

INTEGRAÇÃO DO CONTROLE DE DENSIDADE,
COBERTURA E ROTEAMENTO EM REDES DE
SENSORES SEM FIO:
MODELOS MATEMÁTICOS, ALGORITMOS DE
OTIMIZAÇÃO E SIMULAÇÃO

IURI BUENO DRUMOND DE ANDRADE

INTEGRAÇÃO DO CONTROLE DE DENSIDADE,
COBERTURA E ROTEAMENTO EM REDES DE
SENSORES SEM FIO:
MODELOS MATEMÁTICOS, ALGORITMOS DE
OTIMIZAÇÃO E SIMULAÇÃO

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
COORIENTADOR: GERALDO ROBSON MATEUS

Belo Horizonte

Agosto de 2011

© 2011, Iuri Bueno Drumond de Andrade.
Todos os direitos reservados.

A553i Andrade, Iuri Bueno Drumond de
Integração do Controle de Densidade, Cobertura e
Roteamento em Redes de Sensores Sem Fio: Modelos
Matemáticos, Algoritmos de Otimização e Simulação /
Iuri Bueno Drumond de Andrade. — Belo Horizonte,
2011

xx, 82 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Alexandre Salles da Cunha

Coorientador: Geraldo Robson Mateus

1. Computação - Teses. 2. Otimização Combinatória
- Teses. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*61 (043)



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

FOLHA DE APROVAÇÃO

Integração do controle de densidade, cobertura e roteamento em redes de sensores sem fio: modelos matemáticos, algoritmos de otimização e simulação

IURI BUENO DRUMOND DE ANDRADE

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. CARLOS ROBERTO VENÂNCIO DE CARVALHO
Departamento de Engenharia de Produção - UFMG

PROF. LUIZ FILIPE MENEZES VIEIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 04 de agosto de 2011.

Agradecimentos

Gostaria de agradecer primeiramente a Deus por me dar coragem para enfrentar meus medos. Aos meus pais, Celso e Vânia, e irmãos, Taigrom e Haron, por sempre estarem ao meu lado nos momentos em que mais precisei. Agradeço-os também por sempre acreditar que eu seria capaz de chegar até aqui. São muito importantes na minha vida e, por eles, me tornei a pessoa que sou hoje.

Agradeço aos meus verdadeiros amigos, que independente do grupo que a pertenciam (Brous ou Zueira) sempre me apoiaram nas minhas escolhas, ficando ao meu lado nos momentos bons e ruins. Não menos importante, agradeço também aos amigos que fiz no LAPO, que me ajudaram no que puderam para que eu conseguisse terminar este trabalho.

Ao Prof. Robson, não só por ser meu co-orientador, mas por ele sempre acreditar em mim. Agradeço-o por me guiar durante todo meu trajeto acadêmico, auxiliando-me nas escolhas que deveria fazer para chegar até aqui.

Finalmente, gostaria de agradecer ao meu orientador, Prof. Alexandre, pela dedicação fora do comum que ele teve para que eu alcançasse tudo o que consegui neste trabalho.

Agradeço a todos porque sem a ajuda de vocês não estaria onde estou hoje. Obrigado.

Resumo

Nesta dissertação, introduzimos modelos e algoritmos de otimização visando reduzir o consumo de energia das Redes de Sensores Sem Fio, através da resolução integrada de problemas de otimização comuns a esse tipo de rede. Um simulador de eventos discretos é implementado e testado computacionalmente com o intuito de avaliar, a partir de determinadas métricas de Qualidade de Serviço, como as soluções encontradas pelos algoritmos de otimização se comportam em um ambiente dinâmico. Por fim, o modelo proposto é comparado a um modelo alternativo apresentado na literatura. O Problema de Otimização tratado neste trabalho consiste em integrar os problemas de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro através de uma abordagem multi-período. Conhecendo-se *a priori*, o tempo inicial em que a rede permanecerá ativa, o mesmo é dividido em períodos de tempos iguais e, para cada um, uma solução para todos os problemas a serem integrados deve ser assegurada. Para modelar o problema, é apresentada uma formulação em Programação Inteira Mista. Um algoritmo do tipo Branch-and-Cut é proposto e implementado para resolver o problema de forma exata. Devido às dificuldades inerentes à integração, resolver o problema proposto na otimalidade mostrou-se uma tarefa difícil, mesmo para instâncias de pequenas dimensões. Dessa forma, propomos uma heurística para encontrar soluções viáveis, idealmente de boa qualidade, em tempos aceitáveis. Nossos resultados mostram que os algoritmos de otimização permitiram alcançar bons níveis de Qualidade de Serviço, além de reduções significativas no consumo de energia, principalmente quando comparados a uma abordagem alternativa encontrada na literatura.

Abstract

In this work, we introduce models and optimization algorithms to reduce the wireless sensor networks energy consumption, by integrating some common optimization problems to this network. We implemented and tested a discrete event simulator in order to assess, from certain Quality of Service metrics, how the solutions found by optimization algorithms behave in a dynamic environment. Finally, we compared the proposed model with an alternative method in the literature. The optimization problem addressed in this work consists in integrating the Density Control, Coverage, Routing and Sink Location Problem through a multi-period approach. As the initial network lifetime is known, it is divided into equal time periods and for each one, a solution for all the integrated problems must be ensured. This model is presented by a mixed integer programming formulation. A Branch-and-Cut algorithm is proposed and implemented to solve the problem exactly. Due to the difficulties inherent to the integration, to solve the proposed problem optimality is a difficult task, even for small instances. Thus, we propose a heuristic to find feasible solutions, with good quality, in acceptable time. Our results show that the optimization algorithms had achieved good levels of service quality, presenting significant energy reductions, especially when it is compared to an alternative approach found in the literature.

Lista de Figuras

2.1	Modelo de nó sensor Mica2 da linha Motes [mic, 2011]	6
3.1	Um exemplo de como mapear Θ em Θ'	21
5.1	Um exemplo da reestruturação da rede após a inserção de um novo nó	36
5.2	Exemplo de uma solução ao aplicar o critério de diversificação	40
5.3	Melhora da solução ocasionada pelo aumento da cardinalidade de M para 400 sensores	46
5.4	Melhora da solução ocasionada pelo aumento da cardinalidade de M para 500 sensores	46
5.5	Melhora da solução ocasionada pelo aumento da cardinalidade de M para 600 sensores	47
6.1	Fluxograma do simulador	52
6.2	Comparação entre Atraso Médio, Taxa de Entrega e Distância Percorrida pelo Sorvedouro para 400 sensores	57
6.3	Comparação entre Atraso Médio, Taxa de Entrega e Distância Percorrida pelo Sorvedouro para 600 sensores	58

Lista de Tabelas

4.1	Parâmetros	29
4.2	Relação entre a corrente gasta ao se transmitir uma mensagem e a distância	31
4.3	Primeira solução viável e após uma hora de execução do algoritmo BC . .	32
4.4	Solução viável encontrada após 10 horas de execução	32
5.1	Resultados obtidos pela Heurística proposta para o PICDCR	45
5.2	Comparação entre as soluções encontradas pela heurística com aquelas encontradas pelo algoritmo exato	48
6.1	Comparação entre a energia residual da rede encontrada pela heurística em relação àquela retornada pela simulação	59
7.1	Diferenças entre o PICDCR e o LSRP	64
7.2	Soluções encontradas pelo CPLEX para o LSRP-M	71
7.3	Soluções encontrada para o PICDCR	73

Lista de Algoritmos

5.1	Procedimento de Busca Local	37
5.2	Heurística construtiva para o PICDCR	42
5.3	Pseudo-código do procedimento ILS	43
5.4	Pseudo-código do Procedimento COBERTURA_COMPLETA	44

Sumário

Agradecimentos	vii
Resumo	ix
Abstract	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Organização da Dissertação	3
2 Redes de Sensores Sem Fio	5
2.1 Conceitos e Aplicações	5
2.2 Organização de uma RSSF	7
2.2.1 Problema de Cobertura	9
2.2.2 Problema de Controle de Densidade	9
2.2.3 Problema de Roteamento	10
2.2.4 Problema de Localização do Sorvedouro	12
2.2.5 Problemas Integrados	14
2.3 A Nossa Contribuição	15
3 Um Modelo Integrado para o Problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro em RSSFs	17
3.1 Motivação	17
3.2 Modelo Proposto	18
3.3 Formulação baseada em Desigualdades de Eliminação de Circuitos Direcionada	20

4	Métodos de Solução Exata para o PICDCR	27
4.1	Um Algoritmo Branch-And-Cut	27
4.2	Resultados do Algoritmo Branch-and-Cut	28
4.2.1	Instâncias	29
4.2.2	Resultados computacionais	30
4.3	Comentários	33
5	Método heurístico para o PICDCR	35
5.1	Busca Local	35
5.1.1	Inserção	37
5.1.2	Troca	37
5.1.3	Remoção	38
5.1.4	Função de Avaliação	39
5.2	Critério de Diversificação	40
5.3	ILS	41
5.4	Resultados Computacionais	43
5.5	Comentários	49
6	Simulação de uma RSSF	51
6.1	Aspectos Gerais do Simulador	51
6.2	Resultados Computacionais	54
6.2.1	Métricas de QoS Avaliadas	55
6.3	Comentários	60
7	Comparação com a Literatura	63
7.1	Modelo da Literatura	63
7.2	Resultados computacionais	68
7.3	Comentários	73
8	Conclusão e Trabalhos Futuros	75
	Referências Bibliográficas	77

Capítulo 1

Introdução

Com o desenvolvimento de técnicas cada vez mais avançadas e econômicas para a fabricação de componentes eletrônicos tais como processadores embutidos, rádios, e micro sistemas eletromecânicos as Redes de Sensores Sem Fio (RSSF) vêm se concretizando como uma realidade [Ruiz et al., 2004]. As RSSFs são compostas de centenas ou mesmo milhares de dispositivos computacionais pequenos, autônomos e compactos, chamados de nós sensores¹, e dotados de processador, memória, rádio, dispositivo de sensoriamento e bateria.

Devido às limitações no seu hardware, os sensores necessitam trabalhar de forma colaborativa. Para isso, cada sensor monitora um fenômeno ao seu redor (por exemplo, temperatura), o processa e dispersa os dados coletados pela rede via rádio (através de um protocolo de comunicação sem fio), de forma que esses possam ser roteados até os sorvedouros através de outros nós sensores. As informações recebidas pelos sorvedouros são então enviadas para algum observador externo (central de processamento de dados, por exemplo), o qual é responsável por analisar os dados coletados e realizar, caso necessário, tomada de decisões. Dessa maneira, estas redes passam a apresentar grande potencial em várias aplicações, desde o rastreamento e vigilância de alvos militares à exploração de ambientes perigosos ou inóspitos [Yick et al., 2008].

Por serem classificadas como redes *ad-hoc*, as RSSFs apresentam características específicas que as diferem das redes tradicionais, tais como: alta densidade de nós, alta dependência de aplicações, fortes restrições de recursos incluindo limitada quantidade de energia, curto raio de comunicação entre os sensores, baixa largura de banda e limitado poder de processamento e armazenamento [Yick et al., 2008].

Dentre todas as características citadas, a capacidade limitada de energia das RSSFs é, com certeza, a que apresenta maior impacto durante sua organização. Por

¹Nesse texto, os termos nó sensor e nó serão utilizados como sinônimos.

isso mesmo, a maioria dos trabalhos encontrados na literatura procura abordar formas de redução no consumo de energia destas redes, com conseqüente preservação em seu tempo de vida [Anastasi et al., 2009].

Vários são os mecanismos utilizados para alcançar reduções significativas no consumo de energia pela rede. Todos, entretanto, procuram explorar características específicas desse tipo de rede para se chegar a esse objetivo. Em uma RSSF onde os sorvedouros são fixos, aliada a limitações no alcance de comunicação dos sensores, as informações coletadas pelos mesmos não conseguem, por vezes, chegar diretamente ao sorvedouro. Ao invés disso, faz-se necessário a utilização de outros sensores para que a informação coletada possa ser roteada ao seu destino, em um paradigma conhecido como *Multi-hop* [Akyildiz et al., 2002]. Diversos autores ([Pottie & Kaiser, 2000], [Raghunathan et al., 2002], entre outros) argumentam que o consumo de energia relacionado ao uso do rádio pelos nós, quando esses transmitem ou recebem informações de outros nós sensores, é superior ao consumo exigido pelos mesmos durante suas funções de sensoriamento.

Sendo assim, algumas estratégias são utilizadas como meio de diminuir o tráfego de dados na rede, com subseqüente redução no consumo de energia, gerando alguns problemas, tais como:

- a escolha de um sub-conjunto de sensores que permanecerão inativos enquanto os demais continuarão ativos sensoriando a região monitorada;
- a definição de rotas para que os dados gerados pelos sensores cheguem até o sorvedouro;
- a possibilidade de mobilidade dos sorvedouros, escolhendo localizações (posições cujas coordenadas são conhecidas) para estes na RSSF, a partir de um conjunto pré-definido de posições, de forma que haja redução no tamanho das rotas dos dados coletados pelos sensores.

Por levarem em conta a necessidade de redução no consumo de energia da rede, esses problemas passam a apresentar por si só um elevado grau de dificuldade, uma vez que são considerados NP-difíceis ([Yang et al., 2006b], [Park & Sahni, 2006]). O intuito desta dissertação é integrar a resolução de todos esses problemas de forma a garantir redução no consumo de energia da rede com conseqüente prolongamento em seu tempo de vida, ao mesmo tempo em que se assegura certas Qualidades de Serviço (QoS²), tais como cobertura e conectividade.

²Da sigla em inglês *Quality of Service*.

Neste trabalho, o problema de integração foi formalizado através de um Problema de Otimização em Grafos. A partir disso, foi proposta uma formulação em Programação Inteira Mista, empregando exponencialmente muitas Desigualdades de Eliminação de Circuitos Direcionada (*DCSET*³).

Foi proposto também um algoritmo exato, do tipo Branch-and-Cut ([Padberg & Rinaldi, 1991], [Lucena & Beasley, 1996]), baseado na identificação das *DCSETs* como planos de corte. Uma vez que o método exato desenvolvido exige um elevado tempo computacional para resolver o problema na otimalidade, o que inviabiliza sua utilização em aplicações de interesse prático, foi desenvolvido também um método heurístico para a problema, baseado na metaheurística ILS [Ramalhinho-Lourenço et al., 2000].

Como parte final do trabalho, implementamos um simulador, sobre o arcabouço de simulação JIST/SWAMS [jis, 2007], a fim de avaliar algumas métricas de QoS importantes em RSSFs, além de compararmos o problema proposto neste trabalho com o apresentado por TürkogullarI et al. [Türkoğullari et al., 2010].

Os resultados obtidos pelos algoritmos de otimização e via simulação demonstram que reduções no consumo de energia e bons níveis de QoS para as RSSFs foram alcançados. Ainda mais, quando comparado a um trabalho semelhante apresentado por TürkogullarI et al. [Türkoğullari et al., 2010], o problema que propomos permitiu ganhos energéticos significativos, possibilitando que o tempo de vida da rede pudesse ser estendido em até 50%.

As principais contribuições desta dissertação são:

- A integração de alguns problemas de otimização comuns em RSSFs, tais como o problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro, apresentando uma formulação matemática para isso;
- A apresentação de algoritmos de otimização para resolver o problema proposto, além da implementação de um ambiente de simulação dinâmico, onde as redes e as propriedades dos nós sensores sofrem alterações com o tempo.

1.1 Organização da Dissertação

O restante da dissertação está organizado da seguinte forma. No Capítulo 2, introduzimos alguns conceitos e aplicações de RSSFs. Uma revisão bibliográfica é apresentada e vários problemas relacionados à organização de uma RSSF são discutidos. Ao final do

³Da sigla em inglês *Direct Cut Set*.

capítulo, destacamos o problema central desta dissertação: A integração do problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro utilizando uma abordagem Multi-Período. No Capítulo 3, formalizamos o problema, aqui denominado Problema Integrado para Controle de Densidade, Cobertura e Roteamento (PICDCR) e discutimos um problema semelhante presente na literatura. Ao longo do capítulo, apresentamos uma formulação em Programação Inteira Mista para modelar o PICDCR.

No Capítulo 4, discutimos um algoritmo exato para o PICDCR. Resultados e experimentos computacionais são apresentados e as dificuldades encontradas, inerentes ao problema proposto, são discutidas. Tais dificuldades estimularam o desenvolvimento da heurística proposta no Capítulo 5, que é basicamente uma metaheurística, composta por mecanismos de diversificação e intensificação. Os resultados da heurística implementada são utilizados como entrada no simulador de eventos discretos, introduzido no Capítulo 6. Com base nos resultados, é possível afirmar que melhoras nos indicadores de QoS de uma RSSF são ocasionadas quando o PICDCR, e conseqüentemente, os algoritmos que o resolvem, são utilizados durante a organização desse tipo de rede. Além disso, no Capítulo 7 comparamos o PICDCR com um problema semelhante encontrado na literatura e mostramos que o primeiro alcançou reduções significativas no consumo de energia, com conseqüente prolongamento no tempo de vida de uma RSSF.

Finalmente, encerramos o texto no Capítulo 8, apresentando as principais conclusões extraídas deste trabalho e as direções de pesquisa futura.

Capítulo 2

Redes de Sensores Sem Fio

Neste capítulo discutimos os principais conceitos envolvidos na organização das RSSFs. Apresentaremos aplicações práticas que ilustram a importância desses tipos de redes na atualidade assim como especificidades que as distinguem das redes tradicionais. Em função dessas especificidades, vários problemas relacionados à organização ótima das RSSFs são apresentados e discutidos. Concluimos o capítulo apresentando as principais motivações deste trabalho.

2.1 Conceitos e Aplicações

Além dos nós sensores (um equipamento comercial é mostrado como exemplo na Figura 2.1), uma RSSF apresenta um ou mais nós especiais conhecidos como nós sorvedouros, dotados de características distintas em relação aos demais, tais como bateria ilimitada (considerando-a recarregável), maior alcance de rádio, melhor unidade de processamento e armazenamento, entre outros.

Esses nós podem ser fixos ou se movimentarem pela rede e são responsáveis por receber os dados coletados dos sensores, processá-los e enviá-los até um observador externo para que esses possam ser analisados.

Dependendo da aplicação, as RSSFs podem apresentar centenas, ou mesmo milhares de nós sensores. Entretanto, devido a esse tipo de rede apresentar pouca ou nenhuma infraestrutura, estes sensores necessitam trabalhar de forma colaborativa, cada um obtendo dados sobre o ambiente ao seu redor, no intuito de conseguir o monitoramento completo de uma região.

Os nós sensores apresentam diversos tipos distintos de dispositivos de sensoriamento. Entre os mais comuns, podemos destacar aqueles que apresentam funções sísmicas, visuais, infravermelho ou acústicas. Sendo assim, uma grande variedade de



Figura 2.1. Modelo de nó sensor Mica2 da linha Motes [mic, 2011]

eventos podem ser monitorados tais como: temperatura, umidade, movimentação, pressão, acidez do solo, nível de ruído, ou presença (ausência) de certos tipos de objetos.

Com funcionalidades tão diversificadas, aliadas ao constante progresso no desenvolvimento de tecnologias sem fio, as RSSFs passam a apresentar potencial para serem utilizadas em inúmeros tipos de aplicações, quer sejam em ambientes externos ou internos.

Aplicações envolvendo ambientes externos, em geral, usam uma densa coleção de sensores. Isto porque, nesse tipo de aplicação, há uma grande dispersão aleatória de nós sensores em uma região (através de um avião, por exemplo) no intuito de garantir que a mesma seja completamente monitorada por estes. Por não existirem posições pré-definidas para os sensores nessa região, esse tipo de dispersão apresenta algumas vantagens, tais como a redução no custo operacional da rede (uma vez que não há a necessidade de posicionar manualmente os sensores na região) e a facilidade de utilizá-la em áreas apresentando características que dificultem a dispersão manual (regiões inóspitas, de difícil acesso ou mesmo perigosas). Em contrapartida, aplicações envolvendo ambientes internos, em geral, utilizam poucos nós sensores, permitindo que sua distribuição pela região a ser monitorada seja feita manualmente. Embora a utilização de poucos nós sensores possibilite reduzir o custo de manutenção da rede, aplicações envolvendo uma grande dispersão de nós de forma aleatória são usualmente mais convenientes. Isso é verdade pois, além de reduzir o custo operacional da rede, tal dispersão possibilita maior grau de redundância nos dados gerados pelos nós sensores, o que permite a utilização de mecanismos que se beneficiem de tal característica como forma de reduzir o consumo de energia da rede [Anastasi et al., 2009].

De acordo com [Yick et al., 2008], há várias áreas de aplicações para as RSSFs.

Podemos citar algumas como:

- Militar: RSSFs podem ser usados na detecção ou no rastreamento de inimigos, veículos, no reconhecimento de terrenos, no monitoramento do campo de batalha ou detecção de ataques biológicos, entre outros.
- Meio Ambiente: Podem ser usados na detecção e prevenção de acidentes sísmicos, químicos, queimadas, no detecção de pessoas em desastres naturais ou mesmo no controle de poluentes de uma cidade e ainda no monitoramento da biodiversidade de uma determinada área.
- Saúde: Dados como temperatura, pressão, umidade, posição de um paciente (a forma como ele está deitado, por exemplo) ou localização do mesmo em sua residência são coletados através de sensores posicionados em suas roupas ou mesmo em sua casa e são enviados a um centro médico.
- Público/Industrial: Podem ser usados no rastreamento de veículos ou no monitoramento do tráfego. Na indústria, os sensores podem ser usados para controle de inventário, monitoramento de poluentes, de máquinas ou mesmo de pessoas.

Como pode-se perceber, as RSSFs são bastante dependentes da aplicação ao qual foram destinadas. Dessa forma, o projeto de uma RSSF deve levar em conta vários fatores tais como: tolerância a falha, controle de topologia, escalabilidade da rede, restrições de hardware, ambiente de atuação, Qualidade de Serviço (QoS), mobilidade, consumo de energia, etc. A seguir, discutiremos alguns desses fatores.

2.2 Organização de uma RSSF

A organização de uma RSSF deve levar em conta as características exclusivas desse tipo de rede, já que protocolos genéricos utilizados em redes tradicionais costumam ser ineficazes nas RSSFs. Segundo [Yick et al., 2008], as RSSFs diferenciam-se das redes convencionais por apresentarem os seguintes aspectos:

- Densa coleção de nós sensores;
- Restrições no hardware (principalmente por questões econômicas), limitando o poder de processamento, armazenamento, rádio e energia dos sensores;
- Grande dependência da aplicação;
- Largura de banda reduzida;

- Mudança na topologia da rede, devido a mobilidade dos sensores ou necessidade de reestruturação da mesma, ocasionada por variações no ambiente.

Devido a alguns desses aspectos, as RSSFs podem apresentar sensores com características de hardware diferentes em sua composição, como forma de melhorar a coleta de dados ou mesmo prolongar seu tempo de vida. Esse tipo de rede, chamado rede *heterogênea*, é comum em aplicações que necessitem de informações em larga quantidade para se chegar a uma conclusão (tais como a detecção de queimadas em uma floresta, feita a partir de dados coletados sobre a temperatura e a umidade da mesma). Já aquelas redes que apresentam sensores com as mesmas características de hardware são chamadas *homogêneas*.

Dentre todas as características das RSSFs citadas, a restrição de energia é certamente a que apresenta maior impacto durante a organização desse tipo de rede [Anastasi et al., 2009]. Isso acontece pois, em muitas aplicações, necessita-se monitorar regiões de difícil acesso (como florestas densas, por exemplo), o que torna inviável a troca das baterias dos sensores, quando a mesma for necessária. Dessa forma, enquanto as redes tradicionais se preocupam em garantir QoS, como atraso médio de mensagem ou taxa de mensagens entregues por exemplo, tais métricas têm seu valor reduzido em termos de prioridade durante a organização de uma RSSF, já que mecanismos que reduzam o consumo de energia da rede tornam-se mais importantes.

Contudo, conflitando com a redução de energia, a garantia de conectividade da rede além da taxa de cobertura (porcentagem total da área que é coberta por pelo menos um nó sensor em um determinado instante de tempo) são QoS que devem, também, ser levadas em conta durante a organização dessas redes. O motivo para isso é evidente, já que o objetivo das RSSFs consiste em basicamente monitorar, a partir dos nós sensores que as compõem, uma determinada região, permitindo que os dados coletados por eles cheguem (através de uma abordagem *multi-hop*, em geral) até um sorvedouro. Dessa forma, uma redução de energia ocasionada pela desativação de alguns sensores, por exemplo, poderia causar redução na taxa de cobertura ou mesmo impedir que os dados coletados pelos nós pudessem ser roteados corretamente até o sorvedouro.

Como consequência disso, alguns problemas específicos em RSSFs surgem no intuito de atender algumas das QoS, tais como taxa de cobertura e garantia de conectividade, ao mesmo tempo em que se visa a redução no consumo de energia da mesma. Alguns desses problemas são explicados a seguir.

2.2.1 Problema de Cobertura

Um das questões fundamentais em RSSFs trata do Problema de Cobertura. Em geral, a cobertura reflete o quão bem uma área é monitorada pelos sensores, o que a torna uma das mais importantes medições de QoS [Seapahn Megerian & Srivastava, 2005]. Tseng e Huang [Huang & Tseng, 2003] definem o problema de cobertura, chamando-o de *Problema da k-cobertura*, como aquele consistindo em *verificar* se toda região em uma dada área é coberta por pelo menos k sensores. Apesar desse problema ser resolvido em tempo polinomial ([Huang & Tseng, 2003], [So & Ye, 2005], [Fang & Low, 2007]), durante o processo não é levado em conta critérios para redução no consumo de energia da rede.

Outra abordagem consiste em dividir a área a ser monitorada em um conjunto finito de pontos de demanda. A região monitorada por cada sensor, por sua vez, é modelada como uma circunferência de raio R_s , conhecido como *Raio de Sensoriamento*, cujo centro é o próprio nó. Dessa forma, uma vez que os sensores são distribuídos em uma determinada área, deve-se assegurar que qualquer região pertencente à área monitorada contenha ao menos um nó sensor monitorando-a.

Chakrabarty et al. [Chakrabarty et al., 2002], Meguerdichian e Potkonjak [Miodrag & Potkonjak, 2003] levam em conta indiretamente o consumo de energia ao formularem e resolvem modelos de Programação Linear Inteira (PLI) para o problema de cobertura, objetivando a minimização do custo dos sensores e do número necessário desses em uma RSSF, respectivamente. Yang et al. [Yang et al., 2006b] definem o problema de *encontrar* um sub-conjunto mínimo de sensores que garantam a k -Cobertura como o *Problema da k-cobertura mínima*. Além disso, demonstram que esse problema é NP-difícil uma vez que é uma extensão do problema de conjuntos dominantes [Garey & Johnson, 1979]. Hefeeda e Bagheri [Bagheri et al., 2007] propõem um algoritmo aproximativo para esse problema e mostram que a solução obtida por seu algoritmo está a um fator logarítmico do ótimo.

A partir do que foi apresentado, definimos o *Problema de Cobertura* como o problema de encontrar um subconjunto de nós sensores que permanecerão ativos, garantindo que cada ponto de demanda seja coberto (esteja dentro da área de sensoriamento) por pelo menos um desses sensores. O subconjunto pode ser mínimo ou não, dependendo de como as QoS são considerados.

2.2.2 Problema de Controle de Densidade

Como consequência de uma distribuição aleatória dos sensores em uma dada região, há a necessidade de se utilizar uma rede com alta densidade de nós para que uma

determinada taxa de cobertura seja garantida. Isso cria a possibilidade de se explorar a redundância no grau de cobertura com o intuito de reduzir o consumo de energia da rede, e conseqüentemente, permitir que sua vida útil não seja muito pequena. Uma técnica comum para isso consiste em selecionar um sub-conjunto de sensores, dentre aqueles dispersados na área a ser monitorada, para permanecerem ativos, sensoriando a região e transmitindo as informações coletadas, enquanto os demais são agendados a permanecerem em um estado de baixo consumo de energia ou mesmo desativados por certo intervalo de tempo. Dessa maneira, os nós sensores podem se alternar entre serem ativados ou desativados periodicamente a partir de algum critério, o que permite aumentar o tempo de vida da rede por um fator de 1,5-2 quando comparado a uma rede apresentando todos os nós ativos [Warrier et al., 2007]. O problema de definir um escalonamento ótimo para os sensores permanecerem desativados no intuito de melhorar o tempo de vida de uma RSSF é conhecido como *Problema de Controle de Densidade*.

O problema de Controle de Densidade é bastante tratado na literatura ([Marcos et al., 2003], [Zhang & Hou, 2005], [Siqueira et al., 2006], [Yang et al., 2006a], [Zeng et al., 2010]), e, basicamente pode ser encontrado sob duas formas. A primeira, através de uma abordagem centralizada, onde as decisões são tomadas por um agente central (o sorvedouro da rede). A segunda, utiliza-se de uma abordagem descentralizada (resumidos em [Anastasi et al., 2009], [Fan & Jin, 2010]), onde os próprios sensores decidem, de forma colaborativa, quais permanecerão ativos e quais serão desativados.

Apesar de a primeira abordagem ser mais precisa, por contar com informações globais sobre o estado da rede, para casos onde o sorvedouro não apresenta mobilidade, a rede consome mais energia que a abordagem descentralizada. Isso acontece pois o sorvedouro precisa enviar periodicamente informações sobre o controle de densidade a todos os sensores. Já no caso da abordagem descentralizada, como a decisão é tomada localmente (através de troca de mensagens entre os sensores), há um consumo menor de energia.

2.2.3 Problema de Roteamento

Um dos principais objetivos durante a construção das RSSFs é permitir a transmissão dos dados coletados pelos sensores até um sorvedouro enquanto tenta-se prolongar o tempo de vida da rede, evitando a degradação da conectividade da mesma [Al-karaki & Kamal, 2004]. Contudo, devido à limitações estruturais e de recursos inerentes a esse tipo de rede, como alcance de comunicação do rádio por exemplo, o

monitoramento de uma determinada região deve ser feito de forma colaborativa. Desta maneira, os sensores monitoram a região ao seu redor e enviam os dados coletados até algum sensor dentro de seu *Raio de Comunicação*, R_c , de forma que tal nó possa retransmitir a informação recebida para algum outro, e assim sucessivamente, até que essa alcance o sorvedouro. Essa abordagem de transmissão utilizando vários sensores para retransmitir os dados coletados é chamada de *Multi-hop* [Akyildiz et al., 2002].

Uma vez que o consumo de energia do sensor na transmissão e retransmissão de dados é superior ao gasto nas funções de sensoriamento ([Pottie & Kaiser, 2000], [Raghunathan et al., 2002]), deve-se tentar encontrar soluções que reduzam o tráfego de dados da rede a fim de minimizar seu consumo total de energia.

Embora pesquisas recentes sobre o tema concentrem-se no estudo da utilização de protocolos eficientes (veja [Al-karaki & Kamal, 2004]), alguns autores apresentam formulações lineares ou não-lineares para a determinação ótima de rotas de dados, objetivando a maximização do tempo de vida da rede. Younis et al. [Younis et al., 2002], por exemplo, formulam o problema de roteamento como o problema de um-para-muitos caminhos mais curtos (por exemplo, menor energia), depois de determinar o peso, em unidades de energia, que está relacionado com cada par de nós. Outros autores apresentam formulações que alcançam o mesmo objetivo ao minimizarem a energia total gasta por cada sensor [Chang & Tassiulas, 2004], ao agregarem as rotas entre os sensores e o sorvedouro [Xue et al., 2005] ou ao minimizarem o consumo de energia máximo normalizado dos sensores [Hua & Peter Yum, 2008]. Ciciriello et al. [Ciciriello et al., 2007] também alcançam esse objetivo ao apresentarem um modelo multi-fluxo para o problema de roteamento onde existem muitos sorvedouros. A idéia proposta pelos autores é procurar minimizar o número de caminhos explorados entre cada par sensor-sorvedouro de forma a reduzir o consumo de energia da rede.

Apesar de questões como agregação de dados ([Hua & Yum, 2008], [Lin et al., 2009]), múltiplas rotas ([Yahya & Ben-Othman, 2009], [Nesa Sudha & John, 2011]), protocolo de enlace ([Cui et al., 2005], [gang Gong et al., 2009]), entre outras possibilitarem reduções no consumo de energia da rede quando analisadas no contexto de redes de sensores, estamos interessados exclusivamente no problema de roteamento como um problema de otimização. Dessa forma, o *Problema de Roteamento* será definido como o problema de se encontrar rotas de dados únicas e ótimas em termos energéticos entre cada sensor e o sorvedouro, a fim de reduzir o consumo de energia da rede e, conseqüentemente, estender seu tempo de vida.

Park e Sahni [Park & Sahni, 2006] provam que o problema de roteamento maximizando o tempo de vida da rede é NP-difícil, visto que o problema de caminhos

disjuntos conectados (DCP¹)[Garey & Johnson, 1979] é redutível polinomialmente a esse.

2.2.4 Problema de Localização do Sorvedouro

O problema de roteamento é diretamente influenciado pela posição do(s) sorvedouro(s) na RSSF, já que o objetivo desse problema é a minimização no consumo de energia da rede através das melhores escolhas de rotas entre cada par sensor-sorvedouro. Dessa forma, ao variar a posição do(s) sorvedouro(s) na rede, uma nova solução para o problema de roteamento é encontrada. Ao analisar o comportamento de cada solução obtida para o problema de roteamento, percebe-se:

1. variação no comprimento das rotas a serem usadas pelos dados;
2. variação no tráfego de dados daqueles sensores próximos (dentro do raio de comunicação) ao sorvedouro;
3. melhor distribuição do consumo de energia da rede;

O primeiro comportamento é bastante intuitivo, uma vez que variando-se a posição do sorvedouro na RSSF, os comprimentos das rotas de dados poderão aumentar ou diminuir, assim como também o consumo de energia da rede. O segundo, atenta-se àqueles sensores próximos ao sorvedouro, já que são eles os responsáveis por, além de transmitir seus dados, retransmitirem todos os demais gerados pela rede até o sorvedouro. Dependendo de como as rotas de dados entre cada par sensor-sorvedouro são definidas, haverá sobrecarga no tráfego de dados nestes sensores, o que pode ocasionar falhas de conectividade na rede, uma vez que tais sensores tendem a extinguir sua energia mais rapidamente [Li & Mohapatra, 2007]. Já no terceiro, à medida que varia-se a posição do(s) sorvedouro(s) na RSSF, diferentes rotas para o par sensor-sorvedouro são geradas, o que reduz a necessidade de utilizar uma única rota sensor-sorvedouro por todo o tempo de vida da rede.

É importante observar que o problema de localização apresenta definições distintas dependendo do contexto ao qual pertence. No contexto de rede de sensores, por exemplo, esse problema consiste em basicamente estimar as coordenadas geográficas de uma entidade que compõe a rede (sorvedouro ou sensor), com base em observações de grandezas físicas (valores de potência, tempo e ângulo de chegada dos sinais recebidos por exemplo) [Moreira, 2007]. Já no contexto de otimização, o problema de localização, em geral, trata decisões sobre onde localizar facilidades (fábricas, escolas, sensores,

¹Sigla em inglês para *Disjoint Connected Problem*

entre outros) em uma rede, considerando que existem demandas (clientes, estudantes, área de cobertura) a serem atendidas, de forma a otimizar um determinado critério ([Daskin, 1995], [Goldbarg & Luna, 2000]).

Focando-se portanto no contexto de otimização, o *Problema de Localização do Sorvedouro*², consiste em escolher, a cada período de tempo, localizações ótimas para posicionar o(s) sorvedouro(s) em uma RSSF, dado um conjunto finito de posições candidatas para o(s) mesmo(s), de forma que o consumo de energia da rede seja minimizado a ponto de permitir prolongar seu tempo de vida.

Esse problema, da forma como definido, permite modelar, apesar de simples, a mobilidade do sorvedouro. Por simples, entende-se a possibilidade do sorvedouro estar, a cada início de um período, em uma posição diferente na rede, ao invés de se movimentar continuamente por essa. Dessa maneira, durante o intervalo de tempo entre um período e o subsequente, o sorvedouro permanece imóvel em uma posição.

O Problema de Localização do Sorvedouro pode ser considerado de difícil resolução, já que apresenta a necessidade de se resolver, como subproblema, um problema de roteamento (comprovadamente NP-difícil, [Park & Sahni, 2006]), para cada posição candidata para o(s) sorvedouro(s) se localizar.

Embora a maioria dos trabalhos recentes concentrem-se na mobilidade contínua do sorvedouro, alguns autores assumem que a movimentação do mesmo poderá ocorrer apenas periodicamente, permitindo que o sorvedouro permaneça parado em uma localização durante toda a duração do período. Bogdanov et al. [Bogdanov et al., 2004] mostram que a localização do sorvedouro tem influência na eficiência energética da RSSF já que o fluxo de dados é afetado. Gandham [Gandham, 2003] discute os benefícios de múltiplos sorvedouros como uma estratégia para economia de energia. De acordo com estes autores, o uso de múltiplos sorvedouros, além da mudança periódica de suas localizações, aumentam o tempo de vida da rede. Partindo disso, propõem dois modelos usando Programação Linear Inteira Mista (PLIM) onde o tempo de vida da rede é dividido em períodos de tempos iguais e determinam a localização do sorvedouro minimizando o consumo de energia de cada sensor por período. Wang et al. [Wang et al., 2005] apresentam um modelo em programação linear para tratar o problema de localização e mobilidade do sorvedouro. Os autores determinam qual será a localização do sorvedouro na RSSF e quanto tempo esse ficará naquela posição, de forma a maximizar o tempo de vida da rede, que consideram ser o tempo até que algum sensor fique sem energia. Guney et al. [Güney et al., 2010] apresentam dois grupos de modelos em PLI para tratar esse problema: o primeiro, cuja preocupação é a minimi-

²Da tradução literal em inglês *Sink Location Problem*

zação da energia gasta pelos sensores, com o objetivo de maximizar o tempo de vida da rede; e o segundo, focado na preocupação quanto ao custo inerente de se posicionar o sorvedouro em uma determinada posição, a fim de minimizar o custo econômico da RSSF. Assim como os trabalhos anteriores, os grupos de modelos procuram encontrar as melhores localizações para o(s) sorvedouro(s) dentro de um conjunto finito de posições.

2.2.5 Problemas Integrados

Ao analisar todos os problemas descritos anteriormente, fica caracterizado a grande preocupação em se reduzir o consumo de energia de uma RSSF, garantindo o prolongamento do seu tempo de vida. Além disso, pelo fato de estes problemas apresentarem algum tipo de inter-dependência, como é o caso do problema de cobertura e controle de densidade e do problema de roteamento e localização do sorvedouro, pode-se pressupor que uma melhor eficiência energética consiga ser alcançada pela rede a partir da integração entre todos esses problemas.

Entretanto, apesar de muitos trabalhos encontrados na literatura aproveitarem esse tipo de relação para formular problemas integrados ([Siqueira et al., 2006], [Yang et al., 2006a], [Zeng et al., 2010],[Güney et al., 2010], entre outros), é interessante observar que a formulação de modelos integrando todos os problemas descritos anteriormente não é encontrado tão facilmente. Um motivo para isso decorre do fato de que tais problemas são, por si só, difíceis de serem resolvidos separadamente, o que torna sua integração ainda mais difícil.

Nakamura et al. [Nakamura et al., 2005] apresentam uma tentativa de integração ao proporem uma formulação em PLIM baseada em multi-fluxo e objetivando a maximização do tempo de vida da rede. A abordagem apresentada integra o problema de cobertura, controle de densidade e roteamento em uma estratégia multi-período. Entende-se por abordagem/estratégia multi-período o fato de se assumir que o tempo de vida útil T de uma RSSF é dividido em períodos iguais, e a solução para o problema proposto deve satisfazer simultaneamente, e em todos os períodos, os problemas que estão sendo integrados. Embora de cunho apenas teórico, os autores afirmam que este modelo pode ser útil para o propósito de oferecer um limite superior para o consumo de energia de uma RSSF durante seu planejamento real. Andrade et al. [de Andrade et al., 2009] apresentam uma heurística para resolver de forma aproximada o problema anterior, baseada na meta-heurística GRASP [Feo & Resende, 1995]. Os resultados mostraram que a abordagem utilizada pela heurística tornou possível encontrar soluções próximas a ótima, com reduzidos tempos computacionais.

Já Türkogulları et al. [Türkogulları et al., 2010] propuseram uma formulação de PLIM baseada em fluxos em redes, cuja proposta consiste em maximizar o tempo de vida da rede a partir da integração dos problemas de Cobertura, Controle de Densidade, Roteamento e Localização do Sorvedouro assim como de Sensores em uma abordagem multi-período. Entretanto, sob o ponto de vista dos autores, o problema de localização do sorvedouro consiste em encontrar a melhor posição para o mesmo a partir de um conjunto de posições possíveis, de forma que, o sorvedouro permaneça na posição escolhida durante todo o tempo de vida da rede. Além disso, como a proposta dos autores é trabalhar com redes heterogêneas e aplicações cuja dispersão dos sensores pode ser feita manualmente (apresentando um custo para isso), eles resolvem simultaneamente com os demais, o problema de encontrar as melhores posições para os sensores na área monitorada com o menor custo possível. Entretanto, a formulação que os autores propõem é ineficiente a partir do aspecto combinatorial, uma vez que apresentam restrições do tipo *big-M*, notoriamente conhecidas por dificultarem a convergência da soluções para o ótimo durante a resolução do modelo de forma exata. Dessa forma, os autores apresentam uma heurística como alternativa para encontrar boas soluções para o problema integrado com reduzido custo computacional. Em [Türkoğulları et al., 2010], os mesmos autores apresentam outra heurística para o problema integrado acima descrito. Nessa heurística, dividida em três partes, primeiramente são determinados as melhores localizações possíveis para o sorvedouro. Em seguida, o problema original é reformulado, levando-se em conta os possíveis subconjuntos de sensores assim como os resultados obtidos na primeira fase. Na terceira parte, a relaxação linear da reformulação é resolvida através do procedimento de geração de colunas [Desaulniers et al., 2005] na raiz. A solução ótima da relaxação é então utilizada para encontrar uma solução viável para o problema original, a partir de testes consecutivos a procura de subconjuntos de sensores que apresentem o maior tempo de vida dentre todos os demais. Os resultados apresentados mostraram que essa heurística foi mais eficiente que a anteriormente proposta.

2.3 A Nossa Contribuição

Como visto nas Seções anteriores, problemas de otimização complexos e, por vezes, com objetivos conflitantes surgem durante o projeto das RSSFs. Com a necessidade de se alcançar bons níveis de requisitos de QoS juntamente com as reduções no consumo de energia da rede, a busca por mecanismos que consigam balancear tais exigências torna-se uma necessidade.

Neste trabalho introduzimos algoritmos de otimização que permitem resolver alguns desses problemas de otimização comuns em uma RSSF, ao mesmo tempo em que minimiza-se o consumo de energia da rede por todo seu horizonte de planejamento. A rede aqui considerada é homogênea, envolve somente *um* nó sorvedouro e centenas de nós sensores distribuídos aleatoriamente em um ambiente externo. A área a ser monitorada, modelada por um quadrado no plano Euclidiano, abrange conjuntos discretizados de pontos de demanda, todos com exigências uniformes de sensoriamento. Por simplicidade de cálculos, será considerado o modelo linear de descarga de bateria, embora modelagens analíticas diferentes da utilizada consigam representar melhor o tempo de vida da bateria [Jongerden & Haverkort, 2008].

Objetivando a minimização no consumo de energia da rede, o que induz a um prolongamento no tempo de vida da mesma, propomos a integração dos problemas de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro através de uma abordagem multi-período e referenciado neste trabalho como PICDCR. Dessa forma, procuramos soluções que, para todos os períodos, garantam que um subconjunto de sensores permanecerá ativo, cobrindo todos os pontos de demanda e roteando eficientemente toda a informação gerada até o sorvedouro, o qual estará otimamente localizado na área monitorada. Apesar de vários trabalhos encontrados na literatura apresentarem certo grau de integração entre os problemas descritos anteriormente, nenhum apresentou o nível de integração aqui considerado.

Considerando os bons resultados obtidos pelos algoritmos de otimização, introduzimos um arcabouço de simulação que recebe como entrada as soluções encontradas pelos algoritmos propostos e avalia alguns dos requisitos de QoS de uma RSSF, tais como atraso médio e taxa de entrega de mensagens. Os resultados de otimização e simulação indicam que os algoritmos permitiram reduzir o consumo de energia da rede, além de apresentarem soluções com bons níveis de QoS. Quando comparado ao problema semelhante proposto por Türkogullari et al. [Türkogullari et al., 2010], o PICDCR apresentou melhoras significativas no consumo de energia, conseguindo estender o tempo de vida útil da rede. No capítulo seguinte, é apresentado o modelo proposto para o PICDCR.

Capítulo 3

Um Modelo Integrado para o Problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro em RSSFs

Neste capítulo, introduzimos um novo problema de Otimização Inteira Mista ao propormos um modelo integrado para simultaneamente tratar as questões de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro em RSSFs através de uma abordagem multi-período. Apesar de muitas tentativas de integração entre alguns dos problemas estudados terem sido propostas na literatura, a julgar pela nossa revisão bibliográfica, nenhuma apresentou o nível de integração que propomos. O problema integrado é formulado através de um Problema de Otimização em Grafos e de Programação Inteira Mista.

3.1 Motivação

Em um trabalho recente, Türkoğulları et al. [Türkoğulları et al., 2010] propõem uma formulação baseada em fluxo em redes objetivando a maximização do tempo de vida de uma RSSF ao integrar os problemas de Cobertura, Controle de Densidade, Roteamento e Localização do Sorvedouro assim como de Sensores, a partir de um conjunto finito de posições candidatas. Para isso, trabalham com uma rede heterogênea, onde a

dispersão dos sensores é feita manualmente. Apesar desse tipo de dispersão ser útil em aplicações envolvendo ambientes internos (como monitoramento de intrusos em uma galeria, por exemplo), as RSSFs são mais comumente usadas em ambientes externos, cuja dispersão usualmente é feita de forma aleatória (sensores são lançados em uma determinada região através de um avião, por exemplo), devido à certas dificuldades encontradas na região monitorada (regiões inóspitas ou de difícil acesso, por exemplo). Dessa forma, o problema de localização dos Sensores torna-se desnecessário, uma vez que não há necessidade de se encontrar posições ótimas para os mesmos na região monitorada.

Além disso, a estrutura topológica assumida para a rede é representada por um grafo conexo, composto por vértices vinculados aos sensores e posições candidatas para o sorvedouro na rede, além de um conjunto de arcos que são gerados levando-se em conta a distância Euclidiana entre os nós e o raio de comunicação, R_c . Isso induz a acreditar que mais energia poderá ser consumida pelos sensores, uma vez que os mesmos podem comunicar-se simultaneamente com mais de um nó por vez (em detrimento à representação através de uma árvore, no qual cada nó comunica-se somente com um outro sensor). A partir do que foi dito, em tese, ganhos adicionais em termos energéticos podem ser obtidos, ao permitir que a representação topológica da rede possa ser feita através de uma árvore, assim como a possibilidade da mobilidade periódica do sorvedouro.

A seguir, introduzimos um novo modelo que integra de forma eficaz o problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro através de uma abordagem multi-período.

3.2 Modelo Proposto

O PICDCR considera que uma RSSF monitora uma dada região por T^1 períodos de igual duração, d . Dessa forma, a rede apresenta um horizonte de planejamento igual a Td . Seja S o conjunto de nós sensores, D o conjunto de pontos de demanda e M o conjunto de posições candidatas para o sorvedouro na rede, todos esses representados por pontos no plano Euclidiano. O problema consiste em minimizar o consumo de energia da rede ao escolher, a cada período $t = \{1, \dots, T\}$, uma posição candidata $m \in M$ para o sorvedouro, enquanto esse recebe dados, através de rotas eficientes em termos energéticos, de um conjunto $S' \subseteq S$ de nós sensores que permanecem ativos e cobrindo todos os pontos de demanda $p \in D$.

¹Assuma que, por um abuso de notação, $T = |T|$ durante toda a dissertação.

Para formular o PICDCR como um problema de Otimização em Grafos, considere primeiramente um digrafo $\Theta = (V, A)$. Assuma que, por um abuso de notação, o conjunto de vértices, V , seja definido como $V = S \cup M$, onde cada sensor $s \in S$, e cada posição candidata para o sorvedouro, $m \in M$, representem um vértice em Θ . Seja $A := \{(i, j) : d_{ij} \leq R_c, \forall i \in V, j \in S\}$ o conjunto de arcos que representa todas as possíveis conexões entre os vértices, com d_{ij} sendo a distância Euclidiana entre i e j . Assumimos que o custo de um arco é definido como a energia gasta ao se transmitir uma mensagem entre seus vértices e está relacionado à distância Euclidiana entre os mesmos. Considere também que o subconjunto de S , $\alpha(p) := \{i \in S : d_{ip} \leq R_s\}$, represente os sensores capazes de cobrir o ponto de demanda $p \in D$. Assuma que $\Upsilon^t = (V^t, A^t)$ seja um subgrafo de Θ que induza uma arborescência não geradora no período $t \in T$, tendo-se como raiz o sorvedouro, localizado em uma única posição candidata $m \in M$ no período t , e formada por um conjunto de arcos A^t e de vértices V^t , cada um dos quais associados somente a nós sensores que estão ativos durante o intervalo de tempo que inicia em $(t-1)d$ e termina em td .

Dadas as características do PICDCR, o conjunto de vértices V^t que induzem uma determinada arborescência não geradora no período $t \in T$ e raiz $m \in M$, pode mudar entre um intervalo de tempo e o subsequente. Isto é verdade pois, uma vez que um nó sensor $s \in S$ pertença a alguma arborescência, esse passa a apresentar um consumo energético relacionado às operações de comunicação, sensoriamento e processamento. Dessa maneira, o mesmo pode vir a ser desativado em algum período posterior a t , caso a energia disponível em sua bateria, e_s^t , não seja suficiente para comportar o gasto de energia, g_s^t , necessário para mantê-lo ativo durante tal período.

Assim, o PICDCR consiste em encontrar T arborescências não geradoras em Θ , uma para cada $t = \{1, \dots, T\}$, de forma que, ao final do horizonte de planejamento, a energia residual da rede (soma da energia que resta na bateria de todos os nós sensores, $\sum_s^S e_s^T$) seja maximizada.

Além disso, escolhida uma arborescência não geradora para um período $t \in T$ qualquer, a intersecção entre o conjunto de vértices que a constitui, V^t , e $\alpha(p)$, $\forall p \in D$ deve ser maior ou igual a um, indicando que todo ponto de demanda, $p \in D$, foi coberto por pelo menos um nó sensor. Uma vez que cada sensor ativo apresenta um custo energético para permanecer nesse estado, deve-se atentar também para seu consumo de energia a cada período t , de forma que esse não exceda a energia disponível em sua bateria.

Diante do exposto, o PICDCR consiste em:

$$\max \sum_s^S e_s^T \quad (3.1)$$

tal que

$$t \in T : (V^t, A^t) \text{ induza uma arborescência não geradora de } \Theta, \quad (3.2)$$

$$|V^t \cap \alpha(p)| \geq 1, \quad \forall t \in T, p \in D, \quad (3.3)$$

$$e_s^t \geq g_s^t, \quad \forall s \in S, t \in T. \quad (3.4)$$

Uma vez que a topologia da solução, para cada período $t \in T$, é um subgrafo de Θ que induz uma arborescência não geradora (3.2), conectando o sorvedouro (raiz), localizado em uma única posição candidata $m \in M$ no período t , a cada sensor ativo, existe um caminho (rota) para o envio de todas as informações roteadas pela rede até o sorvedouro, como definido. Nossa meta é portanto, encontrar soluções que sejam eficientes em termos energéticos, de forma a maximizar a energia residual da rede (3.1) sem que o gasto de energia dos sensores no período $t \in T$ exceda a energia restante em suas baterias no início desse período (3.4). Ao mesmo tempo, deve-se garantir que todo ponto de demanda $p \in D$ seja coberto por pelo menos um nó sensor ativo a cada período $t \in T$ (3.3).

O PICDCR é de difícil resolução exata, uma vez que apresenta uma extensão do problema de conjuntos dominantes, comprovadamente NP-Completo [Garey & Johnson, 1979], como subproblema.

3.3 Formulação baseada em Desigualdades de Eliminação de Circuitos Direcionada

Para apresentarmos uma formulação para o PICDCR baseada em Desigualdades de Eliminação de Circuitos Direcionada (*DCSET*), primeiro consideramos o digrafo $\Theta = (V, A)$ como definido anteriormente.

Em seguida, estendemos Θ , gerando o digrafo Θ' da mesma forma como proposto por [Quintão et al., 2010] ao trabalhar com o problema da árvore de cardinalidade k .

Tal procedimento foi realizado para que pudéssemos modelar a escolha das arborescências não geradoras Υ^t induzidas pelo grafo Θ a cada período $t \in T$ através de uma arborescência geradora. Para isso, foram adicionados ao grafo original dois vértices artificiais, $n + 1$ e $n + 2$, ambos sem funcionalidade na rede, além de arcos artificiais com custos nulos. Dessa maneira, o conjunto de arborescências geradoras

induzidas em Θ' , uma para cada período $t \in T$, pode ser mapeado diretamente como uma solução do PICDCR. Isso é possível ao fazer-se com que $n + 1$ seja a raiz de cada uma dessas arborescências, e que toda Υ^t encontrada para o período t se conecte à raiz através do vértice $n + 2$.

Para isso, considere $\Theta' = (V', A')$, tal que o conjunto de vértices seja definido como $V' = V \cup \{n + 1, n + 2\}$. Seja o conjunto de arcos $A' := A^1 \cup A^2 \cup A^3$, com $A^1 := A$, $A^2 := \{(n + 2, i) : i \in M\}$, $A^3 := \{(n + 1, i) : i \in V\} \cup \{(n + 1, n + 2)\}$, sendo os dois últimos conjuntos de arcos artificiais com custos nulos, enquanto que o primeiro terá em cada arco (i, j) o custo, Et_{ij} , definido como a energia gasta ao se transmitir uma mensagem entre seus vértices i e j . De posse disso, considere que uma arborescência geradora, com raiz em $n + 1$, seja induzida por Θ' a cada período de tempo $t \in T$, onde todos os vértices em S que estejam sendo utilizados por essa arborescência (nós sensores ativos) estarão conectados direta ou indiretamente à única posição candidata escolhida para o sorvedouro, $m \in M$, naquele período t . Além disso, m estará conectado diretamente a $n + 2$, e todos os demais vértices que não estiverem sendo usados (sensores desativados ou posições candidatas para o sorvedouro não escolhidas) estarão diretamente conectados à raiz dessa arborescência. Assuma também que, um subconjunto próprio W de V' , $(W, V' \setminus W) := \{(i, j) \in A : i \in W, j \in V' \setminus W\}$, define o corte implicado por W .

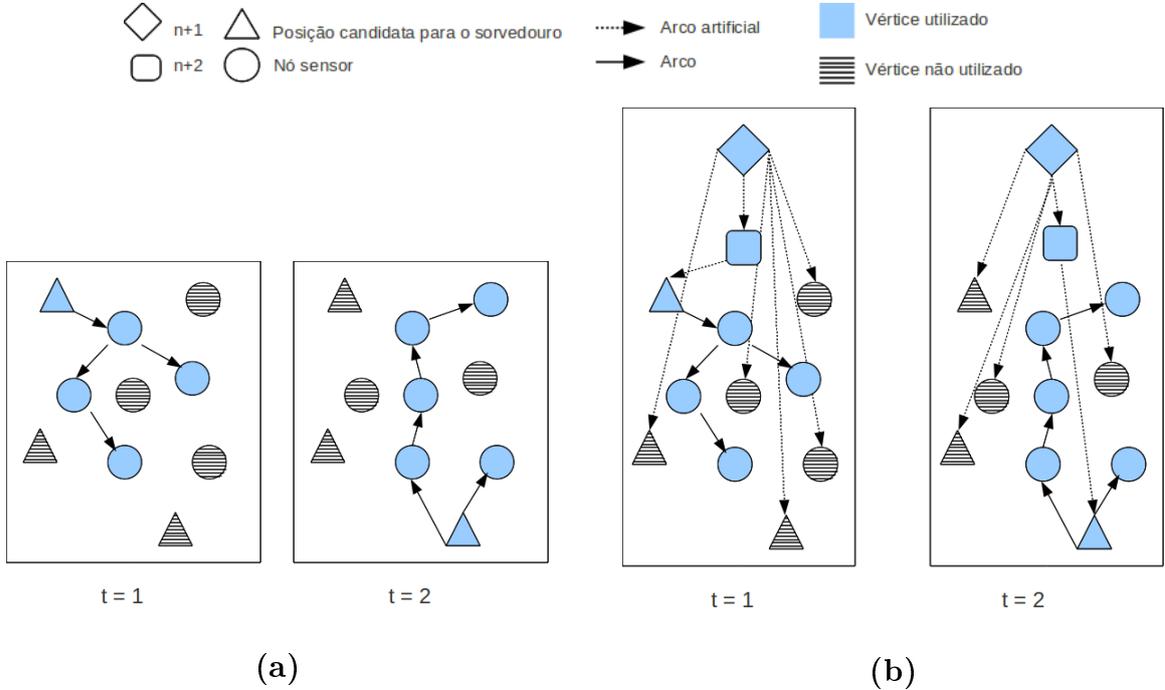


Figura 3.1. Um exemplo de como mapear Θ em Θ'

A Figura 3.1 mostra um exemplo, contendo dois períodos, de como uma arbores-

cência não geradora induzida em Θ pode ser mapeada em uma arborescência geradora induzida em Θ' , com raiz em $n + 1$. Na Figura 3.1a é mostrado duas arborescências não geradoras induzidas em Θ (uma para cada período). Perceba na Figura 3.1b que nas arborescências geradoras somente há um arco $(n + 2, m)$ saindo de $n + 2$ e indo para uma Υ^t , $t = \{1, 2\}$, cuja raiz é uma posição candidata para o sorvedouro, $m \in M$. Se $(n + 2, m)$ é removido dessas arborescências, dois componentes fracamente conectados aparecem. O primeiro contém os vértices $n + 1$, $n + 2$ e todos os vértices $V' \setminus V^t$, $t = \{1, 2\}$. O segundo componente é a arborescência induzida em Θ (Figura 3.1a), cuja raiz é uma posição candidata para o sorvedouro, $m \in M$.

Para apresentar uma formulação baseada em Desigualdades de Eliminação de Circuitos Direcionada para o PICDCR, empregamos os seguintes conjuntos de variáveis:

- $z_v^t \in \{0, 1\}, \forall v \in M \cup S, t \in T$ assumindo valor 1 se o vértice v está ativo no período t (se $v \in M$ implica que uma posição candidata para o sorvedouro foi escolhida e se $v \in S$ implica que o nó sensor *está* ativado no período t , $z_v^t = 0$ caso contrário);
- $y_v^t \in \{0, 1\}, \forall v \in S, t \in T$ assumindo valor 1 se o sensor v foi ativado no período t ($y_v^t = 0$ caso contrário);
- $x_{(i,j)}^t \in \{0, 1\}, \forall (i, j) \in A^1 \cup A^2 \cup A^3, t \in T$ assumindo o valor 1 se o arco (i, j) é usado no roteamento durante o período t (0, caso contrário);
- $e_v^t \in \mathbb{R}, \forall v \in S, \forall t \in T$ representa a energia disponível no sensor v , ao final do período t .

Os parâmetros usados no modelo são dados a seguir.

- $\alpha(p)$, como definido anteriormente e sendo o conjunto de nós sensores que cobrem o ponto de demanda p ;
- Ea , uma constante que representa a energia gasta para ativar o rádio de um nó sensor (mAh);
- $Et_{ij}, i \in S, j \in SUM$, energia gasta com a transmissão de dados entre os sensores i e j (mAh);
- Er , energia gasta pelo sensor com a recepção de dados (mAh);
- Em , energia gasta para o sensor se manter ativo (relacionada a energia gasta no processamento e sensoriamento) (mAh);

- Eb , Capacidade inicial da bateria dos nós sensores (mAh);

Uma formulação baseada em Desigualdades de Eliminação de Circuitos Direcionada para o PICDCR é dada por:

$$w = \max \left\{ \sum_{s \in S} e_s^T : (x, y, z, e) \in P_{CUTS} \cap \mathbb{B}^{|A'|} \times \mathbb{B}^{|S|} \times \mathbb{B}^{|V|} \times \mathbb{R}^{|S|} \right\} \quad (3.5)$$

onde P_{CUTS} é o poliedro definido pelas restrições (3.6)-(3.19):

$$\begin{aligned} y_s^t &\geq z_s^t - z_s^{t-1}, \quad s \in S, t \in T, t \geq 1, \\ y_s^t + z_s^{t-1} &\leq 1, \quad s \in S, t \in T, t \geq 1, \\ y_s^t &\leq z_s^t, \quad s \in S, t \in T, t \geq 1, \\ y_s^0 - z_s^0 &\geq 0, \quad s \in S, \end{aligned} \quad (3.6)$$

$$\begin{aligned} e_s^t &= Eb - Eay_s^t - Er \left(\sum_{j \in S \setminus \{s\}} x_{sj}^t \right) - \sum_{v \in SUM \setminus \{s\}} Et_{vs} x_{vs}^t - Em.z_s^t, \quad s \in S, t \in T, t = 0, \\ e_s^t &= e_s^{t-1} - Eay_s^t - Er \left(\sum_{j \in S \setminus \{s\}} x_{sj}^t \right) - \sum_{v \in SUM \setminus \{s\}} Et_{vs} x_{vs}^t - Em.z_s^t, \quad s \in S, t \in T, t \geq 1, \end{aligned} \quad (3.7)$$

$$z_v^t = 1 - x_{n+1,v}^t, \quad \forall v \in M \cup S, \forall t \in T, \quad (3.8)$$

$$\begin{aligned} x_{n+1,v}^t + x_{vj}^t &\leq 1, \quad \forall (v, j) \in A^1 \cup A^2, \\ \forall v \in M \cup S, \forall j \in S, \forall t \in T, \end{aligned} \quad (3.9)$$

$$x_{n+1,n+2}^t = 1, \quad \forall t \in T, \quad (3.10)$$

$$\sum_{m \in M} x_{n+2,m}^t = 1, \quad \forall t \in T, \quad (3.11)$$

$$\sum_{i \in W, j \in V' \setminus W} x_{ij}^t \geq 1, \quad \forall W \subset V', n+1 \in W, \forall t \in T. \quad (3.12)$$

$$\sum_{((j,i) \in A^1 \cup A^2)} x_{ji}^t = z_i^t, \forall i \in S, \forall t \in T, \quad (3.13)$$

$$x_{n+2,m}^t = z_m^t, \forall m \in M, \forall t \in T, \quad (3.14)$$

$$\sum_{v \in \alpha(p)} z_v^t \geq 1, \forall p \in D, \forall t \in T, \quad (3.15)$$

$$0 \leq z_v^t \leq 1, \forall v \in M \cup S, t \in T, \quad (3.16)$$

$$0 \leq y_v^t \leq 1, \forall v \in S, t \in T, \quad (3.17)$$

$$0 \leq x_{(i,j)}^t \leq 1, \forall (i,j) \in A, t \in T, \quad (3.18)$$

$$e_s^t \geq 0, \forall s \in S, \forall t \in T. \quad (3.19)$$

Como existe a preocupação quanto ao consumo de energia da rede, a função objetivo (3.5) é dada pela maximização da soma da energia disponível pelos nós sensores ao final do último período. Em outras palavras, pretende-se maximizar a energia residual da rede ao final do horizonte de planejamento.

O conjunto de restrições (3.6) relacionam o estado dos nós sensores, evitando que um sensor possa ser ativado em um período t caso ele já tenha sido ativado no período $t-1$. O balanço de energia é feito nas restrições (3.7). As Desigualdades de Eliminação de Circuitos Direcionada (3.12) e as restrições (3.8)-(3.11) asseguram a conectividade entre cada sensor ativo e o vértice $n+1$. As restrições (3.13),(3.14) garantem que a todo vértice ativo deverá incidir um arco. As desigualdades de cobertura (3.15) garantem que, a cada período, deverá existir pelo menos um nó sensor ativo coletando os dados de cada ponto de demanda. O domínio das variáveis é definido em (3.16)-(3.19).

O modelo, como proposto, impossibilita a retirada de informações acerca da quantidade de dados que cada sensor envia/recebe. Dessa forma, assumimos que cada nó utilizado para retransmitir os dados até o sorvedouro primeiramente empacotará todas as mensagens recebidas de seus filhos juntamente com as suas próprias, e em seguida enviará essa nova mensagem agrupada pela rede. Consideramos que o tempo para realizar tal operação é irrisório e seu gasto energético está embutido em Em . Assim, como forma de simplificar os cálculos do consumo de energia com a transmissão e recepção, ao início de cada período, os sensores ativos enviam um único pacote de dados pela rede. Além disso, o consumo de energia ao se enviar/receber tal pacote será sempre igual ao consumo máximo de energia gasto para isso, e não proporcional ao tempo de

envio, assim como feito em um ambiente simulado.

Como pode ser verificado, P_{CUTS} emprega exponencialmente muitas restrições do tipo (3.9) e (3.12). Dessa forma, no próximo capítulo será apresentado um algoritmo exato, do tipo Branch-and-Cut, que procura adicionar tais restrições ao modelo à medida que as mesmas forem violadas.

Capítulo 4

Métodos de Solução Exata para o PICDCR

Neste capítulo apresentamos um algoritmo exato para a resolução do PICDCR, baseado na formulação *DCSET*. Trata-se de um algoritmo Branch-and-Cut. Os resultados obtidos com o algoritmo indicam que a dificuldade inerente ao se integrar alguns dos problemas de otimização encontrados em uma RSSF tornaram o problema muito difícil de ser resolvido, mesmo para instâncias de pequenas dimensões.

4.1 Um Algoritmo Branch-And-Cut

O algoritmo Branch-and-Cut (BC) [Padberg & Rinaldi, 1991] pode ser visto como um algoritmo Branch-and-bound [A.Land & Doig, 1960] no qual tenta-se, de forma sistemática, obter relaxações cada vez mais fortes através da introdução de planos de corte, em cada nó da árvore de busca [Lucena & Beasley, 1996].

Como dito anteriormente, P_{CUTS} apresenta um número exponencial de restrições. Dessa forma, adicionar todas ao modelo e resolvê-lo através de um algoritmo Branch-and-Bound é inviável mesmo para instâncias do PICDCR de tamanhos reduzidos. Ao invés disso, introduziremos as *DCSETs* em relaxações lineares para (3.5) na medida em que forem necessárias.

Assim sendo, implementamos o algoritmo BC para o PICDCR baseado na formulação de desigualdades de eliminação de circuitos direcionada, a partir de todas as funcionalidades oferecidas pelo pacote de otimização CPLEX [sol, 2010] (versão 10.2 com configuração padrão). Todos os pré-processamentos e rotinas de separação de desigualdades válidas efetuadas pelo CPLEX foram desabilitadas, de forma que, somente os cortes explicados a seguir foram separados.

De posse disso, o algoritmo BC inicia resolvendo o problema linear:

$$\max \left\{ \sum_{s \in S} e_s^T : (x, y, z, e) \in \overline{P}_{CUTS} \right\} \quad (4.1)$$

onde $x := (x^1, \dots, x^T)$, $y := (y^1, \dots, y^T)$, $z := (z^1, \dots, z^T)$, $e := (e^1, \dots, e^T)$ e o poliedro \overline{P}_{CUTS} é dado pela intersecção de (3.6)-(3.8),(3.10)-(3.11),(3.13)-(3.15).

Assuma que $(\bar{x}, \bar{y}, \bar{z}, \bar{e}) \in \overline{P}_{CUTS}$ seja uma solução para (4.1) e $\Omega^t = (\overline{V}^t, \overline{A}^t)$, o subgrafo de Θ' que represente cada período $t \in T$, induzidos por $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$ (onde $\overline{V}^t := \{s \in S \cup M : \bar{z}_s^t > 0, t \in T\}$ e $\overline{A}^t := \{(i, j) \in A : \bar{x}_{(i,j)}^t \geq 0, t \in T\}$). Se $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$ é inteiro e se existe um caminho direto de $n+1$ para todo $s \in V' \setminus \{n+1\}$ em Ω^t , $\forall t \in T$, $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$ resolve (3.5). Caso contrário, procuramos identificar as desigualdades (3.9) e (3.12) violadas por $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$, para reforçar \overline{P}_{CUTS} e (4.1) é reotimizado. Esse processo segue até que nenhuma desigualdade violada seja identificada.

A separação das desigualdades (3.9) é conduzida por inspeção. Todas as desigualdades nessa classe que forem violadas serão incluídas em \overline{P}_{CUTS} . Na sequência, procuramos por violações no conjuntos de cortes (3.12) em tempo polinomial, através da execução de uma série de $|V'| - 1$ algoritmos de fluxo máximo (corte mínimo), um para cada par $n+1$ e $s \in V' \setminus \{n+1\}$, a cada $t \in T$. Mais precisamente, dado um $t \in T$, procuramos um corte de capacidade mínima que separa $n+1$ de $s \in V' \setminus \{n+1\}$, na rede dada por Ω^t e capacidade dos arcos $\{\bar{x}_{(i,j)}^t : (i, j) \in \overline{A}^t\}$.

Considere que $(W, \overline{V}^t \setminus W)$ defina o corte de capacidade mínima ($n+1 \in W$). Sempre que a capacidade $\sum_{i \in W, j \in \overline{V}^t \setminus W} \bar{x}_{ij}^t$ for menor que 1, uma desigualdade *DCSET*

$$\sum_{i \in W, j \in \overline{V}^t \setminus W} x_{ij}^t \geq 1, \forall W \subset V', n+1 \in W, \forall t \in T, \quad (4.2)$$

é violada por $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$. Se nenhuma desigualdade violada por $(\bar{x}, \bar{y}, \bar{z}, \bar{e})$ for encontrada, a árvore de busca é ramificada, priorizando-se a variável z para isso, com a opção *strong branching* do CPLEX ativada.

Para a identificação dos cortes de mínima capacidade, empregamos uma implementação do algoritmo de fluxo máximo direcionado, descrito em Dinic [Dinitz, 2006].

4.2 Resultados do Algoritmo Branch-and-Cut

Nesta seção serão apresentados os resultados obtidos através do nosso algoritmo BC. Os resultados foram gerados a partir de um conjunto de instâncias cujos parâmetros refletem aqueles encontrados durante a organização idealizada de uma RSSF.

4.2.1 Instâncias

Para conduzir os experimentos computacionais, instâncias de testes foram geradas priorizando-se a variação dos seguintes parâmetros: número de sensores (entre 400 a 600), quantidade de possíveis localizações para o sorvedouro na rede (entre 5 a 20) e número de períodos (entre 4 a 8). Para isso, primeiramente é considerada uma área de monitoramento modelada como um quadrado de lado igual a $200m$, possuindo 40000 pontos de demanda ($1 \text{ ponto}/m^2$), separados por uma distância de $1m$. Os parâmetros de configuração dos sensores foram baseados no modelo de sensor comercialmente disponível, Mica2 [mic, 2011], e é sumarizado na Tabela 4.1.

Parâmetro	Valor
Energia da bateria	400mAh
Raio de sensoriamento (R_s)	15m
Raio de comunicação (R_c)	30m
Duração de um período (d)	1h
Energia máxima gasta com transmissão (Et)	8.9mAh
Energia gasta com recepção (Er)	10mAh
Energia gasta com a ativação do rádio (Ea)	7mAh
Energia gasta no processamento	8mAh
Energia gasta no sensoriamento	5mAh
Energia gasta na manutenção do sensor (Em)	13mAh
Tamanho do <i>buffer</i>	4 Kbytes
Velocidade do sorvedouro	1m/s
Frequência do rádio	916MHz
Largura de banda	38.4kbps

Tabela 4.1. Parâmetros

As instâncias foram geradas de forma aleatória, a partir de uma distribuição uniforme. São geradas três instâncias de teste diferentes a partir de uma mesma quantidade específica de sensores. Para cada uma, é gerado um conjunto de posições candidatas para o sorvedouro, M , de forma que cada posição contenha uma grande quantidade de nós em seu entorno. Essa estratégia mostra-se importante pois, quanto mais sensores próximos o suficiente do sorvedouro, menor o consumo de energia gasto pela rede para a transmissão de dados (uma vez que a transmissão pode ser feita diretamente). Além disso, menor também será a sobrecarga de tráfego nesses sensores, o que permite estender o tempo de vida da rede [Li & Mohapatra, 2007]. Dessa forma, os elementos de M são gerados a partir dos seguintes passos:

1. Escolha aleatoriamente um nó sensor, $s \in S$;
2. Selecione todos os nós sensores que estão a no máximo R_c de distância de s ;
3. Calcule o centro de massa, Cm_s , do conjunto de nós selecionados;
4. Crie um conjunto C_s , cujos elementos são sensores que estão a no máximo R_c de distância de Cm_s ;

5. Repita os passos anteriores para cada sensor disponível na rede de forma a criar vários conjuntos C_s , porém lembrando-se que, uma vez que algum sensor já esteja contido em algum C_s , o mesmo não poderá participar da construção de um outro conjunto;
6. Escolhida uma cardinalidade para M , ordene os conjuntos C_s gerados, e selecione os de maior cardinalidade, de forma que seu centro de massa seja um elemento de M ;

Caso a cardinalidade escolhida para M seja maior que a quantidade de conjuntos C_s gerados, calcula-se o ponto médio entre cada par de elementos pertencentes a M e, aqueles que apresentarem as maiores quantidades de sensores em seu entorno serão selecionados para completar tal conjunto. Apesar de simples, essa abordagem mostrou-se bem eficaz em reduzir a distância entre as posições candidatas para o sorvedouro, o que permite diminuir o tempo de deslocamento do sorvedouro em um ambiente simulado.

Uma vez que são geradas três instâncias para cada combinação entre os parâmetros que estão sendo variados, em toda a dissertação, será apresentada a média das soluções encontradas entre essas três instâncias.

No intuito de explicitar como o consumo de energia é reduzido com o aumento tanto do número de sensores quanto da quantidade de posições candidatas para o sorvedouro na rede, as instâncias foram criadas levando-se em consideração os valores anteriores de tais parâmetros. Por exemplo, as instâncias com 500 sensores foram geradas tomando-se como base as de 400 sensores anteriormente criadas, acrescidas com mais 100 sensores, gerados aleatoriamente. O mesmo aconteceu para instâncias onde o tamanho de M variou. Por exemplo, instâncias com $|M|$ igual a 10 foram geradas a partir daquelas cuja cardinalidade é igual a 5, bastando acrescentar ao conjunto mais 5 diferentes posições candidatas para o sorvedouro na rede. Isso permite garantir que o resultado obtido para uma dada instância será, no pior caso, não pior que sua instância geradora.

A relação entre a energia gasta na transmissão de uma mensagem, Et , e a distância entre os sensores é dada pela Tabela 4.2. Esses valores se baseiam no consumo de energia gasto na transmissão de dados do sensor Mica2 [mic, 2011].

4.2.2 Resultados computacionais

É importante mencionar que todos os experimentos computacionais apresentados nessa dissertação foram conduzidos em um computador com processador *Intel Xeon Quad*

Energia (mAh)	Distância (metros)
5,3	5,3347
6,9	5,9855
7,1	6,7162
7,1	7,5355
7,1	8,4549
7,4	9,4866
7,4	10,644
7,4	11,943
7,6	13,401
7,6	15,035
7,9	16,870
7,9	18,928
8,2	21,238
8,4	23,829
8,7	26,737
8,9	30,000

Tabela 4.2. Relação entre a corrente gasta ao se transmitir uma mensagem e a distância

Core, que opera em 2.27GHz e possui 16 Gb de memória RAM. O sistema operacional Linux foi utilizado. Os algoritmos foram implementados em C/C++ e compilados com o gcc/g++, usando-se as *flags* -O3 de otimização.

Para a execução do algoritmo, foi imposto um tempo limite de no máximo 10 horas. Caso esse tempo seja alcançado, a execução do algoritmo BC é interrompida, e se possível, os melhores limites inferior e superior são recuperados. Devido à complexidade inerente à integração dos problemas tratados neste trabalho, dentre todas as instâncias testadas, o algoritmo BC proposto foi capaz de encontrar soluções viáveis para somente 5 destas.

A Tabela 4.3 mostra a primeira solução viável encontrada pelo algoritmo BC e aquela após 1 hora de execução. As primeiras colunas ($|S|$, $|M|$ e T) apresentam a quantidade de sensores, de posições candidatas para o sorvedouro na rede e de períodos utilizadas pelas instâncias, respectivamente. As colunas w , \bar{w} , mostram, respectivamente, a função objetivo da melhor solução viável encontrada, assim como o melhor limite dual superior encontrado ao longo da árvore de enumeração. Já as colunas *Gap* e t apresentam o *gap* de dualidade ($\frac{\bar{w}-w}{w}$) e o tempo de execução necessário até que tais soluções fossem encontradas. Por último, as colunas N_p e N_a mostram a quantidade de nós da árvore de enumeração que foram processados e aqueles a processar após o algoritmo ter sido executado por 1 hora, respectivamente. Os resultados mostram que o algoritmo proposto encontrou grandes dificuldades para resolver o PICDCR, mesmo para as menores instâncias testadas ($|S| = 400$ e $T = 4$). Isso é evidenciado ao analisar o tempo computacional gasto para o algoritmo encontrar a primeira solução viável. Apesar dessas soluções apresentarem um *gap* de dualidade baixo (no máximo 1,77%), o tempo computacional exigido para encontrá-las foi superior a 3 horas (o que justi-

fica nenhuma solução ter sido apresentada após 1 hora de execução). Além disso, ao aumentar-se a cardinalidade de M , houve uma redução tanto no número de nós processados quanto o de nós a serem processados, indicando que o aumento da dimensão das instâncias (valores maiores para os parâmetros $|S|$, $|M|$ e T) implica em maior esforço computacional para se processar cada nó da árvore de enumeração.

Instâncias			Primeira Solução Viável				Solução encontrada após 1 hora				
$ S $	$ M $	T	w	\bar{w}	$Gap(\%)$	$t(s)$	w	\bar{w}	$Gap(\%)$	N_p	N_a
400	5	4	145867,67	147676,67	1,24	13919,07	-	147676,67	-	36,00	37,00
400	5	5	142273,33	144783,33	1,77	23464,95	-	144783,33	-	39,67	40,67
400	10	4	145704,00	147747,67	1,40	20008,87	-	147747,67	-	17,00	18,00
400	15	4	145770,00	147752,00	1,36	19920,10	-	147758,00	-	7,00	8,00
400	20	4	145766,33	147752,33	1,36	22601,17	-	147776,00	-	1,33	2,33

Tabela 4.3. Primeira solução viável e após uma hora de execução do algoritmo BC

Na Tabela 4.4 são apresentados os resultados obtidos após 10 horas de execução do algoritmo proposto. As colunas $\#C_1$, $\#C_2$, tC_1 , tC_2 , tM , indicam respectivamente: a quantidade de restrições pertencentes ao conjunto de restrições (3.9) e (3.12) que foram adicionadas ao modelo, o tempo acumulado gasto para se adicionar tais restrições, e o tempo acumulado para se resolver o algoritmo de fluxo máximo. Percebe-se que a diferença entre a primeira solução viável encontrada e àquela retornada após 10 horas foi pequena, uma vez que seus *gaps* de dualidade ficaram bem próximos (Tabela 4.3). Isso pode indicar que o algoritmo, após encontrar a primeira solução viável, está gastando muito tempo para provar a otimalidade da mesma. Como era de se esperar, à medida que as instâncias tornam-se maiores em dimensão, o número de desigualdades adicionadas ao modelo aumenta. Observando os tempos gastos para adicionar o conjunto de restrições violadas ao modelo e executar o algoritmo de fluxo máximo, conclui-se que o alto custo computacional exigido pelo algoritmo para encontrar soluções viáveis não está relacionado aos procedimentos implementados para isso, uma vez que os mesmos consumiram um tempo bem inferior ao de execução.

Instâncias			Solução viável encontrada após 10 horas							
$ S $	$ M $	T	w	\bar{w}	$Gap(\%)$	$\#C_1$	$\#C_2$	$tC_1(s)$	$tC_2(s)$	$tM(s)$
400	5	4	145870,33	147654,33	1,22	9097,33	1782,00	21,19	23,72	795,39
400	5	5	142273,33	144763,67	1,75	10233,00	1976,67	21,76	35,41	952,47
400	10	4	145704,00	147723,67	1,39	9089,67	1745,33	19,05	26,04	706,86
400	15	4	145770,00	147743,33	1,35	9216,33	1340,33	18,40	12,47	687,32
400	20	4	145766,33	147747,67	1,36	9233,00	1205,33	16,48	16,32	658,13

Tabela 4.4. Solução viável encontrada após 10 horas de execução

4.3 Comentários

O método de solução exato apresentado neste capítulo não permitiu, ao nosso ver, resolver de forma satisfatória o PICDCR. Mesmo após horas de execução, em muitas instâncias não se pode encontrar soluções viáveis, principalmente quando o número de períodos e sensores era incrementado.

Entretanto, acreditamos que o desempenho do algoritmo BC possa ser melhorado através das seguintes ações:

- Estudo poliedral do problema;
- Investigação do uso de outras classes de desigualdades válidas como planos de corte;
- Implementação de heurísticas para serem utilizadas durante o processamento dos nós da árvore de enumeração pelo algoritmo BC.

Todas essas ações, entretanto, serão conduzidas posteriormente, como continuação dessa pesquisa.

Devido à dificuldade de se resolver o PICDCR de forma exata, no próximo capítulo apresentaremos uma heurística que resolve o problema de forma satisfatória, apresentando boas soluções com um custo computacional reduzido.

Capítulo 5

Método heurístico para o PICDCR

Neste capítulo apresentamos uma heurística para resolver o PICDCR. Devido ao alto custo computacional exigido pelo algoritmo exato proposto anteriormente, nosso objetivo é propor um algoritmo capaz de encontrar soluções viáveis, idealmente de boa qualidade, com tempos computacionais reduzidos o bastante para que instâncias de dimensões mais próximas das encontradas na prática possam ser testadas. A heurística proposta, descrita a seguir, é composta por uma fase construtiva, além de mecanismos de busca local e diversificação.

5.1 Busca Local

O mecanismo de Busca Local consiste em realizar determinadas operações em uma solução corrente no intuito de encontrar novas soluções no espaço de busca. O procedimento de Busca Local que propomos será o mesmo apresentado em [de Andrade et al., 2009]. Esse consiste em encontrar, para um determinado período $t \in T$, uma solução em árvore tendo o sorvedouro como raiz, de forma a obter reduções no consumo de energia da rede ao garantir que os problemas de controle de densidade e roteamento de dados sejam atendidos, enquanto tenta-se assegurar a cobertura completa da área monitorada. Para isso, o procedimento aplica, de forma ordenada e iterativa, os operadores de inserção, trocas e remoção de nós (explicados a seguir), sobre uma solução parcial inicialmente inviável. Entende-se por solução parcial inviável aquela que representa um período $t \in T$ e que inicialmente é composta por poucos ou mesmo nenhum nó sensor ativo, dessa forma não assegurando questões como cobertura, controle de densidade e roteamento. Como a solução é construída adicionando-se nós a essa, faz-se necessário que a aplicação dos operadores seja seguida na ordem estipulada.

Além disso, a aplicação dos operadores na ordem descrita anteriormente foi ne-

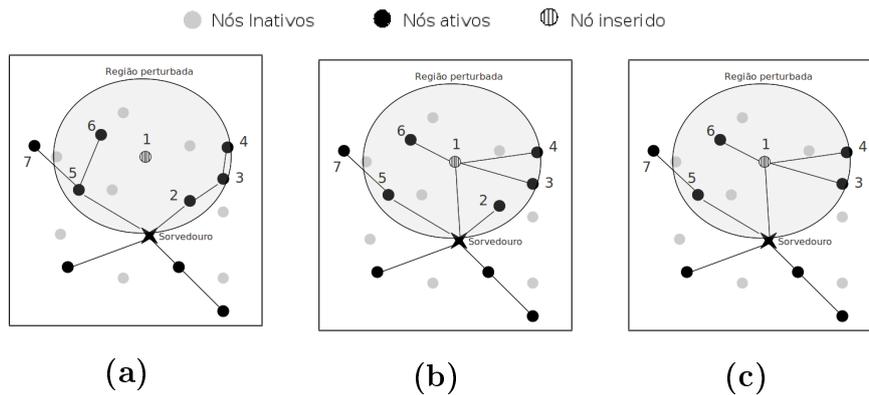


Figura 5.1. Um exemplo da reestruturação da rede após a inserção de um novo nó

cessária pois, ao se inserir (ativar) um novo nó sensor à solução corrente, esse aparecerá na vizinhança dos nós ativos que estão dentro da região circular, com centro sendo o novo nó e o raio igual a R_p . Esta região é chamada *região perturbada* e o raio R_p , ou *raio de perturbação*, é equivalente ao raio de comunicação, R_c . Essa perturbação na vizinhança dos nós ativos pode modificar a topologia da rede, uma vez que o aparecimento de um novo nó permite que algum sensor sobrecarregado com muitos filhos possa ceder alguns ao nó inserido. Isso possibilita o aparecimento de nós sem função de roteamento (não são necessários para retransmitir informações de outros sensores). Se, além disso, tais nós apresentarem áreas de monitoramento *completamente redundantes*, quer dizer, todos os pontos de demanda que esses sensores cobrem são cobertos por outros nós ativos, os mesmos poderão ser removidos (desativados) da solução.

A Figura 5.1 mostra a inserção de um nó sensor e a consequência disso sobre a topologia da rede. A inserção do nó 1 causa uma perturbação na vizinhança dos nós ao seu redor $\{\text{sorvedouro}, 2, 3, 4, 5, 6\}$ (Figura 5.1a). Essa perturbação permite que os nós 3, 4 e 6 sejam adotados pelo nó inserido (Figura 5.1b), o que deixa o nó 2 com nenhum filho e o nó 5 somente com um. Assim, o nó 2 perde sua contribuição no roteamento de dados gerados por outros sensores, e considerando que o mesmo tem sua área de cobertura completamente redundante, ele pode ser removido (desativado) da solução (Figura 5.1c).

O procedimento de Busca Local pode ser visto no Algoritmo 5.1. Ele recebe como entrada uma solução inicial, sol , e aplica sobre a mesma os operadores de inserção, troca e remoção (linhas 6-8), de forma a encontrar possíveis melhoras na solução corrente após a mesma ter sido perturbada pela inserção do sensor s . Isso é repetido até que todos os nós que não pertençam à solução corrente sol_tmp (estejam desativados), tenham sido testados (linha 3). Para avaliar a solução gerada (linhas 9-14), a cada

```

1 Procedimento Busca_Local(sol)
2 sol* ← sol;
3 para cada Sensor desativado, s, faça
4   | sol_tmp ← sol*;
5   | Aplicando operadores na vizinhança da solução corrente
6   | Insira s em sol_tmp;
7   | Realize, em sol_tmp, trocas entre sensores pertencentes à região perturbada por s;
8   | Remova, em sol_tmp, alguns sensores pertencentes à região perturbada por s;
9   | se  $f(sol\_tmp) < f(sol^*)$  então
10  |   | sol* ← sol_tmp;
11  |   fim
12  | senão
13  |   | sol_tmp ← sol*;
14  |   fim
15 fim
16 retorna sol*

```

Algoritmo 5.1: Procedimento de Busca Local

iteração faz-se uso de uma função de avaliação $f(\cdot)$, descrita em detalhes em 5.1.4. A melhor solução encontrada após todos os testes é retornada (linha 16).

5.1.1 Inserção

Como mencionado anteriormente, a operação de inserção é usada para ativar nós sensores que possam melhorar a solução corrente através da redução no consumo de energia da rede, causado pela reestruturação de sua topologia.

Para isso, o operador de inserção escolhe aleatoriamente um sensor a partir de um conjunto de nós candidatos (desativados) e, em seguida, ativa-o. O nó escolhido procura, em sua vizinhança, algum nó pai que possa adotá-lo, ou seja, que seja capaz de retransmitir seus dados. O nó pai escolhido deve ser aquele tal que a conexão entre ele e o sensor inserido cause o menor incremento possível no consumo de energia da rede. Se o novo nó sensor não puder encontrar um nó pai com essa propriedade, ele será desativado e um novo nó sensor será escolhido em seu lugar. Perceba que nós que já foram analisados, mas que não causaram melhorias à solução corrente, não precisam ser testados novamente até que a mesma seja modificada. A Figura 5.1a mostra o nó 1 sendo inserido e adotado pelo nó sorvedouro.

5.1.2 Troca

A operação de troca pode ser vista como uma tentativa de se reduzir o consumo de energia da rede ao possibilitar que o comprimento das rotas sensor-sorvedouro e/ou a quantidade de sensores conectados a um mesmo nó sejam decrementados. Assim,

considerando C o conjunto de nós ativos pertencentes à região perturbada, para cada $c \in C$, todas as possíveis conexões entre c e $c' \in C, c \neq c'$ são testadas, e aquelas que permitam que a reestruturação da árvore de roteamento proporcione a maior economia de energia, são escolhidas.

Para assegurar que o roteamento seja sempre garantido, a operação de troca exclui possibilidades que possam causar ciclos ou problemas de conectividade (nós que não podem suportar o desconto de energia após a reestruturação da árvore de roteamento) na rede. Logo, essa operação mantém pelo menos uma rota, sem ciclos, de todos os nós ativos até o sorvedouro. Na Figura 5.1b os nós 2, 3, 4, 5, 6 pertencem à vizinhança do nó inserido 1. No exemplo considerado, após vários testes de troca, os nós 3, 4 e 6 escolhem 1 como seu novo pai.

5.1.3 Remoção

Uma vez que o procedimento de Busca Local visa obter reduções no consumo de energia da solução atual, assegurando o roteamento de dados juntamente com a tentativa de se cobrir todos os pontos de demanda pelos nós ativos, pode-se afirmar que os nós sensores desempenharão dois tipos de papéis, não necessariamente excludentes, na solução: o de rotear dados e o de garantir cobertura. Dessa forma, a operação de remoção é usada para desativar nós que:

- apresentem área de monitoramento completamente redundante, e
- não sejam necessários para garantir o roteamento de dados.

Para a desativação de um nó sensor é necessário que ambos os critérios anteriores sejam atendidos, pois, uma solução pode conter sensores que são necessários para assegurar o roteamento de dados, mas desnecessários para garantir cobertura, uma vez que os pontos de demanda que os mesmos cobrem também são cobertos por outros nós ativos.

De posse disso, a operação de remoção procura primeiramente, dentro da região perturbada, nós sensores que atendam aos dois critérios, removendo-os da solução. Em seguida, procura-se por nós que atendam somente ao primeiro critério. Caso algum seja encontrado, aplica-se a esse nó um procedimento que procura por sensores ativos que possam adotar todos os seus filhos e cujo incremento no consumo de energia da rede seja o menor possível. Se após tal procedimento os nós encontrados passarem a atender também ao segundo critério, eles serão desativados. Caso contrário, serão mantidos na solução.

A Figura 5.1b mostra os nós 3, 4 e 6 sendo adotados pelo nó inserido 1, o que torna o nó 2 desnecessário para o roteamento de dados. Considerando que o nó 2 não contribui para a cobertura da área monitorada, ele será removido (5.1c). Perceba que, mesmo se o nó 5 possuir área completamente redundante, o mesmo não poderá ser desativado, uma vez que ele é importante para o roteamento dos dados gerados por 7.

5.1.4 Função de Avaliação

A função de avaliação é utilizada para medir a qualidade de uma solução encontrada para o período $t \in T$, a partir de algumas de suas características, tais como o consumo de energia por ponto de demanda coberto, a taxa de cobertura e a conectividade entre os sensores. Dessa forma, soluções idealmente de boa qualidade serão aquelas que apresentarem baixos valores numéricos para tal função, implicando que poucos nós sensores estarão ativos, mas que possuirão energia suficiente em suas baterias para manterem-se conectados a uma rota até o sorvedouro, além de cobrirem todos os pontos de demanda. Formalmente a função é definida como:

$$f(g, o, l) = \sum_{i \in S} \frac{g_i^t}{o_i^t} + (1 - \sum_{i \in S} \frac{o_i^t}{|D|}) + \sum_{i \in S} l_i^t \quad (5.1)$$

com

o_i^t , a quantidade de pontos de demanda que somente o sensor i cobre no período t ,

g_i^t , a energia consumida pelo sensor i no período t ,

l_i^t , variável de decisão que possui valor 1 caso o sensor i não possa garantir conectividade no período t , e 0 caso contrário.

A primeira soma na definição de f dado por (5.1) prioriza nós sensores que possuem baixo consumo de energia e grande área de cobertura. Dessa forma, tal somatório procura restringir, indiretamente, a ativação de vários sensores, já que do contrário, a área que cada sensor cobre com exclusividade poderia ser reduzida. Isso é reforçado no segundo somatório, no qual procura-se priorizar a maior área coberta pelos sensores a fim de reduzir falhas de cobertura na rede. A terceira soma é responsável por contabilizar quantos nós na solução não possuem energia suficiente para garantir o roteamento de dados, o que permite escolher soluções que apresentem poucos ou mesmo nenhum desses nós. Devido as diferentes magnitudes apresentadas por tais critérios, a função foi normalizada através do método *Min-max* [Han, 2005]. Esse método consiste em efetuar uma transformação linear a partir de vários critérios com magnitudes diferentes, porém preservando-se a relação entre os dados originais.

5.2 Critério de Diversificação

Como forma de garantir que melhores soluções sejam retornadas pelo procedimento de Busca Local, faz-se necessário que o mesmo receba soluções iniciais diversificadas o suficiente a fim de garantir que a busca não fique presa à mínimos locais. Usualmente, a construção de um conjunto de soluções diversificadas pode ser feita perturbando-se uma solução inicial. Para isso, utiliza-se algum procedimento que explore características específicas do problema tratado. Entretanto, um dos principais problemas envolvendo critérios de diversificação se dá na forma como tais perturbações são realizadas e com qual frequência as mesmas devem ser aplicadas. A segunda questão é especialmente importante, pois dependendo da intensidade da perturbação (se há uma pequena ou grande diferença entre a solução inicial e a que foi perturbada), a busca por novas soluções apresentará características distintas. Se a perturbação for forte (grande diferença entre a solução inicial e a perturbada), a solução será muito diversificada e o procedimento de busca local terá dificuldades em encontrar soluções melhores. Já se a perturbação for muito fraca, a busca local terá dificuldades de sair de mínimos locais.

Para resolver tais questões, propomos como critério de diversificação a perturbação de uma solução através da retirada (desativação) de alguns sensores ativos de forma aleatória. Como a solução para o problema tratado nesse trabalho é representada por uma árvore, ao se retirar um sensor ativo, deve-se retirar todos os demais (seus descendentes na árvore) que utilizam tal nó para retransmitir seus dados.

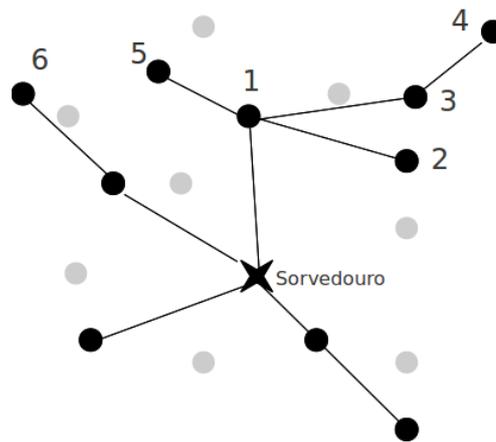


Figura 5.2. Exemplo de uma solução ao aplicar o critério de diversificação

A Figura 5.2 mostra uma solução para um determinado período. Ao perturbar tal solução, escolhendo o nó 1 para ser retirado, os nós 2, 3, 4 e 5 deverão ser retirados para preservar a integridade da rede. Medindo a intensidade da perturbação através do número de nós retirados, a escolha do nó 1 possibilitou uma perturbação de 50%. Já ao

escolher o nó 6 a intensidade será de apenas 10%. Dessa forma, a escolha aleatória do nó que será retirado permite que varie-se a intensidade da perturbação, possibilitando variar tanto o tamanho do espaço de busca quanto o esforço computacional exigido pela busca local.

5.3 ILS

Nesta seção será apresentado um procedimento construtivo que combina uma busca local e o critério de diversificação, previamente descritos nas seções 5.1 e 5.2, e que são guiados para explorar o espaço de soluções através da metaheurística ILS [Ramalhinho-Lourenço et al., 2000].

O ILS (*Iterated Local Search*) [Ramalhinho-Lourenço et al., 2000] é uma metaheurística cujo objetivo é obter iterativamente melhores soluções no espaço de busca a partir da aplicação de procedimentos de busca local e perturbação à uma solução inicial até que um critério de parada seja satisfeito. Para isso, o ILS é composto basicamente por três passos: Perturbação, Busca Local e critério de aceitação. Esta metaheurística recebe como entrada uma solução inicial e a ela aplica um procedimento de busca local. Em sequência, o ótimo local em relação à vizinhança escolhida é submetido a um procedimento de perturbação como forma de diversificar a solução, gerando uma solução intermediária. O procedimento de busca local é novamente aplicado à solução intermediária gerando uma nova solução. Esta é então avaliada por um critério de aceitação que pode ser, por exemplo, a melhor solução encontrada até o momento. Escolhida uma solução a partir de tal critério, os passos anteriores são repetidos. Este processo de alternar entre os procedimentos de busca local e perturbação é repetido no intuito de alcançar um mínimo local e continua até que algum critério de parada seja satisfeito.

A heurística que propomos consiste em encontrar soluções viáveis, partindo-se de um conjunto (possivelmente vazio) de nós sensores ativos que não assegura a viabilidade do problema tratado. Entende-se por soluções viáveis aquelas que garantem para todos os períodos, a escolha de um subconjunto de sensores que permanecerão ativos, rotas de dados sensor-sorvedouro eficientes em termos energéticos até a posição do sorvedouro, enquanto assegura-se que toda a área a ser monitorada seja coberta. Para alcançar isso, o algoritmo constrói uma solução período a período, garantindo que as condições de viabilidade sejam atendidas. Isso é mostrado no Algoritmo 5.2. A cada período t (linha 2), o algoritmo executa dois procedimentos (explicados detalhadamente a seguir): o ILS (Algoritmo 5.3) e o COBERTURA_COMPLETA (Algoritmo 5.4). O

```

1 Solucao()  $\leftarrow \emptyset$ ;
2 para  $t \leftarrow 1$  até  $T$  faça
3    $sol() \leftarrow \emptyset$ ;
4   para  $m \leftarrow 1$  até  $|M|$  faça
5      $sol(m) \leftarrow \text{ILS}(M(m))$ ;
6   fim
7   Atribui a  $sol^*$  a melhor solução contida em  $sol()$  (através da função de avaliação  $f(\cdot)$ );
8    $sol^* \leftarrow \text{COBERTURA\_COMPLETA}(sol^*)$ ;
9   se  $sol^*$  não cobrir completamente a área monitorada então
10     $Não\ existe\ solução$ ;
11  fim
12  senão
13     $Solucao(t) \leftarrow sol^*$ ;
14  fim
15 fim
16 retorna Solucao

```

Algoritmo 5.2: Heurística construtiva para o PICDCR

primeiro (linha 5), se preocupa em assegurar que todos os problemas que procuramos integrar com exceção do relacionado à localização do sorvedouro sejam alcançados. Entretanto, o procedimento ILS não garante que todos os pontos de demanda sejam cobertos, uma vez que o mesmo utiliza o procedimento de Busca Local descrito na Seção 5.1, e que como visto, não assegura esse aspecto. Por esta razão, o procedimento COBERTURA_COMPLETA (linha 8) é chamado no intuito de procurar por sensores que possam ser ativados, afim de garantir que toda a área a ser monitorada seja coberta. Caso isso não seja possível, o algoritmo retorna uma solução vazia (linha 9-14).

Como o problema da localização do sorvedouro afeta diretamente a eficiência energética da rede, ao construir uma solução periodicamente, devemos garantir que, a cada período, o consumo de energia seja reduzido. Portanto, a questão da localização do sorvedouro é tratada de forma exaustiva, testando-se todas as posições candidatas para o sorvedouro (linhas 4-6). Cada posição m é passada como entrada para o procedimento ILS e a melhor solução retornada por esse (linha 7) será a escolhida.

O Algoritmo 5.3 apresenta o procedimento ILS. No início, esse procedimento recebe como parâmetro a posição, *Sink*, na qual o sorvedouro estará localizado na rede, em um dado período t . Em seguida, é chamado (linha 3) o procedimento de Busca Local (descrito na seção 5.1), passando para o mesmo uma solução inicial contendo somente o sorvedouro localizado na posição *Sink* (linha 2). Isso é possível pois a heurística e conseqüentemente todos os seus procedimentos, partem da inviabilidade a fim de tentar encontrar uma solução que seja viável. O próximo passo consiste em chamar *MAXITER* vezes os procedimentos de Perturbação (descrito na seção 5.2) e Busca Local (linha 5 e 6), passando para os mesmos a solução corrente sol . A escolha

da próxima solução a ser perturbada será a melhor encontrada até o momento (critério de aceitação) (linhas 7-12).

```

1 Procedimento ILS(Sink)
2  $sol^* \leftarrow \{Sink\}$ ;
3  $sol \leftarrow$  BUSCA_LOCAL( $sol^*$ );
4 para  $l \leftarrow$  até  $\overline{MAXITER}$  faça
5   |  $sol \leftarrow$  Perturbação( $sol$ );
6   |  $sol \leftarrow$  BUSCA_LOCAL( $sol$ );
7   | se  $f(sol) < f(sol^*)$  então
8     |   |  $sol^* \leftarrow sol$  ;
9     |   fim
10  | senão
11  |   |  $sol \leftarrow sol^*$  ;
12  |   fim
13 fim
14 retorna  $sol^*$ 

```

Algoritmo 5.3: Pseudo-código do procedimento ILS

O Algoritmo 5.4 apresenta o procedimento COBERTURA_COMPLETA que recebe como parâmetro a solução, sol , retornada pelo procedimento ILS. Uma vez que a heurística que propomos visa encontrar uma solução viável, deve-se assegurar que todos os pontos de demanda sejam cobertos pelos sensores ativos. Dessa forma, o procedimento procura na solução de partida, algum ponto de demanda que não tenha sido coberto (linha 4). Caso exista algum, o procedimento testa, para cada ponto de demanda não coberto, todos os sensores que o cobrem (linha 5). O teste consiste em aplicar todos os operadores do procedimento de Busca Local (linha 6), descrito na Seção 5.1, escolhendo aquela solução que retorne o melhor valor (linha 7-9), a partir função de avaliação, $f(\cdot)$ (descrita na Seção 5.1.4). Como tal função leva em conta os critérios de energia, cobertura e conectividade, o procedimento de COBERTURA_COMPLETA procura ativar nós a fim de garantir a cobertura completa da área monitorada, reduzindo ao máximo o incremento de energia decorrente de tal ativação.

5.4 Resultados Computacionais

Nesta seção, apresentamos os experimentos computacionais conduzidos para avaliar a heurística introduzida para o PICDCR. O propósito é validar o algoritmo para que ele possa ser utilizado na resolução de instâncias de dimensões mais próximas das encontradas na prática.

Os testes foram executados trinta vezes, uma vez que a heurística apresenta fatores aleatórios em sua execução. A melhor e a pior soluções encontradas foram recu-

```

1 Procedimento COBERTURA_COMPLETA(sol)
2 sol* ← ∅;
3 para p ← 1 até |D| faça
4   | se p não foi coberto por nenhum sensor ativo em sol então
5   |   | para cada s ∈ α(p) faça
6   |   |   | Aplicar os operadores de busca local em s;
7   |   |   | se f(sol) < f(sol*) então
8   |   |   |   | sol* ← sol;
9   |   |   |   | fim
10  |   |   | senão
11  |   |   |   | Desfaça as alterações feitas em sol;
12  |   |   |   | fim
13  |   |   | fim
14  |   | fim
15 fim
16 retorna sol*

```

Algoritmo 5.4: Pseudo-código do Procedimento COBERTURA_COMPLETA

peradas, assim como a média de todas as execuções. As instâncias de testes utilizadas foram as mesmas explicadas na Seção 4.2.1.

A Tabela 5.1 mostra os resultados obtidos para todas as instâncias testadas. As colunas *Melhor* (Coluna 5), *Pior* (Coluna 6), *Média* (Coluna 7), Δ e *Tempo* apresentam a energia residual da melhor e da pior soluções encontradas, assim como a energia residual média das trinta execuções, o desvio padrão e o tempo médio para cada execução (em segundos), respectivamente. Uma vez que a heurística proposta apresenta somente um parâmetro, *MAXITER*, atribuímos ao mesmo os valores 1 e 20, no intuito de analisarmos a melhora ocasionada por tal variação. Dessa forma, as colunas *Melhor* (Coluna 10), *Pior* (Coluna 11), *Média* (Coluna 12) apresentam a diferença percentual entre as energias residuais obtidas pelo algoritmo com o parâmetro *MAXITER* = 1 em relação àquelas obtidas fazendo-se esse mesmo parâmetro igual a 20.

Ao se comparar, por exemplo, as instâncias para $|M| = 5$ e $T = 4$ (linhas 1, 21, 41), percebe-se que, quanto maior o número de sensores menor é o consumo de energia das soluções, uma vez que tanto o conjunto de sensores ativos quanto o comprimento das rotas de dados estão sendo reduzidos. Deixando os demais parâmetros fixos e variando-se somente $|M|$, verifica-se a mesma tendência explicada anteriormente, principalmente quando a coluna *Média* é analisada (linhas 43, 48, 53, 58, por exemplo).

Nas Figuras 5.3, 5.4 e 5.5 isso pode ser visto mais claramente, já que essas mostram a relação entre o aumento do número de períodos (eixo *Periodos*) e a melhora das soluções médias (eixo *Incremento*), a partir daquelas soluções cujo valor de $|M|$ permaneceu igual a 5. A inclinação positiva das curvas indica que à medida que o número de períodos é incrementado, o algoritmo torna-se cada vez mais sensível ao

Instâncias			Soluções									
			MAXITER = 20					MAXITER = 1				
			Energia Residual (mAh)				Energia Residual					
S	M	T	Melhor	Pior	Média	Δ	Tempo	Melhor	Pior	Média	Tempo	
1	400	5	4	146407,57	145998,23	146214,33	102,36	821,97	-0,11%	-0,21%	-0,14%	134,60
2	400	5	5	143204,77	142716,80	142974,73	121,65	1013,00	-0,12%	-0,25%	-0,16%	162,39
3	400	5	6	140007,13	139440,90	139740,68	142,03	1228,96	-0,13%	-0,25%	-0,18%	195,42
4	400	5	7	136804,50	136172,57	136506,78	160,82	1209,49	-0,13%	-0,25%	-0,20%	197,54
5	400	5	8	133600,03	132896,60	133275,67	178,97	1640,01	-0,14%	-0,24%	-0,22%	266,23
6	400	10	4	146448,17	146052,70	146256,98	88,20	1430,68	-0,10%	-0,15%	-0,12%	265,40
7	400	10	5	143249,03	142795,90	143027,16	103,93	1755,61	-0,11%	-0,18%	-0,14%	321,03
8	400	10	6	140041,13	139542,33	139796,64	119,37	2199,29	-0,11%	-0,22%	-0,15%	400,12
9	400	10	7	136853,33	136293,07	136569,39	136,62	2660,14	-0,12%	-0,25%	-0,16%	455,32
10	400	10	8	133658,03	133039,03	133343,44	152,39	2914,99	-0,14%	-0,28%	-0,18%	462,05
11	400	15	4	146452,63	146104,23	146260,15	89,68	2264,65	-0,11%	-0,15%	-0,11%	391,20
12	400	15	5	143234,53	142835,93	143025,56	105,39	2873,53	-0,11%	-0,19%	-0,12%	420,36
13	400	15	6	140035,80	139565,60	139791,37	123,61	3685,32	-0,12%	-0,22%	-0,13%	504,88
14	400	15	7	136843,13	136301,73	136559,66	143,91	3750,92	-0,14%	-0,27%	-0,14%	535,80
15	400	15	8	133646,07	133023,50	133331,59	162,20	4395,00	-0,15%	-0,28%	-0,15%	788,12
16	400	20	4	146446,47	146059,47	146266,61	93,55	3314,41	-0,09%	-0,13%	-0,11%	531,16
17	400	20	5	143255,60	142809,93	143035,88	110,59	3266,54	-0,12%	-0,17%	-0,13%	653,34
18	400	20	6	140063,27	139562,10	139808,93	124,98	4573,68	-0,14%	-0,19%	-0,15%	723,85
19	400	20	7	136873,20	136312,17	136585,94	141,17	4806,23	-0,16%	-0,22%	-0,17%	926,17
20	400	20	8	133683,20	133065,83	133363,24	156,67	6030,52	-0,17%	-0,25%	-0,19%	1027,52
21	500	5	4	187173,60	186686,83	186920,38	117,43	892,67	-0,17%	-0,16%	-0,15%	166,73
22	500	5	5	184158,57	183561,57	183854,70	141,48	1149,47	-0,20%	-0,17%	-0,17%	180,36
23	500	5	6	181139,97	180433,97	180793,71	165,37	1487,93	-0,21%	-0,21%	-0,19%	252,20
24	500	5	7	178124,30	177293,07	177734,45	191,31	1637,40	-0,22%	-0,22%	-0,21%	296,59
25	500	5	8	175102,03	174162,93	174675,23	216,40	1675,64	-0,23%	-0,22%	-0,22%	291,52
26	500	10	4	187193,20	186637,07	186954,85	128,67	1737,57	-0,12%	-0,11%	-0,14%	283,55
27	500	10	5	184182,30	183538,23	183891,62	151,33	2399,27	-0,15%	-0,14%	-0,15%	301,81
28	500	10	6	181169,10	180444,47	180835,06	174,51	2917,79	-0,16%	-0,17%	-0,17%	436,78
29	500	10	7	178159,87	177331,67	177777,46	202,36	3445,68	-0,17%	-0,19%	-0,18%	503,26
30	500	10	8	175156,97	174219,30	174721,53	230,00	3461,30	-0,18%	-0,21%	-0,19%	585,90
31	500	15	4	187180,73	186739,47	186976,03	107,59	2372,38	-0,06%	-0,13%	-0,13%	490,91
32	500	15	5	184163,93	183625,17	183916,40	132,51	3680,93	-0,07%	-0,15%	-0,14%	628,95
33	500	15	6	181147,67	180518,03	180856,93	156,30	4345,73	-0,08%	-0,16%	-0,16%	736,73
34	500	15	7	178129,93	177433,13	177799,89	179,14	4739,39	-0,08%	-0,19%	-0,17%	877,88
35	500	15	8	175123,27	174345,87	174745,46	203,13	4925,81	-0,09%	-0,22%	-0,18%	919,66
36	500	20	4	187191,80	186683,23	186974,56	106,70	3266,21	-0,06%	-0,11%	-0,12%	591,72
37	500	20	5	184177,27	183591,93	183915,51	126,00	4322,35	-0,06%	-0,14%	-0,13%	840,41
38	500	20	6	181158,77	180506,60	180857,81	146,93	4731,03	-0,08%	-0,16%	-0,15%	898,66
39	500	20	7	178136,53	177412,37	177801,87	168,69	6157,87	-0,08%	-0,19%	-0,16%	1094,49
40	500	20	8	175118,87	174308,17	174746,82	189,45	6544,96	-0,08%	-0,20%	-0,18%	1365,56
41	600	5	4	227338,27	226866,07	227106,82	104,13	1305,70	-0,15%	-0,19%	-0,16%	213,07
42	600	5	5	224352,83	223806,23	224089,68	124,41	1624,28	-0,19%	-0,22%	-0,19%	236,56
43	600	5	6	221369,40	220746,97	221073,26	146,24	1938,72	-0,20%	-0,25%	-0,21%	322,41
44	600	5	7	218397,67	217669,57	218060,36	171,49	2267,22	-0,21%	-0,23%	-0,22%	377,17
45	600	5	8	215428,63	214605,87	215050,71	194,71	2381,69	-0,23%	-0,24%	-0,24%	449,13
46	600	10	4	227367,73	226952,63	227140,91	101,62	2419,48	-0,10%	-0,18%	-0,13%	421,65
47	600	10	5	224390,47	223915,17	224129,45	121,60	3188,46	-0,11%	-0,20%	-0,15%	539,85
48	600	10	6	221408,63	220875,93	221120,29	143,06	3052,76	-0,10%	-0,23%	-0,16%	561,56
49	600	10	7	218430,53	217825,80	218113,94	166,28	3590,45	-0,10%	-0,24%	-0,17%	756,10
50	600	10	8	215453,57	214772,87	215108,37	189,27	4541,98	-0,08%	-0,24%	-0,18%	775,59
51	600	15	4	227351,40	226961,20	227156,10	98,83	3753,22	-0,10%	-0,16%	-0,13%	546,50
52	600	15	5	224385,77	223919,03	224145,26	118,61	4667,26	-0,11%	-0,18%	-0,15%	591,26
53	600	15	6	221425,10	220869,60	221138,72	141,07	5669,82	-0,13%	-0,20%	-0,16%	948,76
54	600	15	7	218458,60	217808,73	218132,41	162,05	6434,90	-0,15%	-0,21%	-0,18%	990,73
55	600	15	8	215499,20	214747,33	215127,44	184,56	6958,42	-0,16%	-0,21%	-0,19%	1167,84
56	600	20	4	227373,30	226989,50	227167,09	96,12	4253,09	-0,11%	-0,17%	-0,13%	833,60
57	600	20	5	224409,33	223938,63	224162,42	119,38	6410,54	-0,13%	-0,19%	-0,15%	1102,80
58	600	20	6	221448,93	220875,20	221159,12	143,88	6862,38	-0,14%	-0,20%	-0,17%	1132,89
59	600	20	7	218494,97	217826,33	218159,56	168,36	6693,58	-0,15%	-0,22%	-0,19%	1327,78
60	600	20	8	215539,67	214765,27	215160,97	193,66	9039,77	-0,16%	-0,23%	-0,21%	1085,51

Tabela 5.1. Resultados obtidos pela Heurística proposta para o PICDCR

tamanho do conjunto $|M|$, ou seja, o problema de localização do sorvedouro torna-se, com o aumento de T , cada vez mais importante para garantir redução no consumo de energia da rede.

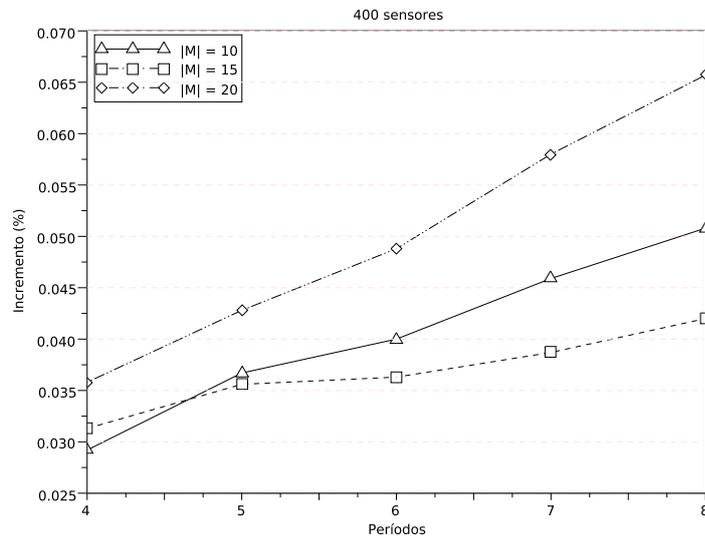


Figura 5.3. Melhora da solução ocasionada pelo aumento da cardinalidade de M para 400 sensores

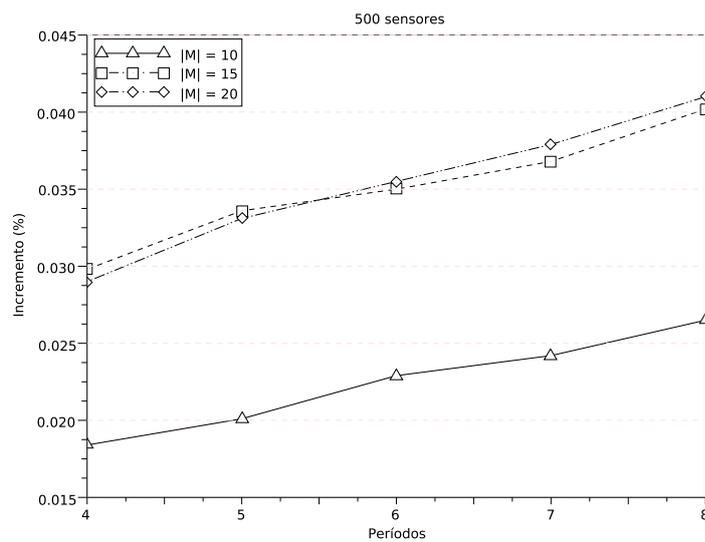


Figura 5.4. Melhora da solução ocasionada pelo aumento da cardinalidade de M para 500 sensores

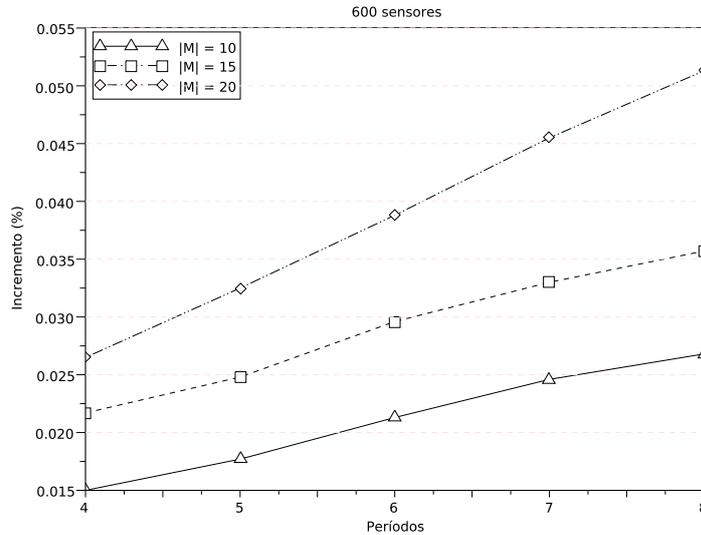


Figura 5.5. Melhora da solução ocasionada pelo aumento da cardinalidade de M para 600 sensores

Percebe-se também, uma relação direta entre a redução na quantidade de sensores e o problema de localização do sorvedouro, uma vez que as soluções com 400 sensores apresentaram o melhor incremento na qualidade das soluções (0,042% na média), quando comparadas às soluções com quantidades maiores de sensores (0,030% e 0,0297% na média para 500 e 600 sensores, respectivamente). Já ao analisar as curvas que representam $|M|$, percebe-se que as mesmas tendem a não se interceptarem com o aumento na quantidade de sensores. Isso indica que, embora tal aumento permita ao algoritmo buscar, mais facilmente, por soluções que visem reduções no consumo de energia através de melhores rotas de dados, o problema de localização do sorvedouro tem seu papel reduzido nisso, tornando-se mais importante a determinação do conjunto de sensores que permanecerão ativos.

Era de se esperar que para todas as instâncias, e não somente para as com 600 sensores, houvesse uma distinção clara na melhora da qualidade das soluções geradas com o aumento de $|M|$. Isso decorre do fato de que cada conjunto M é criado a partir daquele de tamanho exatamente inferior, o que permite que as instâncias que utilizem o novo conjunto apresente soluções, no pior caso, iguais àquelas que utilizam o conjunto gerador (Seção 4.2.1). Entretanto, deve-se lembrar que isso é garantidamente verdadeiro quando comparamos as soluções ótimas, que podem não ter sido encontradas.

Dessa forma, o tamanho do parâmetro $|M|$ é crucial para que o algoritmo possa encontrar soluções melhores, em termos energéticos, principalmente quando trabalha-se

com redes apresentando quantidades reduzidas de sensores e/ou com muitos períodos.

Apesar de a heurística ter sido capaz de encontrar soluções viáveis para todas as instâncias propostas, o custo computacional observado foi muito elevado para isso. O tempo computacional médio para se executar a heurística variou de 821,97s (linha 1) até 9039,77s (linha 60), considerando $MAXINTER = 20$, e de 134,60s (linha 1) até 1365,56s (linha 40), ao considerar $MAXITER = 1$. Essa variação extrema entre os tempos se deve ao fato de as instâncias testadas apresentarem diferentes níveis de dificuldade, caracterizadas por diferentes valores para os parâmetros $|S|$, $|M|$, T . Portanto, aquelas que contêm mais sensores, mais períodos e mais posições possíveis para o sorvedouro se localizar, tendem a requerer maior esforço computacional para sua resolução. Isso também pode ser verificado ao se analisar o desvio padrão (coluna Δ). À medida que o número de períodos é incrementado (linhas 5 a 10, por exemplo), o desvio padrão tende a aumentar, indicando a dificuldade de obter soluções sistematicamente próximas ao ótimo.

Entretanto, quando comparado ao tempo exigido pelo algoritmo exato (Capítulo 4), a heurística foi muito mais eficiente. A Tabela 5.2 mostra uma comparação entre as energias residuais das soluções viáveis encontradas pelo algoritmo BC em relação às melhores e a piores soluções encontradas pela heurística com $MAXITER = 20$, assim como as médias. Como o algoritmo exato não foi capaz de encontrar soluções viáveis para todas as instâncias testadas ao final de dez horas de execução, foram comparados com a heurística somente àquelas que puderam ser encontradas. As colunas 1ª Viável e Após 10 horas mostram a diferença em porcentagem, respectivamente, da primeira solução viável encontrada e daquela após o algoritmo BC ter sido executado por dez horas. Os valores negativos mostrados indicam que todas as soluções encontradas pela heurística, mesmo as piores, apresentaram energia residual melhor do que as encontradas pelo algoritmo exato. Com isso, pode-se concluir que os mecanismos propostos na construção da heurística foram eficazes em permitir que uma solução viável, e idealmente de boa qualidade, fosse encontrada.

Instâncias			Melhor solução		Pior Solução		Solução Média	
$ S $	$ M $	T	1ª Viável	Após 10 horas	1ª Viável	Após 10 horas	1ª Viável	Após 10 horas
400	5	4	-0,370%	-0,368%	-0,090%	-0,088%	-0,238%	-0,236%
400	5	5	-0,655%	-0,655%	-0,312%	-0,312%	-0,493%	-0,493%
400	10	4	-0,511%	-0,511%	-0,239%	-0,239%	-0,380%	-0,380%
400	15	4	-0,468%	-0,468%	-0,229%	-0,229%	-0,336%	-0,336%
400	20	4	-0,467%	-0,467%	-0,201%	-0,201%	-0,343%	-0,343%

Tabela 5.2. Comparação entre as soluções encontradas pela heurística com aquelas encontradas pelo algoritmo exato

Fazendo $MAXINTER = 1$, permitimos analisar a primeira solução viável encon-

trada pela heurística proposta. A partir disso, percebemos que já na primeira iteração o algoritmo encontra soluções boas a ponto de serem, no máximo, 0,24% piores que as encontradas quando tal parâmetro é estabelecido como 20. Entretanto, a diferença no custo computacional é evidente quando *MAXITER* é mudado de 1 para 20, chegando a ser 8,33 vezes pior (linha 60, Tabela 5.1). Dessa forma, se o tempo de execução da heurística não puder ser relevado durante a geração de uma RSSF, é possível utilizar a mesma fazendo *MAXITER* igual a 1, uma vez que as soluções obtidas com esse valor foram quase todas melhores que às viáveis obtidas pelo algoritmo exato, após o mesmo ter sido executado por dez horas. Uma conta simples permite provar tal argumento, bastando-se, para isso, subtrair a diferença das soluções encontradas variando-se *MAXITER* (Tabela 5.1) da diferença entre a solução encontrada pelo algoritmo exato (Tabela 5.2). Apesar de os resultados mostrados na Tabela 5.1 apresentarem pouca variação com o incremento de *MAXITER*, afirmamos estatisticamente, com 95% de confiança que as médias das soluções, ao se variar o parâmetro em questão, são diferentes entre si. Tal afirmação decorre dos resultados obtidos após a realização do *teste t pareado* [Bussab & Morettin, 2003], que consiste em um teste estatístico para definir o nível de semelhança ou diferença entre duas amostras dependentes.

5.5 Comentários

O método de solução heurístico permitiu encontrar soluções viáveis, e de boa qualidade, para o PICDCR, principalmente quando comparadas àquelas obtidas pelo algoritmo exato após o mesmo ter sido executado por dez horas. Isso indica que os mecanismos propostos na construção da heurística foram eficazes em alcançar tal objetivo.

Ao analisar os resultados obtidos, percebe-se que algoritmo apresenta maior sensibilidade ao parâmetro $|M|$, principalmente quando o número de sensores é reduzido ou o número de períodos é aumentado. Isso acontece pois, nesses casos o papel do problema de localização do sorvedouro fica mais destacado em relação aos outros problemas. Com poucos sensores, por exemplo, mais energia é consumida devido à geração de rotas de dados mais longas, enquanto que com o aumento de T mais energia deve ser gasta para manter a rede funcional durante todo o horizonte de planejamento.

Apesar de bons resultados terem sido obtidos com a utilização da heurística, o tempo computacional exigido por essa variou muito. Dessa forma, caso exista limitações quanto ao tempo necessário para a organização de uma RSSF, pode-se, ainda, utilizar a heurística simplesmente reduzindo o parâmetro *MAXITER* para 1, sem que isso cause perdas significativas na qualidade das soluções encontradas.

Com base nos resultados apresentados, foi escolhido atribuir o valor 20 ao parâmetro *MAXITER*, para que a heurística possa encontrar soluções viáveis para as instâncias de testes a serem usadas no arcabouço de simulação para RSSFs. Como a execução da heurística será feita uma única vez durante a simulação, e considerando que dispomos de um tempo limite igual a dez horas para organizar a rede, não há a necessidade de melhorar o tempo computacional gasto pela heurística através da redução no valor do parâmetro *MAXITER*, uma vez que seu tempo de execução ficou muito aquém do limite imposto. Além do mais, a escolha por utilizar a heurística em detrimento ao algoritmo exato é evidente, já que a primeira foi a única a sempre obter soluções viáveis, em um tempo computacional inferior à dez horas.

O arcabouço de simulação para as RSSFs será discutido no capítulo que se segue.

Capítulo 6

Simulação de uma RSSF

Capturar e analisar de forma exata as interações e relações entre entidades pertencentes a sistemas dinâmicos complexos, como é o caso das RSSFs, é extremamente difícil, uma vez que não é possível prever com precisão o modelo matemático para várias propriedades do sistema. Assim sendo, ao longo deste capítulo propomos o desenvolvimento de um Simulador de Eventos Discreto para avaliar o modelo proposto, permitindo analisar como o modelo de organização da rede e os algoritmos de otimização propostos neste trabalho permitiram assegurar alguns dos requisitos de QoS de uma RSSF.

6.1 Aspectos Gerais do Simulador

Foi escolhido um Simulador de Eventos Discretos como ferramenta para avaliar como o modelo proposto reage quando métricas comuns de QoS, tais como atraso médio de mensagens e taxa média de mensagens entregues, são levadas em conta durante a operação de uma RSSF. Para isso, desenvolvemos uma arcação para RSSFs, implementado sobre o simulador *JIST/SWAMS* [jis, 2007]. Os protocolos de energia, transmissão, sensoriamento e armazenamento são previamente disponibilizados na biblioteca de funções do simulador empregado.

O modelo de simulação apresentado compreende uma série de operações periódicas, no intuito de representar virtualmente como a rede se comportaria em um ambiente real. O fluxograma das atividades realizadas durante a simulação é apresentado na Figura 6.1. A simulação inicia recebendo como entrada a resolução do PICDCR, logo após a distribuição dos sensores pela rede. Tal resolução é realizada *a priori*, através de algum dos algoritmos de otimização propostos nos Capítulos 4 e 5, e portanto, assume-se que tanto quem irá resolvê-lo (alguma central de processamento, por exemplo) quanto o sorvedouro conheçam a posição geográfica de todos os sensores. A solução de como

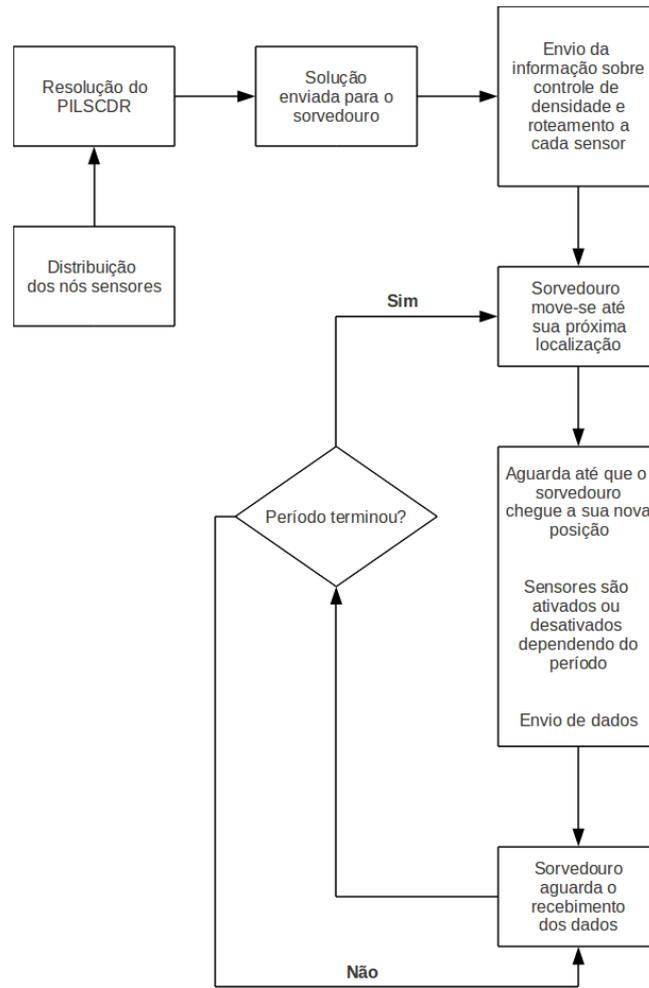


Figura 6.1. Fluxograma do simulador

será a topologia da rede por todos os períodos é inserida no sorvedouro, e esse, por sua vez, a transmite para todos os sensores via *broadcast*. Antes de receber tal informação, cada sensor apresenta-se, inicialmente, em um estado onde somente suas funções de rádio e processamento estão ativas. Após o recebimento da informação sobre a topologia da rede, cada sensor periodicamente se coloca em um estado ativo (com as funções de rádio, processamento e sensoriamento ativas) ou inativo (em *standby*).

Porém, antes de alterar seu estado, os sensores aguardam 30 segundos, a partir do início do envio das informações pelo sorvedouro, a fim de se certificarem que todos recebam tal informação. Esse tempo é importante pois, caso algum sensor não tenha recebido as informações sobre a topologia, o mesmo poderá consumir energia desnecessariamente ao permanecer ativo ou impedir o roteamento correto dos dados pela rede.

Pode-se imaginar que os 30 segundos necessários para configurar a rede afetem

negativamente a simulação. Entretanto, como os sensores iniciam suas operações com seus dispositivos de sensoriamento desligados, nenhuma informação é coletada durante esse tempo. Além disso, como a configuração acontece somente uma vez e no primeiro período (o que não deve ser negligenciado), o impacto energético causado por tal operação será diluído nos próximos períodos.

O sorvedouro parte de uma posição inicial na rede (posição $(0,0)$, considerando que a região monitorada é modelada a partir de um plano cartesiano) e se locomove, em linha reta, até a melhor posição candidata escolhida para o mesmo no primeiro período. Enquanto o sorvedouro estiver se movimentando, toda a informação sensorizada pelos sensores ativos é armazenada em seus dispositivos de memória, chamados de *buffer*.

Como a rota que o sorvedouro seguirá ao longo dos períodos é conhecida *a priori* (através da resolução do PICDCR), bem como a velocidade que o mesmo possui para isso, um segundo antes do sorvedouro começar a se locomover, os sensores interrompem a transmissão e retransmissão de mensagens, armazenando toda informação que está sendo transitada pela rede em seus *buffers*. A abordagem de interromper a transmissão de dados um segundo antes do sorvedouro se mover é importante pois evita que mensagens sejam perdidas durante a mudança da topologia da rede. Além disso, caso tal abordagem não fosse utilizada, a cada período, o sorvedouro deveria enviar mensagens avisando sobre o início de sua movimentação, tendo de esperar que todos os sensores a recebessem, uma vez que a perda de alguma destas mensagens de configuração poderia comprometer toda a integridade da rede. Outra vantagem dessa abordagem é a possibilidade de os sensores desativados não precisarem permanecer com o rádio ligado, dessa forma economizando energia.

Enquanto o sorvedouro está se movendo, durante um tempo que depende de sua velocidade e distância a ser percorrida, cada sensor se auto-configura, mudando a escolha de quem será seu pai na árvore de roteamento, assim como seu estado (ativo ou inativo). Um segundo depois de o sorvedouro chegar a posição candidata escolhida para o próximo período, cada sensor inicia o envio de seus dados, começando primeiramente por aqueles armazenados em seus *buffers*. Esse tempo é suficiente para permitir que todos os sensores se auto-configurem antes de iniciarem a transmissão de dados. O sorvedouro permanecerá estacionado na posição escolhida para o mesmo até que o período termine. Acontecido isso, o sorvedouro voltará a se mover, seguindo os passos descritos anteriormente.

Os parâmetros de simulação escolhidos neste trabalho foram definidos de acordo com características dos nós sensores Mica2 [mic, 2011], e sumarizados na Tabela 4.1. Para a aplicação particular aqui tratada, foi assumido que os nós sensores periodicamente coletam informação de temperatura codificadas em 32 bits, a uma taxa constante

de 1/20 Hz, e tamanho da mensagem de dados igual a 16 *bytes*. O cenário para a simulação não inclui nenhum tipo de obstáculo ou interferência ambiental, dessa forma, a perda de mensagens só acontecerá através de colisão, perda de sinal durante a transmissão ou por descarte de mensagens antigas armazenadas no *buffer* dos sensores, quando os mesmos necessitarem armazenar novas mensagens e não houver espaço para isso.

Apesar de cada sensor ativo enviar mensagens em uma mesma frequência, cada um iniciará o envio a partir de um tempo aleatório. Isso permite reduzir, em relação ao envio sincronizado, a perda de mensagens ocasionadas por colisões entre si.

A camada MAC é a IEEE 802.11, disponível no simulador SWANS, uma vez que os sensores Mica2 implementam o protocolo CSMA/CA.

6.2 Resultados Computacionais

Com base no que foi apresentado no Capítulo 5, passamos os resultados obtidos pela heurística proposta, com o parâmetro MAXITER igual a 20, como entrada para o arcabouço de simulação.

Como visto na formulação para o PICDCR (Capítulo 3), abstrações tiveram que ser assumidas a fim de simplificar as restrições (3.7) que tratam o consumo de energia gasto pelos sensores. Questões como o tempo em que o sensor passa transmitindo/recebendo mensagens ou gasto processando e/ou coletando dados são alguns dos exemplos de informações difíceis de serem capturadas através de modelos matemáticos, porém, mais fáceis de serem mensuradas na simulação. Dessa forma, assumimos durante a resolução do PICDCR, que os parâmetros utilizados no cálculo do consumo de energia sempre serão superestimados, considerando seus valores máximos, ao invés de proporcionais ao tempo que um determinado hardware do sensor está sendo utilizado. Uma vantagem obtida por tal abordagem é a de sempre garantir que as energias residuais encontradas pelo modelo proposto para o PICDCR possam ser consideradas limites inferiores para àquelas retornadas pela simulação. Se os valores dos parâmetros fornecidos para o modelo matemático e heurística forem subestimados, falhas de sensores ocasionadas pela falta de energia poderiam acontecer durante a simulação, sem que estas tivessem sido previstas durante a resolução do PICDCR.

Uma forma de melhor estimar os valores para os parâmetros do modelo pode ser a seguinte: executamos, para uma instância particular, a heurística proposta para o PICDCR com os valores dos parâmetros superestimados, e em seguida passamos a solução obtida como entrada para o arcabouço de simulação. Uma vez que a simulação seja executada, capturamos os valores reais gastos pelos sensores após o término da

simulação. A seguir, utilizamos os novos valores para realimentar a heurística e obter uma nova solução para outras instâncias. Porém, isso não permitiria garantir que os valores obtidos não ficassem subestimados, uma vez que os parâmetros ideais são extremamente dependentes da instância que está sendo avaliada. A mesma conclusão é obtida caso os passos anteriores fossem repetidos para um conjunto de instâncias, a média dos valores obtidos para os parâmetros fosse tomada, e a partir disso, utilizadas na resolução de outras instâncias.

Assim sendo, primeiramente avaliaremos como a topologia gerada ao se resolver o PICDCR se comporta quando métricas de QoS, tais como atraso médio de mensagens entregues e taxa de mensagens entregues são levadas em consideração durante a organização de uma RSSF. Em seguida, analisaremos as diferenças entre as energias residuais encontradas através da resolução do PICDCR e aquelas retornadas pela simulação.

Para isso, o tempo de simulação variou entre 4 e 8 horas, dependendo da quantidade de períodos de cada instância testada.

6.2.1 Métricas de QoS Avaliadas

O atraso médio de mensagens entregues é uma métrica importante quando as RSSFs são utilizadas em aplicações onde há a necessidade de que a informação sensoriada seja gerada com uma alta frequência ou mesmo regularmente (ao invés de ser gerada sob demanda ou baseada em eventos). Um exemplo seriam aplicações para detecção de incêndio ou contaminação química do ar, já que nestes casos o atraso médio deve ser baixo o suficiente para permitir que ações sejam tomadas rapidamente. Além disso, para tais aplicações, valores baixos de atraso médio evitam que perdas de dados aconteçam devido à baixa capacidade de memória dos sensores.

Outra métrica de QoS avaliada neste trabalho é a taxa de mensagens entregues. Tal métrica é importante em aplicações críticas, onde a perda de informações deve ser levada em conta durante a organização das RSSFs. Um exemplo para isso se daria em aplicações para controle do nível de radiação em uma usina nuclear ou mesmo para controle de fronteiras, onde uma determinada taxa de informação que chega ao sorvedouro precisa ser garantida a fim de permitir que ações possam ser tomadas.

Apesar de ambas as métricas parecerem correlacionadas, elas são independentes. Dessa forma, nada se pode afirmar de fato, por exemplo, sobre a taxa de mensagens entregues em uma rede apresentando baixo atraso médio, uma vez que a rede poderia estar perdendo muitas mensagens e somente as poucas entregues estariam sendo computadas. Entretanto, quando analisadas em conjunto, tais métricas são um ferramental importante para permitir inferências sobre a qualidade da topologia analisada,

principalmente em relação ao tráfego de mensagens.

Nas Figuras 6.2 e 6.3 são apresentados, respectivamente, como o atraso médio (*Atraso*) e a taxa de mensagens entregues (*Taxa*) se comportam ao variar-se o número de períodos, de sensores e na quantidade de posições candidatas para o sorvedouro na rede, $|M|$. Além disto, como tais métricas estão relacionadas ao tempo total com que o sorvedouro passa se movimentando, o mesmo também será mostrado (*Deslocamento*). Essa relação é evidente, já que se o sorvedouro passar muito tempo se movimentando, mensagens armazenadas no *buffer* dos sensores começarão a ser descartadas para que novas sejam armazenadas, o que reduz a taxa de mensagens entregues. Além disso, quanto maior o tempo que o sorvedouro gastar se movimentando, maior será o atraso médio de entrega de mensagens, uma vez que as informações armazenadas nos *buffers* só são enviadas para o sorvedouro depois que este parar de se movimentar. Para fins de clareza na apresentação dos resultados, somente as soluções contendo quantidades extremas de sensores, ou seja, para 400 e 600 nós, serão analisados, uma vez que foram elas as que mais sofreram com os impactos causados pelas variações dos parâmetros.

Como pode-se verificar nas Figuras, à medida que o parâmetro $|M|$ é incrementado até 20, as soluções encontradas tendem a apresentar melhorias tanto nas taxas de mensagens entregues quanto no atraso médio. Porém, tal melhora não se mostrou tão evidente quanto àquelas observadas ao se comparar o consumo de energia (como mostrado na seção 5.4). Isso se deve, principalmente, ao fato de as métricas que estão sendo analisadas não serem tratadas diretamente no PICDCR. Além disso, à medida que o número de períodos aumenta, melhoras nas QoS das soluções geradas também podem ser evidenciadas. Observe que, quando $|M| = 5$ para 400 sensores, a taxa de mensagens entregues chega a no máximo 91%, enquanto que para valores maiores de $|M|$ elas ficam a, no mínimo, igual a 93%. Ou seja, por mais que o atraso médio na entrega de mensagens esteja diminuindo no decorrer dos períodos, os resultados mostram que qualquer valor para $|M|$ maior do que 10 proporciona soluções bem melhores, a partir das métricas de QoS que estamos considerando, do que aquelas encontradas para $|M| = 5$.

As conclusões anteriores podem ser percebidas com mais clareza aumentando-se o número de sensores. Para 600 sensores, por exemplo, ao analisar as duas métricas simultaneamente, observa-se que com o incremento do número de períodos, a curva para $|M| = 20$, em *Atraso*, chega em seu mínimo (50s para $T = 8$). Apesar de a curva para $|M| = 15$ ter apresentado um atraso médio menor (29s, para $T = 8$), analisando as curvas para $|M| = 20$ em ambas as métricas, fica claro a melhora das soluções ocasionada pelo aumento no número de períodos. Porém, essa mesma curva varia muito ao se comparar valores extremos de T (4 e 8). Dessa forma, as melhores soluções

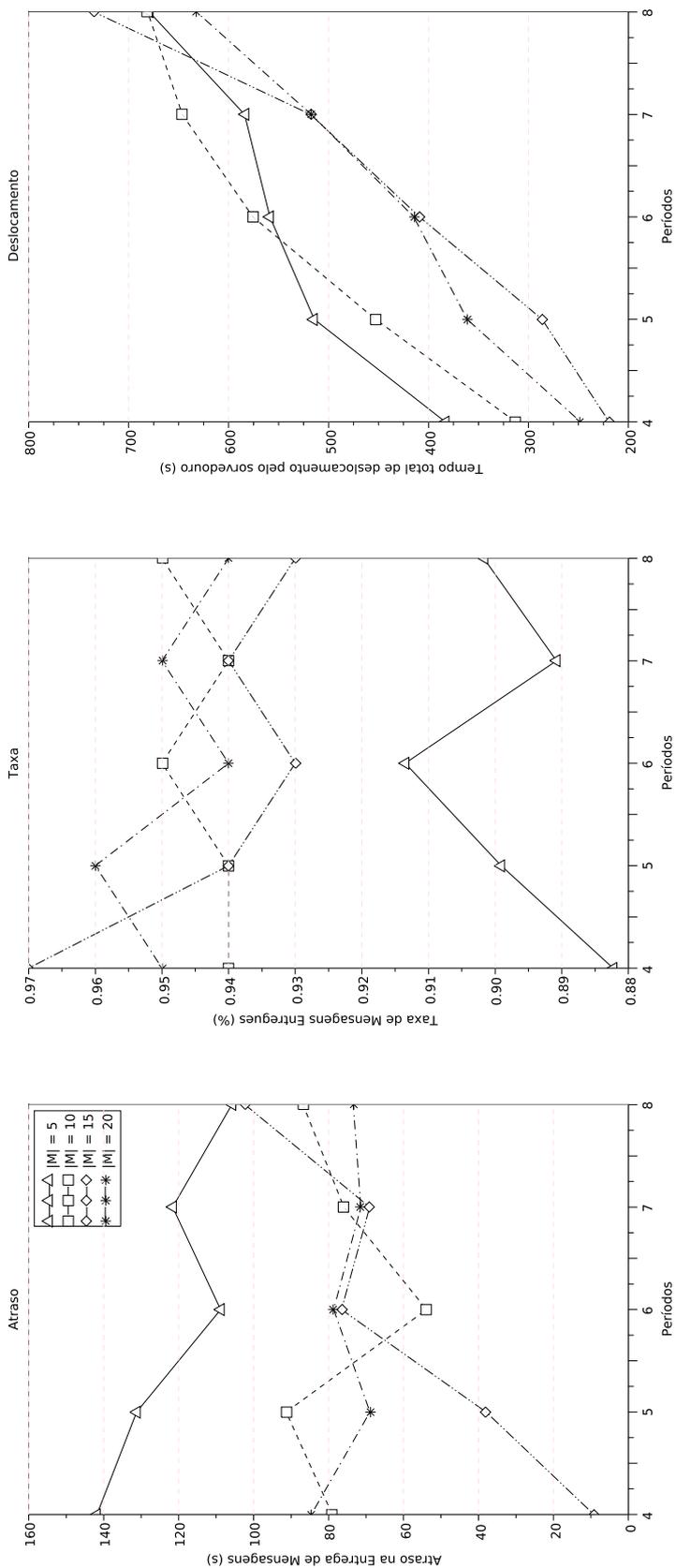


Figura 6.2. Comparação entre Atraso Médio, Taxa de Entrega e Distância Percorrida pelo Sorvedouro para 400 sensores

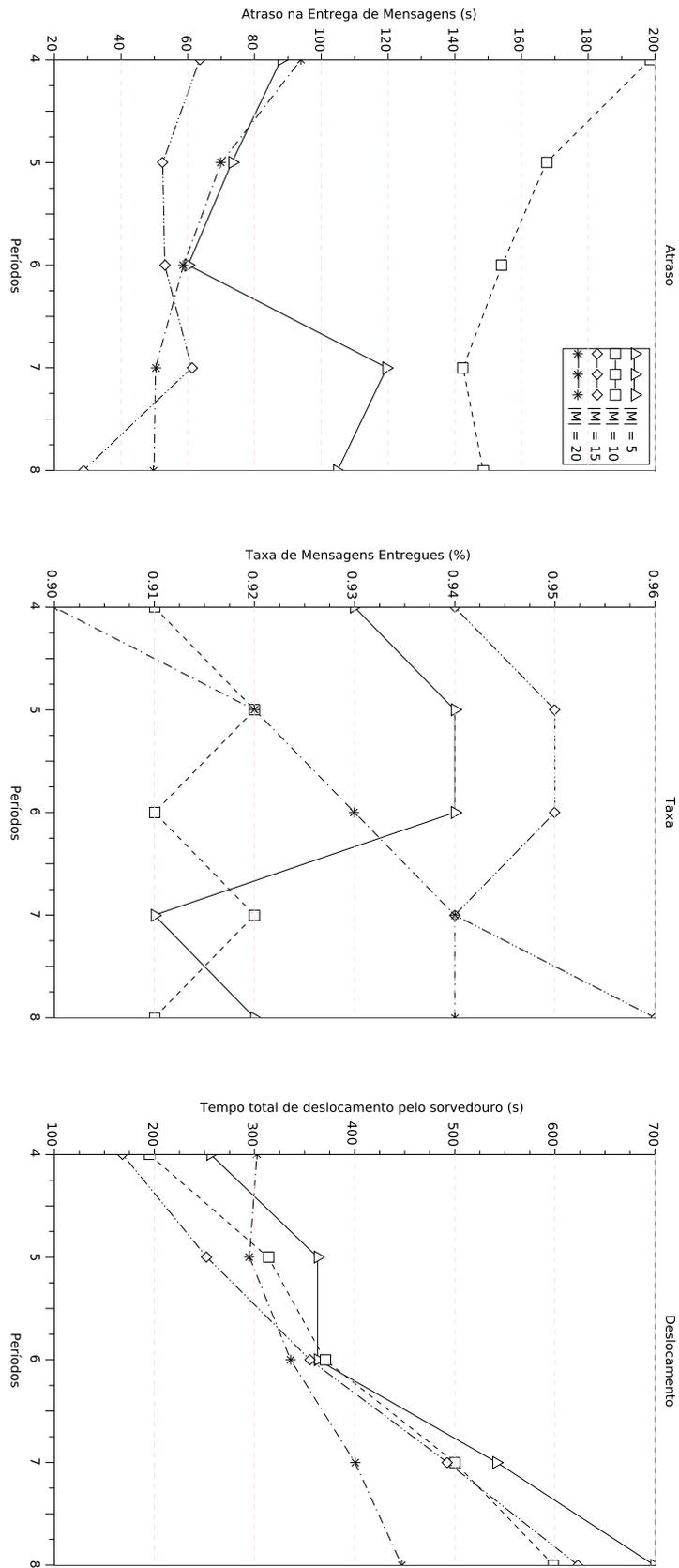


Figura 6.3. Comparação entre Atraso Médio, Taxa de Entrega e Distância Percorrida pelo Sorvedouro para 600 sensores

(sob o ponto de vista das métricas analisadas) para instâncias com 600 sensores, seriam aquelas para $|M| = 15$, uma vez que a curva que o descreve foi a que apresentou menor variação ao longo dos períodos.

Analisando a curva para $|M| = 5$ e 400 sensores, observa-se a relação entre as métricas analisadas e o tempo total percorrido pelo sorvedouro. Verifica-se, por exemplo, que o atraso médio na entrega de mensagens e a taxa de mensagens entregues atingiram uns de seus valores mais baixos e máximos, respectivamente, quando a curva, em *Deslocamento*, apresentou sua menor inclinação (no caso para $T = 6$). Além disso, com o aumento de $|M|$, percebe-se uma diminuição no tempo total de deslocamento do sorvedouro, indicando que maiores quantidades de localizações para o mesmo possibilitam uma redução média nas distâncias entre tais posições. Isso permitiu que o sorvedouro se movimentasse menos entre as posições candidatas escolhidas, ou até que permanecesse mais períodos em uma mesma posição. Tal comportamento é melhor observado nas soluções para para $|M| = 20$ e 600 sensores, já que suas curvas em *Deslocamento* foram a que apresentaram menor variação ao longo dos períodos. Dessa forma, melhoras significativas nas soluções encontradas a partir das métricas de QoS que estamos analisando, podem ser alcançadas caso consideremos que restrições na distância percorrida pelo sorvedouro sejam adicionadas ao PICDCR.

T	400 Sensores				600 Sensores			
	$ M = 5$	$ M = 10$	$ M = 15$	$ M = 20$	$ M = 5$	$ M = 10$	$ M = 15$	$ M = 20$
4	4,00%	4,01%	4,12%	4,08%	2,46%	2,48%	2,56%	2,46%
5	5,15%	5,11%	5,26%	5,21%	3,14%	3,15%	3,25%	3,19%
6	6,35%	6,29%	6,33%	6,32%	3,86%	3,86%	3,91%	3,91%
7	7,52%	7,45%	7,53%	7,50%	4,45%	4,56%	4,60%	4,62%
8	8,74%	8,66%	8,66%	8,71%	5,15%	5,25%	5,28%	5,33%

Tabela 6.1. Comparação entre a energia residual da rede encontrada pela heurística em relação àquela retornada pela simulação

A Tabela 6.1 compara os valores de energia residual das soluções encontradas pela heurística ao se resolver o PICDCR em relação àquelas retornadas pela simulação. Observa-se que a diferença entre as energias residuais foi, no máximo, igual a 8,74% e, no mínimo, igual a 2,46%, ou seja, o consumo de energia real da rede chegou a ser 8,74% menor do que do aquele encontrado pela heurística.

À medida que o número de períodos aumenta, percebe-se que as simplificações assumidas para o modelo de consumo de energia tornam-se mais críticas, ou seja, a diferença entre os valores de energia residual obtidos ao se resolver o PICDCR e pela simulação aumenta à medida que o número de períodos cresce. Um motivo para isso se deve ao fato do PICDCR ter, com o incremento de T , de se preocupar cada vez mais com reduções no consumo de energia da rede, a fim de garantir que soluções viáveis

sejam encontradas.

Contudo, tal diferença tende a diminuir com o aumento do número de sensores, indicando que os efeitos colaterais ocasionados pelas simplificações de energia assumidas neste trabalho podem ser atenuados pela redução do número de sensores ativos, assim como a redução no tamanho das rotas de dados.

Isso é evidente ao se comparar as instâncias para 400 e 600 sensores com $T = 4$. Neste caso é possível observar que a energia residual obtida pelo simulador encontra-se, em média, 4,05% superior àquela obtida pela heurística para uma instância com 400 sensores e $T = 4$, enquanto que a energia residual obtida pelo simulador encontra-se, em média, 2,49% superior àquela obtida pelo modelo para uma instância com 600 sensores e o mesmo número de períodos.

Além disso, observa-se que variações no parâmetro $|M|$ causam pouca diferença entre as energias residuais comparadas, principalmente em relação aos demais parâmetros (quantidade de sensores e número de períodos).

6.3 Comentários

O simulador de eventos discretos implementado neste capítulo permitiu avaliar a topologia das soluções encontradas pela heurística para o PICDCR. Como tal problema desconsidera falhas de cobertura e de conectividade (sensores sem energia), requisitos de QoS relacionadas à tais questões e inerentes a organização de uma RSSF, são automaticamente garantidos ao se trabalhar com soluções viáveis.

Os resultados apresentados mostram que os requisitos de QoS avaliados neste capítulo alcançaram bons níveis de qualidade, apresentando taxas de mensagens entregues médias superiores a 0.90%, enquanto que o atraso médio na entrega de mensagens, apesar de poder ser considerado elevado, não ultrapassou 162 segundos, para todas as instâncias testadas. Para se ter uma ideia quantitativa desses resultados, em uma área de 40000 m^2 , cerca de 4000 m^2 em média não estariam sendo cobertos pelos sensores, enquanto as demais regiões estariam sendo monitoradas a cada 3 minutos, aproximadamente.

Além disso, os testes computacionais mostraram que resultados melhores de QoS podem ser alcançados aumentando-se a quantidade de sensores, assim como o número de períodos. Isso indica que, com a variação de tais parâmetros, a heurística proposta focou-se em encontrar soluções que reduzissem o número de sensores ativos assim como o comprimento das rotas de dados, escolhendo posições candidatas para o sorvedouro que ocasionassem baixa ou mesma nenhuma movimentação do mesmo ao longo dos

períodos.

Observou-se também que, apesar de o aumento da quantidade de posições candidatas para o sorvedouro não provocar grandes variações no consumo de energia da rede, o mesmo provocou melhoras significativas na qualidade dos níveis de QoS avaliadas.

A partir da análise das diferenças entre as energias residuais das soluções encontradas pela heurística e àquelas retornadas pelo simulador, verificou-se que as simplificações assumidas no modelo de consumo de energia mostraram-se demasiadamente sensíveis às variações nos valores dos parâmetros relacionados a quantidade de sensores e períodos.

Dessa forma, como continuidade dessa pesquisa, pretende-se reduzir tais diferenças através de melhores simplificações no modelo de consumo de energia. Além disso, como as diferenças entre as energias residuais mostraram-se mais significativas com o aumento no número de períodos, será avaliado também formas de se trabalhar com abordagens que dividam o horizonte de planejamento da rede em um número menor de períodos, onde cada período pode ser subdividido em sub-períodos. Portanto, uma maneira seria executar um dos algoritmos de otimização propostos em ciclos de no máximo r períodos, ou seja, ao invés de executá-lo para T períodos, o executaríamos $\frac{T}{r}$ vezes, passando o estado final de um ciclo como entrada para o próximo.

No próximo capítulo será mostrado as melhoras energéticas alcançadas pelas RS-SFs ao se utilizar o problema proposto nesta dissertação em detrimento a um problema semelhante encontrado na literatura.

Capítulo 7

Comparação com a Literatura

Neste Capítulo realizamos uma comparação entre a organização da rede introduzida com o PICDCR e a proposta de Türkogulları et al. [Türkogulları et al., 2010], que, a julgar pela nossa revisão bibliográfica, é o trabalho que mais se aproxima da nossa proposta de integração.

7.1 Modelo da Literatura

Como explicado anteriormente, o trabalho proposto por Türkogulları et al. [Türkogulları et al., 2010] consiste em apresentar um modelo de PLIM baseado em fluxo em redes, que concentra-se na integração dos problemas de Cobertura, Controle de Densidade, Roteamento e Localização do Sorvedouro assim como de Sensores, através de uma abordagem multi-período. Esse problema será apresentado aqui como o *Problema de Localização, Controle de Densidade e Roteamento* (LSRP¹). O LSRP escolhe posições ótimas para os sensores e sorvedouros se localizarem na região monitorada, determina o subconjunto ótimo de sensores que permanecerão ativos, assim como seleciona as melhores rotas de dados sensor-sorvedouro, de forma a maximizar o tempo de vida de uma RSSF. Para isso, os autores optaram por trabalhar com uma rede heterogênea, cuja dispersão dos sensores é feita de forma manual, e composta por um ou mais nós sorvedouros. A área a ser monitorada é formada por J^2 pontos de demanda indexados como $j = 1, \dots, J$, cada um exigindo uma quantidade mínima de sensores necessário para cobri-lo (exigências de sensoriamento), d_j . O conjunto de sensores e de posições para o sorvedouro se localizar consiste desses mesmos J pontos. Os autores assumem que existem K diferentes tipos de sensores indexados como $k = 1, \dots, K$, cada

¹sigla em inglês para *Location, Scheduling and Routing Problem*.

²Assuma que, por um abuso de notação, $J = |J|$.

um apresentando especificações técnicas próprias, tais como raio de sensoriamento e comunicação, bem como a quantidade de mensagens, h_k , geradas durante o início de cada período. Além disso, cada tipo de sensor k possui sua própria equação de consumo de energia, caso o mesmo esteja ativo (não há gasto energético para a ativação de um sensor). Como existe um custo operacional, c_k , gasto para posicionar um sensor do tipo k na área monitorada, há um valor monetário máximo, B , limitando tal operação. Uma vez que o objetivo do LSRP consiste na maximização do tempo, L , de vida da rede, ou seja, o tempo máximo que a rede pode garantir cobertura completa da área que está sendo monitorada, os autores assumem um horizonte de planejamento suficientemente longo, consistindo de T períodos de igual duração. Dessa forma, T se torna um limite superior de L .

De posse disso, percebe-se algumas diferenças entre o modelo que propomos neste trabalho, sintetizado pelo problema de otimização PICDCR e aquele introduzido por Türkogulları. A Tabela 7.1 mostra uma comparação entre as diferenças encontradas.

	PICDCR	LSRP
Tipo de rede	Homogênea	Heterogênea
Tipo de dispersão	Aleatória	Manual
Existe custo associado ao tipo de dispersão?	Não	Sim
Quantidade de nós sorvedouros	Um	Possibilidade de vários
Existe gasto energético para se ativar um sensor?	Sim	Não
Exigência de sensoriamento	Constante	Pode ser variável
Quantidade de mensagens enviadas por um sensor ao início de cada período	Uma	Pode ser várias
Cálculo do consumo de energia	Baseado na quantidade de arcos que chegam e saem de um sensor	Baseado na quantidade de fluxos que chegam e saem de um sensor
Problema de localização do sorvedouro	Escolha de uma posição candidata a cada período	Escolha de uma posição candidata única para todos os períodos
Função objetivo	Maximizar a energia residual da rede ao final de seu horizonte de planejamento	Maximizar o tempo de vida da rede

Tabela 7.1. Diferenças entre o PICDCR e o LSRP

Para que possamos comparar as RSSFs organizadas segundo PICDCR e LSRP, tivemos que adaptar o último problema, sob dois pontos de vista diferentes: estrutural e matemático, e, as vezes, de uma forma não trivial. O primeiro consiste em alterar a estrutura da rede utilizada pelo LSRP, adequando-a àquela que consideramos neste trabalho, da seguinte forma:

- somente sensores de um único tipo (rede homogênea) serão utilizados ($K = 1$);
- a rede será composta por somente um nó sorvedouro;

- cada ponto de demanda possui uma exigência constante de sensoriamento, ou seja, cada um precisa ser coberto por pelo menos um nó sensor ($d_j = 1, \forall j \in J$);
- cada sensor gera um único pacote de dados no início de cada período ($h_k = 1$);
- o custo em se posicionar cada sensor é unitário ($c_k = 1$);
- assumimos recursos monetários ilimitados ($B = \infty$), uma vez que não estamos preocupados em reduzir o gasto operacional com o posicionamento dos sensores em uma dada região.

O segundo ponto de vista se preocupa em modificar a formulação matemática apresentada para o LSRP, de forma que uma solução para esse possa ser comparada com aquela encontrada para o PICDCR. Entretanto, no intuito de não modificar as características originais do LSRP, em sua formulação, foram alteradas as restrições relacionadas ao cálculo do consumo de energia gasto pelos sensores, já que as mesmas são diferentes daquelas que apresentamos neste trabalho (3.7). Além disso, como não estamos preocupados em reduzir o gasto operacional com o posicionamento dos sensores em uma dada região, desconsideramos a restrição da mochila associada. Após tais modificações, haverão, ainda, três diferenças entre os modelos.

A primeira é em relação à função objetivo, já que para o PICDCR, procuramos, diretamente, maximizar a energia residual da rede, enquanto que para o LSRP tal preocupação é alcançada indiretamente, através da maximização do tempo de vida da rede. A segunda está relacionada à topologia gerada pelo problema, já que enquanto o PICDCR gera soluções representadas por uma árvore, o LSRP representa suas soluções através de um grafo conexo, possivelmente envolvendo mais de um caminho entre os sensores ativos e o sorvedouro. A última diferença se dará em como o problema de Localização do Sorvedouro é tratado. Isso pois, como já visto anteriormente, em nosso trabalho consideramos que tal problema consiste em encontrar localizações ótimas para o sorvedouro na rede a cada período, enquanto que no LSRP, escolhe-se a melhor posição para que o sorvedouro, de forma que esse permaneça fixo nessa localização durante todo o tempo de vida da rede.

Para apresentar a formulação baseada em fluxo em redes para o LSRP após as modificações anteriores, é empregado o seguinte conjunto de variáveis:

- $z_i^t \in \{0, 1\}, \forall i \in J, \forall t \in T$, assumindo valor 1 se o sensor posicionado no ponto de demanda i está ativo no período t ($z_i^t = 0$, caso contrário);
- $s_i \in \{0, 1\}, \forall i \in J$, assumindo valor 1 se o sorvedouro é posicionado no ponto de demanda i ($s_i = 0$, caso contrário);

- $y^t \in \{0, 1\}, t \in T$, assumindo valor 1 se o período t está dentro do tempo, L , de vida da rede ($y_t = 0$, caso contrário);
- $e_i^t \in \mathbb{R}, \forall i \in J, \forall t \in T$ representa a energia disponível no sensor posicionado no ponto de demanda i , ao final do período t .
- $f_{ij}^t \in \mathbb{R}, \forall i, j \in J, \forall t \in T$, sendo a quantidade de fluxo de dados no período t do sensor localizado no ponto de demanda i para aquele localizado no ponto de demanda j ;
- $g_{ij}^t \in \mathbb{R}, \forall i, j \in J, \forall t \in T$, sendo a quantidade de fluxo de dados no período t do sensor localizado no ponto de demanda i para o sorvedouro localizado no ponto de demanda j ;
- $w_{ij}^t \in \{0, 1\}, \forall i, j \in J, \forall t \in T$, assumindo valor 1 se o arco, cujas extremidades são sensores posicionados nos pontos de demanda i e j , existe ($w_{ij}^t = 0$, caso contrário);
- $r_{ij}^t \in \{0, 1\}, \forall i, j \in J, \forall t \in T$, assumindo valor 1 se o arco, cuja extremidade contém um sensor posicionado no ponto de demanda i e, na outra, o sorvedouro posicionado no ponto j , existe ($r_{ij}^t = 0$, caso contrário);
- $L \in \mathbb{R}$, sendo o tempo de vida de uma RSSF;

Os parâmetros usados no modelo são apresentados a seguir:

- J , o número de pontos de demanda na área monitorada;
- u , o limite superior da quantidade de dados que são enviados entre os sensores ou entre sensores e sorvedouro;
- a_{ij} , assume valor 1 se o sensor posicionado no ponto de demanda i cobre o ponto j , ou seja, se j está no alcance do raio de sensoriamento (R_s) do sensor (0, caso contrário);
- b_{ij} , assume valor 1 se o ponto de demanda j está dentro do alcance do raio de transmissão (R_c) do sensor posicionado no ponto de demanda i (0, caso contrário);
- T , horizonte de planejamento da rede;

A fim de apresentarmos uma formulação mais clara para o LSRP, os parâmetros $K, d_j, \forall j \in J, c_k, \forall k \in K, h_k, \forall k \in K$, não serão mostrados no modelo, uma vez que os mesmos apresentam valores unitários. De posse disso, a formulação simplificada para o LSRP, a partir de agora chamada de LSRP-M, é dada por:

$$\max L \quad (7.1)$$

$$\sum_{i=1}^J a_{ij} z_i^t \geq y^t, \quad \forall j \in J, t \in T \quad (7.2)$$

$$T y_t \geq L + 1 - t, \quad \forall t \in T \quad (7.3)$$

$$z_i^t + \sum_{j=1, j \neq i}^J b_{ji} f_{ji}^t = \sum_{j=1, j \neq i}^J b_{ij} f_{ij}^t + \sum_{j=1}^J b_{ij} g_{ij}^t, \quad \forall i \in J, t \in T \quad (7.4)$$

$$w_{ij}^t \geq z_i^t + z_j^t - 1, \quad \forall i, j \in J, i \neq j, t \in T \quad (7.5)$$

$$w_{ij}^t \leq f_{ij}^t, \quad \forall i, j \in J, i \neq j, t \in T \quad (7.6)$$

$$r_{ij}^t \geq z_i^t + s_j - 1, \quad \forall i, j \in J, t \in T \quad (7.7)$$

$$r_{ij}^t \leq g_{ij}^t, \quad \forall i, j \in J, t \in T \quad (7.8)$$

$$e_i^t = Eb - Er \left(\sum_{j \in J \setminus \{i\}} w_{ji}^t \right) - \left(\sum_{j \in J \setminus \{i\}} Et_{ij} w_{ij}^t + \sum_{i \in J} Et_{ij} r_{ij}^t \right) - Em.z_i^t, \quad i \in J, t \in T, t = 0$$

$$e_i^t = e_i^{t-1} - Er \left(\sum_{j \in J \setminus \{i\}} w_{ji}^t \right) - \left(\sum_{j \in J \setminus \{i\}} Et_{ij} w_{ij}^t + \sum_{j \in J} Et_{ij} r_{ij}^t \right) - Em.z_i^t, \quad i \in J, t \in T, t \geq 1 \quad (7.9)$$

$$\sum_{i=1}^J s_i = 1 \quad (7.10)$$

$$f_{ij}^t \leq ub_{ij} z_i^t, \quad \forall i, j \in J, i \neq j, t \in T \quad (7.11)$$

$$f_{ij}^t \leq ub_{ij} z_j^t, \quad \forall i, j \in J, i \neq j, t \in T \quad (7.12)$$

$$g_{ij}^t \leq ub_{ij} z_i^t, \quad \forall i, j \in J, t \in T \quad (7.13)$$

$$g_{ij}^t \leq ub_{ij} s_j, \quad \forall i, j \in J, t \in T \quad (7.14)$$

$$z_i^t, s_i, y^t, w_{ij}^t, r_{ij}^t \in \{0, 1\}, \quad e_i^t \geq 0, \quad f_{ij}^t \geq 0, \quad g_{ij}^t \geq 0, \quad L \geq 0, \quad \forall t \in T, i, j \in J \quad (7.15)$$

A função objetivo (7.1) maximiza o tempo de vida em que a rede permanece funcional, garantindo cobertura completa da área monitorada. As restrições (7.2) e (7.3) garantem que, durante o tempo de vida da RSSF (quer dizer, para $t \leq L$), cada ponto de demanda será coberto por pelo menos um nó sensor. As restrições em (7.4) asseguram o balanço de fluxo. Garantem que se um sensor i está ativo no período t , a soma entre os dados gerados por esse e as informações que chegam em i por outros sensores é igual ao total de dados que sai desse sensor, sendo transmitidas para

outros sensores ou sorvedouro. As restrições (7.5)-(7.8) foram adicionadas ao modelo original do LSRP no intuito de verificar a existência ou não de um arco qualquer através de variáveis binárias (w e r). Isso foi necessário, uma vez que o conjunto de restrições adicionados a partir de (3.7), e relacionadas ao cálculo do consumo de energia gasto pelos sensores (7.9), são feitas a partir da existência ou não de arcos que chegam e saem de um determinado nó. Em (7.10), as restrições garantem que apenas um nó sorvedouro exista na rede. O conjunto de restrições (7.11)-(7.14) asseguram que, caso um arco exista, o mesmo possuirá em suas extremidades nós sensores ativos comunicando-se entre si ou um nó sensor ativo se comunicando com o sorvedouro. O domínio das variáveis são definidos em (7.15).

Percebe-se que as restrições (7.9) quando comparadas àquelas em (3.7), não apresentam gasto com a ativação de um sensor (Ea), já que no LSRP isso não é considerado. Dessa forma, durante os resultados, será desconsiderado o gasto com Ea no PICDCR para que os problemas sejam comparados em termos de consumo de energia.

Apesar de nada ter sido dito sobre a topologia da solução encontrada para o LSRP (e consequentemente para o LSRP-M), ao analisar a formulação para tal problema pode-se concluir que essa será representada por um grafo conexo. Isso pode ser verificado através das restrições de balanço de fluxo (7.4), onde permite-se que um sensor possa se comunicar com outros (mas não com ele mesmo, impedindo que o grafo seja desconexo) ao mesmo tempo em que comunica-se com o sorvedouro.

7.2 Resultados computacionais

Nesta seção, conduziremos alguns experimentos computacionais com o objetivo de compararmos a qualidade, em termos energéticos, das soluções geradas ao se resolver o LSRP-M e o PICDCR.

Para isso, foram criadas instâncias cuja área a ser monitorada apresentou dimensões variadas, cada uma possuindo quantidades de pontos de demanda (gerados de forma aleatória e a partir de uma distribuição uniforme) equivalentes, em termos numéricos, a tais dimensões (assim como feito em [TürkogullarI et al., 2010]). Além disso, foi considerado que cada ponto de demanda está no alcance do raio de comunicação de pelo menos um outro ponto (assumindo-se que, em ambos, possam haver sensores posicionados sobre esses), a fim de garantir, para toda instância, a existência de uma solução que possa ser representada por um grafo conexo. Tal consideração é razoável devido à própria natureza do LSRP, uma vez que um ponto de demanda é coberto ou por um nó sensor ativo posicionado sobre esse, ou por algum sensor ativo

localizado em outra posição, mas cuja distância entre ele e o ponto é menor do que R_s .

A formulação para o LSRP-M foi implementada utilizando-se o CPLEX [sol, 2010] (versão 10.2 com configuração padrão) e seus resultados, comparados àqueles encontrados a partir da execução do algoritmo exato BC (Capítulo 4) proposto para o PICDCR. Ambos foram executados impondo-se para eles um limite máximo de 4 horas. Ao final desse tempo, as execuções são interrompidas e os melhores limites inferior e superior são recuperados.

Os parâmetros utilizados foram, em sua maioria, os mesmos apresentados na Tabela 4.1, diferenciando-se somente os valores de raio de comunicação, igual a $2m$ e de sensoriamento, igual a $1m$. Além disso, a relação entre a energia gasta com a transmissão e a distância entre os sensores (Tabela 4.2) teve que ser alterada para melhor condizer aos valores de raio de comunicação utilizados, uma vez que do contrário o consumo de energia gasto com a transmissão seria constante. Isso foi feito simplesmente dividindo os valores de distância por dez.

Já os parâmetros T e u , próprios do modelo proposto para o LSRP-M, assumiram determinados valores a partir de algumas características (explicadas a seguir) do problema em questão.

Restrições do tipo *Big-M* são utilizadas como forma de linearizar restrições condicionais através da adição de uma constante "M", de tal forma a permitir uma relação entre variáveis binárias e reais. Porém, a adição desse tipo de restrição em modelos matemáticos torna-os ruins, uma vez que seus limites de relaxação linear são fracos. Além disso, uma atenção deve ser tomada quanto aos parâmetros vinculados a esse "M", como é o caso de T e u nas restrições desse tipo (7.3), (7.11)-(7.14) encontradas na formulação para o LSRP (LSRP-M). Isso porque valores altos para "M" dificultam a convergência do modelo para a otimalidade durante sua resolução, enquanto que valores baixos possibilitam que soluções viáveis sejam descartadas.

Apesar de os autores em [TürkogullarI et al., 2010] considerarem valores para T e u bem maiores do que os assumidos neste trabalho, tivemos que optar em reduzi-los, uma vez que a adição das restrições (7.5)-(7.9) na formulação para o LSRP tornou a mesma muito mais difícil de ser resolvida na otimalidade. Dessa maneira, a partir de uma análise empírica atribuímos a T valor igual a 8, já que para valores maiores do que esse o LSRP-M apresentou dificuldades em encontrar soluções viáveis. Apesar de soluções viáveis serem descartadas utilizando-se tal valor para T , para algumas instâncias esse valor foi suficiente para permitir que se encontrasse o tempo de vida máximo da rede. A partir disso, conclui-se que as soluções encontradas poderão apresentar valores de L que não mais representem o tempo máximo de vida da rede, ou seja, essas poderão possuir energia residual suficiente para permitir que consigam monitorar

completamente uma região por mais alguns períodos.

Uma vez que a formulação para o LSRP é baseada em fluxo e não impõe-se restrições quanto a topologia das soluções geradas (assume-se simplesmente que a solução seja representada por um subgrafo conexo), há a necessidade de limitar a quantidade de fluxo que passa em um arco qualquer, já que pode haver a formação de ciclos. Na formulação original para o LSRP isso é feito através de um limite u máximo permitido de fluxo que possa passar por um arco qualquer, assim como pelas restrições de cálculo de energia, já que essas consideram a quantidade de fluxo que passam nos arcos. Entretanto, como no LSRP-M estamos interessados somente na existência ou não dos arcos para o cálculo de energia (7.9), e uma vez que as restrições de energia originais do LSRP foram retiradas, assumimos que u terá valor igual a J . Isso permite representar a quantidade máxima de fluxo que poderá passar em um determinado arco caso a solução para o LSRP-M possa ser representada por uma árvore (assim como é feito no PICDCR).

Dessa forma, percebemos que apesar dos problemas comparados apresentarem em suas formulações o mesmo cálculo do consumo de energia, algumas diferenças entre os modelos podem influenciar na melhora ou na piora das soluções encontradas, tais como o foco de suas funções objetivo. Mais ainda, diferentemente do PICDCR, que garante que a topologia da solução seja uma árvore, com o sorvedouro sendo a raiz, o LSRP apresenta soluções representadas através de um grafo conexo.

A partir do que foi exposto, dividimos os testes computacionais em dois grupos diferentes, no intuito de comparar mais precisamente a energia residual encontrada pelas soluções dos problemas que estão sendo analisados. No primeiro grupo, avaliamos o consumo energético ocasionado somente pela topologia que está sendo utilizada para representar as soluções encontradas. Para isso, primeiramente resolvemos o LSRP-M, e a melhor posição escolhida para o sorvedouro é passada como entrada para o PICDCR. Tal metodologia permite que possamos desconsiderar o problema de localização do sorvedouro da forma como definida neste trabalho, e conseqüentemente o possível ganho energético ocasionado pela mobilidade do mesmo.

Já no segundo grupo, comparamos a melhora das soluções ao confrontar a forma como o problema de localização é definido nos dois problemas, ou seja, avaliaremos o benefício energético ao permitir que o sorvedouro se locomova periodicamente em detrimento à abordagem de fazê-lo permanecer fixo durante todo o tempo de vida da rede. Isso será feito simplesmente fazendo com que o conjunto M seja igual ao conjunto J .

As instâncias de teste geradas podem ser utilizadas pelo PICDCR considerando que cada ponto de demanda possui um sensor sobre esse, o que é perfeitamente con-

dizente com o que é feito pelo LSRP-M, já que esse não se preocupa em reduzir a quantidade de sensores utilizados. Além disso, assim como no LSRP-M, o horizonte de planejamento definido para o PICDCR foi considerado como sendo igual a 8.

Na Tabela 7.2 são mostrados os resultados obtidos pelo LSRP-M. A colunas *Dimensão*, *J* e *Tipo* mostram respectivamente a dimensão da área monitorada, a quantidade de pontos de demanda nessa e o tipo de instância, uma vez que foram geradas três instâncias aleatórias para cada dimensão de área utilizada (representando cada dispersão aleatória pelos números 1, 2, 3). L e \bar{L} representam o tempo de vida da rede (podendo ser máximo ou não), assim como o melhor limite dual superior encontrado ao longo da árvore de enumeração. As três últimas colunas representam respectivamente o tempo de execução, a energia residual da rede ao final desse tempo e o gap de dualidade ($\frac{\bar{L}-L}{\bar{L}}$) encontrado.

	Dimensão (m^2)	J	Tipo	L	\bar{L}	Tempo (s)	Energia (mAh)	Gap (%)
1	10	10	1	8	8	0,48	2956,4	0
2	10	10	2	8	8	0,45	2806,5	0
3	10	10	3	8	8	0,39	2960,5	0
4	15	15	1	4	4	1,97	4162	0
5	15	15	2	4	4	2,07	4080,5	0
6	15	15	3	6	6	98,03	3109,2	0
7	20	20	1	6	6	285,83	6093,0	0
8	20	20	2	8	8	55,28	3206,8	0
9	20	20	3	8	8	176,71	4067,3	0
10	25	25	1	7	8	14400,00	3643,0	14,29
11	25	25	2	8	8	276,77	3641,8	0
12	25	25	3	8	8	243,52	3724,0	0
13	30	30	1	5	7,12	14400,00	6644,7	34,73
14	30	30	2	8	8	3874,80	6049,7	0
15	30	30	3	8	8	8192,60	6110,8	0
16	40	40	1	8	8	12566,90	7643,5	0
17	40	40	2	2	6,90	14400,00	12209,1	83,06
18	40	40	3	8	8	3646,80	4839,8	0
19	60	60	1	-	-	-	-	-
20	60	60	2	-	-	-	-	-
21	60	60	3	-	-	-	-	-

Tabela 7.2. Soluções encontradas pelo CPLEX para o LSRP-M

O fato de assumirmos um valor para T , pequeno o suficiente no intuito de se resolver o LSRP-M para todas as instâncias testadas, implicou na possibilidade do tempo de vida da rede encontrado não poder ser máximo. Dessa forma, ao analisarmos as soluções encontradas para o LSRP-M, levaremos em conta a seguinte relação:

$$L \leq \bar{L} < T \quad (7.16)$$

Tal relação nos diz que o horizonte de planejamento, T , foi longo o suficiente para permitir que o tempo de vida da rede pudesse ser maximizado (caso em que $L = \bar{L}$) ou que, pelo menos, as soluções viáveis encontradas possuam um tempo de vida máximo inferior a T . Já para aqueles resultados onde tal relação não pode ser atendida,

têm-se ao menos uma solução viável para ser comparada com aquelas retornadas pelo PICDCR.

Apesar de o CPLEX ter encontrado soluções ótimas (gap de dualidade igual a zero) para a maioria das instâncias testadas para o LSRP-M, apenas algumas poucas atenderam à relação (7.16) (linhas 4,5,6,7,13,17 da Tabela 7.2). Além disso, na medida em que aumentou-se a dimensão da área a ser monitorada, soluções viáveis para o LSRP-M ficaram cada vez mais difíceis de serem obtidas, ao ponto de nenhuma ter sido encontrada para instâncias de dimensão igual a 60 (linhas 19-21). Isso pode ser verificado comparando-se por exemplo, a partir do tempo de execução gasto, as instâncias com dimensão de área igual a $10m^2$ (linhas 1-3) com aquelas com dimensão igual a $40m^2$ (linhas 16-18).

Na Tabela 7.3 são apresentados os resultados obtidos pelo PICDCR. As linhas ímpares mostram os resultados do primeiro grupo de testes realizados ao se fazer $|M| = 1$, ou seja, a única localização possível para o sorvedouro será aquela retornada pelo LSRP-M (para uma mesma instância). Observa-se que para todas as instâncias testadas, os resultados encontrados pelo PICDCR apresentaram menores consumos de energia do que aqueles encontrados pelo LSRP-M (colunas 8, 9 e 10), chegando-se a uma melhora de até 62,89% (linha 35). Além disso, a maior evidência da melhora ocasionada pela utilização do PICDCR aparece durante a comparação entre seus resultados e aqueles que atenderam à relação (7.16), obtidos pelo LSRP-M (linhas 7, 9, 11, 13, 25, 33). Percebe-se que, para as instâncias de testes relacionadas a tais resultados, o PICDCR não só conseguiu obter melhor eficiência energética ao prolongar o tempo de vida máximo da rede entre 10,89% (linha 25) à 50% (linhas 7 e 9), como conseguiu encontrar soluções apresentando 8 períodos e consumindo menos energia do que às obtidas pelo LSRP-M. Isso indica que a proposta do PICDCR em representar a topologia de sua solução através de uma árvore, aliada ao fato de explicitamente maximizar a energia residual da rede, permitiu que se obtivesse melhores resultados energéticos em relação a abordagem utilizada pelo LSRP-M nesses quesitos. Um motivo para isso considera o fato de que a representação de uma solução através de um grafo conexo, apesar de apresentar vantagens quanto tolerância à falhas, redundâncias de dados, entre outras que fogem do escopo deste trabalho, permite múltiplas comunicações entre os sensores, o que causa degradação, em termos energéticos, na qualidade das soluções encontradas.

Já o segundo grupo de testes realizados, ou seja, fazendo-se $M = J$, é mostrado nas linhas pares da Tabela 7.3. Como era de se esperar, os resultados encontrados pelo PICDCR quando o sorvedouro pode se localizar em uma posição diferente a cada período foi superior àqueles nos quais o mesmo permaneceu em uma única localização.

A melhora ocasionada pela abordagem de mobilidade do sorvedouro chegou a 7,75% (linha 7), evidenciando o que já foi anteriormente dito neste trabalho: que o problema de localização, assim como definido nesta dissertação, possibilita estender o tempo de vida da rede, principalmente quando comparado à forma como tal problema é tratado no LSRP.

	S	M	Tipo	PICDCR			LSRP-M		
				Energia (mAh)	Gap (%)	Tempo (s)	L	L	Energia (mAh)
1	10	1	1	3483,8	0,00	0,02	8	8	2956,4
2	10	10	1	3503,0	0,00	0,14			
3	10	1	2	3498,2	0,00	0,01	8	8	2806,5
4	10	10	2	3500,6	0,00	1,98			
5	10	1	3	3506,2	0,00	0,02	8	8	2960,5
6	10	10	3	3507,0	0,00	1,81			
7	15	1	1	4314,2	0,00	0,04	4	4	4162
8	15	15	1	4648,6	0,00	0,08			
9	15	1	2	4411,0	0,00	0,91	4	4	4080,5
10	15	15	2	4729,4	0,00	0,39			
11	15	1	3	4515,0	0,00	0,30	6	6	3109,2
12	15	15	3	4656,6	0,00	16,77			
13	20	1	1	6412,6	0,00	1,80	6	6	6093,0
14	20	20	1	6570,2	0,70	14440,00			
15	20	1	2	6400,0	0,00	0,15	8	8	3206,8
16	20	20	2	6480,0	0,00	167,74			
17	20	1	3	6652,6	0,00	0,79	8	8	4067,3
18	20	20	3	6652,6	0,00	181,56			
19	25	1	1	7805,2	0,00	2,78	7	8	3643,0
20	25	25	1	7968,4	0,00	988,84			
21	25	1	2	7738,8	0,00	286,55	8	8	3641,8
22	25	25	2	7894,9	1,20	14440,00			
23	25	1	3	8401,6	0,00	0,44	8	8	3724,0
24	25	25	3	8413,6	0,45	14440,00			
25	30	1	1	9396,8	0,00	0,09	5	7,12	6644,7
26	30	30	1	9388,6	1,17	14440,00			
27	30	1	2	9885,2	0,00	0,18	8	8	6049,7
28	30	30	2	9902,2	1,06	14440,00			
29	30	1	3	9575,8	1,72	14576,40	8	8	6110,8
30	30	30	3	9873,9	1,66	14440,00			
31	40	1	1	13471,0	0,00	1532,50	8	8	7643,5
32	40	40	1	13627,0	0,95	14440,00			
33	40	1	2	12882,8	0,00	2,47	2	6,90	12209,1
34	40	40	2	13027,6	0,00	5575,30			
35	40	1	3	13040,4	0,00	185,39	8	8	4839,8
36	40	40	3	13045,2	1,58	14440,00			

Tabela 7.3. Soluções encontrada para o PICDCR

7.3 Comentários

Nesta dissertação apresentamos um novo problema de otimização. Por isso, ao compará-lo neste capítulo com um problema semelhante proposto por TürkogullarI et al. [TürkogullarI et al., 2010] tivemos que considerar modificações tanto nos quesitos estruturais da rede quanto modificações na formulação matemática do problema comparado. Isso impossibilitou que pudéssemos realizar comparações melhores do que aquelas apresentadas.

Apesar disso, o foco deste capítulo foi mostrar quais as vantagens energéticas de uma RSSF organizada segundo PICDCR daquela segundo LSRP. Embora muito dos resultados apresentados não tenham atendido à relação (7.16), pode-se retirar importantes informações dos poucos que o fizeram. Observou-se, por exemplo, que o PICDCR prolongou o tempo de vida de rede em até 50%, além de apresentar uma energia residual para a rede ao final de 8 períodos melhor do que aquela encontrada pelo LSRP-M ao final de 4 períodos. Além disso, reafirmamos a vantagem ocasionada pela movimentação periódica do sorvedouro, uma vez que as reduções de energia chegaram a 7,75% em relação a abordagem de se utilizar o sorvedouro fixo em uma determinada posição. A partir disso, podemos concluir que a utilização do PICDCR durante a organização de uma RSSF proporciona reduções significativas no consumo de energia quando comparado à abordagens alternativas encontradas na literatura.

Uma das explicações para que o PICDCR fosse tão melhor, em termos energéticos, do que o LSRP-M (LSRP) se deve ao fato da topologia das soluções do segundo serem representadas simplesmente através de um grafo conexo. Apesar de apresentar vantagens quanto tolerância à falhas, com redundâncias de dados para alguns nós, entre outras que fogem do escopo deste trabalho, tal topologia permite múltiplas comunicações entre os sensores, o que causa degradação, em termos energéticos, na qualidade das soluções encontradas.

Capítulo 8

Conclusão e Trabalhos Futuros

Neste trabalho, procuramos reduzir o consumo de energia de uma Rede de Sensores Sem Fio (RSSF) ao propor um modelo e algoritmos de otimização que integram vários problemas relacionados à organização ótima desse tipo de rede. O principal problema tratado aqui diz respeito à integração do Problema de Controle de Densidade, Cobertura, Roteamento e Localização do Sorvedouro, de forma a maximizar o tempo de vida da rede. Embora muitos trabalhos na literatura tenham apresentado certo nível de integração entre alguns dos problemas considerados, nenhum apresentou o nível de integração que estamos propondo.

O modelo que propomos consiste em maximizar a energia residual de uma RSSF ao final de seu horizonte de planejamento, a partir da escolha, a cada período, de um subconjunto de sensores que permanecerão ativos, sensoriando completamente a área a ser monitorada, e transmitindo as informações coletadas por rotas de dados eficientes em termos energéticos entre cada par sensor-sorvedouro, no qual o último está otimamente localizado na rede. Este problema, denominado *Problema Integrado para Controle de Densidade, Cobertura e Roteamento* (PICDCR), foi formulado como um Problema em Otimização em Grafos. Além dessa formulação, foi apresentada uma formulação em Programação Inteira Mista, baseada nas Desigualdades de Eliminação de Circuitos Direcionada (DCSET).

A formulação em Programação Inteira Mista foi avaliada computacionalmente, através da implementação de um algoritmo exato, Branch-and-Cut. Os resultados indicam que investigações futuras quanto à utilização de outras desigualdades válidas para o problema, bem como a implementação de heurísticas a serem utilizadas durante o processamento dos nós da árvore de enumeração, podem vir a melhorar seus resultados.

Devido ao elevado custo computacional exigido pelo algoritmo exato para se resolver o PICDCR, uma heurística baseada em na Metaheurística ILS foi proposta para

obter soluções viáveis e de boa qualidade, apresentando tempos de execução mais adequados para a aplicação considerada.

Um simulador de eventos discretos foi implementado a partir de um arcabouço de simulação a fim de avaliar, através de determinadas métricas de QoS, o comportamento das soluções encontradas pela heurística aqui introduzida. Os resultados empíricos, além daqueles encontrados pela simulação, permitiram concluir que o PICDCR foi capaz de encontrar soluções apresentando reduções no consumo de energia, além de bons níveis de QoS (chegando-se a alcançar uma taxa de entrega de mensagens igual a 97% e atraso médio igual a 9,22 segundos). Para reforçar tais conclusões, comparamos o problema proposto com um semelhante encontrado na literatura. As comparações entre os resultados encontrados mostram que as hipóteses feitas pelo PICDCR assim como a mobilidade periódica do sorvedouro permitiram ganhos energéticos de até 62,89%, com a possibilidade de prolongamento no tempo de vida da rede em até 50%.

Nos resultados observados, entretanto, não foram analisadas formas diferentes de se construir o conjunto de posições candidatas para o sorvedouro, tão pouco o comportamento da solução caso os raios de comunicação e sensoriamento variassem durante os períodos e fossem dependentes da energia residual do sensor. Dessa forma, pretendemos avaliar o impacto causado nas soluções geradas pelo PICDCR ao levar-se em conta tais questões. Além disso, pretendemos propor também novos modelos matemáticos que possibilitem variações na taxa de cobertura, ou seja, possibilitem a ocorrência de falhas na cobertura da área a ser monitorada, caso isso venha a permitir que o tempo de vida da rede consiga ser prolongado. Por fim, questões, tais como falhas nos sensores / proteção no roteamento, limitações quanto ao tempo de deslocamento do sorvedouro, além da redivisão dos períodos com otimizações globais e locais serão também abordadas como continuidade desse trabalho.

Referências Bibliográficas

- [jis, 2007] (2007). Java in simulation time / scalable wireless ad hoc network simulator. <http://jist.ece.cornell.edu/>.
- [sol, 2010] (2010). Ilog cplex solver. <http://www.ilog.com/products/cplex>.
- [mic, 2011] (2011). Xbow. mica2 - wireless measurement system. Source: <http://www.xbow.com/>.
- [Akyildiz et al., 2002] Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38:393--422.
- [Al-karaki & Kamal, 2004] Al-karaki, J. N. & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6--28.
- [A.Land & Doig, 1960] A.Land & Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497--520.
- [Anastasi et al., 2009] Anastasi, G.; Conti, M.; Francesco, M. D. & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537--568.
- [Bagheri et al., 2007] Bagheri, M.; Bagheri, C. M. & Bagheri, M. (2007). Efficient k-coverage algorithms for wireless sensor networks. *IEEE Transactions on Networking*.
- [Bogdanov et al., 2004] Bogdanov, A.; Maneva, E. & Riesenfeld, S. (2004). Power-aware base station positioning for sensor networks. In *in Proc. IEEE Infocom*, pp. 575--585.
- [Bussab & Morettin, 2003] Bussab, W. & Morettin, P. (2003). *Estatística Básica*. SA-RAIVA.
- [Chakrabarty et al., 2002] Chakrabarty, K.; Member, S.; Iyengar, S. S.; Qi, H. & Cho, E. (2002). Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51:1448--1453.

- [Chang & Tassiulas, 2004] Chang, J.-H. & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12:609--619.
- [Ciciriello et al., 2007] Ciciriello, P.; Mottola, L. & Picco, G. P. (2007). Efficient routing from multiple sources to multiple sinks in wireless sensor networks. In *Proceedings of the 4th European conference on Wireless sensor networks, EWSN'07*, pp. 34--50, Berlin, Heidelberg. Springer-Verlag.
- [Cui et al., 2005] Cui, S.; Madan, R.; Goldsmith, A. J. & Lall, S. (2005). Joint routing, mac, and link layer optimization in sensor networks with energy constraints. In *in Proc. IEEE ICC 2005*, pp. 725--729.
- [Daskin, 1995] Daskin, M. S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley-Interscience, 1 edição.
- [de Andrade et al., 2009] de Andrade, I. B. D.; Mateus, G. R. & Nakamura, F. G. (2009). A grasp heuristic to density control: Solving multi-period coverage and routing problems in wireless sensor networks. In *ISCC*, pp. 493--499.
- [Desaulniers et al., 2005] Desaulniers, G.; Desrosiers, J. & Solomon, M. M. (2005). *Column Generation*. Springer.
- [Dinitz, 2006] Dinitz, Y. (2006). Dinitz' Algorithm: The Original Version and Even's Version. In Goldreich, O.; Rosenberg, A. & Selman, A., editores, *Theoretical Computer Science*, volume 3895 of *Lecture Notes in Computer Science*, chapter 10, pp. 218--240. Springer Berlin / Heidelberg, Berlin, Heidelberg.
- [Fan & Jin, 2010] Fan, G. & Jin, S. (2010). Coverage problem in wireless sensor network: A survey. *Journal of Networks*, 5(9).
- [Fang & Low, 2007] Fang, C. & Low, C. P. (2007). Redundant coverage in wireless sensor networks. *ICC 2007 proceedings*, pp. 3535--3540.
- [Feo & Resende, 1995] Feo, T. & Resende, M. (1995). Greedy randomized adaptive search procedures. *J. of Global Optimization*, 6:109--133.
- [Gandham, 2003] Gandham, S. D. M. P. R. V. S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. *EEE Global Telecommunications Conference, GLOBECOM '03*, 1:377--381.
- [gang Gong et al., 2009] gang Gong, H.; Cao, J.; Liu, M.; Chen, L. & Xie, L. (2009). A traffic aware, energy-efficient mac protocol for wireless sensor networks. *IJAHUC*, 4(3/4):148--156.

- [Garey & Johnson, 1979] Garey, M. & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*.
- [Goldbarg & Luna, 2000] Goldbarg, M. C. & Luna, H. P. L. (2000). *Otimização Combinatória e Programação Linear – Modelos e algoritmos*. Editora Campus, 2ed edição.
- [Güney et al., 2010] Güney, E.; Aras, N.; AltInel, I. K. & Ersoy, C. (2010). Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks. *Computer Networks*, 54(11):1805--1822.
- [Han, 2005] Han, J. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [Hua & Peter Yum, 2008] Hua, C. & Peter Yum, T.-S. (2008). Data aggregated maximum lifetime routing for wireless sensor networks. *Ad Hoc Netw.*, 6:380--392.
- [Hua & Yum, 2008] Hua, C. & Yum, T.-S. P. (2008). Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Trans. Netw.*, 16:892--903.
- [Huang & Tseng, 2003] Huang, C.-F. & Tseng, Y.-C. (2003). The coverage problem in a wireless sensor network. In *Wireless Sensor Networks and Applications*, pp. 115--121.
- [Jongerden & Haverkort, 2008] Jongerden, M. R. & Haverkort, B. R. H. M. (2008). Which battery model to use? In *24th UK Performance Engineering Workshop, London, UK*, number DTR08-9 in Technical Report Series of the Department of Computing, Imperial College London, pp. 76--88. Imperial College London.
- [Li & Mohapatra, 2007] Li, J. & Mohapatra, P. (2007). Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive Mob. Comput.*, 3:233--254.
- [Lin et al., 2009] Lin, F. Y.-S.; Yen, H.-H. & Lin, S.-P. (2009). Delay qos and mac aware energy-efficient data-aggregation routing in wireless sensor networks. *Sensors*, 9(10):7711--7732.
- [Lucena & Beasley, 1996] Lucena, A. & Beasley, J. (1996). In *Advances in Linear and Integer Programming*, pp. 187--221. Oxford University Press.
- [Marcos et al., 2003] Marcos; Luiz; Ruiz, L. B.; Loureiro, A. A. F.; Fernandes, A. O. & Jose; (2003). Scheduling nodes in wireless sensor networks: A voronoi approach. In

- LCN '03: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, Washington, DC, USA. IEEE Computer Society.
- [Miodrag & Potkonjak, 2003] Miodrag, S. M. & Potkonjak, M. (2003). Low power 0/1 coverage and scheduling techniques in sensor networks. Technical report, University of California at Los Angeles.
- [Moreira, 2007] Moreira, R. B. (2007). Análise de técnicas de localização em redes de sensores sem fio. Master's thesis, Universidade Estadual de Campinas . Faculdade de Engenharia Elétrica e de Computação.
- [Nakamura et al., 2005] Nakamura, F. G.; Quintão, F. P.; Menezes, G. C. & Mateus, G. R. (2005). An optimal node scheduling for flat wireless sensor networks. In *4th International Conference on Networking - Lecture Notes in Computer Science*, volume 3420, pp. 104--483.
- [Nesa Sudha & John, 2011] Nesa Sudha, M. & John, S. E. (2011). Energy efficient cost function for wireless sensor networks. *IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI)*, (8):24--28. Published by Foundation of Computer Science.
- [Padberg & Rinaldi, 1991] Padberg, M. & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.*, 33:60--100.
- [Park & Sahni, 2006] Park, J. & Sahni, S. (2006). An online heuristic for maximum lifetime routing in wireless sensor networks. *IEEE Trans. Computers*, pp. 1048--1056.
- [Pottie & Kaiser, 2000] Pottie, G. J. & Kaiser, W. J. (2000). Wireless integrated network sensors. *Commun. ACM*, 43:51--58.
- [Quintão et al., 2010] Quintão, F. P.; da Cunha, A. S.; Mateus, G. R. & Lucena, A. (2010). The k-cardinality tree problem: Reformulations and lagrangian relaxation. *Discrete Appl. Math.*, 158:1305--1314.
- [Raghunathan et al., 2002] Raghunathan, V.; Schurgers, C.; Park, S.; Srivastava, M. & Shaw, B. (2002). Energy-aware wireless microsensor networks. In *IEEE Signal Processing Magazine*, pp. 40--50.
- [Ramalhinho-Lourenço et al., 2000] Ramalhinho-Lourenço, H.; Martin, O. C. & Stützle, T. (2000). Iterated local search. Economics Working Papers 513, Department of Economics and Business, Universitat Pompeu Fabra.

- [Ruiz et al., 2004] Ruiz, L. B.; Correia, L. H. A.; Vieira, L. F. M.; Macedo, D. F.; Nakamura, E. F.; Figueiredo, C. M. S.; Vieira, M. A. M.; Mechelane, E. H.; Camara, D.; Loureiro, A. A.; Nogueira, J. M. S. & da Silva Jr, D. C. (2004). *Architecture for wireless sensor network*, chapter 4, pp. 167--218. SBRC Tutorial, in portuguese edição.
- [Seapahn Megerian & Srivastava, 2005] Seapahn Megerian, F. K. M. P. & Srivastava, M. B. (2005). Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4:84--92.
- [Siqueira et al., 2006] Siqueira, I. G.; Figueiredo, C. M. S.; Loureiro, A. A. F.; Nogueira, J. M. & Ruiz, L. B. (2006). An integrated approach for density control and routing in wireless sensor networks. In *Proceedings of the 20th international conference on Parallel and distributed processing*, IPDPS'06, pp. 104--104, Washington, DC, USA. IEEE Computer Society.
- [So & Ye, 2005] So, A. M.-C. & Ye, Y. (2005). On solving coverage problems in a wireless sensor network using voronoi diagrams. In *WINE*, pp. 584--593.
- [Türkogulları et al., 2010] Türkogulları, Y. B.; Aras, N.; Altinel, I. K. & Ersoy, C. (2010). A column generation based heuristic for sensor placement, activity scheduling and data routing in wireless sensor networks. *European Journal of Operational Research*, 207(2):1014--1026.
- [Türkoğulları et al., 2010] Türkoğulları, Y. B.; Aras, N.; Kuban Alt, I. & Ersoy, C. (2010). An efficient heuristic for placement, scheduling and routing in wireless sensor networks. *Ad Hoc Netw.*, 8:654--667.
- [Wang et al., 2005] Wang, Z. M.; Basagni, S.; Melachrinoudis, E. & Petrioli, C. (2005). Exploiting sink mobility for maximizing sensor networks lifetime. *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, p. 287a.
- [Warrier et al., 2007] Warrier, A.; Park, S.; Min, J. & Rhee, I. (2007). How much energy saving does topology control offer for wireless sensor networks? - a practical study. *Comput. Commun.*, 30:2867--2879.
- [Xue et al., 2005] Xue, Y.; Cui, Y. & Nahrstedt, K. (2005). Maximizing lifetime for data aggregation in wireless sensor networks. *Mob. Netw. Appl.*, 10:853--864.

- [Yahya & Ben-Othman, 2009] Yahya, B. & Ben-Othman, J. (2009). An energy efficient and qos aware multipath routing protocol for wireless sensor networks. In *LCN*, pp. 93--100.
- [Yang et al., 2006a] Yang, B.; Yu, H.; Li, H. & Hou, H. (2006a). A coverage-preserving density control algorithm based-on cooperation in wireless sensor networks. *Wireless Communications, Networking and Mobile Computing (WiCOM)*.
- [Yang et al., 2006b] Yang, S.; Dai, F.; Cardei, M.; Wu, J. & Patterson, F. (2006b). On connected multiple point coverage in wireless sensor networks. *International Journal of Wireless Information Networks*, 13:289--301. 10.1007/s10776-006-0036-z.
- [Yick et al., 2008] Yick, J.; Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292--2330.
- [Younis et al., 2002] Younis, M.; Youssef, M. & Arisha, K. (2002). Energy-aware routing in cluster-based sensor networks. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS '02*, pp. 129--, Washington, DC, USA. IEEE Computer Society.
- [Zeng et al., 2010] Zeng, Y.; Sreenan, C.; Xiong, N.; Yang, L. & Park, J. (2010). Connectivity and coverage maintenance in wireless sensor networks. *The Journal of Supercomputing*, 52:23--46. 10.1007/s11227-009-0268-7.
- [Zhang & Hou, 2005] Zhang, H. & Hou, J. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2).