransmitidas pelos próprios nós sensores.

Vemos também que para  $H = 2$  e  $H = 4$ , a nossa abordagem apresentou menor gasto de energia, uma vez que a energia residual total foi maior durante a maior parte

da simulação. Pode-se perceber inclusive que a RSSF da abordagem MHS, para  $H = 4$ , "morre" antes do final do tempo de simulação.

Entretanto, para  $H = 3$ , durante a maior parte da simulação, o MHS-I apresentou maior gasto de energia do que o trabalho anterior. Os resultados referentes ao atraso sugerem um motivo para tal comportamento. Como o MHS-I para  $H = 3$  apresentou a melhor melhora de atraso (cerca de 25% para 600 nós), a comunicação entre os elementos da rede foi ainda mais frequente, gerando maior tráfego de dados e, portanto, maior gasto de energia. O ganho obtido com a minimização da floresta de comunicação não foi suficiente para compensar o gasto de energia oriundo do aumento da frequência de comunicação.

Os resultados para o percentual da área coberta em função do tempo de simulação são mostrados na Figura 5.7. Percebe-se que, para as abordagens aqui mostradas, a cobertura apresenta comportamento similar ao gasto de energia: o MHS-I melhorou a cobertura para  $H = 2$  e  $H = 4$ . Para  $H = 2$ , o MHS-I apresentou cobertura 20% maior do que o trabalho anterior, ao final da simulação. É possível ver também que, para  $H = 3$ , o MHS-I apresentou melhor cobertura do que o MHS apenas no final da simulação, assim como ocorreu com o gasto de energia. Ainda assim, para  $H = 3$ , o MHS-I obteve o dobro da cobertura do trabalho anterior após 25 horas de tempo de simulação.

Na Figura 5.8 apresentamos os resultados referentes à taxa de mensagens entregues, medida que avalia a confiabilidade da RSSF. Esses resultados sugerem que uma RSSF operada através do método MHS-I apresenta maior confiabilidade do que o MHS com CPT, quando comparamos as abordagens com mesmo número limitado de saltos  $H$ . Entretanto, a diferença entre as abordagens não foi significativa. O MHS-I apresentou melhora de 3%, 2% e 3% para  $H = 2$ ,  $H = 3$  e  $H = 4$ , respectivamente.

Os resultados mostram também que a confiabilidade da rede diminui à medida que o número limitado de saltos  $H$  aumenta. Isto se deve à existência de árvores com profundidade maior quando aumentamos o limite de saltos. Árvores mais profundas aumentam a probabilidade de uma mensagem ser perdida, ao ser retransmitida pelos nós existentes no seu caminho até a raiz da árvore.

Percebemos que, pelos resultados apresentados nessa seção, o MHS-I apresentou uma melhora sistemática em relação ao MHS com mecanismo de CPT, para as medidas aqui consideradas. Para continuarmos a avaliação de outros fatores sobre o desempenho da RSSF, apresentamos a seguir a comparação entre as duas abordagens para o MHS-I consideradas nesse trabalho, `SimInvert_MHS-I` e `Sim_MHS-I`, as quais resolvem o

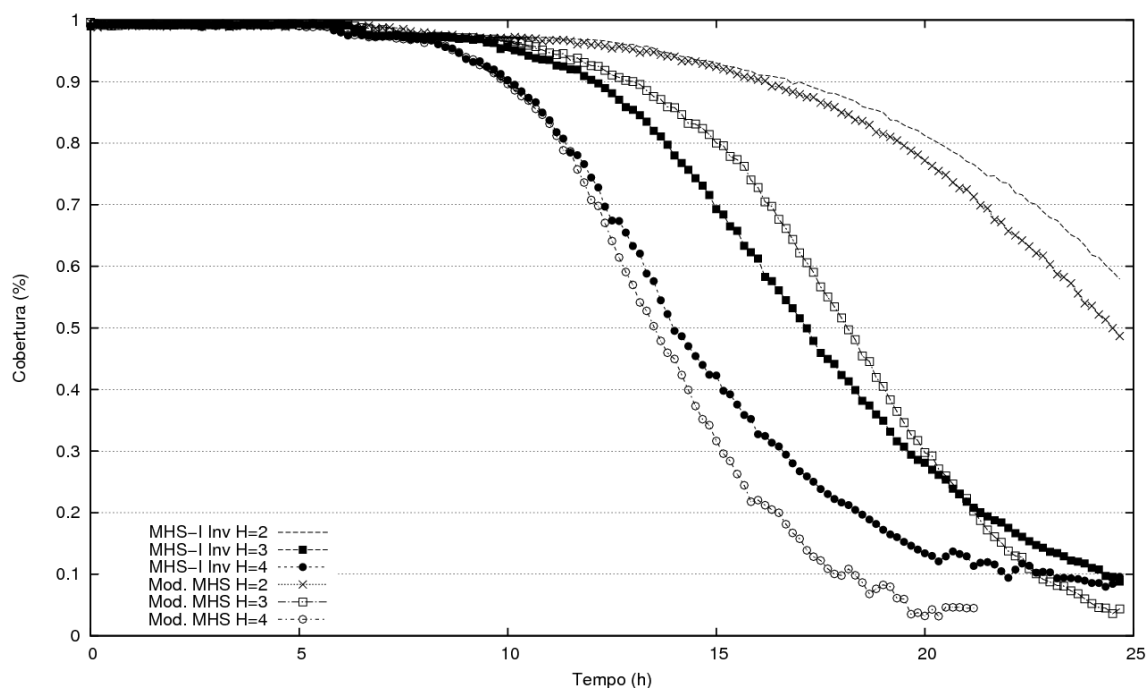


Figura 5.7: SimInvert\_MHS-I: Cobertura em função do tempo de simulação.

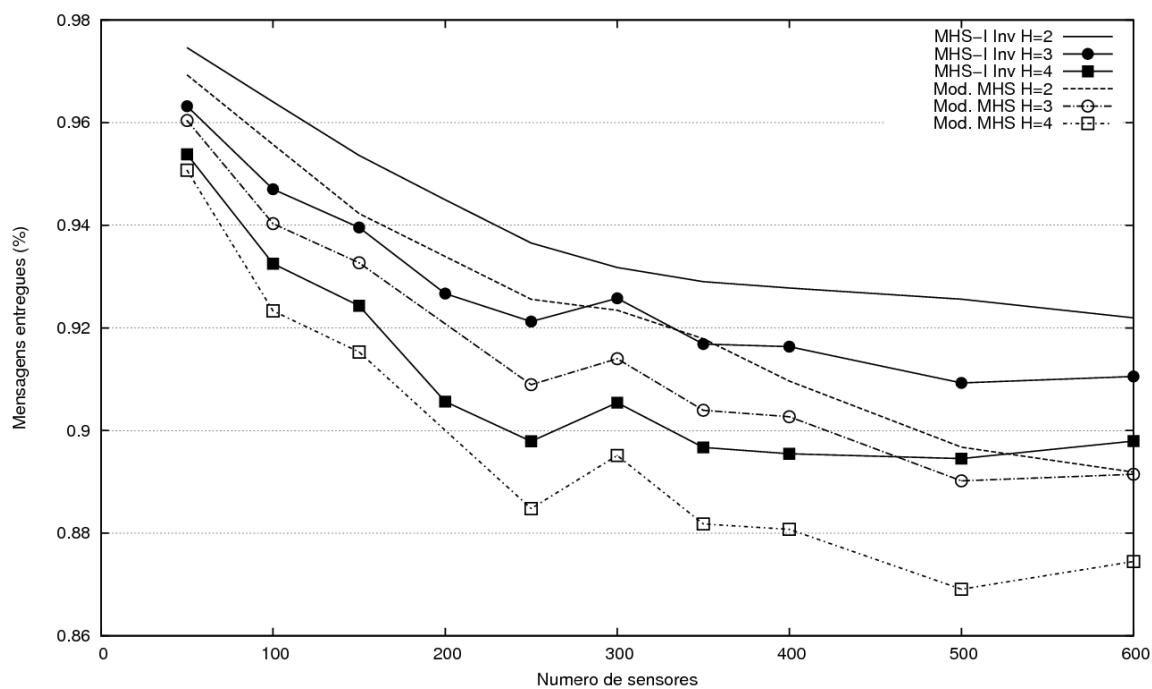


Figura 5.8: SimInvert\_MHS-I: Taxa de mensagens entregues em função do número de nós sensores.

Problema de Controle de Densidade (PCD) em momentos distintos em um período de simulação.

### 5.7.2.1 Impacto do momento da resolução do PCD

Na simulação `Sim_MHS-I`, o PCD é resolvido antes dos problemas de roteamento e agrupamento. Como explicado anteriormente, isso implica que menos nós sensores são utilizados para retransmitir mensagens pela rede. Dessa forma, esperamos que o gasto de energia da simulação `Sim_MHS-I` seja menor do que o gasto apresentado pela simulação `SimInvert_MHS-I`.

De fato, o gráfico da Figura 5.9 mostra que a cobertura da rede foi bem maior para a simulação `Sim_MHS-I`. A cobertura ao final de 25 horas de tempo de simulação do `Sim_MHS-I` foi quase igual à cobertura do `SimInvert_MHS-I` para  $H = 2$ , a despeito do maior consumo de energia de comunicação esperado no caso  $H = 4$ .

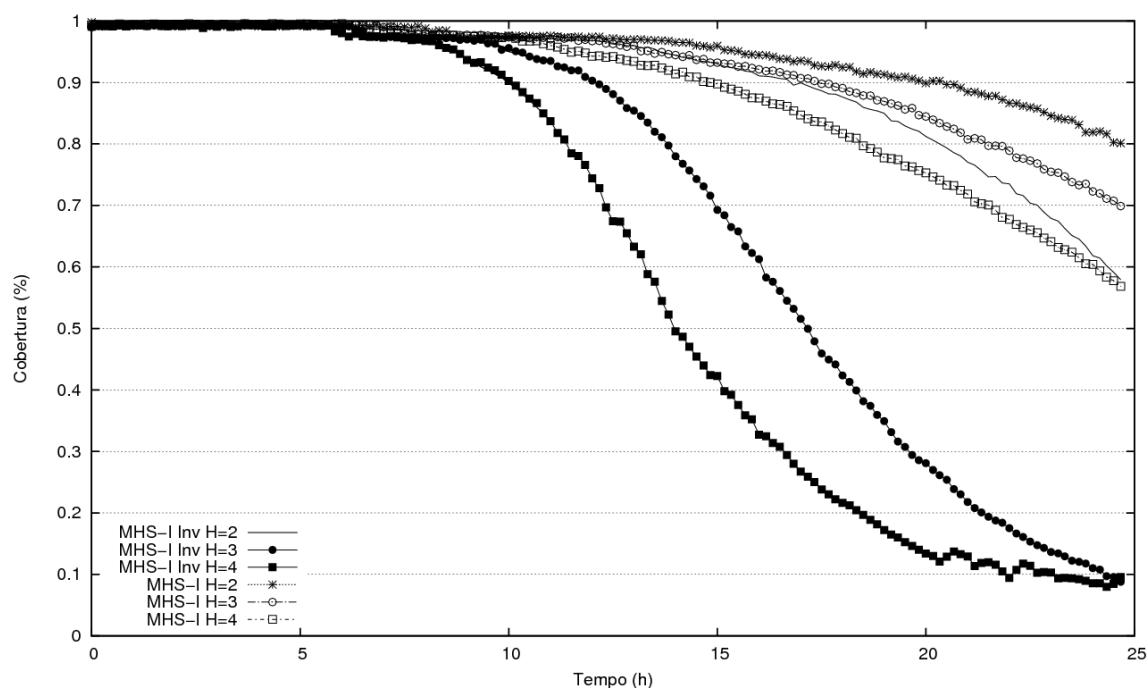


Figura 5.9: As duas abordagens para o MHS-I: Cobertura em função do tempo de simulação.

Os resultados da Figura 5.9 evidenciam a importância da racionalização do gasto de energia do rádio para o aumento da cobertura e o despêndio de energia total da rede. Afinal, menos nós sensores estão sendo utilizados para a comunicação entre os

elementos da rede. O impacto observado de energia residual segue o mesmo padrão de comportamento, e não são mostrados aqui para maior simplicidade do texto.

Outro resultado interessante sobre o impacto da inversão do momento em que o PCD é resolvido é apresentado na Figura 5.10. O atraso na entrega de mensagens da simulação `Sim_MHS-I` é sistematicamente menor do que o atraso da simulação `SimInvert_MHS-I`, chegando a uma diferença de 33% para  $H = 3$ . Isso se deve ao fato de que as árvores de comunicação não precisam incluir nós sensores desnecessários para o desempenho das funções da RSSF, diminuindo ainda mais a rota do sorvedouro, pois não precisa alcançar certos pontos da rede.

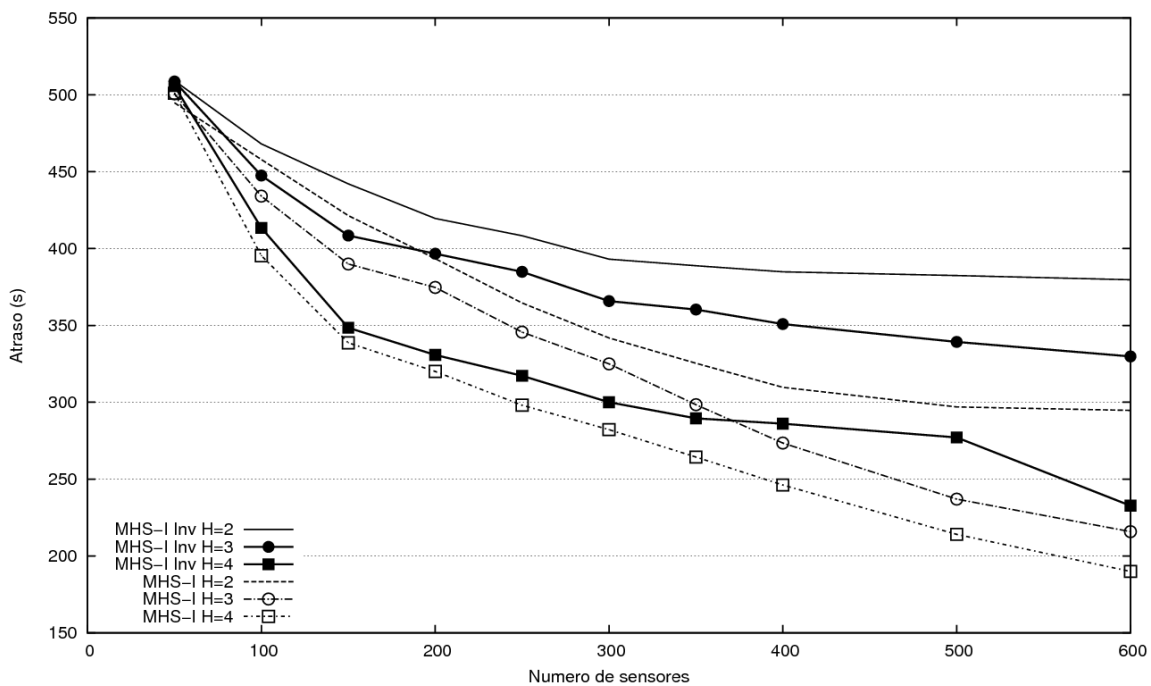


Figura 5.10: MHS-I: Atraso em função do número de nós sensores.

A confiabilidade da RSSF, medida pela taxa de mensagens entregues, também aumentou sensivelmente, como podemos observar pela Figura 5.11.

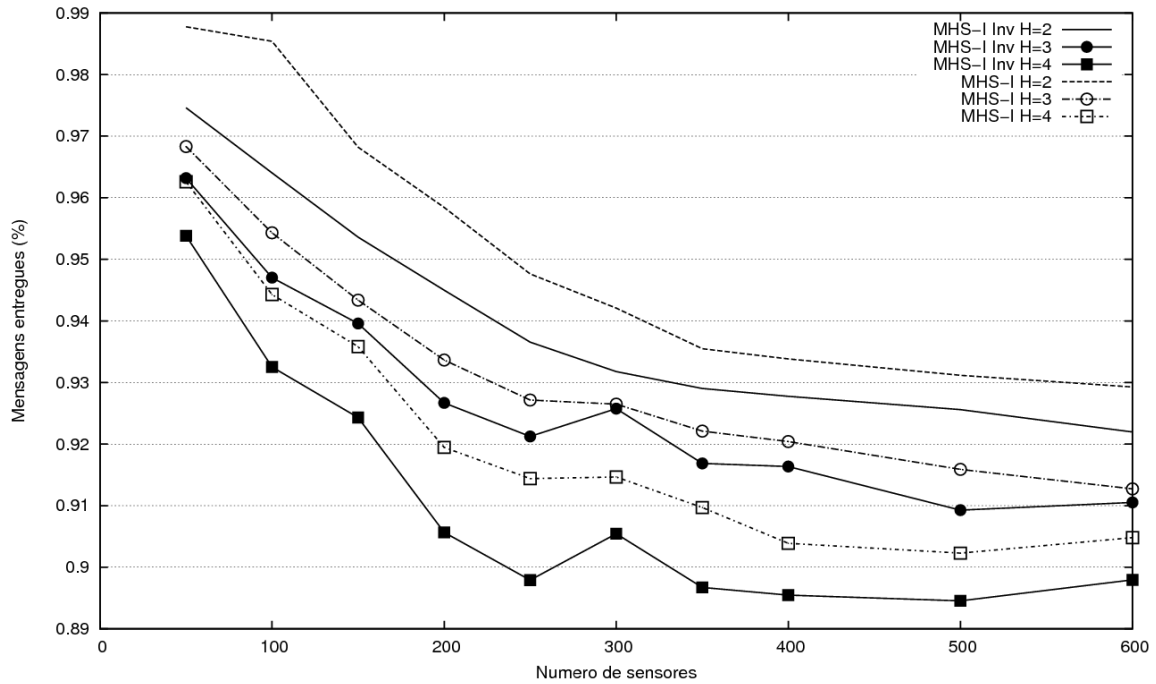


Figura 5.11: MHS-I: Taxa de mensagens entregues em função do número de nós sensores.

Os resultados apresentados nessa seção validam a nossa proposta de resolver o PCD antes do PFMR. Assim, deste ponto em diante, essa será a nossa estratégia para abordar o PCD, dado que suas vantagens são bastante claras.

Como nos resultados apresentados para a simulação *Sim\_MHS-I*, o gasto de energia não se mostrou grande o suficiente para exaurir os recursos da RSSF, deste ponto em diante redefinimos a energia inicial dos nós sensores como 5 mAh. Queremos com isso iniciar a simulação em um ponto onde os nós sensores têm estado de energia mais baixo, adquirido após um tempo de operação da RSSF. Poderemos assim avaliar melhor as abordagens (com menor tempo de computação), permitindo nos concentrar na análise do comportamento da rede quando os nós têm menos energia.

### 5.7.3 HRFI-1

Apresentamos nessa sessão a comparação dos resultados associados ao HRFI-1 e à simulação *Sim\_MHS-I*. Como dito anteriormente, o HRFI-1 pode ser visto como um MHS-I em que pode-se controlar o tamanho da rota do nó sorvedouro. Realizamos uma bateria de testes em que o valor de  $D_{max}$  é igual ao limite superior do comprimento das rotas calculadas pelo MHS-I (obtido para cada valor de  $H$  e para cada número de

nós sensores na rede).

Para essa bateria de testes, os resultados referentes a todas as medidas foi bastante similar para as duas abordagens. A maior diferença se deu apenas no atraso na entrega de mensagens. Entretanto, essa diferença foi pouco sensível, não chegando nem mesmo a 5%. Esses fatos sugerem que as abordagens HRFI-1 e MHS-I são de fato bastante similares, se utilizarmos um valor de  $D_{max}$  obtido através do MHS-I.

Com esta comparação, pretendemos também avaliar o impacto que a frequência de comunicação entre os elementos da rede apresenta sobre o gasto de energia total da RSSF. Dessa forma, realizamos outra bateria de testes, na qual o valor de  $D_{max}$  foi acrescido de 100 metros, em relação ao valor desse parâmetro na bateria de testes anterior. Esperamos com isso diminuir o gasto de energia total da rede, ao diminuir a frequência de comunicação entre os elementos da rede através de uma rota mais longa do nó sorvedouro.

As Figuras 5.12 e 5.13 apresentam a energia residual total e a cobertura da rede em função do tempo de simulação, para o valor de  $D_{max}$  aumentado. De fato, o aumento do valor do parâmetro  $D_{max}$  culminou em menor gasto energético total da rede e melhor cobertura. Os valores para a cobertura, por exemplo, mostraram-se 14%, 23% e 28% maiores para  $H = 2$ ,  $H = 3$  e  $H = 4$ , respectivamente, após 25 horas de simulação.

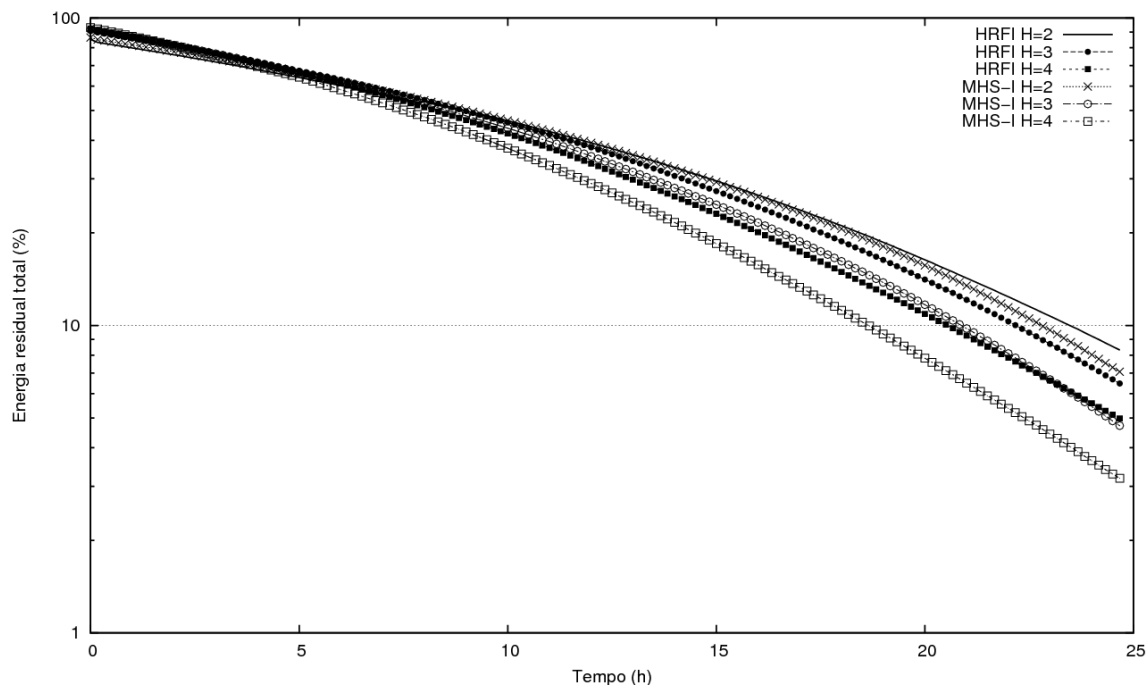


Figura 5.12: Aumentando o  $D_{max}$ : Energia residual total dos nós sensores.



Os resultados de cobertura e energia residual também evidenciam que a diferença entre os gastos de energia para  $H = 2$ ,  $H = 3$  e  $H = 4$  diminui quando resolvemos o PCD antes do PFMR. O controle de densidade feito dessa maneira diminui a necessidade de utilizar nós sensores que não estão monitorando o ambiente para retransmissão de mensagens. Dessa forma, valores maiores de  $H$  apresentam economia de energia maior, ao poupar tais nós sensores de retransmitir mensagens pelos vários enlaces da RSSF. Isto é justificado pelo fato de, quanto maior o valor de  $H$ , maior o número de filhos que os nós sensores podem ter, na média. Por esse motivo, a diferença do gasto energético para valores diferentes de  $H$  diminui.

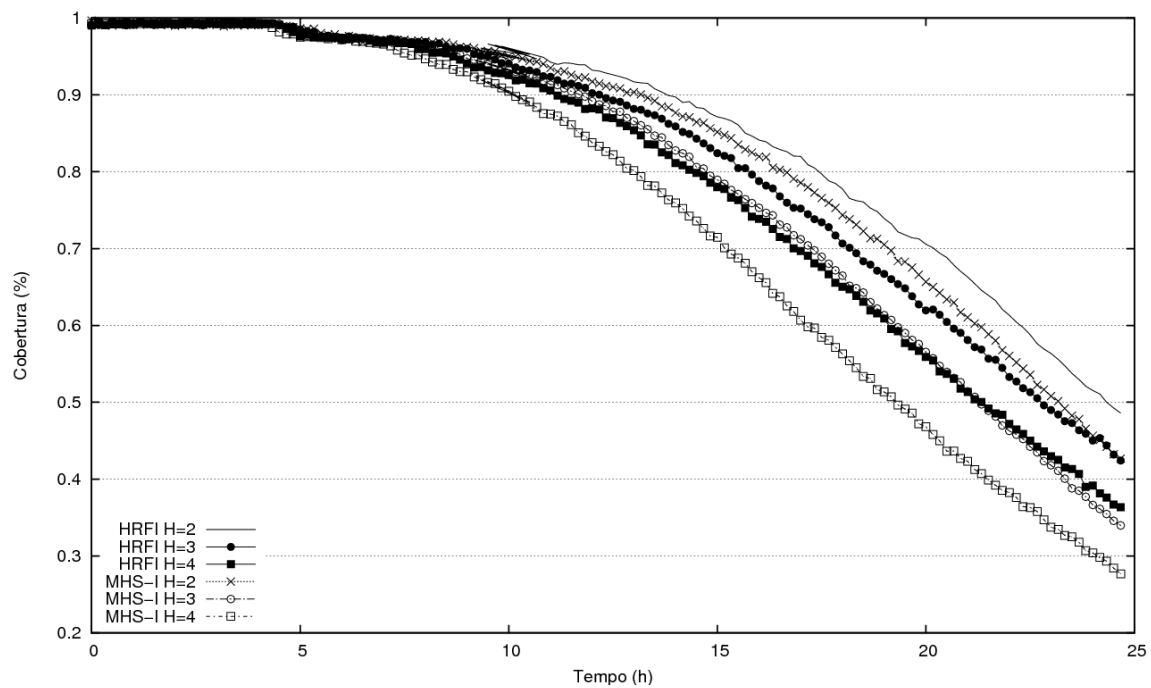


Figura 5.13: Aumentando o  $D_{max}$ : Cobertura em função do tempo de simulação.

A Figura 5.14 ilustra o impacto no atraso na entrega de mensagens provocado pelo aumento do parâmetro  $D_{max}$ . Vemos que, para redes mais esparsas, a diferença no atraso é bem pequena, e vai aumentando na medida em que a densidade da rede aumenta. Para 600 nós, o atraso aumentou em média 75 segundos para  $H = 2$ ,  $H = 3$  e  $H = 4$ . Isso significa que, para os parâmetros utilizados em nossos experimentos, um aumento de  $d$  metros no valor de  $D_{max}$  culmina em um aumento de  $0,75d$  segundos no atraso na entrega de mensagens para redes mais densas, quando temos apenas um sorvedouro móvel na RSSF.

O aumento do parâmetro  $D_{max}$  proporcionou uma melhora sensível na confiabilidade da rede, como podemos ver pela gráfico da Figura 5.15. Como a rota do

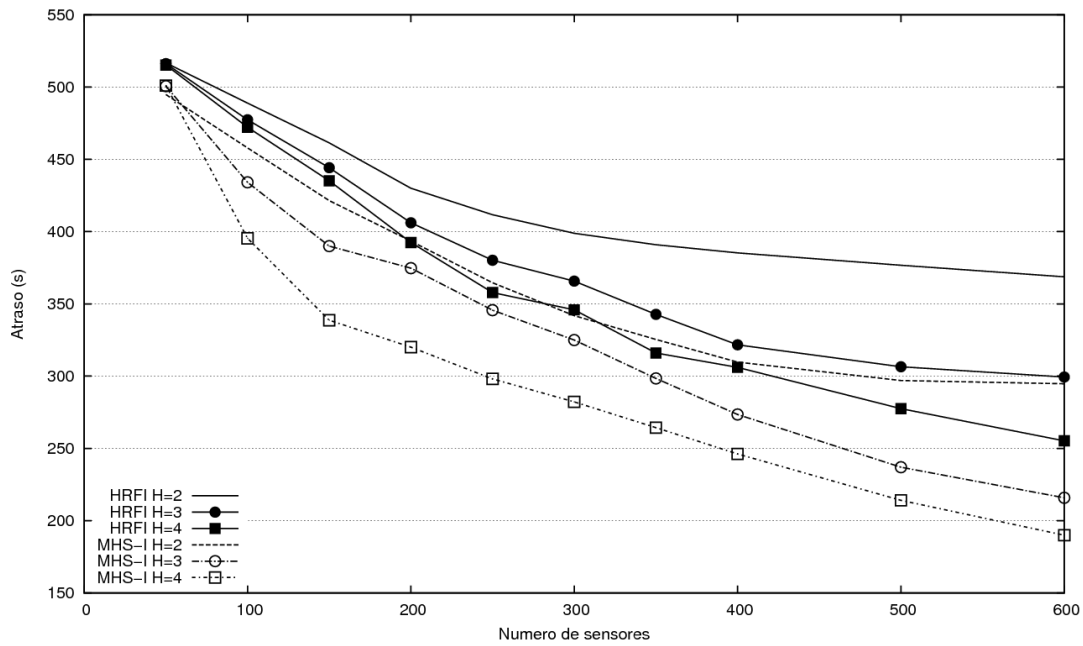


Figura 5.14: Aumentando o  $D_{max}$ : Atraso na entrega de mensagens.

sorvedouro ficou um pouco maior, ela provavelmente inclui mais líderes de grupos do que rotas menores. O resultado disso é a presença de mais árvores de comunicação na rede, cada qual contendo menos sensores, diminuindo a probabilidade de perda das mensagens retransmitidas pela árvore, ao diminuir o número de colisões.

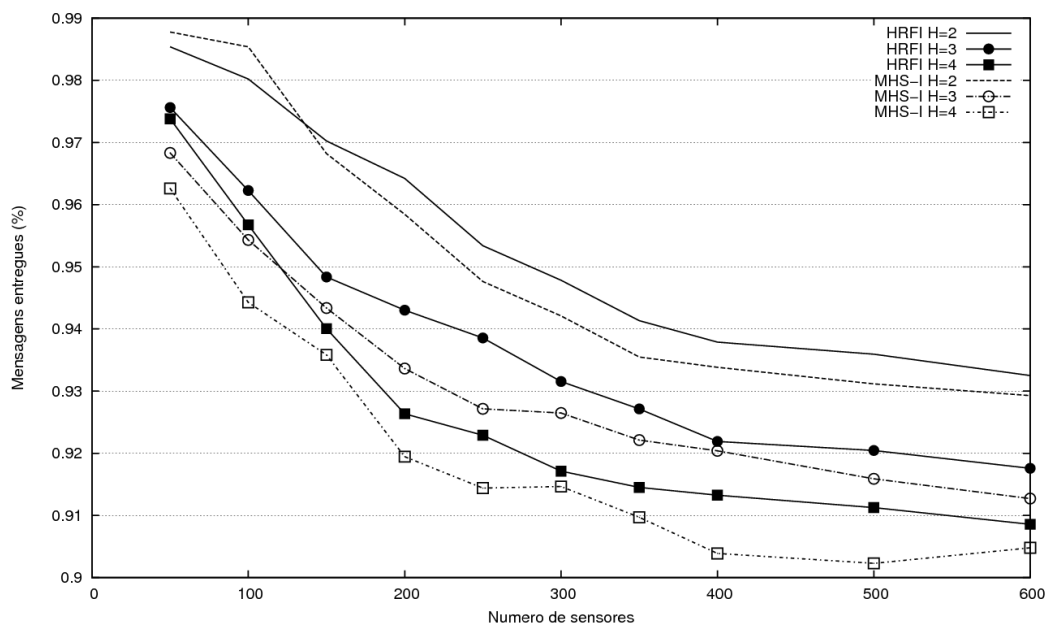


Figura 5.15: Aumentando o  $D_{max}$ : Taxa de mensagens entregues.

Os resultados apresentados nessa seção evidenciam qual o comportamento da rede esperado, em relação às medidas que utilizamos para avaliar a simulação de RSSFs, quando o valor de  $D_{max}$  é alterado. Mais precisamente, todos os resultados apresentados até este ponto do trabalho nos ajudam a entender o papel que a limitação do número de saltos  $H$ , o controle de densidade e o comprimento da rota do nó sorvedouro representam no desempenho da RSSF, segundo as medidas avaliadas.

A abordagem integrada para resolver os problemas de agrupamento e roteamento se mostrou efetiva para melhorar o atraso na entrega de mensagens, ao permitir rotas de menor comprimento para o sorvedouro. Já a minimização do custo da floresta de comunicação representou uma boa melhora no gasto de energia, tempo de vida e cobertura da rede. Entretanto, o que realmente representou uma grande economia de energia foi a mudança do momento em que o controle de densidade é executado em um determinado período de simulação.

Na próxima seção damos continuidade à avaliação da nossa abordagem, ao apresentar alguns resultados obtidos quando mais de um nó sorvedouro são empregados na RSSF. De acordo com o que apresentamos até o momento, é de se esperar, por exemplo, que quanto mais nós sorvedouros estiverem presentes em uma RSSFs, menor será o atraso na entrega de mensagens, ao custo de maior dispêndio de energia.

#### 5.7.4 Múltiplos sorvedouros

Apresentamos nessa seção resultados computacionais referentes à simulação de RSSFs com  $K = 2$  e  $K = 3$  nós sorvedouros móveis. O parâmetro  $D_{max}$  foi definido como o menor valor que conseguimos obter para cada valor  $H = 2, 3, 4$ , para cada número de nós sensores presentes na rede. Com isso queremos verificar, principalmente, a melhora que a abordagem com mais de um sorvedouro móvel representa para o atraso na entrega de mensagens, e como o gasto de energia e cobertura se comportam com esse novo cenário.

A Figura 5.16 apresenta o atraso na entrega de mensagens para  $K = 1, 2, 3$  nós sorvedouros, separados por cada valor de limite de saltos  $H$  testado. Os resultados para  $K = 1$  nó sorvedouro referem-se aos parâmetros da simulação com o HRFI-1 citados no início da seção 5.7.3.

Os três gráficos da Figura 5.16 apresentam comportamento bastante similar. Redes com  $K = 2$  sorvedouros obtiveram um atraso cerca de 75 segundos menor do que redes com um sorvedouro apenas, para todos os valores de  $H$  testados. Similarmente, o atraso foi cerca de 125 segundos menor em redes com  $K = 3$  sorvedouros, quando

comparadas com redes com apenas um nó sorvedouro.

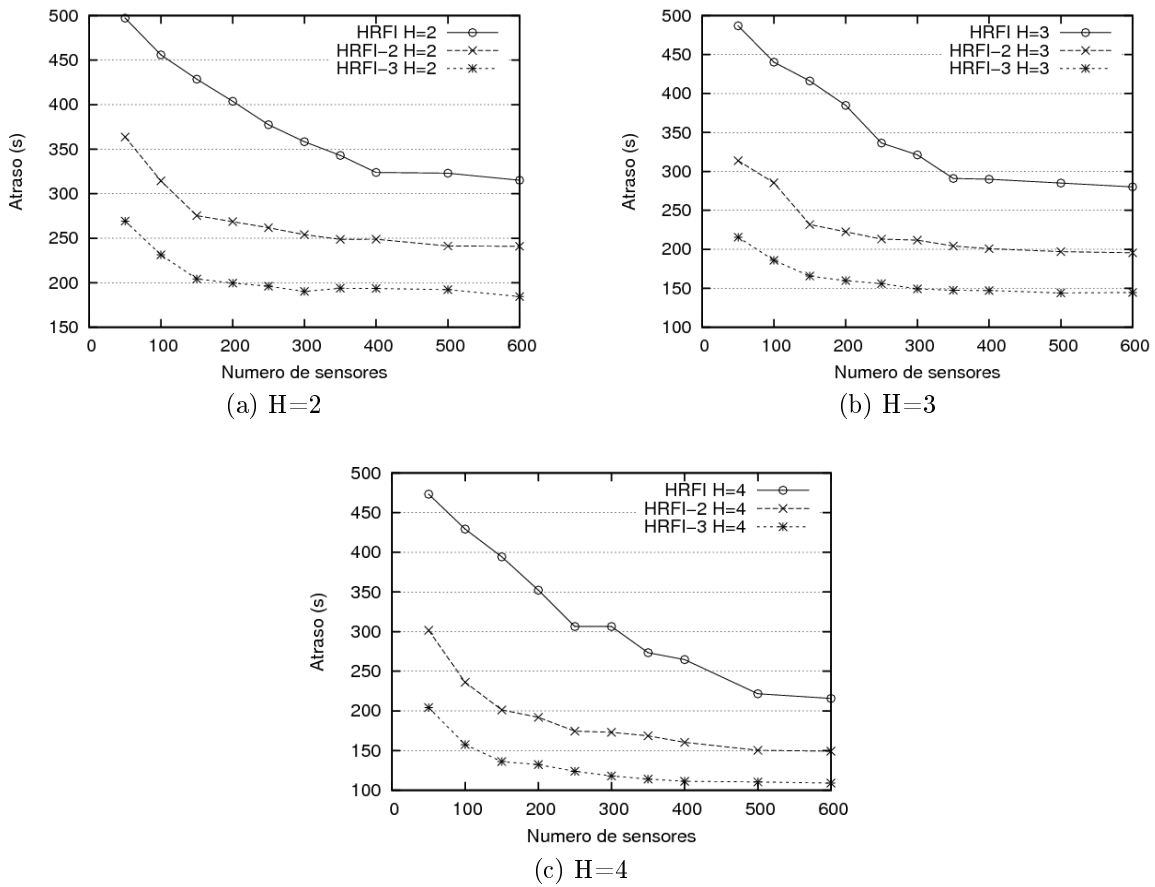


Figura 5.16: Múltiplos sorvedouros: Atraso na entrega de mensagens.

Apesar destes valores de melhora no atraso (em segundos) serem quase equivalentes em termos absolutos, em termos relativos, o percentual de melhora foi aumentando bastante na medida em que o valor de  $H$  aumentou. A Figura 5.17, que apresenta a cobertura real da rede para  $K = 1, 2, 3$ , explicita esse fato. Repare que, quanto maior o valor de  $H$ , maior a diferença obtida com a cobertura da rede, com o aumento do número de nós sorvedouros. O percentual de energia residual também apresentou comportamento semelhante, e não é aqui apresentada para maior simplicidade do texto.

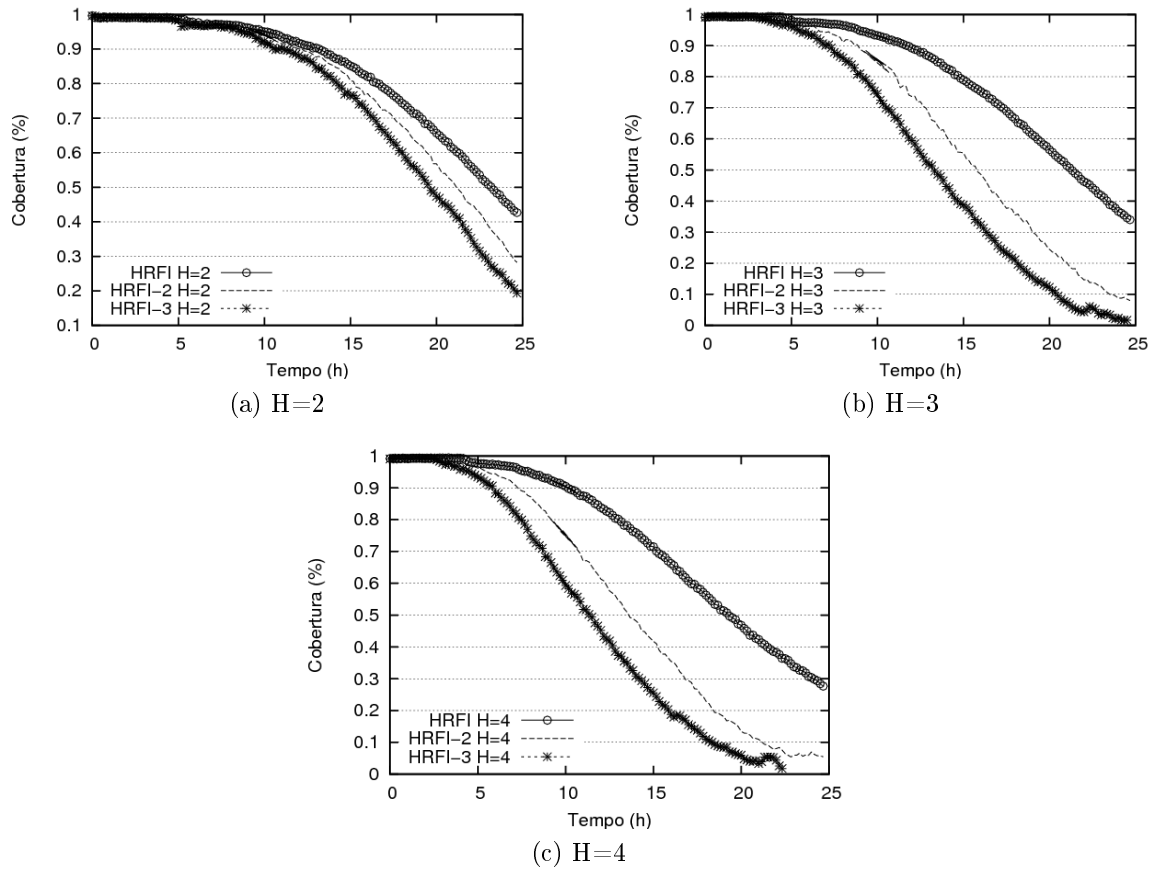


Figura 5.17: Múltiplos sorvedouros: Cobertura em função do tempo de simulação.

A taxa de mensagens entregues apresentou uma sensível melhora com o aumento do número de nós sorvedouros na rede. Quanto mais sorvedouros presentes na rede, menos elementos têm as árvores de comunicação (apesar de, em geral, manterem a mesma profundidade de acordo com o valor de  $H$ ) e, portanto, menores são as chances de as mensagens se perderem ao serem retransmitidas. Na Figura 5.18, apresentamos a taxa de mensagens entregues para  $H = 3$  e  $K = 1, 2, 3$ . Os resultados obtidos para  $H = 2$  e  $H = 4$ , não ilustrados neste texto, apresentam comportamento semelhante.

Os resultados apresentados para RSSFs com mais de um sorvedouro móvel mostraram que pode ser obtida uma melhora significativa no atraso na entrega de mensagens com a introdução de mais sorvedouros móveis na RSSF, caso essa seja a intenção do projetista da rede. Entretanto, tal melhora resulta em maior gasto de energia, como apresentado ao longo desse texto. Dependendo das intenções do projetista quanto ao atraso desejado para a rede, o valor de  $D_{max}$  pode ser aumentado, resultando em economia de energia, como mostrado na seção 5.7.3. Nesse sentido, redes com múltiplos sorvedouros apresentam duas principais vantagens. A primeira é a maior

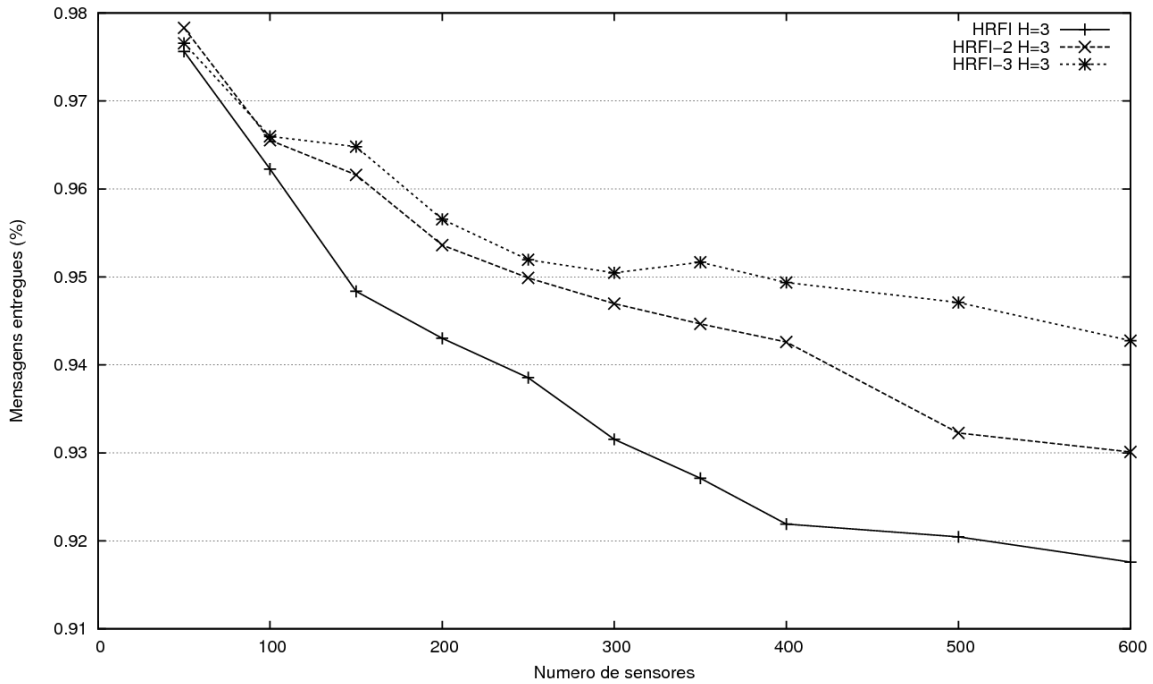


Figura 5.18: Múltiplos sorvedouros: Taxa de mensagens entregues em função do número de nós sensores.

flexibilidade para a escolha do parâmetro  $D_{max}$ , dado que é possível obter menores rotas quando há mais de um nó sorvedouro na rede. A outra vantagem é o fato de as árvores de coleta de dados possuírem menos elementos, melhorando a confiabilidade da rede ao reduzir a probabilidade de colisão durante a retransmissão de mensagens pelos nós sensores.

# Capítulo 6

## Considerações finais

### 6.1 Conclusões

Neste trabalho foram apresentadas abordagens para diminuir o consumo de energia em Redes de Sensores Sem Fio, utilizando um ou mais sorvedouros móveis. Nessas abordagens, o gasto de energia é considerado por meio da atribuição de pesos, que representam a energia gasta com a transmissão de uma mensagem de tamanho fixo, aos enlaces das árvores de comunicação. Os problemas de roteamento e agrupamento de nós sensores são tratados de forma integrada, permitindo controlar o comprimento da rota do(s) sorvedouro(s) e, ao mesmo tempo, minimizar o custo total das árvores de coleta de dados.

A minimização do consumo de energia em RSSFs com sorvedouros móveis deu origem a um novo Problema de Otimização Combinatória, chamado Problema da Floresta Mínima com Distância Restrita entre as Raízes (PFMR). Apresentamos uma formulação de Programação Inteira Mista para o PFMR, e um algoritmo *Branch-and-bound* (BB), baseado na formulação proposta, para resolver o problema aqui estudado. Entretanto, o BB não se mostrou adequado para resolver o problema, tornando-se impraticável com o aumento do tamanho das instâncias de teste utilizadas neste trabalho. Com isso, uma abordagem heurística para o PFMR foi desenvolvida, a fim de prover um método escalável que possa ser utilizado dentro do contexto dinâmico das RSSFs.

Utilizamos o método de simulação de RSSFs para avaliar as abordagens aqui propostas. Primeiramente, a fim de avaliar nossa proposta em relação a trabalhos presentes na literatura (sob as mesmas condições de simulação), desenvolvemos e testamos uma heurística cujo objetivo é o mesmo objetivo dos trabalhos anteriores aqui consid-

erados: minimizar o comprimento da rota do nó sorvedouro. Em seguida, foi realizada uma extensiva avaliação experimental de todas as abordagens aqui propostas.

Os resultados obtidos indicam que considerar o gasto de energia na minimização das árvores de comunicação é vantajoso, tanto para diminuir o consumo de energia quanto para melhorar outros parâmetros de QoS da rede, como taxa de mensagens entregues. Os resultados mostram também que o comprimento das rotas dos nós sorvedouros pode ser controlado de forma a aumentar ou diminuir a frequência de comunicação entre os nós sensores. Disso resulta maior ou menor consumo da energia total da RSSF, ao impactar diretamente a atividade do rádio dos nós sensores.

Neste trabalho também evidenciamos a importância do mecanismo de controle de densidade em RSSFs. Apresentamos o impacto em vários parâmetros de QoS quando tal mecanismo é executado em momentos distintos da simulação.

Finalmente, verificamos que usar mais de um sorvedouro móvel na rede é uma estratégia vantajosa. Ao permitir menores rotas para os nós sorvedouros, tal estratégia provê mais flexibilidade para controlar o atraso na entrega de mensagens e, consequentemente, o consumo de energia da RSSF.

## 6.2 Trabalhos futuros

O problema aqui estudado apresenta oportunidades em várias linhas de pesquisa. Para a modelagem do PFMR, podem ser investigadas outras formulações que nos permitiriam obter algoritmos BB mais efetivos. Já para a abordagem exata deste problema, podem ser explorados outros algoritmos exatos capazes de melhorar os limites inferiores e superiores obtidos, dentro de um espaço de tempo computacional razoável.

Já para a simulação de RSSFs, vários trabalhos podem ser propostos para melhorar os parâmetros de QoS avaliados neste trabalho. Para diminuir o consumo de energia, podem ser estudadas estratégias distribuídas para o mecanismo de controle de potência de transmissão. O gasto de energia com recepção de mensagens também pode passar a ser considerado nos enlaces das árvores de comunicação, o que ainda não foi feito. Além disso, podem ser desenvolvidas outras abordagens heurísticas para obter melhores soluções para o PFMR.

Para melhorar a cobertura da rede, além de buscar diminuir o gasto de energia, pode ser utilizada uma abordagem *on line* para resolver o Problema de Controle de Densidade. Nesse caso, ao invés de o PCD ser resolvido e controlado de forma centralizada (pelos nós sorvedouros), tal problema seria resolvido de forma dinâmica pelos



próprios nós sensores. Isso evitaria perdas momentâneas de cobertura. Outra opção seria permitir que os nós sorvedouros auxiliares resolvam o PCD, e não apenas o nó sorvedouro principal. Isso incluiria estudar outras formas de distribuição de tarefas entre os nós sorvedouros.

Para melhorar o atraso na entrega de mensagens, podem ser utilizadas outras abordagens para tratar as rotas dos sorvedouros móveis, como, por exemplo, métodos em que cada nó sorvedouro tem um ponto de partida/chegada distinto. Nesse caso, cada um poderia ser responsável por uma área da RSSF. A forma de comunicação entre os sorvedouros também poderia ser tratada, de forma que eles possam se comunicar e tomar decisões sem a necessidade de definir um sorvedouro principal ou um ponto de encontro entre eles. Isso resultaria em melhora não apenas no atraso na entrega de mensagens, como em vários outros parâmetros de QoS da RSSF.



# Referências Bibliográficas

- Aioffi, W. M. (2007). Métodos Integrados para Organização de Redes de Sensores sem Fio com Sink móvel e Controle de Densidade. Master's thesis, UFMG.
- Aioffi, W. M.; Quintão, F. P. & Mateus, G. R. (2007a). Integrated Methods for Organization of Wireless Sensor Networks with Mobile Sink. In *International Workshop on Design and Reliable Communication Networks*, La Rochelle, France.
- Aioffi, W. M.; Quintão, F. P. & Mateus, G. R. (2007b). Optimization issues and Algorithms for Wireless Sensor Networks with Mobile Sink. In *International Network Optimization Conference*, Spa, Belgium.
- Bajaj, L.; Takai, M.; Ahuja, R.; Tang, K.; Bagrodia, R. & Gerla, M. (1999). GloMoSim: A Scalable Network Simulation Environment. Technical Report Technical Report 990027, UCLA Computer Science Department.
- Balanis, C. A. (2003). *Antenna Theory*. John Wiley and Sons Inc.
- Barr, R.; Haas, Z. J. & van Renesse, R. (2005). *Scalable Wireless Ad hoc Network Simulation*, chapter 19, pp. 297--311. CRC Press.
- Berkeley Sensor and Actuator Center (2009). Cots Dust, Large Scale Models for Smart Dust. Fonte: . [http://siliconrobot.com/macro\\_motes/macromotes.html](http://siliconrobot.com/macro_motes/macromotes.html).
- Bilò, V.; Caragiannis, I.; Kaklamanis, C. & Kanellopoulos, P. (2005). Geometric Clustering to Minimize the Sum of Cluster Sizes. In *13th Annual European Symposium on Algorithms (ESA 05)*, pp. 460--471. LNCS 3669, Springer.
- Bryant A. Julstrom (1999). Coding TSP Tours as Permutations Via an Insertion Heuristic. In *SAC '99: Proceedings of the 1999 ACM Symposium on Applied Computing*, pp. 297--301, San Antonio, Texas, United States. ACM Press.

- Bulusu, N.; Heidemann, J. & Estrin, D. (2000). GPS-Less Low Cost Outdoor Location for Very Small Devices. *IEEE Personal Communication*, 7(5):28--34.
- Cardei, M.; Maccallum, D.; Cheng, M.; Min, M.; Jia, X.; Li, D. & Du, D. (2002a). Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3(3-4):213--229.
- Cardei, M.; MacCallum, D.; Cheng, M. X.; Min, M.; Jia, X.; Li, D. & Du, D.-Z. (2002b). Wireless Sensor Networks with Energy Efficient Organization. *Journal of Interconnection Networks*, 3(3-4):213--229.
- CDT - Eclipse C/C++ Development Tooling (2009). <http://www.eclipse.org/cdt/>.
- Chakrabarti, A.; Sabharwal, A. & Aazhang, B. (2003). Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. In *The 2nd Intl. Workshop on Information Processing in Sensor Networks (IPSN)*.
- Correia, L. H. A.; Macedo, D. F.; Silva, D. A. C.; dos Santos, A. L.; Loureiro, A. A. F. & Nogueira, J. M. S. (2005). Transmission power control in MAC protocols for Wireless Sensor Networks. In *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 282--289, New York, NY, USA. ACM Press.
- CotsBots (2006). <http://www-bsac.eecs.berkeley.edu/projects/cotsbots/>.
- Croes, G. A. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6):791--812.
- Dantzig, G.; Fulkerson, D. & Johnson, S. (1954). Solution of a large scale traveling salesman problem. *Operations Research*, 2:393--410.
- de Oliveira, H.; Nakamura, E.; Loureiro, A. & Boukerche, A. (2005). Directed position estimation: a recursive localization approach for wireless sensor networks. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pp. 557--562.
- de Oliveira, H. C. B.; Vasconcelos, G. C.; Alvarenga, G. B.; Mesquita, R. V. & Souza, M. M. (2007). A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis. In *SCIS '07: IEEE Symposium on Computational Intelligence in Scheduling*, pp. 198--205.

- Ekici, E.; Gu, Y. & Bozdogan, D. (2006). Mobility-based communication in wireless sensor networks. *Communications Magazine, IEEE*, 44(7):56–62.
- Feo, T. & Resende, M. (1995). Greedy randomized adaptive search procedures. *J. of Global Optimization*, 6:109--133.
- Figueiredo, C.; Nakamura, E. & Loureiro, A. (2004). Protocolo Adaptativo Híbrido para a Disseminação de Dados em Redes de Sensores Sem Fio Auto-Organizáveis. *22o Simpósio Brasileiro de Redes de Computadores*, 1:43--56.
- Gandham, S.; Dawande, M.; Prakash, R. & Venkatesan, S. (2003). Energy Efficient Schemes for wireless Sensor Networks With Multiple Mobile Base Stations. In *IEEE GLOBECOM*, San Francisco, USA.
- Garey, M. R. & Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- Gomez, J. & Campbell, A. (2004). A case for variable-range transmission power control in wireless multihop networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pp. 1425–1436 vol.2.
- Gouveia, L. (1996). Multicommodity flow models for Spanning Trees with Hop Constraints. *European Journal of Operational Research*, 95:178--190.
- Gouveia, L.; Paias, A. & Sharma, D. (2007). Local search heuristics for the hop-constrained minimum spanning tree problem. In *International Network Optimization Conference*.
- Heinzelman, W.; Chandrakasan, A. & Balakrishnan, H. (2002). An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Trans. on Wireless Communications*, 1(4):660--670.
- ILOG Concert Technology (2009). <http://www.ilog.com/products/optimization/tech/concert.cfm>.
- ILOG CPLEX Solver (2009). <http://www.ilog.com/products/cplex>.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley- Interscience, New York, NY.

- Jain, S.; Shah, R.; Brunette, W.; Borriello, G. & Roy, S. (2006). Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. *Mobile Networks and Applications*, 11(3):327--339.
- JiST/SWANS (2007). Java in Simulation Time - Scalable Wireless Ad Hoc Network Simulator. <http://jist.ece.cornell.edu/>.
- Jung, E.-S. & Vaidya, N. H. (2005). A power control mac protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):55--66.
- Khepera-III (2009). <http://www.k-team.com>.
- Kim, H. S.; Abdelzaher, T. F. & Kwon, W. H. (2003). Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 193--204, New York, NY, USA. ACM.
- Kwon, T. J. & Gerla, M. (1999). Clustering with power control. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 2, pp. 1424--1428 vol.2.
- Luo, J. & Hubaux, J.-P. (2005). Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *The 24th IEEE INFOCOM*.
- Meguerdichian, S. & Potkonjak, M. (2003). Low Power 0/1 Coverage and Scheduling Techniques in Sensor Networks. Technical Report 030001, University of California, Los Angeles.
- Meguerdichian, S.; Slijepcevic, S.; Karayan, V. & Potkonjak, M. (2001). Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pp. 106--116, New York, NY, USA. ACM.
- Mhatre, V.; Rosenberg, C.; Kofman, D.; Mazumdar, R. & Shroff, N. B. (2005). A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Trans. Mob. Comput.*, 4(1):4--15.
- MICA2 (2009). MICA2 - Wireless Measurement System. Fonte: . [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf).
- MICA2DOT (2009). MICA2DOT Datasheet. Fonte: . [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2DOT\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf).

- MICAZ (2009). MICAZ Datasheet. Fonte: . [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICAZ\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf).
- Mirchandani, P. & Francis, R. (1990). *Discrete Location Theory*. John Wiley and Sons.
- Mladenović, N. & Hansen, P. (1997). Variable Neighborhood Search. *Comps. in Opns. Res.*, 24:1097--1100.
- Monks, J. P. (2001). *Transmission Power Control For Enhancing The Performance Of Wireless Packet Data Networks*. PhD thesis, Department of Electrical and Computer Engineering, University of Illinois.
- Nakamura, F.; Quintão, F.; Menezes, G. & Mateus, G. (2005). An Optimal Node Scheduling for Flat Wireless Sensor Networks. In *IEEE ICN05*, volume 3420, pp. 475–483.
- ns-2 (2007). ns-2 - The Network Simulator 2. <http://nslam.isi.edu/nslam/index.php>.
- Papadimitriou, I. & Georgiadis, L. (2005). Maximum lifetime routing to mobile sink in wireless sensor networks. In *International Conference on Software, Telecommunications and Computer Networks, SoftCOM*.
- Park, S.; Savvides, A. & Srivastava, M. B. (2001). Simulating Networks of Wireless Sensors. In *33rd Conference on Winter Simulation*, pp. 1330–1338. IEEE Computer Society.
- Polastre, J.; Hill, J. & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 95--107, New York, NY, USA. ACM Press.
- Quintão, F.; Nakamura, F. G. & Mateus, G. R. (2004). A Hybrid Approach to solve the Coverage and Connectivity Problem in Wireless Sensor Networks. In *4th European Workshop on Metaheuristics: Design and Evaluation of Advanced Hybrid Metaheuristics*, Nottingham, UK.
- Rajaraman, R. (2002). Topology control and routing in ad hoc networks: a survey. *SIGACT News*, 33(2):60--73.
- Shah, R.; Roy, S.; Jain, S. & Brunette, W. (2003). Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 30–41.

- Siqueira, I.; Figueiredo, M.; Loureiro, A.; Nogueira, J. & Ruiz, L. (2006). An Integrated Approach for Density Control and Routing in Wireless Sensor Networks. In *Parallel and Distributed Processing Symposium*, pp. 10--19, Greece.
- Small, T. & Haas, Z. J. (2003). The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 233--244, New York, NY, USA. ACM.
- Small, T. & Haas, Z. J. (2005). Resource and performance tradeoffs in delay-tolerant wireless networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 260--267, New York, NY, USA. ACM.
- Tirta, Y.; Li, Z.; Lu, Y.-H. & Bagchi, S. (2004). Efficient collection of sensor data in remote fields using mobile collectors. In *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pp. 515--519.
- Tong, L.; Zhao, Q. & Adireddy, S. (2003). Sensor networks with mobile agents. In *Military Communications Conference, 2003. MILCOM 2003. IEEE*, volume 1, pp. 688--693 Vol.1.
- Venkatasubramaniam, P.; Adireddy, S. & Tong, L. (2004). Sensor Networks with Mobile Agents: Optimal Random Access and Coding. *IEEE Journal on Sel. Areas in Comm.: Special Issue on Sensor Networks*.
- Wang, W.; Srinivasan, V. & Chua, K. (2005a). Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. In *MobiCom '05*, pp. 270--283, New York, NY, USA. ACM Press.
- Wang, W.; Srinivasan, V. & Chua, K.-C. (2008). Extending the Lifetime of Wireless Sensor Networks Through Mobile Relays. *Networking, IEEE/ACM Transactions on*, 16(5):1108--1120.
- Wang, Z. M.; Basagni, S.; Melachrinoudis, E. & Petrioli, C. (2005b). Exploiting sink mobility for maximizing sensor networks lifetime. In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, p. 287.1, Washington, DC, USA. IEEE Computer Society.
- Wifly (2007). Virtual coordinates, mobile sink and R/C planes. <http://www.citi.insa-lyon.fr/research/sensors/wifly>.



- Ye, F.; Zhong, G.; Lu, S. & Zhang, L. (2002). Peas: A robust energy conserving protocol for long-lived sensor networks. In *ICNP '02: Proc. of the 10th IEEE Intl. Conf. on Network Protocols*, pp. 200--201, Washington, DC, USA. IEEE Computer Society.
- Ye, J. (2006). A Scalable Sensor Localization Algorithm based on SDP subproblems. In *International Symposium on Mathematical Programming*, p. 125.
- Zhang, H. & Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Intl. Journal of Wireless Ad Hoc and Sensor Networks*, 1(1-2):89–124.

