



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

**HEURÍSTICAS PARA O PROBLEMA DE
COBERTURA EM REDES DE SENSORES SEM
FIO HIERÁRQUICAS COM SORVEDOURO
MÓVEL**



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

ANDRÉ RICARDO MELO ARAÚJO

**HEURÍSTICAS PARA O PROBLEMA DE
COBERTURA EM REDES DE SENSORES SEM
FIO HIERÁRQUICAS COM SORVEDOURO
MÓVEL**

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

ORIENTADOR: FABÍOLA GUERRA NAKAMURA

Manaus - AM

Março de 2013

Agradecimentos

Antes de tudo, agradeço a Deus, por me permitir concluir este trabalho.

Agradeço a minha namorada, Adriana Gomes Peñaranda que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades.

Agradeço também a minha orientadora, Prof. Dr^a. Fabíola Guerra Nakamura, pelo estímulo contínuo e paciência. Obrigada por me mostrar outras maneiras de tentar compreender aquilo que aparentemente não pode ser explicado. A você, o meu carinho, a minha admiração e o meu respeito.

Agradeço a todos os meus amigos, essenciais em todas as etapas da minha vida.

Finalmente, obrigado aos meus pais, Ana e Ricardo, que mais do que me proporcionar uma boa infância e vida acadêmica, formaram os fundamentos do meu caráter e me apontaram uma vida eterna. Obrigada por serem a minha referência de tantas maneiras e estarem sempre presentes na minha vida de uma forma indispensável

Resumo

As Redes de Sensores Sem Fio (RSSFs) são um tipo especial de redes *ad hoc* constituídas por dispositivos capazes de processar, armazenar, sensoriar o ambiente e transmitir dados via interface de comunicação sem fio, denominados nós sensores. Os nós sensores possuem várias limitações, dentre elas, a capacidade de energia devido ao tamanho reduzido. Por isto, muitas pesquisas foram feitas tendo em vista a melhoria no consumo de energia dos nós sensores.

Este trabalho tem como objetivo tratar o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM) em RSSF com nó sorvedouro móvel, que consiste em: dado um conjunto de nós sensores e uma área de monitoramento, desenvolver algoritmos para encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possíveis e encontrar a menor rota para um nó sorvedouro móvel percorrer. O PCAR-SM é uma estratégia utilizada para diminuir o consumo de energia dos nós sensores, a colisão de dados, as interferências e os dados redundantes em redes com alta concentração de nós sensores por área.

A proposta deste trabalho é resolver cada problema separadamente e em conjunto, de modo a avaliar o impacto de cada problema na solução do outro. O Problema de Cobertura foi resolvido com duas metaheurísticas: um Algoritmo Genético (AG) e um algoritmo *Greedy Randomized Adaptive Search Procedure* (GRASP). Neste último foram utilizadas duas representações de solução: (a) representação por sensor, onde cada elemento do vetor de solução representa um nó sensor que estará ligado ou desligado; (b) representação por demanda, onde cada elemento do vetor de solução representa um ponto de demanda no qual indicará qual o nó sensor o cobre. O AG utiliza apenas a representação por demanda.

Os resultados computacionais para o Problema de Cobertura utilizaram o benchmark da *Beasley's OR Library* e foi possível constatar que o GRASP com representação por demanda obteve melhores resultados que o AG e o GRASP com representação por sensor quando o critério de otimização é minimizar a soma total dos custos de cada nó sensor utilizado na solução.

Para o Problema de Agrupamento foi criada uma abordagem de grades virtuais. Nesta abordagem dividimos a área em grades e os grupos são formados por um conjunto de grades adjacentes (no máximo 5 grades) formando um esquema de cruz. O objetivo do problema é minimizar o número de grupos na área.

A partir desta abordagem, pode-se modelar o Problema de Agrupamento como um Problema de Cobertura de Conjuntos (PCC) sem sobreposição (um elemento não pertence a mais de um conjunto), que foi tratada por uma heurística gulosa denominada *Greedy Clustering Algorithm* (GCA). Os grades virtuais provou ser uma boa solução por ser simples para um nó identificar a qual grade ele pertence. Sua simplicidade ainda o torna uma método adequado para uma versão distribuída.

O Problema de Roteamento do nó sorvedouro foi modelado como o Problema do Caixeiro Viajante (PCV), onde o nó sorvedouro móvel parte de um canto da área de monitoramento, percorre a área visitando todos os grupos e retorna ao ponto inicial. Para isto, propomos duas abordagens gulosas baseadas no vizinho mais próximo, o *Routing Greedy Algorithm - Center* (RGA-C) e o *Routing Greedy Algorithm - Border* (RGA-B). A rota do nó sorvedouro também foi resolvida por uma heurística baseada no algoritmo *Centralized Spatial Partitioning* (CSP). Na abordagem CSP, a rota é fixa e lembra o movimento de uma cobra. Os resultados mostram que a rota fixa gera um percurso com tamanho menor em comparação com as heurísticas gulosas para o PCV.

Analisamos, ainda, o PCAR-SM, criando estratégias heurísticas. A união dos Problema de Agrupamento e Roteamento, provou ser mais benéfica em relação ao tamanho da rota do nó sorvedouro, já a união do Problema de Cobertura com o Problema de Agrupamento só mostrou ser benéfica quando o raio de comunicação era aproximadamente 3,9 vezes maior que o raio de sensoriamento.

Nossos resultados, mostram que resolver os problemas em conjunto permite que algumas mudanças nos algoritmos levem a melhores resultados.

Palavras-chave: Redes de sensores sem fio, Problema de Cobertura, Problema de Agrupamento, Problema de Roteamento, Sorvedouro móvel, Redes hierárquicas

Abstract

Wireless Sensor Network (WSN) is a special kind of ad hoc networks composed of devices capable of processing, storing, sensing the environment, and transmitting data via wireless communication interface. The sensor nodes have several limitations, among them the capacity of energy because to the reduced size. For this reason, many searches have been done with a view to improving the energy consumption of sensor nodes.

This work aims to address the Problem of Coverage, Clustering and Routing with Mobile Sink (PCAR-SM, in portuguese Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel) in WSN with mobile sink consisting of: given a set of sensor nodes and a monitoring area, develop algorithms to find the best subset of sensor nodes to cover the monitoring area, group them in a smaller number of clusters and find the shortest route to mobile sink navigate. The PCAR-SM is a strategy used to reduce the energy consumption of sensor nodes, data collisions, interference and redundant data in networks with high concentration of sensor nodes per area.

The purpose of this paper is to solve each problem separately and together, in order to evaluate the impact of each problem on the other. The Coverage Problem has been solved with two metaheuristics: an Genetic Algorithm (GA) and a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm. In the latter we used two representations of solution: (a) representation by sensor, where each element of the solution vector represents a sensor node that must be switched on or off; (b) representation by demand, where each element of the solution vector represents a demand point will indicate which sensor node cover it. The AG uses only the representation by demand.

The computational results for Coverage Problem used the benchmark of Beasley's OR Library and it was possible seen that the GRASP with representation by demand achieved better results than the GA and the GRASP with representation by sensor when the optimization criterion is to minimize the total cost of each sensor node used in the solution.

For Clustering Problem was created approach of virtual grids. In this approach, we divide the area into grids and clusters are formed by a set of adjacent grids (maximum 5 grids in group) forming a cross schematic. The aim of the problem is to minimize the number of clusters in the area.

With this approach, we can model the Clustering Problem as a Set Cover Problem (SCP) without overlapping (an element does not belong to more than one set), which was treated by a greedy heuristic called Greedy Clustering Algorithm (GCA). The virtual grids proved to be a good solution because it is simple to identify a node which grid it belongs. Its simplicity also makes it a appropriate method for a distributed version.

The Routing Problem of sink was modeled as the Travelling Salesman Problem (TSP), where the mobile sink part of a corner of the monitoring area, runs through the area visiting all clusters and returns to the starting point. For this, we propose two greedy approaches based on nearest neighbor, the Routing Greedy Algorithm - Center (RGA-C) and Routing Greedy Algorithm - Border (RGA-B). The route of the sink was also solved by a heuristic based on algorithm Centralized Spatial Partitioning (CSP). In CSP approach, the route is fixed and reminds the movement of a snake. The results show that fixed route produces a path with smaller size compared to the greedy heuristic for TSP.

We analyze also the PCAR-SM, creating heuristic strategies. The union of the Clustering Problem and Routing Problem proved more beneficial in relation to the size of the sink's route. The union of Coverage Problem and Clustering Problem only proved beneficial when the communication radius was about 3,9 times greater than the sensing radius.

Our results show that solve problems together allows some changes in the algorithms will lead to better results.

Keywords: Wireless Sensor Network, Coverage Problem, Clustering Problem, Routing Problem, Mobile sink, Hierarchical networks

Sumário

Agradecimentos	v
Resumo	vii
Abstract	xi
Lista de Abreviaturas	xvii
Lista de Figuras	xix
Lista de Tabelas	xxi
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo	3
1.3 Contribuição	4
1.4 Estrutura do Documento	5
2 Fundamentos	7
2.1 Características das RSSFs	7
2.1.1 Redes de Sensores sem Fios Planas e Hierárquicas	7
2.1.2 Sorvedouros Móveis e Fixos	9
2.2 Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel	9
2.2.1 Definição do Problema de Cobertura em Rede de Sensores Sem Fio	10

2.2.2	Definição do Problema de Agrupamento em Rede de Sensores Sem Fio	10
2.2.3	Definição do Problema de Roteamento em Rede de Sensores Sem Fio	11
2.2.4	Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel	12
2.3	Trabalhos Relacionados	12
2.4	Considerações Finais	15
3	Problema de Cobertura	17
3.1	Definição do Problema de Cobertura em Redes de Sensores sem Fio	17
3.1.1	Modelo Matemático	19
3.2	Representações da Solução	20
3.3	Abordagens de resolução para o Problema de Cobertura	21
3.3.1	Algoritmo Genético	21
3.3.2	<i>Greedy Randomized Adaptive Search Procedure</i>	24
3.4	Testes e Resultados Obtidos	27
3.4.1	Parâmetros e Métricas	27
3.4.2	Bateria de Testes do Algoritmo Genético	28
3.4.3	Bateria de Testes do GRASP	29
3.4.4	Análise dos Testes	30
3.5	Considerações Finais	31
4	Problema de Agrupamento	33
4.1	Definição do Problema de Agrupamento em Redes de Sensores sem Fio	33
4.1.1	Abordagem de Grades Virtuais para o Agrupamento	34
4.1.2	Modelo Matemático	36
4.2	Abordagens para o Problema de Agrupamento	38
4.2.1	<i>Greedy Clustering Algorithm</i>	39
4.3	Resultados Computacionais	40
4.3.1	Parâmetros e Métricas	41
4.3.2	Resultados	41

4.4	Considerações Finais	42
5	Problema de Roteamento	45
5.1	Definição do Problema de Roteamento em Rede de Sensores Sem Fio com Sorvedouro Móvel	45
5.1.1	Modelo Matemático do Problema do Caixeiro Viajante	47
5.2	Abordagens para o Problema de Roteamento	48
5.2.1	Abordagens Gulosas	48
5.2.2	<i>Centralized Spatial Partitioning</i>	50
5.2.3	<i>Centralized Spatial Partitioning Grid</i>	51
5.3	Resultados Computacionais	51
5.3.1	Métricas	52
5.3.2	Resultados	52
5.4	Considerações Finais	54
6	Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel	57
6.1	Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel em Rede de Sensores Sem Fio (RSSF)	58
6.2	Problema de Cobertura + Problema de Agrupamento	58
6.2.1	Resultados - Problema de Cobertura + Problema de Agrupamento	59
6.3	Problema de Agrupamento + Problema de Roteamento	64
6.3.1	<i>Greedy Clustering Algorithm - Border</i>	64
6.3.2	<i>Centralized Spatial Partitioning Grid - Border</i>	65
6.3.3	Resultados - Problema de Agrupamento + Problema de Roteamento	66
6.4	Considerações Finais	69
7	Considerações Finais	71
7.1	Conclusões	71
7.2	Trabalhos Publicados	73
7.3	Trabalhos Futuros	74

Lista de Abreviaturas

AG Algoritmo Genético

CSP *Centralized Spatial Partitioning*

CSPG *Centralized Spatial Partitioning Grid*

CSPG-B *Centralized Spatial Partitioning Grid - Border*

GCA *Greedy Clustering Algorithm*

GCA-B *Greedy Clustering Algorithm - Border*

GRASP *Greedy Randomized Adaptive Search Procedure*

LRC Lista Restrita de Candidatos

PCAR-SM Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel

PCC Problema de Cobertura de Conjuntos

PCV Problema do Caixeiro Viajante

PLIM Programação Linear Inteira Mista

RGAB *Routing Greedy Algorithm - Border*

RGAC *Routing Greedy Algorithm - Center*

RSSF Rede de Sensores Sem Fio

Lista de Figuras

1.1	Problema de Cobertura em Redes de Sensores Sem Fio.	2
1.2	PCAR-SM: sensores agrupados com uma rota tracejada para o nó sorvedouro.	4
2.1	Redes Planas x Redes Hierárquicas.	8
2.2	Área de demanda discretizada em pontos, nós sensores e seus relacionamentos.	10
2.3	Rede com agrupamentos com um nó líder.	11
2.4	Rota para um nó sorvedouro percorrer no Problema de Roteamento. . .	11
3.1	Grafo bipartido com o conjunto universo e os n conjuntos.	18
3.2	Área de demanda discretizada em pontos, nós sensores e seus relacionamentos.	19
3.3	Grafo bipartido para o Problema de Cobertura em RSSF.	19
3.4	Representações da solução: (Sensor \times Demanda).	21
3.5	Processo Evolutivo do Algoritmo Genético.	22
3.6	Cruzamento para o Algoritmo Genético.	23
3.7	Mutação para o Algoritmo Genético.	24
4.1	Rede Hierárquica.	34
4.2	Área de monitoramento dividida em grades.	35
4.3	Grade virtual determinado pelo raio de comunicação definido no algoritmo GAF.	35
4.4	Grupo formado por cinco grades.	36
4.5	Rede hierárquica composta por grades e grupos.	36

4.6	Modelo de grupo no esquema de Hexágono.	36
4.7	Todos os 16 possíveis grupos.	37
4.8	Possíveis grupos: Cantos 4 grupos e Laterais 8 grupos.	37
4.9	Passo a passo do <i>Greedy Clustering Algorithm</i>	40
4.10	Grade Irregular	41
5.1	Rota do caixeiro viajante.	46
5.2	Rota para um nó sorvedouro percorrer no Problema de Roteamento. . .	46
5.3	Sub-rotas no Problema do Caixeiro Viajante.	47
5.4	Modelo dos pontos utilizados para as abordagens gulosas.	49
5.5	Rota gerada no <i>Centralized Spatial Partitioning</i>	50
5.6	Rota gerada no <i>Centralized Spatial Partitioning Grid</i>	51
5.7	Raio de Comunicação \times Tamanho da Rota - Algoritmo CSPG.	52
5.8	Tamanho da Área \times Tamanho da Rota - Algoritmos RGA-B e RGA-C. . .	53
5.9	Tamanho da Área \times Tamanho da Rota - Algoritmos CSPG e CSP. . .	54
6.1	PCAR-SM: sensores agrupados com uma rota tracejada para o nó sorvedouro.	59
6.2	Passo a passo do <i>Greedy Clustering Algorithm - Border</i>	65
6.3	Rota gerada no <i>Centralized Spatial Partitioning Grid - Border</i>	66
6.4	Tamanho da Área \times Tamanho da Rota - Algoritmos CSPG-B, CSPG e CSP.	68

Lista de Tabelas

3.1	Detalhes dos conjunto de problemas do Benchmark.	27
3.2	Resultados para o AG em cada problema "SCP"do benchmark.	29
3.3	Resultados para os GRASPs em cada problema "SCP"do benchmark.	30
4.1	Resultados do GCA comparado com o Modelo Matemático	42
6.1	Resultados para o GRASP+GCA, onde $R_c \cong R_s$	61
6.2	Descrição dos Raios de Comunicação e Sensoriamento ($R_c \cong 3,9R_s$).	63
6.3	Resultados para o GCA+GRASP, onde $R_c \cong 3,9R_s$	63
6.4	Resultados para o GCA-B comparado com o GCA e o Modelo Matemático.	67
6.5	Diferenças relativas entre as abordagens: (GCA \times GCA-B) e (CSPG \times CSPG-B).	69

Capítulo 1

Introdução

1.1 Motivação

As Redes de Sensores Sem Fio (RSSFs) são um tipo especial de redes *ad hoc* constituídas por dispositivos capazes de processar, armazenar, sensoriar o ambiente e transmitir dados via interface de comunicação sem fio, denominados nós sensores. Nestas redes, cada nó sensor é equipado com uma variedade de nós sensores, tais como acústico, sísmico, infravermelho, vídeo-câmera, calor, temperatura e pressão. Por esta razão este tipo de rede pode ser encontrado em diversas áreas, como no ambiente industrial fazendo o controle do processo de manufatura, ou em centros comerciais fazendo a segurança, ou, ainda, fazendo o rastreamento de animais numa floresta [Loureiro et al., 2002].

As RSSFs são diferentes das redes tradicionais em pontos como limitações de processamento, memória e energia, além de serem altamente sujeita a falhas. Loureiro et al. [2002] afirmam que em muitas aplicações, os nós sensores serão colocados em áreas remotas, o que não permitirá facilmente o acesso a esses elementos para manutenção. Neste cenário, o tempo de vida de um nó sensor depende da quantidade de energia disponível. Para isso, um controle do consumo de energia é necessário.

Muitas pesquisas foram feitas tendo em vista a melhoria no consumo de energia. Dentre estas pesquisas temos o Problema de Cobertura (PC) em RSSF, que busca encontrar um subconjunto de nós dispostos na área de monitoramento

para fazer parte da rede, cobrindo toda a área de monitoramento necessitada pela aplicação, conforme visto na Figura 1.1. Uma estratégia para tratar o Problema de Cobertura é quantificar a área de monitoramento em pontos de demanda. São estes pontos que devem ser cobertos pelos nós sensores.

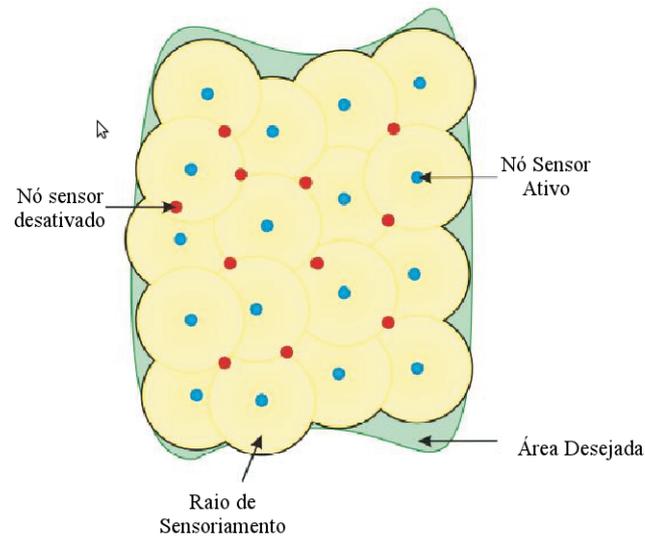


Figura 1.1. Problema de Cobertura em Redes de Sensores Sem Fio.

Araújo et al. [2011] trabalharam com um modelo de Programação Linear Inteira Mista (PLIM) para o Problema de Cobertura em Rede de Sensores sem Fio Planas Voltado ao Rastreamento de Animais. O objetivo da aplicação de rastreamento é manter uma rede de nós sensores fixa na floresta para coletar dados de microclima da região e ao mesmo tempo detectar macacos da espécie Sauim-de-Coleira que passem pela área. Para que isto seja possível, os macacos possuiriam nós sensores implantados em seu corpo, que serão responsáveis por enviar sinais periodicamente e quando detectados pelos nós fixos poderão auxiliar na localização e identificação do animal. O modelo matemático proposto no trabalho deve determinar qual a melhor localização dos nós sensores fixos para que a aplicação possa ser viabilizada.

Além da restrição de energia, outros problemas comuns nas RSSFs são a colisão de dados e a interferência, que ocorrem principalmente, em redes com alta concentração de nós sensores por área, as chamadas redes densas. A alta densidade

de nós além de influenciar na colisão de pacotes e interferências, gera dados redundantes e aumenta o consumo de energia da rede. Boa parte do consumo de energia é associada à quantidade de dados transmitidos na rede, pois a comunicação é a atividade que demanda uma grande quantidade de energia [Villas et al., 2011].

Uma das possíveis estratégias é trabalhar com Redes Sem Fio Hierárquicas. Neste tipo de rede os nós são classificados hierarquicamente, um subconjunto destes nós formam um grupo com um líder, e os nós do grupo se comunicam somente com este líder, diminuindo o tráfego de dados e, conseqüentemente, a colisão de dados, as interferências, os dados redundantes e o consumo de energia. O problema de trabalhar com redes hierárquicas é definir como agrupar os nós, pois devem definir a quantidade de grupos, a quantidade de nós por grupo e quantos níveis de hierarquia.

Outra questão a ser definida em RSSF, é como enviar os dados coletados para fora da rede. O nó sensor responsável por fazer esta transferência é chamado de nó sorvedouro. Ele pode ser fixo ou móvel, o nó sorvedouro fixo fica em uma posição da área esperando que os dados coletados na rede sejam transmitidos de nó em nó até ele. O nó sorvedouro móvel caminha pela área coletando estes dados.

Para utilizar o nó sorvedouro móvel é necessário definir qual a rota que ele deve percorrer de modo que consiga recolher o máximo de dados possível. Este problema da definição da rota pode ser modelado como Problema do Caixeiro Viajante (PCV), onde o nó sai da base, percorre a área e volta para a base para deixar os dados coletados. Para diminuir a latência deve-se gerar a menor rota possível.

1.2 Objetivo

O objetivo deste trabalho é, dado um conjunto de nós sensores e uma área de monitoramento, desenvolver algoritmos para encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possível e encontrar a menor rota para um nó sorvedouro móvel percorrer. Iremos chamar esse problema de Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM). A Figura 1.2 mostra um exemplo de solução para o PCAR-SM.

- Representações da Solução: Sensor \times Demanda.
- Heurísticas:
 - * *Greedy Randomized Adaptive Search Procedure* (GRASP).
 - * Algoritmo Genético (AG).
- Problema de Agrupamento:
 - Modelagem de Grades Virtuais.
 - Heurísticas:
 - * *Greedy Clustering Algorithm* (GCA).
- Problema de Roteamento Com Sorvedouro Móvel:
 - Abordagens baseadas no Vizinho mais próximo:
 - * *Routing Greedy Algorithm - Center* (RGA-C).
 - * *Routing Greedy Algorithm - Border* (RGA-B).
 - Abordagens baseadas no *Centralized Spatial Partitioning* (CSP):
 - * *Centralized Spatial Partitioning Grid* (CSPG).
- Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel:
 - Problema de Cobertura + Problema de Agrupamento.
 - Problema de Agrupamento + Problema de Roteamento:
 - * *Greedy Clustering Algorithm - Border* (GCA-B).
 - * *Centralized Spatial Partitioning Grid - Border* (CSPG-B).

Todas as meta-heurísticas são centralizadas e consideramos que é conhecido a posição de todos os nós sensores.

1.4 Estrutura do Documento

A divisão dos capítulos e um resumo de cada um é apresentado abaixo:

- **Capítulo 2: Fundamentos.** Neste capítulo são apresentados os fundamentos necessários sobre RSSF para o entendimento do problema, definições dos problemas do PCAR-SM e os trabalhos relacionados.
- **Capítulo 3: Problema de Cobertura.** Este capítulo define o Problema de Cobertura, assim como as possíveis representações de solução (Sensor \times Demanda). Apresentando duas heurísticas para tratar o problema: o Algoritmo Genético (AG) e o *Greedy Randomized Adaptive Search Procedure* (GRASP).
- **Capítulo 4: Problema de Agrupamento.** Neste capítulo apresentamos o Problema de Agrupamento junto com a abordagem de grades virtuais utilizada como base para o modelo matemático e para heurística *Greedy Clustering Algorithm* (GCA).
- **Capítulo 5: Problema de Roteamento.** Este capítulo mostramos o problema de roteamento e as estratégias utilizadas para o nó sorvedouro móvel percorrer a área. Implementamos duas heurísticas baseadas no Algoritmo do Vizinheiro Mais Próximo, o *Routing Greedy Algorithm - Center* (RGA-C) e o *Routing Greedy Algorithm - Border* (RGA-B). Implementamos, também, uma heurística baseada no Algoritmo *Centralized Spatial Partitioning* (CSP), o *Centralized Spatial Partitioning Grid* (CSPG).
- **Capítulo 6: Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel.** Neste capítulo são apresentadas as estratégias de união dos Problemas de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel que compõem o PCAR-SM. A união foi feita em duas partes primeiro avaliando os Problemas de Cobertura e Agrupamento, e em seguida os problemas de Agrupamento e Roteamento.
- **Capítulo 7: Considerações Finais.** Este capítulo apresenta as considerações finais, a lista de trabalhos publicados e trabalhos futuros.

Capítulo 2

Fundamentos

2.1 Características das RSSFs

As RSSF podem ser classificadas em função de diferentes características, como composição, organização, distribuição, etc. Por exemplo, se uma rede possui diferentes tipos de nós sensores, esta rede é chamada de rede com composição heterogênea. Já se esta rede possuir agrupamentos ele se torna também uma rede com organização hierárquica [Loureiro et al., 2002]. Dentre as diversas características das RSSF, as que possuem relevância para o PCAR-SM são:

- Organização:
 - Plana: quando a rede não possui agrupamentos.
 - Hierárquica: quando a rede possui agrupamentos.
- Sorvedouro:
 - Móvel: quando o nó sorvedouro caminha pela área.
 - Fixo: quando o nó sorvedouro fica em um ponto fixo na área.

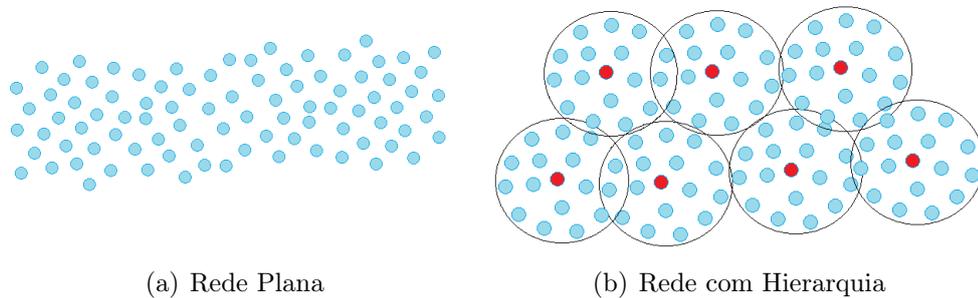
2.1.1 Redes de Sensores sem Fios Planas e Hierárquicas

Dois grandes problemas encontrados nas RSSF são a colisão de dados e a interferência, que ocorrem principalmente, em redes com alta concentração de nós sensores

por área, as chamadas redes densas. A alta densidade de nós além de influenciar na colisão de pacotes e interferências, gera dados redundantes e aumenta o consumo de energia da rede. Boa parte do consumo de energia é associado à quantidade de dados transmitidos na rede, pois a comunicação é a atividade que demanda uma grande quantidade de energia [Villas et al., 2011].

Uma das possíveis estratégias para diminuir a colisão de dados e a interferência é trabalhar com redes sem fio hierárquicas. Nas redes hierárquicas os nós são classificados hierarquicamente, um subconjunto destes nós formariam um grupo com um líder, e os nós do grupo se comunicariam somente com este líder (*Cluster Head*). Nas redes planas os nós não possuem hierarquia [Aioffi & Mateus, 2007]. A Figura 2.1 mostra exemplos de redes Plana e Hierárquica.

O líder é o nó responsável pelo gerenciamento do grupo e por transmitir os dados coletados para o nó sorvedouro [Turgut et al., 2002]. Com essa configuração a rede elimina a necessidade de rotas dedicadas para o envio de dados dos nós sensores ativos até o nó sorvedouro, pois os líderes de cada grupo são responsáveis por esta comunicação.



(a) Rede Plana

(b) Rede com Hierarquia

Figura 2.1. Redes Planas x Redes Hierárquicas.

As redes hierárquicas possuem vantagem sobre as planas, pois elas diminuem o tráfego de dados e, conseqüentemente, a colisão de dados, as interferências, os dados redundantes e o consumo de energia. A utilização de estratégias de fusão de dados ajuda diminuir, ainda mais, o tráfego de dados. O problema de trabalhar com redes hierárquicas é definir como agrupar os nós, pois devem definir a quantidade de grupos, a quantidade de nós por grupo e quantos níveis de hierarquia.

2.1.2 Sorvedouros Móveis e Fixos

Aliado ao Problema de Cobertura e Agrupamento é possível integrar outras características, como por exemplo, a utilização de nós sorvedouros móveis. Com esta característica pode-se utilizar a comunicação *single-hop* onde os nós sensores não roteiam mensagens de outros nós sensores, ou seja, se comunicam diretamente com o nó sorvedouro, ou utilizar a comunicação *multi-hop* onde os nós sensores têm capacidade de roteamento de mensagens.

No caso de nós sorvedouros fixos os dados devem ser roteados até o nó sorvedouro. Com isso os nós sensores mais próximos a ele tendem a gastar mais energia transmitindo dados. Com o intuito de distribuir esse consumo de energia, Andrade et al. [2009] colocam o nó sorvedouro no centro da área de demanda, desta forma o roteamento de dados é mais distribuído e as rotas são menores.

Aioffi & Mateus [2007] utilizam comunicação *multi-hop*, porém com um número limitado de saltos. Eles usam um nó sorvedouro móvel que é responsável por percorrer a área de monitoramento e coletar os dados dos nós sensores. O consumo de energia dos nós sensores é diminuído, pois quando o nó sorvedouro percorre a área de monitoramento a transmissão dos dados pela rede é reduzida.

2.2 Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel

Nesta Seção definimos o PCAR-SM, primeiramente, cada problema é definido separadamente. Após isto, definimos os problemas em conjunto. Neste trabalho, é considerado que temos o conhecimento de toda área que será monitorada, inclusive o posicionamento de todos os nós sensores. O posicionamento dos nós sensores pode ser obtido através de algoritmos de localização como *Ad Hoc Positioning System* [Niculescu & Nath, 2001] ou DV-Loc [Boukerche et al., 2009].

2.2.1 Definição do Problema de Cobertura em Rede de Sensores Sem Fio

O Problema de Cobertura em RSSF é derivado do Problema de Cobertura de Conjuntos (PCC) [Huang & Tseng, 2003]. Assim, o Problema de Cobertura é: *Dados os pontos de demanda, um conjunto de nós sensores e suas relações de cobertura, encontrar um subconjunto de nós sensores que consigam cobrir todos os pontos de demanda.* A Figura 2.2 exemplifica o problema.

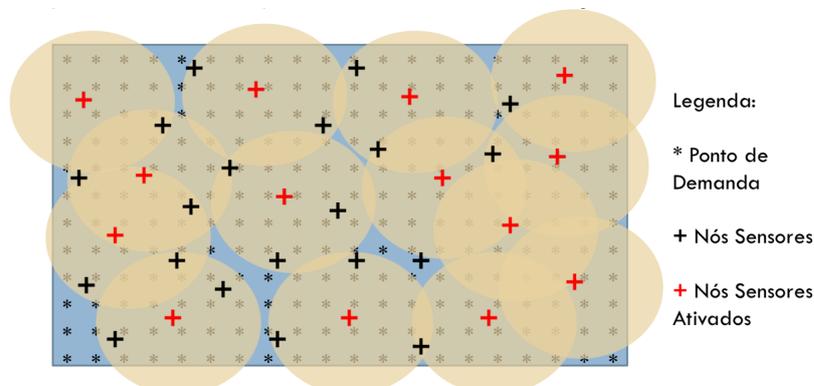


Figura 2.2. Área de demanda discretizada em pontos, nós sensores e seus relacionamentos.

2.2.2 Definição do Problema de Agrupamento em Rede de Sensores Sem Fio

O Problema de Agrupamento em RSSF pode ser descrito como encontrar grupos de nós sensores que estão, relativamente, próximos e conseguem se comunicar com um nó central (não necessariamente no centro do grupo) [Aioffi et al., 2011]. Cada grupo é chamado, também, de *cluster*. Cada grupo tem um líder, o *cluster head*. A Figura 2.3 exemplifica o problema.

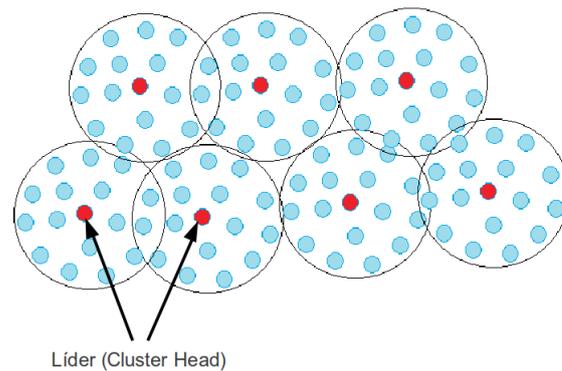


Figura 2.3. Rede com agrupamentos com um nó líder.

2.2.3 Definição do Problema de Roteamento em Rede de Sensores Sem Fio

O Problema de Roteamento apresentado neste trabalho consiste em encontrar uma rota para que um nó sorvedouro móvel percorra a área com o objetivo de coletar os dados dos nós sensores. Este problema pode ser modelado como o Problema do Caixeiro Viajante (PCV) [Dantzig et al., 1954].

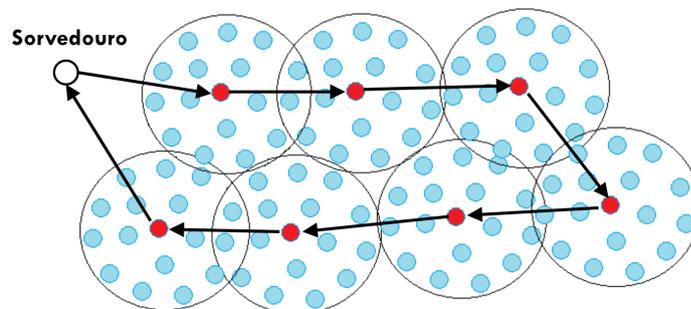


Figura 2.4. Rota para um nó sorvedouro percorrer no Problema de Roteamento.

O Problema de Roteamento (representado na Figura 2.4) consiste em se definir uma rota para o nó sorvedouro percorrer que passe por todos os grupos e retorne ao ponto inicial. Esta rota deverá possuir o menor custo [Cormen et al., 2001]. Este custo pode ser a distância, custo financeiro, ou, ainda, o custo de energia.

2.2.4 Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel

O PCAR-SM pode ser definido formalmente como: *Dados um conjunto de nós sensores e uma área de monitoramento desenvolver algoritmos para encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possível e encontrar a menor rota para um nó sorvedouro móvel percorrer.*

2.3 Trabalhos Relacionados

O Problema de Cobertura de Conjuntos que deriva o Problema de Cobertura em RSSF tem sido estudado nas mais diversas áreas da computação.

Bagnall et al. [2001] formulam o *Next Release Problem* (NRP, problema da próxima release). Este problema envolve a seleção de um conjunto de requisitos de clientes que deverão ser entregues na próxima release. Bagnall et al. [2001] usam varias técnicas de otimização, incluindo *simulated annealing* e algoritmo guloso, para um conjunto de dados, características de modelos e a relação entre eles. Guanzhou Lu e Bahsoon [2010] resolvem o problema usando o algoritmo Hill Climbing.

No Problema da Seleção de Dados de Testes, o objetivo é selecionar um conjunto de casos de teste que cobrem as características do projeto [Harman, 2007]. Dias-Neto et al. [2011] descrevem a técnica de testes de seleção como um problema de otimização, visando maximizar a cobertura de características de projetos de software e requisitos de teste e cobertura das competências necessárias para utilizar as técnicas selecionadas e minimizar o esforço para construir os modelos de teste. Dias-Neto et al. [2011] modelam o Problema de Seleção de Dados de Teste como o Problema do Menor Conjunto Dominante (uma variante do PCC), eles elaboram um algoritmo guloso usando uma estratégia de busca local, esta última baseada na heurística proposta por Rodrigues et al. [2008] e que utiliza a vizinhança *2-opt* para a geração de soluções de compromisso, não-dominadas entre si, para os objetivos envolvidos.

Araújo et al. [2011] propõem uma heurística para resolver o Problema de

Cobertura em RSSF Voltado para Rastreamento de Animais. Nesta heurística, eles têm duas áreas de demanda sobrepostas para cobrir, um para cobrir com um nó sensor e o outro com três nós sensores. A heurística é baseada no algoritmo guloso, mas eles dividem as áreas em pedaços e aplicam seu algoritmo em paralelo. Depois, eles fazem uma busca local, que não teve melhoria significativa na solução.

Há muitos outros trabalhos que tentam resolver o problema de cobertura, alguns deles trabalham em conjunto em outros problemas como o controle de densidade e conectividade [Andrade & Mateus, 2008; Nakamura et al., 2005].

Assim como o Problema de Cobertura, os Problemas de Agrupamento e Roteamento em RSSF com nós sensores móveis tem sido extensivamente pesquisado. O agrupamento é proposto como uma técnica para usar melhor os recursos escassos da rede e reduzindo o consumo de energia e latência [Heinzelman et al., 2000]. Um ou mais nós sorvedouros móveis contribuem para aumentar o tempo de vida do sistema. Um agente móvel que se aproxima dos nós pode ajudar a poupar energia, pois os dados coletados são transmitidos em menos saltos, reduzindo assim o número de pacotes transmitidos. A energia extra gasta para a operação e movimentação do nó sorvedouro não afeta a vida útil da rede pois o nó sorvedouro móvel é considerado um fator externo a rede, na verdade, pode ser um veículo dirigido por um homem ou um robô não tripulado que periodicamente retorna a um centro de apoio, a fim de recarregar-se [Chatzigiannakis et al., 2008].

Chen et al. [2007] apresentam o *Hybrid-Structure Routing Protocol* (HSRP, protocolo de roteamento com estrutura híbrida) que combina os benefícios de estruturas baseadas em grades e baseadas em grupos. Estrutura baseada em grupos é projetado para resolver o problema de seleção de líderes. Eles dividem os líderes na estrutura de grades, cada líder transmite uma simples mensagem *HELLO* para cada nó sensor a sua volta. O nó sensor salva todas as informações para cada mensagem que recebe (incluindo a identificação do líder), após isto, ele retransmite a mensagem incrementando o contador de saltos da mensagem. Se um nó sensor recebe outra mensagem, ele compara o contador de saltos e escolhe o menor. Depois de cada nó sensor ter sido classificado em um grupo virtual, o procedimento termina e os grupos estão formados.

Aioffi et al. [2007] propõem a *Single Hop Strategy* (SHS, estratégia de salto único) para estabelecer um modelo para disseminação, recepção e transmissão de

dados em RSSF. No SHS, o nó sorvedouro se comunica diretamente com cada nó sensor fixo. Um único nó sorvedouro móvel é usado para coletar os dados e presume-se que o nó sorvedouro sabe todas as localizações dos nós sensores. Só quando o nó sorvedouro chega ao centro de um determinado grupo, a comunicação entre o nó sorvedouro e todos os nós abrangidos pelo grupo é realizado. Aioffi et al. [2011] usam algoritmos de otimização para definir políticas de controle de densidade ótima, grupos de nós sensores, e as rotas do nó sorvedouro para a coleta de dados. Eles lidam com a natureza multiobjetivo do problema, explicitamente, minimizar a latência de entrega da mensagem e a imposição de restrições de topologia que ajudam a reduzir o consumo de energia. Os autores propõem diversas metaheurísticas para resolver o problema integrado e incorporar as suas soluções em um ambiente de simulação dinâmica.

Chatzigiannakis et al. [2007] investigam o impacto de múltiplos nós sorvedouros móveis na coleta de dados em redes de nós sensores sem fio. Para melhorar o desempenho, seu protocolo centra-se na minimização de sobreposições das rotas dos nós sorvedouros e no equilíbrio da carga de serviço entre os nós sorvedouros. Eles propõem três protocolos diferentes: a) uma forma centralizada, que equaliza cobertura espacial; b) um protocolo distribuído baseado esquia mútuo dos nós sorvedouros; c) um protocolo de agrupamento que distribui áreas de grupos para equilibrar a carga por nó sorvedouro. O protocolo centralizado, chamado de *Centralized Spatial Partitioning* (CSP, particionamento espacial centralizado), para coletar dados dos nós sensores, o nó sorvedouro móvel repetidamente implementa um ‘movimento de cobra’ sobre a área.

Nakayama et al. [2007] propõem abordagens que, como CSP, sofrem com a falta de integração entre a resolução do agrupamento e os problemas de roteamento. Depois de agrupar os nós sensores usam um esquema chamado *K-means*, uma rota que abrange todos os líderes dos grupos é encontrada por uma heurística de busca local para o PCV.

Em nosso trabalho, a integração entre os algoritmos de agrupamento e roteamento gera pequenas mudanças nos algoritmos e leva a melhores resultados, em relação ao comprimento da rota.

2.4 Considerações Finais

Este capítulo apresenta as características da Rede de Sensores Sem Fio e quais destas são mais relevantes para o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel. O capítulo traz as definições formais de cada problema e os trabalhos relacionados. O capítulo seguinte apresenta o Problema de Cobertura e as soluções propostas para resolvê-lo.

Capítulo 3

Problema de Cobertura

Neste capítulo são mostrados os algoritmos para resolver o problema de cobertura do PCAR-SM. O objetivo do problema de cobertura é encontrar um subconjunto de nós sensores que consiga cobrir a área de monitoramento.

Na Seção 3.1 definimos o problema de cobertura em redes de nós sensores sem fio. Na Seção 3.2 são apresentadas as representações da solução que foram consideradas para este trabalho. Na Seção 3.3.1, é mostrado o Algoritmo Genético (AG). Na Seção 3.3.2 é apresentado o algoritmo do *Greedy Randomized Adaptive Search Procedure* (GRASP) e suas características. Por último, a Seção 3.4 apresenta os resultados comparados com um *benchmark* amplamente aceito na comunidade.

3.1 Definição do Problema de Cobertura em Redes de Sensores sem Fio

O Problema de Cobertura em Rede de Sensores Sem Fio (RSSF) é derivado do Problema de Cobertura de Conjuntos (PCC) [Johnson, 1973]. O PCC é o melhor modelo para muitas aplicações importantes, principalmente, no problema de seleção de recursos [Cormen et al., 2001; Caprara et al., 1998]. Seu problema de decisão correspondente generaliza o Problema de Cobertura de Vértices NP-Completo e também é NP-Difícil [Garey & Johnson, 1979]. Ele é um dos 21 problemas que Karp demonstrou ser NP-Completo em 1972 [Karp, 1972].

O Problema de Cobertura de Vértices pode ser descrito como encontrar um conjunto de vértices que cubram todas as arestas de um grafo [Cormen et al., 2001]. Dessa forma, o PCC pode ser descrito da seguinte forma:

Dado um conjunto de elementos $U = \{1, 2, \dots, m\}$ (chamado universo) e n conjuntos cuja a união contém este universo, o objetivo do PCC é encontrar o menor número de conjuntos cuja a união contém todos os elementos do universo.

Por exemplo, dados o seguinte universo de elementos $U = \{1, 2, 3, 4, 5\}$ e os conjuntos $S = \{\{1, 2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$. Claramente, a união de todos os conjuntos de S contém os elementos em U . Embora, nós podemos cobrir todos os elementos com um número menor de conjuntos: $\{\{1, 2, 3\}, \{4, 5\}\}$.

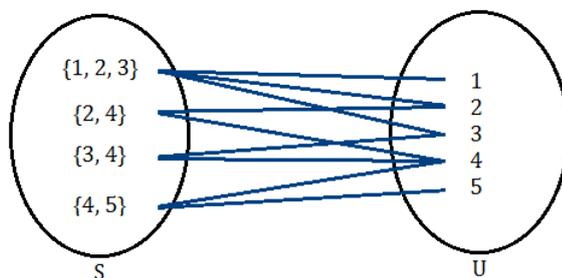


Figura 3.1. Grafo bipartido com o conjunto universo e os n conjuntos.

O PCC também é visto como um grafo bipartido, no qual um dos lados tem todos os elementos de S e no outro lado os elementos de U , como mostrado na Figura 3.1. O relacionamento entre os dois lados é a função de cobertura de um subconjunto de S com um elemento de U . Isto é, se um conjunto de S cobre um elemento de U , então eles são conectados.

Para o Problema de Cobertura em RSSF, o universo de elementos representa a área de monitoramento, discretizada em pontos de demanda, que deverá ser coberta pelos nós sensores. Os conjuntos de pontos de demanda cobertos por cada nó sensor representam os conjuntos do Problema de Cobertura. A Figura 3.2 mostra um exemplo da área de demanda discretizada em pontos, os nós sensores e a relação de cobertura entre eles.

Da mesma forma que o Problema de Cobertura, o grafo bipartido para o Problema de Cobertura em RSSF é, facilmente, construído. A Figura 3.3 demonstra o grafo.

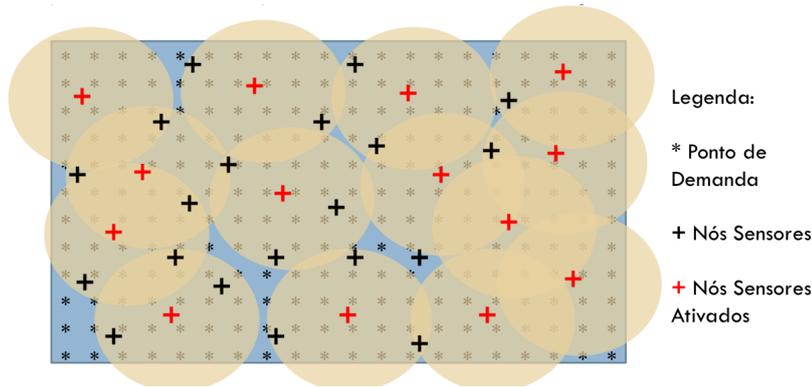


Figura 3.2. Área de demanda discretizada em pontos, nós sensores e seus relacionamentos.

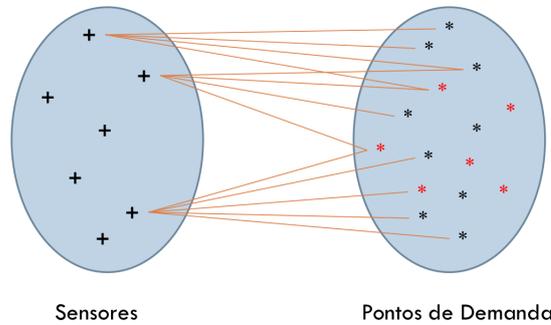


Figura 3.3. Grafo bipartido para o Problema de Cobertura em RSSF.

O Problema de Cobertura em RSSF pode ser definido como: *Dado os pontos de demanda, um conjunto de nós sensores e suas relações de cobertura, encontrar um subconjunto de nós sensores que consiga cobrir todos os pontos de demanda.*

Neste trabalho, tratamos o Problema de Cobertura em RSSF estático. Ou seja, para um determinado período de tempo analisamos qual a melhor configuração para a RSSF. Esta configuração é o subconjunto de nós sensores que ficarão ativos que gaste a menor quantidade de energia.

3.1.1 Modelo Matemático

O PCC pode ser formalmente definido da seguinte forma: *Seja $A = (a_{ij})$ uma $m \times n$ matriz binária, $c = (c_j)$ um vetor n -dimensional de inteiros, $M = \{1, \dots, m\}$ e $N = \{1, \dots, n\}$. Se $a_{ij} = 1$, nós falamos que a coluna ($j \in N$) cobre a linha*

($i \in M$). O valor c_j ($j \in N$) representa o custo da coluna j , e assumido sem perda de generalidade que $c_j > 0$ para $j \in N$. O objetivo do PCC é encontrar subconjunto $S \subseteq N$ de colunas com o menor custo, tal que cada linha $i \in M$ é coberta por no mínimo uma coluna $j \in S$. O modelo de programação linear inteira para o PCC é:

$$\text{Minimizar } z = \sum_{j \in N} c_j x_j \quad (3.1)$$

sujeito a:

$$\sum_{j \in N} a_{ij} x_j \geq 1 \quad i \in M \quad (3.2)$$

$$x_j \in \{0, 1\} \quad j \in N \quad (3.3)$$

onde $x_j = 1$ se $j \in S$, $x_j = 0$, caso contrário. Este modelo foi tirado de Caprara et al. [1998].

A função objetivo do modelo, representada na função 3.1 é a soma dos custos para cada coluna que será escolhida. As restrições 3.2 garantem que cada linha será coberta por pelo menos uma coluna. As restrições 3.3 informam que a variável x , que conterà a solução, será inteira com os valores de 0 ou 1.

Para o problema em RSSF, os nós sensores representam as colunas do PCC, os pontos de demanda representam as linhas e o custo de cada coluna indica o custo dos nós sensores. Este custo pode se referir ao custo financeiro, consumo de energia, entre outros.

3.2 Representações da Solução

Neste trabalho, foram implementadas duas representações de soluções para o Problema de Cobertura em RSSF: (a) Representação por Sensor: a solução é um vetor binário que indica para cada nó sensor se ele ficará ativo ou não, (b) Representação por Demanda: a solução é um vetor de inteiros que indica para cada ponto de demanda, qual é o nó sensor que o cobre. A Figura 3.4, mostra um exemplo para as duas soluções.

O objetivo de utilizar dois modelos de representação de solução é avaliar a qualidade e o tempo de execução em cada uma delas.

1	2	3	4	5	6	7	...	n	Sensores
0	1	0	1	1	0	1	...	1	

1	2	3	4	5	6	7	...	m	Pontos de
10	5	3	1	9	10	9	...	3	Demanda

(a) Representação por Sensor (b) Representação por Demanda

Figura 3.4. Representações da solução: (Sensor \times Demanda).

3.3 Abordagens de resolução para o Problema de Cobertura

Nós propomos duas abordagens para o problema de cobertura, descritas a seguir:

1. Algoritmo Genético (AG) - baseado na teoria de Darwin, onde é criada uma população de indivíduos e a cada iteração suas características são passadas de pais para filhos ou os indivíduos evoluem [Holland, 1975].
2. *Greedy Randomized Adaptive Search Procedure* (GRASP) - seu funcionamento consiste basicamente em aplicar, repetidamente, busca local a diferentes soluções iniciais [Feo & Resende, 1989].

3.3.1 Algoritmo Genético

Dentre as metaheurísticas existentes, Algoritmo Genético (AG) é um dos mais utilizados para resolver os problemas NP-Difíceis [Harman, 2007]. Uma das razões para isso é porque os AG é fácil de implementar e compreender, e em muitos casos é usado quando não há um entendimento completo do problema a ser resolvido [Noronha et al., 2001]. Neste trabalho, o AG foi a primeira metaheurística utilizada para resolver o problema e fez parte de trabalhos desenvolvidos nas disciplinas do mestrado.

O AG é baseado na teoria de Darwin, onde é criada uma população de indivíduos e a cada iteração uma nova população é criada onde as características são passadas de pais para filhos (técnica de cruzamento) ou os indivíduos evoluem (técnica de mutação) [Abraham & Jain, 2005].

Cada iteração da população tem o nome de geração, e as formas evolutivas (cruzamento, mutação e elitismo) são os operadores genéticos. Cada indivíduo

da população é um cromossomo. O cromossomo é uma possível solução para o problema a ser resolvido. Cada solução tem uma função de aptidão (*fitness*) que representa a função objetivo do problema.

Os operadores genéticos podem ser:

- **Elitismo:** Os melhores cromossomos são copiados para a próxima geração.
- **Cruzamento:** Cria novos cromossomos de dois ou mais pais.
- **Mutação:** As características de um cromossomo são modificadas.

O processo evolutivo, mostrado na Figura 3.5, começa com uma população inicial. Até um determinado critério de parada, previamente definido, uma nova população é gerada para substituir a anterior. Em cada iteração é avaliada a aptidão dos cromossomos, em seguida, é selecionado uma amostra da população para aplicar os operadores genéticos.

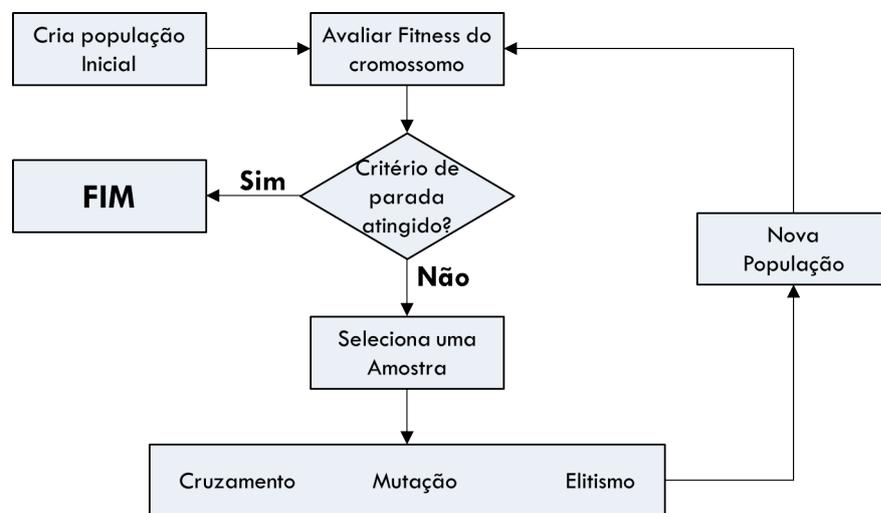


Figura 3.5. Processo Evolutivo do Algoritmo Genético.

Para cada problema é necessário definir como será representada uma solução e como funcionam os operadores genéticos.

3.3.1.1 Representação da solução

Para o PCC, a solução foi definida como um vetor n-dimensional que corresponde a cada ponto de demanda que será coberto. Cada posição (gene) deste vetor indica qual é o nó sensor que cobre o ponto de demanda. Como mostrado na Seção 3.2.A função de aptidão de cada solução é obtida somando os custos de cada sensor ativo.

O AG utiliza, somente, a representação por demanda porque o tempo gasto para executar todas as gerações é grande (mais de 1 hora para as menores instâncias). A representação por sensor demora ainda mais que a representação por demanda por causa do tamanho dos vetores de solução, para os casos de teste, serem maiores.

3.3.1.2 Cruzamento

O cruzamento é um operador genético onde são criados novos cromossomos a partir de genes de dois ou mais pais.

Neste trabalho, criamos um vetor binário aleatoriamente (Máscara) e selecionamos os genes (Pontos de Demanda) a partir do primeiro pai, onde a máscara é 1, e os genes a partir do segundo pai, onde a máscara é 0, e combina os genes para formar o filho. Criamos outro filho da mesma forma, mas os genes que são copiados são invertidos, ou seja, onde a máscara é 1 copiamos o gene do segundo pai, e onde a máscara é 0 copiamos o gene do primeiro pai. Um exemplo do processo é apresentado na Figura 3.6.

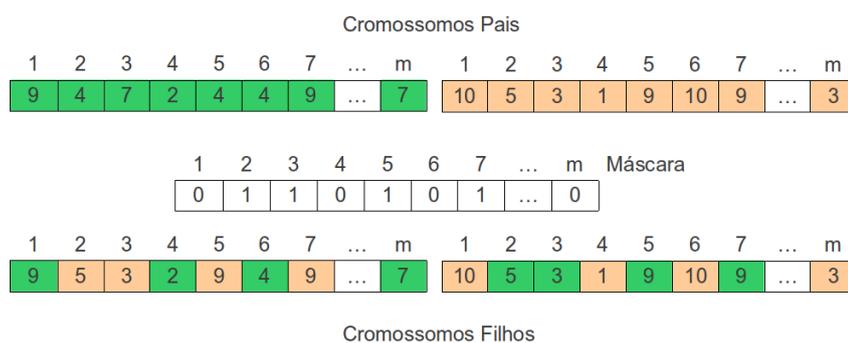


Figura 3.6. Cruzamento para o Algoritmo Genético.

3.3.1.3 Mutação

O cruzamento é um operador genético onde é criado novos cromossomos modificando as características de um cromossomo.

Na mutação, é selecionado um gene (um ponto de demanda) aleatoriamente e escolhe-se um outro nó sensor, dentre os possíveis, para cobrir aquele ponto, como mostrado na Figura 3.7.

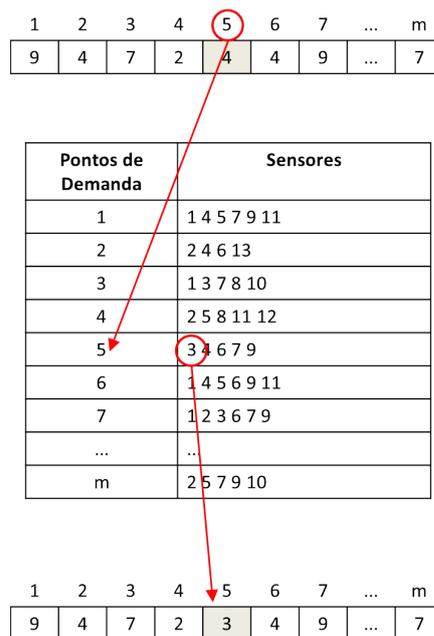


Figura 3.7. Mutação para o Algoritmo Genético.

3.3.2 Greedy Randomized Adaptive Search Procedure

O *Greedy Randomized Adaptive Search Procedure* (GRASP) é uma metaheurística utilizada para encontrar soluções aproximadas para problemas de otimização combinatória [Feo & Resende, 1989]. É composto por duas fases, uma de construção e outra de refinamento. A primeira fase gera uma solução inicial e a segunda fase utiliza mecanismos de busca local para melhorar a solução inicial.

Os algoritmos de busca local (ou busca em vizinhança) [Reeves, 1993] caracterizam-se por partirem de uma solução viável inicial (que pode ser obtida

por outra heurística) e alterá-la, iterativamente, para soluções vizinhas também viáveis, mas com melhor valor da função objetivo, até que um critério de parada seja satisfeito.

Seu funcionamento consiste basicamente em aplicar, repetidamente, busca local à diferentes soluções iniciais, desta forma sempre que houver uma solução melhor esta será guardada como resposta [Resende & Ribeiro, 2003]. O procedimento básico do GRASP é descrito no Algoritmo 1.

Algoritmo 1: GRASP Básico.

Saída: Solução x^*

```

1  $j \leftarrow 0$ ;
2  $x^* \leftarrow \infty$ ;
3 enquanto  $j < N$  faça
4    $x \leftarrow Construção()$ ;
5    $BuscaLocal(x)$ ;
6   se  $x < x^*$  então
7      $x^* \leftarrow x$ ;
8   fim se
9    $j++$ ;
10 fim enqto
11 retorna  $x^*$ 

```

A variável x^* é a solução para o problema de cobertura. Para o GRASP, foram utilizadas as duas representações de solução mostradas da Seção 3.2. O custo da solução é simplesmente a soma dos custos de cada nó sensor ativado.

3.3.2.1 Fase de Construção

Uma solução inicial é gerada na fase de construção através de um algoritmo guloso aleatório. A solução é construída elemento a elemento. Para a escolha de cada elemento é feita uma Lista Restrita de Candidatos (LRC) que contém todos os possíveis elementos que podem ser inseridos na solução parcial e a partir desta lista um elemento é escolhido [Resende & Ribeiro, 2003].

A LRC é definida pelo seguinte conjunto:

$$LCR = \{ e | c(e) \leq c_{min} + \alpha(c_{max} - c_{min}) \} \quad (3.4)$$

onde $0 \leq \alpha \leq 1$ e $c(e)$ é o custo do nó sensor e .

Quanto mais próximo de 0 o valor de α mais gulosa será a construção da LRC, pois será composta pelos nós sensores de melhor custo, e quanto mais próximo de 1 mais aleatória será, pois todos os nós sensores estarão na LRC.

Um algoritmo guloso aleatório pode ser implementado com uma construção gulosa da LRC e escolha aleatória do elemento ou através da construção aleatória da LRC e escolha gulosa do elemento.

Para o Problema de Cobertura, a lista LRC é formada pelos possíveis nós sensores que irão fazer parte da solução. Na representação por demanda, para cada ponto de demanda é feita uma LRC que contém somente os nós sensores que podem cobrir aquele ponto e que se encaixem na definição da lista. Já na representação por sensor, a LRC é feita com todos os nós sensores que se encaixem na definição. A cada nó sensor escolhido verifica-se quais os outros pontos de demanda que ele pode cobrir e o algoritmo marca estes pontos como cobertos.

3.3.2.2 Fase de Refinamento

A fase de refinamento consiste em partir de uma solução inicial aplicar uma busca local, com o objetivo de melhorá-la. A busca local procura no espaço de soluções uma solução vizinha próxima a inicial que tenha uma função de custo menor. Se uma solução melhor for encontrada esta será a solução corrente, caso contrário a solução permanece a mesma. Uma solução vizinha para o problema de cobertura em RSSF é um subconjunto de nós sensores que contenha de um a três sensores diferentes da solução inicial.

A busca local foi implementada em duas fases: (1) Para cada nó sensor ativo é verificado se não existe algum outro nó sensor que cubra os mesmos pontos de demanda, mas que possua um custo menor. Se não houver mais nenhum nó sensor, passa-se para a próxima fase. (2) Para cada nó sensor que não está na solução verifica-se se ele não cobre o mesmo que um ou dois outros nós sensores que estão na solução e que possua um custo menor que o outro ou a soma dos outros dois. As duas fases são realizadas em um ciclo, até não ser possível realizar alguma troca.

A complexidade da fase 1 da busca local é $O(n^2)$, pois para cada nó sensor verifica-se um outro nó sensor que possa realizar a troca. Já da fase 2 é $O(n^3)$,

onde para cada nó sensor verifica-se outros dois nós sensores para realizar a troca.

3.4 Testes e Resultados Obtidos

Esta seção apresenta nossos parâmetros, métricas e resultados computacionais. As heurísticas foram implementadas na linguagem C++.

3.4.1 Parâmetros e Métricas

Foi utilizado o benchmark para o PCC disponível ao público na Beasley's OR Library [Beasley, 1990]. Este benchmark tem sido amplamente utilizado por muitos pesquisadores para verificar a sua eficácia na resolução de métodos para o PCC. Há um total de sete conjuntos de problemas considerados neste trabalho para avaliação. Os detalhes destes conjuntos de problemas são mostrados na Tabela 3.1.

Conjunto de Problemas	Número de Pontos de Demanda	Número de Sensores	Densidade	Número de Testes
SCP-4	200	1000	2%	10
SCP-5	200	2000	2%	10
SCP-6	200	1000	5%	5
SCP-A	300	3000	2%	5
SCP-B	300	3000	5%	5
SCP-C	400	4000	2%	5
SCP-D	400	4000	5%	5

Tabela 3.1. Detalhes dos conjunto de problemas do Benchmark.

Os conjuntos SCP-4, SCP-5 e SCP-6 são problemas produzidos usando um esquema definido por Balas & Ho [1980], enquanto conjunto de problemas SCP-A, SCP-B, SCP-C e SCP-D foram problemas de teste gerados aleatoriamente. Os conjuntos SCP-4 e SCP-5 têm 10 casos de testes e o resto dos conjuntos, o SCP-6, SCP-A, SCP-B, SCP-C e SCP-D, têm 5 casos de testes.

Todos os conjuntos de testes são instâncias para o PCC Ponderado, onde cada nó sensor tem um custo diferente, o que poderia significar que teríamos nós sensores diferentes e cada um com custo diferente. Consideremos os custos como o consumo de energia de cada nó sensor.

A densidade é o número total de número ‘1s’ na matriz A_{ij} . Por exemplo, o conjunto de problemas SCP-4 tem um total de 4000 ‘1s’ na matriz A_{ij} ($200 \times 1000 = 200.000 \times 2\% = 4.000$ ‘1s’). A matriz A_{ij} é a matriz binária que indica se um nó sensor j cobre um ponto de demanda i .

Foram comparados os resultados da função objetivo do problema de cobertura entre as heurísticas e a solução ótima para cada caso de teste. A métrica utilizada é o somatório dos custos de cada nó sensor.

Para um melhor entendimento das distâncias entre esses resultados foi calculado o GAP das soluções, $GAP = 100 \times (F.O. - OTIMO)/OTIMO$, sendo $F.O.$ a função objetivo da heurística. Foram comparados ainda os tempos de execução de cada teste.

3.4.2 Bateria de Testes do Algoritmo Genético

Para o AG foram utilizadas 200 gerações com uma população de 200 cromossomos. Para a população inicial foram criados 400 cromossomos e dentre eles foram escolhidos os 200 melhores. A criação desta população foi baseada na fase construtiva do GRASP, visto na Seção 3.3.2.1. Os parâmetros dos operadores genéticos são: (a) Elitismo = 15%, (b) Mutação = 5% e (c) cruzamento = 85%. Estes parâmetros foram escolhidos empiricamente, e apresentados os que obtiveram os melhores resultados. Os Resultados são mostrados na Tabela 3.2.

Os resultados mostram que o AG não obteve bons GAPs, exceto para o grupo SCP-4 onde os GAPs estiveram menores ou iguais a 10% em metades das instâncias de teste. Pode se destacar o último conjunto de teste, o SCP-D, aonde o GAP chega a ser de 140%, ou seja, a função objetivo foi mais que o dobro da solução ótima. Acredita-se que ocorreram pelo fato dessas instâncias serem as maiores com 400 pontos de demanda, 4000 nós sensores e densidade de 5%. Com relação ao tempo o AG ficou estável para os casos de testes do seu grupo.

O AG utiliza, somente, a representação por demanda porque o tempo gasto para executar todas as gerações é grande (mais de 1 hora para as menores instâncias). A representação por sensor demora ainda mais que a representação por demanda porque o tamanho do vetor de solução, para os casos de teste, é maior.

SCP	Valor Ótimo	Algoritmo Genético			
		Função Objetivo	GAP (%)	Qtde de Sensores	Tempo (min:seg)
4.1	429	448	4,43	75	62:27,63
4.2	512	586	14,45	73	61:50,35
4.3	516	595	15,31	75	61:48,52
4.4	494	568	14,98	76	62:04,33
4.5	512	575	12,30	72	62:20,58
4.6	560	593	5,89	73	61:49,57
4.7	430	463	7,67	67	61:18,39
4.8	492	536	8,94	71	61:19,61
4.9	641	724	12,95	77	61:22,60
4.10	514	566	10,12	71	61:21,75
5.1	253	306	20,95	73	70:18,30
5.2	302	386	27,81	78	69:22,83
5.3	226	277	22,57	67	67:20,84
5.4	242	299	23,55	71	67:21,87
5.5	211	233	10,43	69	67:21,78
5.6	213	269	26,29	71	67:26,21
5.7	293	336	14,68	74	67:21,71
5.8	288	371	28,82	78	67:51,15
5.9	279	318	13,98	69	69:28,45
5.10	265	312	17,74	72	69:03,93
6.1	138	176	27,54	45	71:23,31
6.2	146	172	17,81	40	71:40,14
6.3	145	185	27,59	44	67:00,55
6.4	131	159	21,37	43	67:04,25
6.5	161	203	26,09	45	67:08,22
A.1	253	336	32,81	88	100:26,11
A.2	252	353	40,08	89	100:40,17
A.3	232	295	27,16	83	103:52,85
A.4	234	323	38,03	86	95:21,58
A.5	236	332	40,68	87	92:01,67
B.1	69	137	98,55	48	100:58,98
B.2	76	139	82,89	50	100:01,33
B.3	80	144	80,00	51	99:56,99
B.4	79	139	75,95	51	100:03,13
B.5	72	133	84,72	49	103:24,99
C.1	227	335	47,58	101	134:53,43
C.2	219	333	52,05	103	133:33,90
C.3	243	350	44,03	97	121:27,60
C.4	219	333	52,05	102	121:28,59
C.5	215	340	58,14	104	122:33,51
D.1	60	144	140,00	54	134:12,40
D.2	66	131	98,48	52	132:58,19
D.3	72	149	106,94	53	132:52,44
D.4	62	141	127,42	56	136:17,86
D.5	61	147	140,98	55	134:22,60

Tabela 3.2. Resultados para o AG em cada problema "SCP" do benchmark.

3.4.3 Bateria de Testes do GRASP

O GRASP executa 100 iterações com um alfa de $\alpha = 2\%$, este alfa foi escolhido após testes empíricos avaliando o custo da solução em diferentes alfas. Os resultados dos testes são apresentados na Tabela 3.3.

Os resultados mostram que o GRASP com representação por sensor não obteve bons resultados ($GAP \geq 20\%$) em todas as instâncias. Já o GRASP com representação por demanda obteve bons GAPs ($\leq 15\%$) na maioria dos casos de testes, ele obteve resultados ruins, somente, nas maiores instâncias. Com relação ao tempo as duas representações não geraram diferenças significativas, a diferença

SCP	GRASP Sensores (alpha = 0.02)					GRASP Demanda (alpha = 0.02)			
	Valor Ótimo	Função Objetivo	GAP (%)	Qtde de Sensores	Tempo (min:seg)	Função Objetivo	GAP (%)	Qtde de Sensores	Tempo (min:seg)
4.1	429	515	20,05	91	00:11,18	452	5,36	74	00:10,82
4.2	512	709	38,48	95	00:11,55	586	14,45	74	00:11,32
4.3	516	686	32,95	94	00:11,29	563	9,11	73	00:10,66
4.4	494	677	37,04	98	00:11,78	550	11,34	74	00:11,10
4.5	512	704	37,50	98	00:11,70	559	9,18	70	00:10,81
4.6	560	740	32,14	92	00:11,35	593	5,89	71	00:10,92
4.7	430	570	32,56	90	00:11,39	500	16,28	72	00:11,80
4.8	492	635	29,07	89	00:10,64	531	7,93	69	00:10,34
4.9	641	906	41,34	104	00:12,44	714	11,39	75	00:11,85
4.10	514	668	29,96	92	00:10,98	551	7,20	71	00:11,35
5.1	253	322	27,27	90	00:11,20	285	12,65	72	00:11,44
5.2	302	428	41,72	98	00:12,60	350	15,89	73	00:11,82
5.3	226	294	30,09	86	00:12,19	244	7,96	67	00:09,70
5.4	242	320	32,23	96	00:12,84	257	6,20	68	00:10,44
5.5	211	282	33,65	93	00:11,78	230	9,00	69	00:10,94
5.6	213	305	43,19	95	00:12,14	241	13,15	69	00:10,41
5.7	293	412	40,61	98	00:13,80	323	10,24	69	00:11,00
5.8	288	404	40,28	99	00:13,92	315	9,38	70	00:11,56
5.9	279	379	35,84	91	00:11,58	317	13,62	67	00:11,12
5.10	265	343	29,43	92	00:11,57	286	7,92	70	00:10,60
6.1	138	190	37,68	55	00:06,97	156	13,04	43	00:06,69
6.2	146	210	43,84	55	00:06,79	164	12,33	41	00:06,33
6.3	145	189	30,34	50	00:06,43	161	11,03	42	00:06,80
6.4	131	166	26,72	54	00:06,68	140	6,87	43	00:06,46
6.5	161	227	40,99	57	00:07,28	179	11,18	41	00:06,57
A.1	253	348	37,55	109	00:15,37	280	10,67	75	00:12,56
A.2	252	367	45,63	115	00:15,10	284	12,70	85	00:13,80
A.3	232	317	36,64	110	00:14,10	256	10,34	81	00:11,79
A.4	234	332	41,88	107	00:13,88	267	14,10	82	00:11,82
A.5	236	352	49,15	115	00:14,50	253	7,20	81	00:12,40
B.1	69	105	52,17	57	00:06,98	82	18,84	44	00:06,86
B.2	76	117	53,95	61	00:08,00	91	19,74	49	00:07,15
B.3	80	110	37,50	56	00:07,55	96	20,00	49	00:06,63
B.4	79	120	51,90	63	00:08,40	96	21,52	48	00:07,65
B.5	72	108	50,00	59	00:07,87	86	19,44	46	00:06,81
C.1	227	303	33,48	121	00:15,00	254	11,89	95	00:14,54
C.2	219	316	44,29	124	00:15,25	262	19,63	96	00:14,44
C.3	243	358	47,33	126	00:15,46	288	18,52	93	00:14,33
C.4	219	324	47,95	128	00:15,38	260	18,72	91	00:13,98
C.5	215	307	42,79	123	00:15,46	253	17,67	98	00:14,70
D.1	60	101	68,33	63	00:08,11	82	36,67	51	00:07,84
D.2	66	103	56,06	65	00:07,83	86	30,30	53	00:07,95
D.3	72	104	44,44	64	00:08,10	89	23,61	50	00:07,63
D.4	62	98	58,06	64	00:08,60	78	25,81	49	00:07,70
D.5	61	90	47,54	60	00:08,30	80	31,15	53	00:08,41

Tabela 3.3. Resultados para os GRASPs em cada problema "SCP" do benchmark.

ficou na média de 1 segundo, sendo a representação por demanda mais rápida.

3.4.4 Análise dos Testes

Analisando os GAPs das heurísticas, é possível notar que o GRASP com representação por demanda obtém melhores resultados na maioria dos casos, isto se dá pelo fato da representação forçar a escolha para cada demanda descoberto de um nó sensor com o valor baixo. Na representação por sensor, a escolha é feita apenas se o valor do nó sensor é considerado baixo e não se ele irá realmente cobrir algum ponto de demanda descoberto.

O AG possui soluções próximas ao GRASP com representação por sensor até o conjunto de casos SCP-4. Em alguns casos, como o SCP-4.1, SCP-4.6 e o SCP-4.7 o resultado do AG conseguiu gerar as melhores soluções entre as três heurísticas. Após este conjunto o AG tem soluções muito inferiores.

Comparando os tempos das heurísticas pode-se ver que o GRASP com as duas representações consegue executar seus testes muito mais rápido que o AG. O AG demora mais de 60 minutos, enquanto o GRASP fica entre 10 e 15 segundos. O tempo de execução também definiu por que não foram utilizadas as duas representações de solução para o AG, já que a representação por sensor demora mais tempo pelo motivo do tamanho do vetor de solução ser maior que o da representação por demanda.

Com isso, conclui-se que para estes testes o GRASP com representação por demanda é mais viável que as outras metaheurísticas utilizadas pois gera os melhores resultados a um custo de tempo rápido e esta será a abordagem escolhida para continuar o trabalho.

3.5 Considerações Finais

Neste capítulo apresentamos o Problema de Cobertura que compõe o PCAR-SM. O objetivo do Problema de Cobertura é encontrar o menor subconjunto de sensores que consiga cobrir a área de monitoramento. Apresentou-se sua definição, modelo matemático e duas heurísticas avaliando seus resultados e tempos de execução. As heurísticas geraram valores próximos do valor ótimo, mas ainda há espaço para melhoras.

O foco do capítulo foi avaliar o impacto na representação da solução para na obtenção de soluções para o problema de cobertura. Foram implementadas duas representações de soluções: (a) representação por sensor, onde cada elemento do vetor de solução representa um nó sensor que deverá estar ligado ou desligado; (b) representação por demanda, onde cada elemento do vetor de solução representa um ponto de demanda no qual indicará qual o nó sensor que o cobre.

Foram utilizadas duas técnicas de otimização, um AG e um GRASP, sendo que somente o GRASP utilizou as duas representações. Cada heurística foi comparada com um benchmark da *Beasley's OR Library* para o problema de cobertura.

Os resultados mostraram que o GRASP com representação por demanda obteve desempenho melhor em relação ao valor da função objetivo e do tempo para cada caso de teste. Isto se dá pelo fato da representação forçar a escolha, para cada ponto de demanda descoberto, de um nó sensor com o custo baixo. Por este motivo essa abordagem foi escolhida para ser utilizada no PCAR-SM.

No próximo capítulo apresentamos o Problema de Agrupamento, junto com a abordagem de grades virtuais que baseia o modelo matemático e as heurísticas utilizadas.

Capítulo 4

Problema de Agrupamento

Neste capítulo são mostrados os algoritmos para resolver o Problema de Agrupamento do PCAR-SM. O objetivo do Problema de Agrupamento é encontrar grupos de nós sensores que estão, relativamente, próximos e que contenham pelo menos um nó que consiga se comunicar com todos os outros nós do grupo.

Na Seção 4.1 definimos o Problema de Agrupamento em Rede de Sensores Sem Fio. Na Seção 4.1.1 apresentamos a abordagem de grades virtuais que serve de base para o modelo matemático implementado para este problema e também para os algoritmos implementados. Na Seção 4.2.1, é mostrado o algoritmo *Greedy Clustering Algorithm* (GCA). Por último, a Seção 4.3 apresenta os resultados dos algoritmos comparados com o modelo matemático para o problema.

4.1 Definição do Problema de Agrupamento em Redes de Sensores sem Fio

O problema de agrupamento em redes de nós sensores sem fio pode ser descrito como encontrar grupos de nós sensores que estão, relativamente, próximos e conseguem se comunicar com um nó central (não necessariamente no centro do grupo). Cada grupo é chamado, também, de *cluster*. Cada grupo tem um líder, o *cluster head*. A Figura 4.1 exemplifica o problema.

De uma maneira mais formal podemos definir o problema de agrupamento

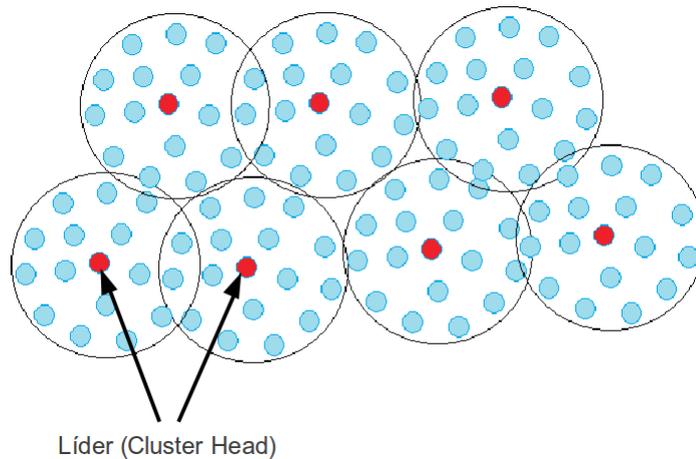


Figura 4.1. Rede Hierárquica.

em redes de nós sensores sem fio como: *Dado um conjunto de nós sensores $S = \{1, 2, \dots, n\}$ e suas coordenadas no espaço. O objetivo do problema é encontrar subconjuntos de S , que são independentes entre si, ou seja, que não exista um nó sensor s_i que pertence a mais de um subconjunto, e cada subconjunto possua pelo menos 1 nó sensor que consiga se comunicar com todos os outros nós sensores do grupo.*

Neste trabalho não há necessidade do líder de um grupo alcançar outro grupo, pois está sendo considerada a utilização de um nó sorvedouro móvel que irá percorrer por todos os grupos. Se o nó sorvedouro for fixo, seria necessária a inclusão da seguinte restrição: *Deverá existir uma rota de comunicação entre os líderes dos grupos e o nó sorvedouro.*

4.1.1 Abordagem de Grades Virtuais para o Agrupamento

Para formamos os grupos, propomos uma abordagem baseada na ideia de grades virtuais. Nós dividimos a área de monitoramento em grades, onde cada um é quadrado de lado r , como mostrado na Figura 4.2.

O tamanho da grade é determinada pelo valor do raio de comunicação R , tal como utilizado no algoritmo GAF (Geographical Adaptive fidelidade) [Xu et al. [2001]]. A formação da grade é apresentada na Figura 4.3.

O comprimento do lado da grade é obtido com a distância da diagonal for-

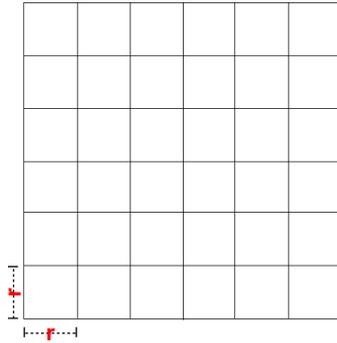


Figura 4.2. Área de monitoramento dividida em grades.

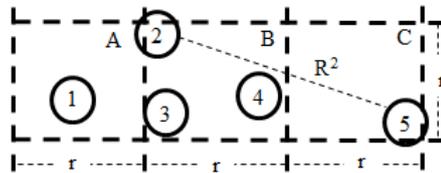


Figura 4.3. Grade virtual determinado pelo raio de comunicação definido no algoritmo GAF.

mada por duas grades adjacentes, utilizando trigonometria como segue:

$$r^2 + (2r)^2 \leq R^2 \quad (4.1)$$

$$r \leq \sqrt{\frac{R^2}{5}} \quad (4.2)$$

onde r é o comprimento do lado da grade e R é o raio de comunicação do nó sensor.

Quando nós fazemos $r \leq \frac{R}{\sqrt{5}}$, permitimos que um nó sensor que está dentro de uma grade consiga se comunicar com qualquer outro nó sensor na grade adjacente da esquerda, da direita, de cima e de baixo. Baseado neste conceito, nós propomos juntar cinco grades para formar um grupo, como mostrado na Figura 4.4. Assim, o líder é um nó sensor que está dentro da grade central para que consiga se comunicar com qualquer outro nó sensor do grupo.

Em alguns casos, o grupo pode ter menos de cinco grades, podendo ter apenas uma grade. A Figura 4.5 mostra uma área dividida em grupos com n -grades, onde

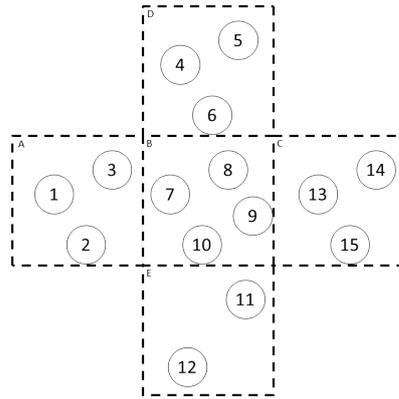


Figura 4.4. Grupo formado por cinco grades.

$n = 1, 2, 3, 4, 5.$

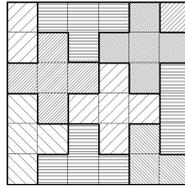


Figura 4.5. Rede hierárquica composta por grades e grupos.

Nós escolhemos o esquema de cruz ao invés do esquema de hexágono (ver Figura 4.6) por que o esquema de cruz é mais simples para o nó sensor encontrar a qual grade ele pertence. O esquema de cruz cobre uma área maior que a do esquema de hexágono ($R^2 > \frac{R^2 3\sqrt{3}}{8}$).

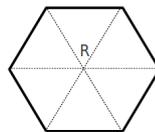


Figura 4.6. Modelo de grupo no esquema de Hexágono.

4.1.2 Modelo Matemático

Transformamos o Problema de Agrupamento com grades virtuais em um Problema de Cobertura de Conjuntos (PCC) [Johnson, 1973], onde cada grupo cobre deter-

minimadas grades, tendo como objetivo de minimizar o número de grupos. Para isto, precisamos simular cada possível grupo tendo uma determinada grade como o centro. Ou seja, para cada grade montamos todos os possíveis grupos. Estes grupos representam os conjuntos do PCC.

Desta forma, todas as grades, que não estejam em uma das laterais ou cantos, poderão ter 16 possíveis grupos, sendo 1 composto de 1 grade, 4 compostos de 2 grades, 6 compostos de 3 grades, 4 compostos de 3 grades e 1 composto de 5 grades, como mostrado na Figura 4.7 onde o 'x' indica a grade central.

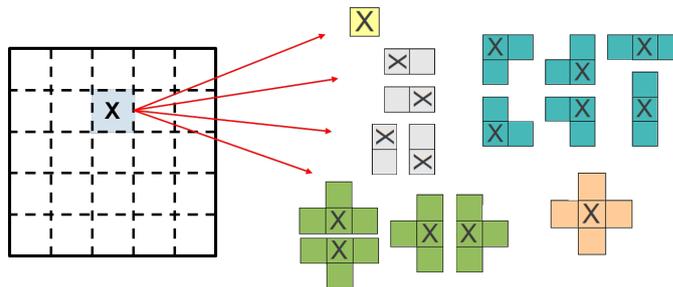


Figura 4.7. Todos os 16 possíveis grupos.

As grades dos cantos e das laterais não podem formar todos os 16 possíveis grupos. Os cantos só podem formar 4 possíveis grupos e as laterais 8 possíveis grupos, como visto na Figura 4.8 onde o 'x' indica a grade central.

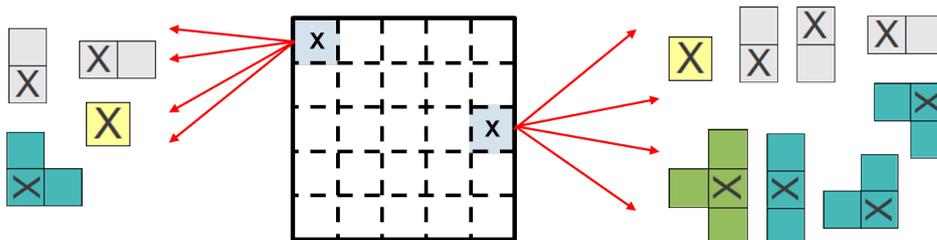


Figura 4.8. Possíveis grupos: Cantos 4 grupos e Laterais 8 grupos.

O número total de grupos pode ser obtido com a seguinte fórmula:

$$GRUPOS = 16 \times (Nl - 2)^2 + 4 \times 8 \times (Nl - 2) + 4 \times 4 \quad (4.3)$$

onde Nl é o número de grades em uma linha da área de monitoramento. A primeira parte da equação $(16 \times (Nl - 2)^2)$ representa todos os grupos onde a grade central não está nos cantos e nas laterais da área. A segunda parte $(4 \times 8 \times (Nl - 2))$ representa os grupos onde a grade central está nas laterais com exceção dos cantos. E a última parte (4×4) representa os grupos onde a grade central está nos cantos.

Um grupo cobre as grades que o compõe. Assim, para a área de monitoramento temos que encontrar o menor número de grupos que cobre todas as grades. O modelo matemático resultante é demonstrado a seguir:

$$\text{Minimizar} \quad \sum_{j \in GRUPOS} x_j \quad (4.4)$$

sujeito a:

$$\sum_{j \in GRUPOS} C_{ij} x_j = 1 \quad i \in GRADES \quad (4.5)$$

$$x_j \in \{0, 1\} \quad j \in GRUPOS \quad (4.6)$$

onde o x_j representa cada possível grupo, e a matriz C_{ij} é a relação grade \times grupo que indica quais grupos cobrem a grade.

Destaca-se que na restrição (Equação 4.5) o somatório tem que ser igual ($=$) a 1, e no modelo clássico do Problema de Cobertura é maior ou igual (\geq) a 1. Isto se deve ao fato de não poder existir sobreposição de grupos, ou seja, uma mesma grade não pode pertencer a mais de um grupo.

4.2 Abordagens para o Problema de Agrupamento

Nós propomos uma abordagem para o problema de agrupamento baseada em grades virtuais, o *Greedy Clustering Algorithm* (GCA).

O GCA é o algoritmo que gera grades e grupos. Seu objetivo é, a cada iteração, escolher um dos grupos que contém mais grades virtuais.

4.2.1 Greedy Clustering Algorithm

O *Greedy Clustering Algorithm* (GCA) é o algoritmo que gera grades e grupos. Seu objetivo é, a cada iteração, escolher um dos grupos que contém mais grades virtuais. Inicialmente, escolhem-se os grupos com cinco grades e elimina estas grades para a próxima escolha. Quando não existem mais grupos com com cinco grades sobrando, ele passa a escolher os grupos com quatro grades, em seguida os com três grades, duas grades e, finalmente, os grupos com somente uma grade.

O algoritmo compõe o centro da área de monitoramento com grupos e, em seguida, os lados da área. O Algoritmo 2 mostra este procedimento.

Algoritmo 2: Algoritmo GCA.

Saída: x

```

1  $r \leq \frac{R}{\sqrt{5}}$ ;
2 Cria grades;
3 para cada  $g \in \text{GRADES}$  faça
4   | Define  $g$  como a grade central do grupo;
5   |  $g.Length =$  Número de grades pertencente ao grupo;
6 fim para cada
7  $i = 0$ ;  $j = 0$ ;
8 enquanto  $j < \text{Numero de grades}$  faça
9   |  $N = \max_{g \in \text{GRADES}}(g.Length)$ ;
10  |  $LRC = \{g \mid g.Length = N\}$ ;
11  |  $s = \text{random}(|LRC|)$ ;
12  |  $x_i = s$ ;
13  |  $\text{GRADES} = \text{GRADES} - s$ ; /* Para todas as grades do grupo
14  |   */
14  |  $i = i + 1$ ;
15  |  $j = j + g.Length$ ;
16 fim enqto
17 retorna  $x$ 
```

O primeiro passo do algoritmo é criar as grades virtuais, e para isto, é necessário calcular o comprimento do lado da grade como definido na Equação 4.2. Este lado é ajustado para assegurar que todas as grades tenham o mesmo tamanho. Em outras palavras, pode-se diminuir o tamanho r para ajustar o comprimento

do lado. Após isto, cada grade é simulada como uma grade central de um grupo e são identificadas quais as grades que formam o grupo.

O próximo passo é a abordagem gulosa, que começa na linha 9. A cada iteração, selecionamos todos os grupos que têm o maior número de grades e escolhemos aleatoriamente um deles. Adicionamos este grupo na solução e removemos todos as grades dele do conjunto de grades, como visto na linha 14.

O procedimento termina quando todas as grades pertencerem a algum grupo. A Figura 4.2.1 ilustra os passos da abordagem.

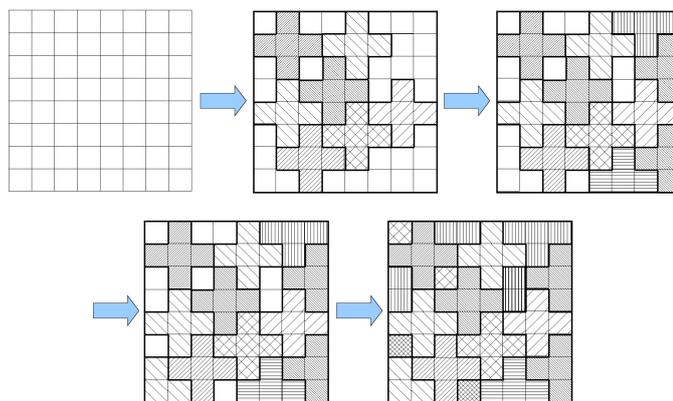


Figura 4.9. Passo a passo do *Greedy Clustering Algorithm*

4.3 Resultados Computacionais

Esta seção apresenta os parâmetros, métricas e resultados computacionais. A área de monitoramento é representada por um quadrado no plano e a comunicação é representada por um círculo de raio R . Esses parâmetros foram variados e para cada par (raio de comunicação \times área de monitoramento) de valores, geramos 33 instâncias diferentes, cada uma representando uma configuração inicial de uma RSSF. Para cada instância, executamos o algoritmo de agrupamento 30 vezes. Para cada par (raio de comunicação \times área de monitoramento) de valores, apresentamos a média das melhores soluções (o menor número de grupos).

A posição dos nós sensores segue uma configuração de grade irregular, onde 100 nós sensores foram colocados em uma grade e erros aleatórios foram aplicados nas suas coordenadas (x, y) como ilustrado na Figura 4.10.

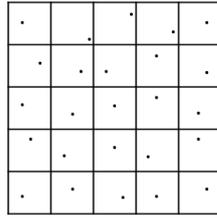


Figura 4.10. Grade Irregular

4.3.1 Parâmetros e Métricas

Considerando $u.d.$ a unidade de distância, nós usamos os seguintes parâmetros:

- Raio de Comunicação: $7u.d.$, $9u.d.$ e $13u.d.$.
- Área de Monitoramento: $30u.d. \times 30u.d.$, $50u.d. \times 50u.d.$, $80u.d. \times 80u.d.$, $100u.d. \times 100u.d.$, $150u.d. \times 150u.d.$, e $200u.d. \times 200u.d.$.

O modelo matemático foi resolvido com o pacote de otimização CPLEX [Documentation, 2012] com o tempo de execução limitado em 2 horas.

A métrica para o Problema de Agrupamento é o menor número de grupos. Dado os aspectos aleatórios do algoritmo do GCA, para cada par (raio de comunicação \times área de monitoramento) de valores, apresentamos a média das melhores soluções.

4.3.2 Resultados

Na Tabela 4.1, apresentamos os principais resultados computacionais da abordagem de agrupamento. Nas primeiras duas colunas, indicamos o raio de comunicação e a área de monitoramento. Nas próximas duas colunas, apresentamos os limites inferior (LI) e superior (LS) alcançados com o modelo matemático. Quando o LI for igual ao LS temos a solução ótima do problema. Na sequência, apresentamos média aritmética do número de grupos e o desvio padrão do GCA. Para a abordagem é calculado um GAP para a melhor solução inteira (LS) encontrada no modelo matemático ($GAP = 100 * (N_c - LS) / N_c$, onde N_c é o número de grupos).

O número de grupos varia de acordo com o tamanho do raio de comunicação, quanto maior o raio menor o número de grupos. Por exemplo, na área $100u.d. \times$

Raio de Comunicação	Tamanho da Área (u.d. x u.d.)	Modelo Matemático		GCA		
		LI	LS	Média	Desvio Padrão	GAP (%)
7	30x30	24,00	24,00	31,45	1,12	23,69
	50x50	58,00	60,00	79,12	2,57	24,17
	80x80	143,43	155,00	207,82	5,42	25,42
	100x100	214,72	227,00	313,42	5,20	27,57
	150x150	474,20	533,00	703,82	8,60	24,27
	200x200	835,85	935,00	1244,33	10,29	24,86
9	30x30	16,00	16,00	20,52	1,03	22,03
	50x50	40,00	40,00	52,58	2,40	23,93
	80x80	87,61	92,00	122,61	3,49	24,97
	100x100	133,09	143,00	192,76	4,04	25,81
	150x150	300,10	326,00	441,03	7,82	26,08
	200x200	513,98	589,00	760,24	9,30	22,52
13	30x30	10,00	10,00	11,97	0,59	16,46
	50x50	20,00	20,00	25,48	1,23	21,51
	80x80	47,00	47,00	61,24	2,37	23,25
	100x100	72,02	76,00	100,85	3,78	24,64
	150x150	143,43	154,00	206,82	5,05	25,54
	200x200	255,48	277,00	373,85	6,22	25,91

Tabela 4.1. Resultados do GCA comparado com o Modelo Matemático

100u.d., se o $R = 7$ a média aritmética do número de grupos é 313,42, mas se o $R = 9$ a média aritmética do número de grupos é 100,85.

A distância entre o GCA e o LS fica, em quase todos os casos de teste, na faixa de 20%, o que mostra que a heurística pode ser melhorada ou trocada por heurísticas mais sofisticadas.

4.4 Considerações Finais

Neste capítulo apresentamos o Problema de Agrupamento do PCAR-SM. Seu objetivo é encontrar grupos de sensores que estão, relativamente, próximos e contenha pelo menos um nó que consiga se comunicar com todos os outros nós do grupo, considerando que estes grupos não tenham sobreposição.

Foi criada uma abordagem de grades virtuais onde se divide a área em grades e os grupos são compostos por um conjunto de grades adjacentes (no máximo 5 grades no grupo) formando um esquema de cruz. O objetivo do problema é minimizar o número de grupos.

A partir desta abordagem, pode-se modelar o Problema de Agrupamento

como um Problema de Cobertura de Conjuntos sem sobreposição. Propomos uma heurística gulosa (*Greedy Clustering Algorithm*, GCA) para resolvê-lo. A ideia do GCA é formar grupos com o maior número de grades possíveis. O GCA encontrou bons resultados quando comparado com os limites do modelo matemático, mas outras técnicas podem ser utilizadas para melhorar estes resultados.

A utilização das grades virtuais provou ser uma boa solução para o agrupamento por ser bastante simples para um nó, sabendo suas coordenadas, identificar a qual grade ele pertence. Sua simplicidade ainda o torna um método adequado para que seja implementado em uma versão distribuída onde alguns dos cálculos serão efetuados pelos nós sensores.

No próximo capítulo apresentamos o Problema de Roteamento Com Sorvedouro Móvel.

Capítulo 5

Problema de Roteamento

Neste capítulo são mostrados os algoritmos para resolver o Problema de Roteamento do PCAR-SM. O objetivo do Problema de Roteamento é definir uma rota para o nó sorvedouro móvel percorrer a área de monitoramento garantindo que o nó alcance todos os grupos.

Na Seção 5.1 definimos o Problema de Roteamento em RSSF. Na Seção 5.2 apresentamos as abordagens utilizadas para resolver o problema. Na Seção 5.2.2, é mostrado o algoritmo *Centralized Spatial Partitioning* (CSP). Na Seção 5.2.3 é apresentado o algoritmo *Centralized Spatial Partitioning Grid* (CSPG). Por último, a Seção 4.3 apresenta os resultados dos algoritmos implementados.

5.1 Definição do Problema de Roteamento em Rede de Sensores Sem Fio com Sorvedouro Móvel

O Problema de Roteamento apresentado neste trabalho consiste em definir uma rota para um nó sorvedouro móvel percorrer a área de demanda com o objetivo de coletar os dados que foram obtidos pelos nós sensores. Uma das abordagens é modelar como o Problema do Caixeiro Viajante (PCV) [Dantzig et al., 1954].

O PCV (representado na Figura 5.1) consiste na procura de uma rota que começa em alguma cidade, passe por todas as outras cidade e retorne a cidade

5.1.1 Modelo Matemático do Problema do Caixeiro Viajante

O PCV pode ser formalmente definido da seguinte forma: *Dado um conjunto de cidades $C = \{1, 2, \dots, n\}$ e seja $D = (d_{ij})$ uma $n \times n$ matriz com as distâncias entre cada cidade (nó), $X = (x_{ij})$ uma $n \times n$ matriz que indica se a rua (aresta) da cidade i para j existe, 1 se sim e 0 se não. O objetivo do PCV é encontrar uma permutação de C que minimize a soma das distâncias das ruas. A programação linear inteira para o PCV é:*

$$\text{Minimizar } z = \sum_{i \in C} \sum_{j \in C} d_{ij} x_{ij} \tag{5.1}$$

sujeito a:

$$\sum_{i \in C} c_{ij} x_{ij} = 1 \quad j \in C \tag{5.2}$$

$$\sum_{j \in C} c_{ij} x_{ij} = 1 \quad i \in C \tag{5.3}$$

$$\sum_{i \in S} \sum_{j \in \bar{S}} c_{ij} \geq 1 \tag{5.4}$$

$$c_{ij} \in \{0, 1\} \quad i, j \in C \tag{5.5}$$

onde c_{ij} indica se a rota de i para j estará na solução e S e \bar{S} são, respectivamente, cada subconjunto de cidades e seu complemento. As restrições 5.2 e 5.3 garantem que cada cidade será visitada apenas uma vez. As restrições 5.4 garantem que não existirá sub-rotas, como ilustrado na Figura 5.3. Já as restrições 5.5 informam que a variável c_{ij} é binária.

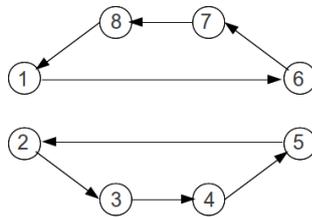


Figura 5.3. Sub-rotas no Problema do Caixeiro Viajante.

5.2 Abordagens para o Problema de Roteamento

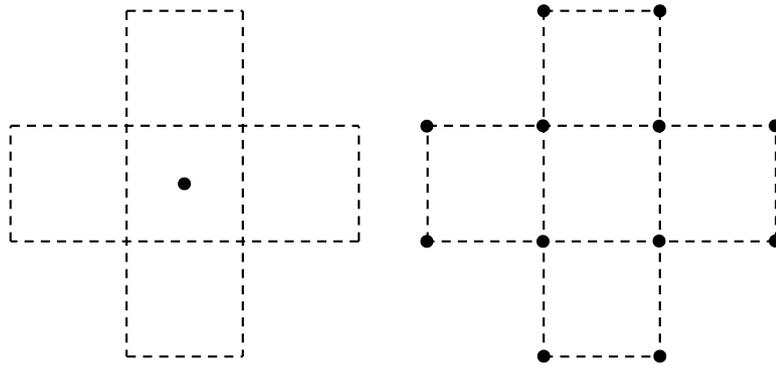
Para resolver o Problema de Roteamento, nós propomos duas abordagens gulosas baseadas na heurística do Vizinho Mais Próximo [Gutin et al., 2002]. É uma heurística baseada no algoritmo *Centralized Spatial Partitioning* (CSP) [Chatzigiannakis et al., 2007]. Na abordagem CSP, a rota é fixa e lembra o movimento de uma cobra. Nós comparamos a heurística desenvolvida com o CSP original. Nossas abordagens são descritas a seguir:

1. *Routing Greedy Algorithm - Center* (RGA-C) define a rota do nó sorvedouro baseado no algoritmo do vizinho mais próximo usando o centro da grade que contém o líder como ponto de coleta de dados;
2. *Routing Greedy Algorithm - Border* (RGA-B) define a rota do nó sorvedouro baseado no algoritmo do vizinho mais próximo usando as bordas do grupo como possíveis pontos de coleta de dados;
3. *Centralized Spatial Partitioning Grid* (CSPG) define a rota fixa do nó sorvedouro com base na estrutura de grades virtuais construído pelo *Greedy Clustering Algorithm* (GCA);

Todas as abordagens são centralizadas e o nó sorvedouro móvel conhece a posição de todos os nós sensores. Nas abordagens CSP, o nó sorvedouro móvel precisa saber, somente, o tamanho da área, porque o percurso é fixo.

5.2.1 Abordagens Gulosas

Nós usamos duas abordagens gulosas baseadas no Algoritmo do Vizinho Mais Próximo para o roteamento do nó sorvedouro móvel. A primeira abordagem, chamada *Routing Greedy Algorithm - Center* (RGA-C), o nó sorvedouro passa exatamente sobre o centro da grade que contém o líder, como mostra a Figura 5.4(a). Na outra abordagem, chamada *Routing Greedy Algorithm - Border* (RGA-B), o nó sorvedouro usa o raio de comunicação para reduzir o comprimento do percurso, passando pelos os lados dos grupos. Para passar ao lado do grupo, consideramos os vértices das grades, como mostra a Figura 5.4(b).



(a) *Routing Greedy Algorithm - Center* - (b) *Routing Greedy Algorithm - Border*

Figura 5.4. Modelo dos pontos utilizados para as abordagens gulosas.

Ambas as abordagens são baseadas nos algoritmos gulosos para resolver o PCV. Em cada iteração, escolhe os grupos mais próximos. O Algoritmo 3 mostra o procedimento.

Algoritmo 3: Abordagem Gulosa para o Problema de Roteamento.

Saída: *rota*

```

1  $p(x, y) = (0, 0)$   $i = 0$ ;
2 enquanto  $i < \text{Numberofgrupos}$  faça
3   |   Calcula distâncias para  $p$ ;
4   |    $Max = \max_{g \in GRUPOS} (Dis(g, p))$ ;
5   |    $Min = \min_{g \in GRUPOS} (Dis(g, p))$ ;
6   |    $LRC = \{g | Dis(g, p) \leq Min + \alpha(Max - Min)\}$ 
7   |    $s = \text{random}(|LRC|)$ ;
8   |    $rota_i = s$ ;
9   |    $GRUPOS = GRUPOS - s$ ;
10  |    $p = s$ ;
11  |    $i = i + 1$ ;
12 fim enquanto
13  $rota_i = p(0, 0)$ ;
14 retorna rota

```

O primeiro passo do algoritmo é encontrar a maior e a menor distância para o ponto atual, como na linha 4 e 5. Depois disso, ele cria uma lista de candidatos

chamada LRC. Esta lista é formada de acordo com um predeterminado α . Este α é uma probabilidade que determina se uma solução é gulosa ou aleatória, se $\alpha = 0$ a solução é gulosa, se $\alpha = 1$ a solução é aleatória.

Na linha 7, o algoritmo escolhe um dos grupos da lista *LRC*. Este grupo é adicionado na solução e removido do conjunto de *GRUPOS*, como visto na linha 8 e 9. O procedimento termina quando todo o grupo pertencente à rota.

As diferenças entre as abordagens estão nas linhas 4, 5 e 6. O ponto do grupo na primeira abordagem é o centro. Na outra abordagem, verificamos todos os pontos da lateral do grupo.

No RGA-B, se um ponto é escolhido e este ponto é um vértice de dois ou mais grupos, todos os grupos são incluídos na solução. Para encontrar estes grupos, verificamos se um grupo tem um ponto em que a distância para o ponto atual é 0.

5.2.2 Centralized Spatial Partitioning

A abordagem *Centralized Spatial Partitioning* (CSP) foi formulada por Chatzigiannakis et al. [2007]. Nesta abordagem a trajetória do nó sorvedouro móvel é composta de segmentos paralelos horizontais e assemelha-se ao movimento de uma cobra sobre a área. A Figura 5.5 ilustra a rota do nó sorvedouro com o CSP.

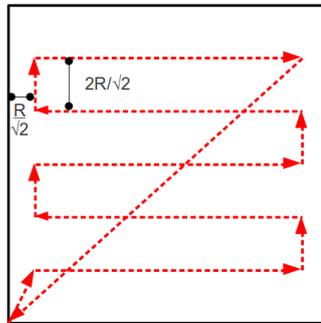


Figura 5.5. Rota gerada no *Centralized Spatial Partitioning*.

Cada segmento de reta é separado por uma distância de $2R/\sqrt{2}$ para assegurar que qualquer nó sensor possa ser alcançado, independente de onde esteja localizado na área, em pelo menos um ponto da rota. No CSP, a posição dos nós sensores ou grupos não alteram o custo da rota do nó sorvedouro. Em outras palavras, a rota

é fixa e o nó sorvedouro não precisa saber a localização dos nós sensores, somente o tamanho da área de monitoramento e do raio de comunicação.

5.2.3 *Centralized Spatial Partitioning Grid*

A abordagem *Centralized Spatial Partitioning Grid* (CSPG) usa a mesma ideia do CSP. O nó sorvedouro também gera segmentos paralelos horizontais para percorrer a área, mas esta abordagem a ideia de grades virtuais (apresentada no Capítulo 4) para desenhar a trajetória. Cada segmento é separado pela distância de quatro grades ($4r$, onde r é o tamanho da grade). A rota é projetada para permitir que o nó sorvedouro alcance todas as grades na área, pois qualquer um deles pode ser o líder de um grupo. A Figura 5.6 mostra a rota do nó sorvedouro móvel com CSPG.

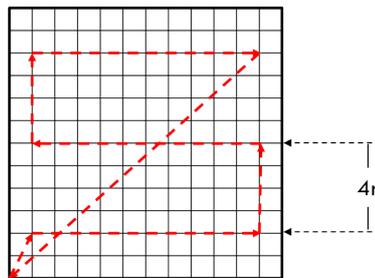


Figura 5.6. Rota gerada no *Centralized Spatial Partitioning Grid*.

Cada segmento tem 4 grades de distância para os outros segmentos. Isto é possível porque um nó sensor que esteja em uma grade consegue se comunicar com qualquer outro nó sensor na grade adjacente. Assim, o nó sorvedouro pode se comunicar com duas grades em cada lado do segmento.

Desta forma, a distância entre dois segmentos horizontais é maior do que no CSP ($2R/\sqrt{2} \leq 4R/\sqrt{5}$), o que diminui o tamanho da rota do nó sorvedouro.

5.3 Resultados Computacionais

Esta seção apresenta as métricas e os resultados computacionais. Foram utilizadas as mesmas instâncias descritas na Seção 4.3.

5.3.1 Métricas

A métrica para o Problema de Roteamento é o tamanho da rota do nó sorvedouro, que inicia e retorna ao canto inferior esquerdo da área de monitoramento.

Para o RGA-C e RGA-B, nós executamos os algoritmos 30 vezes para cada uma das 33 instâncias e apresentamos a média dos melhores resultados das 33 instâncias. O CSP, CSPG a rota é fixa e depende, somente, do tamanho da área de monitoramento e do raio de comunicação.

5.3.2 Resultados

Quanto ao Problema de Roteamento, as Figuras 5.7, 5.8 e 5.9 mostram que, como esperado, o tamanho da área de monitoramento é diretamente proporcional ao comprimento da rota do nó sorvedouro. Além disso, o raio de comunicação (R_c) é inversamente proporcional ao comprimento da rota (Figura 5.7). Isto acontece porque o raio de comunicação é o parâmetro que define o tamanho da grade ($r \leq R/\sqrt{5}$).

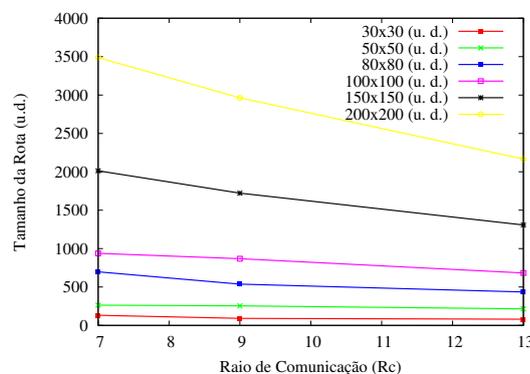


Figura 5.7. Raio de Comunicação \times Tamanho da Rota - Algoritmo CSPG.

Na Figura 5.8 vemos que o RGA-B consegue melhores soluções que o RGA-C. Isto acontece porque utilizar as bordas dos grupos permite que o nó sorvedouro móvel não precise ir até o centro do grupo, somente aos cantos e laterais da grade para comunicar-se com o líder. Isto indica que é melhor utilizar os lados dos grupos para criar a rota, como RGA-B, em vez de usar o centro do grupo.

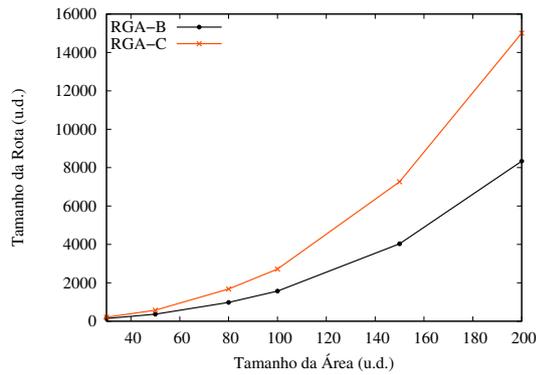
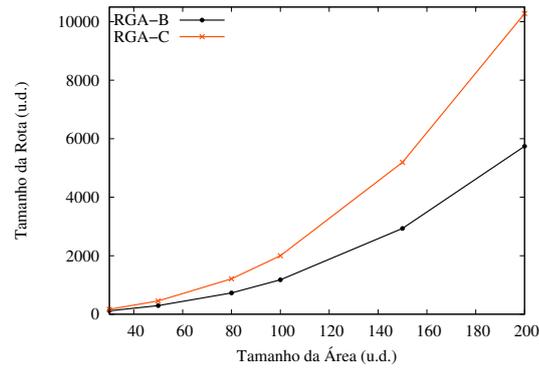
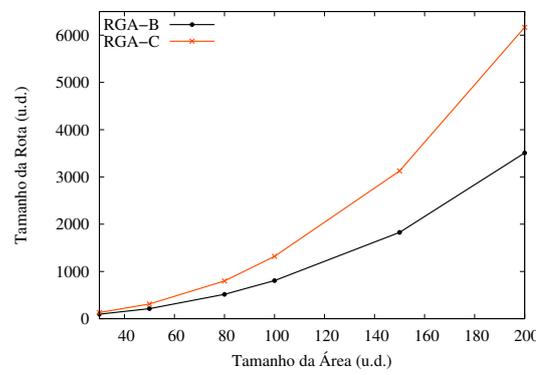
(a) Raio de Comunicação $R_c = 7$ (b) Raio de Comunicação $R_c = 9$ (c) Raio de Comunicação $R_c = 13$

Figura 5.8. Tamanho da Área \times Tamanho da Rota - Algoritmos RGA-B e RGA-C.

Na Figura 5.9 a abordagem CSPG alcança melhores resultados que o CSP, pelo motivo que a diferença entre os segmentos é maior no CSPG ($4r > 2R/\sqrt{2}$). Somente em três instâncias ($R = 9$ Área = $50 \text{ u.d.} \times 50 \text{ u.d.}$, $R = 13$ Área = $50 \text{ u.d.} \times 50 \text{ u.d.}$ e $100 \text{ u.d.} \times 100 \text{ u.d.}$) o CSP tem custos menores, isto acontece porque quando o nó sorvedouro volta para o ponto de origem, tem que percorrer a diagonal da área de monitoramento enquanto o CSP somente a lateral da área.

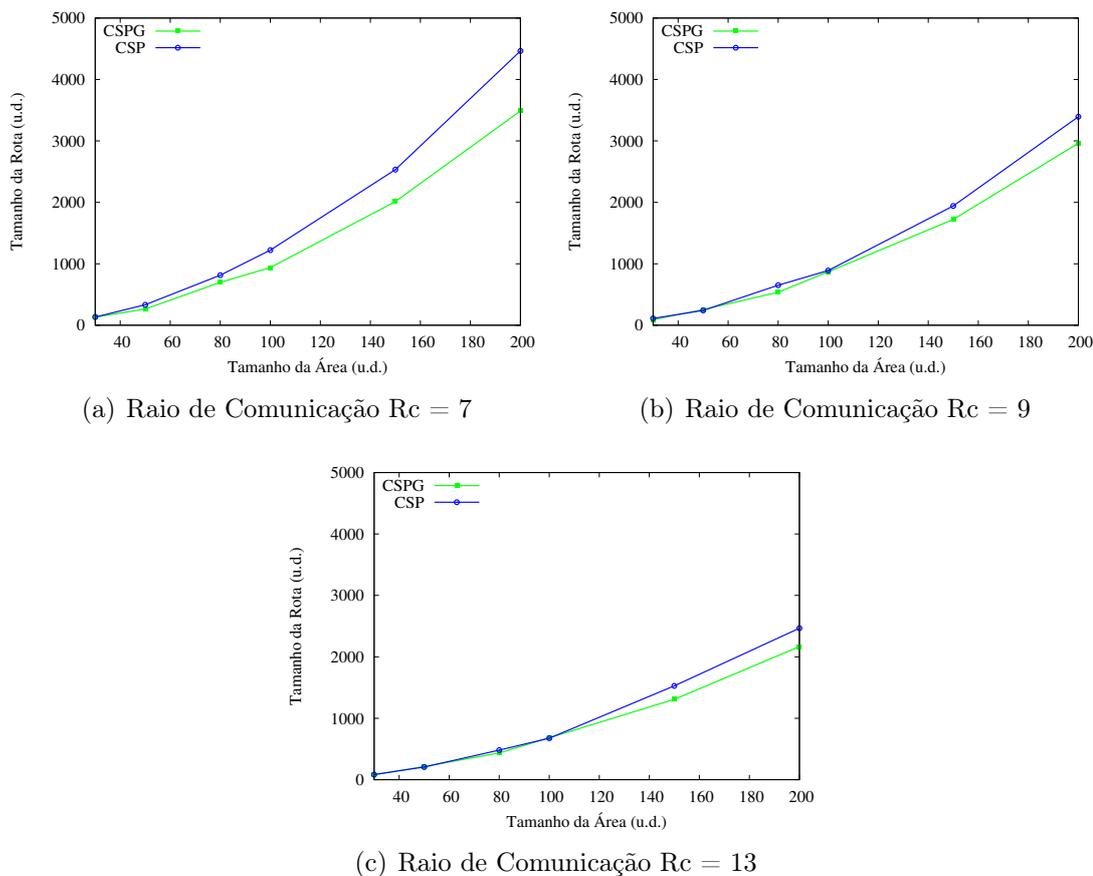


Figura 5.9. Tamanho da Área \times Tamanho da Rota - Algoritmos CSPG e CSP.

5.4 Considerações Finais

Neste capítulo apresentamos o Problema de Roteamento Com Sorvedouro Móvel do PCAR-SM. O objetivo do Problema de Roteamento é encontrar uma rota para o nó sorvedouro percorrer a área de monitoramento garantindo que o nó alcance todos os grupos.

Modelado como o Problema do Caixeiro Viajante, o nó sorvedouro móvel parte de um canto da área de monitoramento, percorre todos os grupos e retorna ao ponto inicial. Neste cenário, propomos duas abordagens gulosas, baseadas no Algoritmo do Vizinho Mais Próximo para o PCV, a abordagem *Routing Greedy Algorithm - Center* (RGA-C) e *Routing Greedy Algorithm - Border* (RGA-B).

Como alternativa a modelagem via PCV, implementamos uma heurística baseada no algoritmo *Centralized Spatial Partitioning* (CSP). Na abordagem CSP, a rota é fixa e lembra o movimento de uma cobra.

Em relação a abordagem baseada no PCV, os resultados indicam que é melhor utilizar os lados dos grupos para criar a rota, como utilizado no Algoritmo RGA-B, em vez de usar o centro do grupo como ponto a ser visitado, como acontece no Algoritmo RGA-C. Porém, para os cenários utilizados, a rota fixa gerada pelos Algoritmos baseados no CSP alcança todos os grupos e tem tamanho menor em comparação as heurísticas gulosas. Além disso no caso de mudança na estrutura dos grupos ela não precisa ser recalculada como acontece nos Algoritmos Gulosos.

No próximo capítulo apresentamos a união dos Problemas de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel que juntos formam o PCAR-SM. A união foi feita em duas partes primeiro avaliando os Problemas de Cobertura e Agrupamento, e em seguida os problemas de Agrupamento e Roteamento.

Capítulo 6

Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel

Neste capítulo são mostradas as estratégias para resolver em conjunto o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM) em RSSF. O objetivo do PCAR-SM é encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possível e encontrar a menor rota para um nó sorvedouro móvel percorrer. Ao tratar os problemas juntos queremos observar que algumas mudanças nos algoritmos levem a melhores resultados.

Na Seção 6.1 definimos o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM). Na Seção 6.2 apresentamos a estratégia para unir os Problemas de Cobertura e Agrupamento, junto com as avaliações sobre essa união. Por último, a Seção 6.3 apresenta a estratégia para unir os Problemas de Agrupamento e Roteamento, junto com as avaliações sobre essa união.

6.1 Definição do Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel em RSSF

O PCAR-SM é a união de três problemas:

- Problema de Cobertura - o objetivo é encontrar um subconjunto de nós sensores que consiga cobrir a área de monitoramento.
- Problema de Agrupamento - o objetivo é encontrar o menor número grupos de nós sensores que estão, relativamente, próximos e contenha pelo menos um nó que consiga se comunicar com todos os outros nós do grupo.
- Problema de Roteamento - o objetivo é encontrar uma rota para o nó sorvedouro percorrer a área de monitoramento garantindo que o nó alcance todos os grupos.

O objetivo do PCAR-SM é: *Dado um conjunto de nós sensores e uma área de monitoramento desenvolver algoritmos para encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possível e encontrar a menor rota para um nó sorvedouro móvel percorrer.* A Figura 6.1 mostra um exemplo de solução do para o PCAR-SM.

6.2 Problema de Cobertura + Problema de Agrupamento

A união dos problemas de cobertura e de agrupamento é feita, simplesmente, executando o problema de cobertura antes do problema de agrupamento, resultando no GRASP+GCA. Ou seja, para cada instância executamos o algoritmo GRASP com representação por demanda (heurística com melhor resultado avaliado neste trabalho, apresentado no Capítulo 3). Com o conjunto de nós sensores obtido inserimos cada um em sua respectiva grade, e para cada grade “vazia” escolhemos aleatoriamente um nó sensor não ativo para ser ativado.

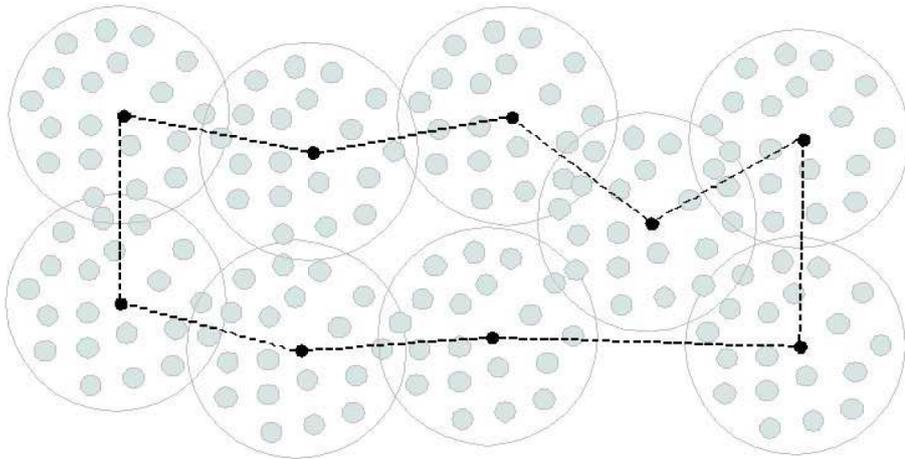


Figura 6.1. PCAR-SM: sensores agrupados com uma rota tracejada para o nó sorvedouro.

Em seguida, executamos o algoritmo GCA (apresentado no Capítulo 4) para formar os grupos.

6.2.1 Resultados - Problema de Cobertura + Problema de Agrupamento

Esta seção apresenta nossos parâmetros, métricas e resultados computacionais. A área de monitoramento é representada por um quadrado no plano e a comunicação é representada por um círculo de raio R . Esses parâmetros foram variados e para cada par (raio de comunicação \times área de monitoramento) de valores, geramos 33 instâncias diferentes, cada uma representando uma configuração inicial de uma RSSF.

Dividimos os testes em duas baterias: (1) A razão entre o Raio de Comunicação (R_c) e o Raio de Sensoriamento (R_s) é menor que o 3; (2) O Raio de Comunicação (R_c) é 3,9 vezes maior que o Raio de Comunicação (R_c).

6.2.1.1 Métricas

A métrica utilizada, nas duas baterias, foi a média do número de nós sensores obtido com o GRASP e com o GRASP+GCA.

Dado o aspecto aleatório do GRASP e do GCA foram feitas 30 execuções para cada instância e tirado a média das melhores soluções de cada instância.

6.2.1.2 Bateria 1 - $R_c \cong R_s$

A. Parâmetros

Considerando $u.d.$ a unidade de distância. Os parâmetros para a Bateria 1 são mostrados abaixo:

- Número de Sensores: 2500.
- Raio de Comunicação (R_c): $10u.d.$, $20u.d.$ e $30u.d.$.
- Raio de Sensoriamento (R_s): $15u.d.$, $25u.d.$ e $35u.d.$.
- Área de Monitoramento: $50u.d. \times 50u.d.$, $80u.d. \times 80u.d.$, $100u.d. \times 100u.d.$, $150u.d. \times 150u.d.$.

B. Resultados

Na Tabela 6.1 apresentamos os resultados computacionais para união dos Problemas de Cobertura e Agrupamento. Nas primeiras três colunas, indicamos a área de monitoramento, raio de comunicação e o raio de nó sensoriamento. Nas próximas duas colunas, apresentamos média aritmética do número de nós sensores ativos e o desvio padrão de 30 execuções do GRASP. Na sequência, apresentamos os resultados para combinação do GRASP com o GCA, destacando o número de grades exigido pelo GCA, a média dos nós sensores e o desvio padrão desta média.

Os resultados nos mostram que quando a razão entre o Raio de Comunicação e o Raio de Sensoriamento é menor que o 3, o numero de nós sensores resultante da união dos problemas é sempre superior, chegando a diferença de 1143 nós sensores, sendo 1156 nós sensores ativos no GRASP+GCA e, somente, 13 nós sensores ativos com o GRASP (Instância: Área 150×150 Raio de Comunicação: 10 Raio de Sensoriamento: 35). Isto acontece porque a estratégia garante que para cada grade teremos no mínimo um nó sensor ativo.

Tamanho da Área	Raio de Comunicação	Raio de Sensoriamento	GRASP		GRASP+GCA		
			Média de Sensores	Desvio Padrão	Número de Grades	Média de Sensores	Desvio Padrão
50	10	15	8,21	0,48	144	144,03	0,17
		25	3,88	0,33		144,00	0,00
		35	1,58	0,50		144,00	0,00
	20	15	7,91	0,58	36	36,06	0,24
		25	3,76	0,44		36,00	0,00
		35	1,58	0,50		36,00	0,00
	30	15	8,15	0,44	16	16,33	0,54
		25	3,88	0,33		16,00	0,00
		35	1,58	0,50		16,00	0,00
80	10	15	20,79	1,22	324	324,03	0,17
		25	7,73	0,45		324,00	0,00
		35	4,00	0,00		324,00	0,00
	20	15	20,76	0,79	81	81,24	0,56
		25	7,55	0,56		81,00	0,00
		35	4,00	0,00		81,00	0,00
	30	15	20,42	0,83	36	37,03	0,77
		25	7,61	0,50		36,06	0,24
		35	4,03	0,17		36,00	0,00
100	10	15	32,12	1,27	529	529,06	0,24
		25	11,76	0,90		529,00	0,00
		35	5,94	0,43		529,00	0,00
	20	15	32,76	1,17	144	144,76	0,79
		25	11,82	0,73		144,03	0,17
		35	6,00	0,50		144,00	0,00
	30	15	32,58	1,15	64	65,52	1,25
		25	12,06	0,61		64,06	0,24
		35	5,91	0,38		64,03	0,17
150	10	15	73,39	1,46	1156	1156,15	0,36
		25	26,91	1,10		1156,00	0,00
		35	13,73	0,67		1156,04	0,24
	20	15	76,93	1,60	289	289,11	0,16
		25	27,03	0,88		289,21	0,42
		35	13,61	0,93		289,00	0,00
	30	15	73,13	0,93	144	144,19	0,51
		25	27,06	1,20		144,39	0,79
		35	13,94	0,83		144,06	0,24

Tabela 6.1. Resultados para o GRASP+GCA, onde $R_c \cong R_s$.

Nos casos onde o Raio de Comunicação e o dobro do Raio de Sensoriamento (Raio de Comunicação: 30 Raio de Sensoriamento: 15), o número de nós sensores no GRASP+GCA é o dobro do número de nós sensores do GRASP.

Uma possível estratégia para diminuir a quantidade de nós sensores é considerar apenas as grades onde estão os nós sensores ativos pelo GRASP no algoritmo GCA. Desta forma garantiria a cobertura e formaria grupos só com as grades

ocupadas.

Outra estratégia é equilibrar o número mínimo de nós sensores com o número de grades. Este equilíbrio é feito modificando os valores de raio de comunicação e raio de nó sensoriamento. A bateria 2 apresenta esta estratégia.

6.2.1.3 Bateria 2 - $R_c \cong 3,9R_s$

A. Parâmetros

A bateria 2 utiliza a estratégia para equilibrar o número mínimo de nós sensores com o número de grades.

O número mínimo de nós sensores é:

$$\frac{A}{R_s^2\pi} \tag{6.1}$$

O número de grades na área:

$$\frac{A}{r^2} \tag{6.2}$$

Este equilíbrio é feito modificando os valores de raio de comunicação e raio de nó sensoriamento. Os valores dos raios são obtidos pela equação:

$$\frac{A}{R_s^2\pi} \cong \frac{A}{r^2} \rightarrow \frac{A}{R_s^2\pi} \cong \frac{A}{\frac{R_c^2}{5}} \rightarrow \tag{6.3}$$

$$R_s^2\pi \cong \frac{R_c^2}{5} \rightarrow R_c \cong R_s\sqrt{5\pi} \rightarrow \tag{6.4}$$

$$R_c \cong 3,9R_s \tag{6.5}$$

onde R_c e R_s são, respectivamente, o raio de comunicação e raio de nó sensoriamento, A é o tamanho da área de monitoramento e r é tamanho de uma grade. Dessa forma, foram criados instâncias com os seguintes parâmetros:

- Número de Sensores: 400.
- Área de Monitoramento: $50u.d. \times 50u.d.$, $80u.d. \times 80u.d.$, $100u.d. \times 100u.d.$, $150u.d. \times 150u.d.$.
- Raio de Sensoriamento \times Raio de Comunicação:

Raio de Sensoriamento	Raio de Comunicação
15u.d.	58,5u.d.
20u.d.	78,0u.d.
25u.d.	97,5u.d.
30u.d.	117,0u.d.

Tabela 6.2. Descrição dos Raios de Comunicação e Sensoriamento ($R_c \cong 3,9R_s$).

B. Resultados

Na Tabela 6.3 apresentamos os resultados computacionais para união dos Problemas de Cobertura e Agrupamento. Nas primeiras três colunas, indicamos a área de monitoramento, raio de sensoriamento e o raio de comunicação. Nas próximas duas colunas, apresentamos média aritmética do número de nós sensores e o desvio padrão de 30 execuções do GRASP. Na sequência, apresentamos os resultados para combinação do GRASP com o GCA, destacando o número de grades exigido pelo GCA, a média dos nós sensores resultante e o desvio padrão desta média.

Tamanho da Área	Raio de Sensoriamento	Raio de Comunicação	GRASP		GRASP+GCA		
			Média de Sensores	Desvio Padrão	Número de Grades	Média de Sensores	Desvio Padrão
50	15	58,5	8,24	0,56	4	8,24	0,56
	20	78,0	4,15	0,36	4	4,15	0,36
	25	97,5	4,00	0,00	4	4,03	0,17
	30	117,0	2,48	0,51	1	2,48	0,51
80	15	58,5	21,21	1,08	16	22,03	1,33
	20	78,0	12,09	0,63	9	12,27	0,72
	25	97,5	7,88	0,55	4	7,88	0,55
	30	117,0	5,09	0,58	4	5,09	0,58
100	15	58,5	32,82	1,07	16	32,82	1,07
	20	78,0	18,61	0,93	9	18,67	0,96
	25	97,5	12,12	0,78	9	12,30	0,92
	30	117,0	8,70	0,59	4	8,70	0,59
150	15	58,5	70,18	1,42	36	70,30	1,49
	20	78,0	41,30	1,36	25	41,79	1,52
	25	97,5	27,21	1,14	16	27,42	1,23
	30	117,0	19,09	1,07	9	19,09	1,07

Tabela 6.3. Resultados para o GCA+GRASP, onde $R_c \cong 3,9R_s$.

Os resultados nos mostram que, para as instâncias onde o $R_c \cong 3,9R_s$, o número de nós sensores permanece o mesmo após o algoritmo de agrupamento. Isto acontece, porque o número de grades é menor que o número de nós sensores necessários para cobertura. Ocasionalmente em grades contendo mais de um nó

sensor, o que, no primeiro conjunto de instâncias, acontece de maneira inversa.

Esta última estratégia não ativou nós desnecessários para a cobertura como aconteceu na primeira bateria.

6.3 Problema de Agrupamento + Problema de Roteamento

Quando analisamos o problema de agrupamento e roteamento juntos percebemos que a posição da grade que contém o líder é o que define a rota do nó sorvedouro. Então para diminuir o tamanho da rota é preciso aproximar do centro da área os líderes. Para isto, criamos duas heurísticas, uma para o agrupamento e outra para o roteamento do nó sorvedouro:

- *Greedy Clustering Algorithm - Border* (GCA-B) - similar ao GCA, mas evitamos que as grades que contém os líderes estejam localizadas em um canto ou em uma lateral da área de monitoramento.
- *Centralized Spatial Partitioning Grid - Border* (CSPG-B) - define a rota fixa do nó sorvedouro com base na estrutura de grades virtuais construída pelo GCA-B.

6.3.1 *Greedy Clustering Algorithm - Border*

Nesta seção, apresentamos o *Greedy Clustering Algorithm - Border* (GCA-B), uma variação do GCA. Quando analisamos o problema de agrupamento e roteamento juntos, percebemos que a posição da grade que contém o líder é o que define a rota que o nó sorvedouro (nó responsável por levar os dados coletados pelos nós sensores para fora da rede) deverá fazer, já que o nó sorvedouro deve percorrer a área a ser monitorada e recolher os dados com os líderes.

No GCA-B, nós forçamos alguns grupos a serem escolhidos com o objetivo de diminuir o tamanho da rota do nó sorvedouro. Nós evitamos que a grade central esteja localizada em um canto ou em uma lateral da área de monitoramento.

O primeiro passo do GCA-B é escolher os grupos dos cantos da área de monitoramento. Nesta abordagem, os primeiros grupos não têm cinco grades, isto

é feito para garantir que os líderes estejam localizados na segunda e na penúltima coluna. O algoritmo completa a lateral com grupos compostos de três grades em linha. No final, completamos a área com o mesmo algoritmo do GCA. A Figura 6.2 explica os passos do GCA-B. O 'x' representa a posição da grade central no grupo.

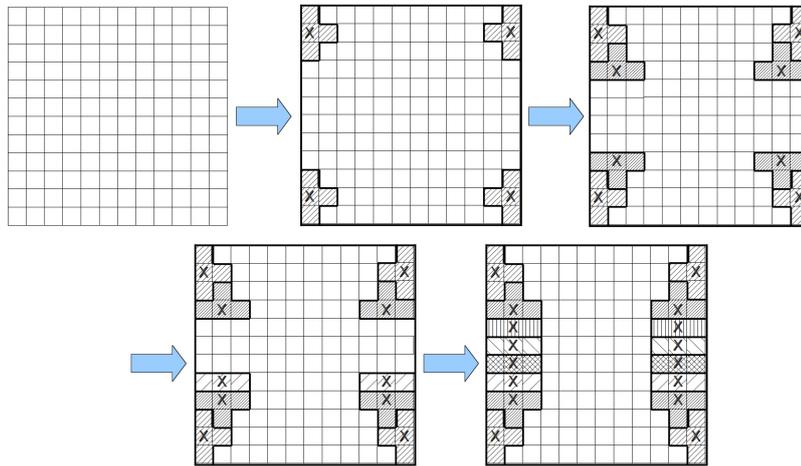


Figura 6.2. Passo a passo do *Greedy Clustering Algorithm - Border*.

6.3.2 *Centralized Spatial Partitioning Grid - Border*

A abordagem *Centralized Spatial Partitioning Grid - Border* (CSPG-B) utiliza a mesma ideia do CSPG, onde cada segmento é separado por uma distância de 4 grades do outro segmento. A diferença está entre as duas abordagens está em qual algoritmo de agrupamento é utilizado. A abordagem CSPG usa o algoritmo GCA (apresentado na Seção 4.2.1) para os agrupamentos, já abordagem CSPG-B utiliza o algoritmo GCA-B (apresentado na Seção 6.3.1).

No CSPG-B cada segmento horizontal inicia no final da segunda coluna de grades e termina no início da penúltima coluna, uma vez que é garantido que o líder do grupo não estará nos cantos e nem nas laterais da área de monitoramento. A Figura 6.3 mostra a rota do nó sorvedouro com o CSPG-B. Com esta abordagem, conseguimos reduzir o custo, para cada segmento, em $2r$ (onde r é o tamanho da grade). Isto demonstra que o custo da rota do nó sorvedouro pode se beneficiar da integração do problema de agrupamento e de roteamento. Diminuir a rota do

nó sorvedouro proporciona a redução do atraso e latência nos dados, no consumo de energia do nó sorvedouro, entre outros.

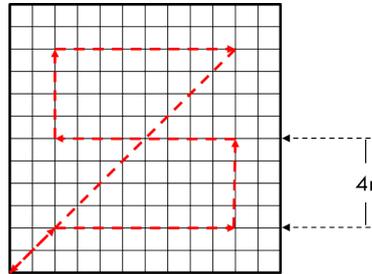


Figura 6.3. Rota gerada no *Centralized Spatial Partitioning Grid - Border*.

6.3.3 Resultados - Problema de Agrupamento + Problema de Roteamento

Esta seção apresenta nossas métricas e resultados computacionais. Foram utilizados as mesmas 33 instâncias descritas na Seção 4.3.

6.3.3.1 Métricas

A métrica para o Problema de Agrupamento é o menor número de grupos para o GCA e o GCA-B. Dado os aspectos aleatórios dos algoritmos, nós apresentamos a média aritmética e o desvio padrão das 30 execuções.

A métrica para o Problema de Roteamento é o tamanho da rota do nó sorvedouro, que inicia e retorna ao canto inferior esquerdo da área de monitoramento. O CSPG-B tem a rota fixa para cada uma das 33 instâncias, a rota depende, somente, do tamanho da área de monitoramento e do raio de comunicação.

6.3.3.2 Resultados

A Tabela 6.4 apresenta os resultados do GCA-B comparando com os limites inferiores (LI) e superiores (LS) do modelo matemático do Problema de Agrupamento e o GCA, apresentados no Capítulo 4.

GCA obteve um número de grupos menor que o GCA-B e ambos tiveram resultados próximos ao modelo matemático. A maior diferença absoluta entre as

Raio de Comunicação	Tamanho da Área (u.d.)	Modelo Matemático		GCA			GCA-B		
		LI	LS	Média	Desvio Padrão	GAP (%)	Média	Desvio Padrão	GAP (%)
7	30x30	24,00	24,00	31,45	1,12	23,69	33,83	1,42	29,06
	50x50	58,00	60,00	79,12	2,57	24,17	83,35	2,53	28,01
	80x80	143,43	155,00	207,82	5,42	25,42	215,12	6,67	27,95
	100x100	214,72	227,00	313,42	5,20	27,57	324,28	5,29	30,00
	150x150	474,20	533,00	703,82	8,60	24,27	703,27	8,22	24,21
	200x200	835,85	935,00	1244,33	10,29	24,86	1261,26	10,18	25,87
9	30x30	16,00	16,00	20,52	1,03	22,03	20,19	1,32	20,75
	50x50	40,00	40,00	52,58	2,40	23,93	53,95	2,28	25,86
	80x80	87,61	92,00	122,61	3,49	24,97	117,13	3,56	21,45
	100x100	133,09	143,00	192,76	4,04	25,81	194,91	4,42	26,63
	150x150	300,10	326,00	441,03	7,82	26,08	439,03	7,80	25,75
	200x200	513,98	589,00	760,24	9,30	22,52	786,31	9,84	25,09
13	30x30	10,00	10,00	11,97	0,59	16,46	12,79	0,50	21,81
	50x50	20,00	20,00	25,48	1,23	21,51	29,92	1,82	33,16
	80x80	47,00	47,00	61,24	2,37	23,25	63,83	2,39	26,37
	100x100	72,02	76,00	100,85	3,78	24,64	99,30	3,08	23,46
	150x150	143,43	154,00	206,82	5,05	25,54	218,98	5,51	29,67
	200x200	255,48	277,00	373,85	6,22	25,91	366,53	6,05	24,43

Tabela 6.4. Resultados para o GCA-B comparado com o GCA e o Modelo Matemático.

abordagens GCA e GCA-B é de 26 grupos ($R = 9$ e $200 \text{ u.d.} \times 200 \text{ u.d.}$), mas ela representa, somente, 3,32% de diferença relativa.

Dados que os algoritmos alcançaram resultados similares, ambos podem ser usados para resolver o problema de agrupamento.

A Figura 6.4 mostra os resultados para o CSPG-B comparando-o com o CSP e o CSPG, apresentados no Capítulo 5.

A abordagem CSPG-B alcança melhores resultados em todas as instâncias. Isto acontece porque a distância entre os segmentos é maior ($4r > 2R/\sqrt{2}$) que o CSP e também porque cada segmento reduz $2r$ (r tamanho da grade) no tamanho da rota do nó sorvedouro quando comparado com o CSPG.

Na Tabela 6.5, apresentamos as diferenças relativas entre as abordagens para o Problema de Agrupamento ($GCA-B - GCA / GCA-B$) e para o Problema de Roteamento ($CSPG-B - CSPG / CSPG-B$) para cada instância. A maior diferença relativa entre as abordagens GCA e GCA-B é de 14,84% ($R = 13$ e $50 \text{ u.d.} \times 50 \text{ u.d.}$) e a menor é de -4,68% ($R = 9$ e $80 \text{ u.d.} \times 80 \text{ u.d.}$). Já para as abordagens CSPG e CSPG-B, a maior diferença relativa é de 18,69% ($R = 13$ e $30 \text{ u.d.} \times 30 \text{ u.d.}$) e a menor diferença é de 2,81% ($R = 7$ e $200 \text{ u.d.} \times 200 \text{ u.d.}$).

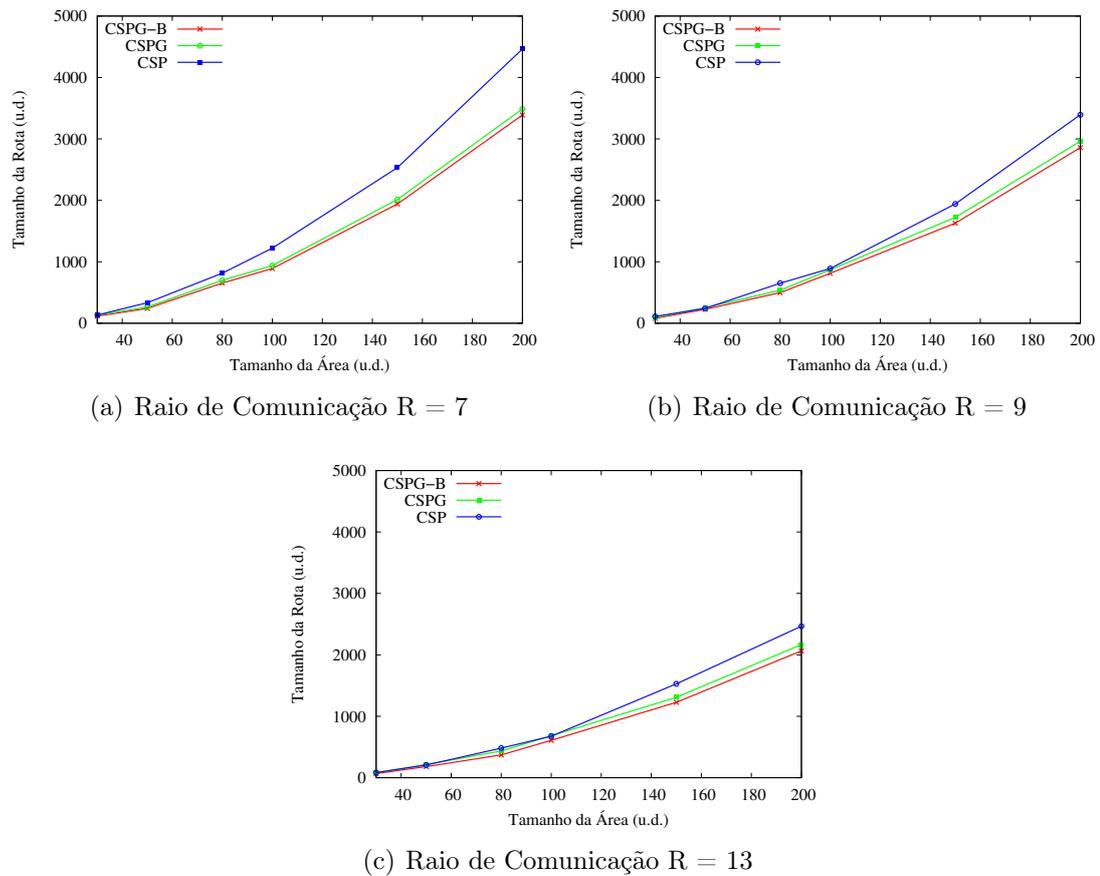


Figura 6.4. Tamanho da Área \times Tamanho da Rota - Algoritmos CSPG-B, CSPG e CSP.

Estas diferenças nos mostram, que quando a rota do nó sorvedouro móvel é fixa, o número de grupos não interfere no tamanho da rota do nó sorvedouro, mas em compensação quanto menor o número de grupos, menor o consumo de energia gasto.

Desta forma, a decisão de qual estratégia utilizar depende da aplicação sabendo que a está relacionada ao nó sorvedouro e o número de grupos a rede.

Dados os resultados do problema de roteamento, se a aplicação requer uma solução com o menor percurso, ele deve usar a combinação de GCA-B com CSPG-B para os problemas de agrupamento e roteamento, respectivamente.

Raio de Comunicação	Tamanho da Área (u.d.)	Diferença GCA ↔ GCA-B (%)	Diferença CSPG ↔ CSPG-B (%)
7	30x30	7,04	-13,86
	50x50	5,07	-8,67
	80x80	3,39	-6,22
	100x100	3,35	-5,11
	150x150	-0,08	-3,63
	200x200	1,34	-2,81
9	30x30	-1,63	-13,02
	50x50	2,54	-10,97
	80x80	-4,68	-7,53
	100x100	1,10	-6,49
	150x150	-0,46	-5,36
	200x200	3,32	-3,53
13	30x30	6,41	-18,69
	50x50	14,84	-15,79
	80x80	4,06	-14,79
	100x100	-1,56	-11,01
	150x150	5,55	-6,22
	200x200	-2,00	-4,77

Tabela 6.5. Diferenças relativas entre as abordagens: (GCA × GCA-B) e (CSPG × CSPG-B).

6.4 Considerações Finais

Neste capítulo apresentamos as estratégias para resolver o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM). O objetivo do PCAR-SM é encontrar o melhor subconjunto de sensores que cubra a área de monitoramento, agrupá-los no menor número de grupos e encontrar a melhor rota para um sorvedouro móvel percorrer. A união foi feita em duas partes primeiro avaliando os Problemas de Cobertura e Agrupamento, e em seguida os problemas de Agrupamento e Roteamento.

Criamos duas heurísticas considerando a união dos Problemas de Agrupamento e Roteamento: (1) o *Greedy Clustering Algorithm - Border* (GCA-B) para o agrupamento; e (2) o *Centralized Spatial Partitioning Grid - Border* (CSPG-B) para roteamento. As heurísticas, juntas, conseguiram reduzir a rota que o nó sorvedouro e mostrando que resolver os problemas em conjunto permite que algumas mudanças nos algoritmos levem a melhores resultados.

Já união dos Problemas de Cobertura e Roteamento só mostrou ser benéfica quando o raio de comunicação era aproximadamente 3,9 vezes maior que o raio

de sensoriamento. Em casos onde o raio de comunicação era menor ou muito próximo ao raio de sensoriamento, o problema de cobertura ativa um número de nós, porém o Problema de Agrupamento, para garantir um nó sensor por grade, ativa muitos nós e gerando um consumo de energia desnecessário na rede. Para estes casos propomos que seja executado o algoritmo de cobertura e que o algoritmo de agrupamento crie os grupos apenas com os grades ocupados.

No próximo capítulo apresentamos nossas conclusões, os trabalhos publicados durante a pesquisa e os trabalhos futuros.

Capítulo 7

Considerações Finais

7.1 Conclusões

Este trabalho mostrou a união de três problemas, que muitas vezes são vistos separadamente, o Problema de Cobertura, Agrupamento e Roteamento com Sorvedouro Móvel (PCAR-SM) em redes de nós sensores sem fio. O PCAR-SM consiste em dado um conjunto de nós sensores e uma área de monitoramento desenvolver algoritmos para encontrar o melhor subconjunto de nós sensores que cubra a área de monitoramento, juntá-los no menor número de grupos possível e encontrar a menor rota para um nó sorvedouro móvel percorrer.

A proposta deste trabalho foi resolver cada problema separadamente e, também, em conjunto e avaliar o impacto de cada problema na solução do outro.

O objetivo do Problema de Cobertura é encontrar o menor subconjunto de nós sensores que consiga cobrir a área de monitoramento. Este problema foi explorado com duas heurísticas, o Algoritmo Genético e o GRASP, este último com duas representações de solução: (a) representação por nós sensores, onde cada elemento do vetor de solução representa um nó sensor que deverá estar ligado ou desligado; (b) representação por demanda, onde cada elemento do vetor de solução representa um ponto de demanda no qual indicará qual o nó sensor que o cobre. O AG utilizou, somente, a representação por demanda, por causa do tempo gasto para executar todas as gerações ser muito grande (mais de 1 hora para as menores instâncias).

Cada meta-heurística foi comparada com um benchmark da *Beasley's OR Library* para o problema de cobertura. Os resultados mostraram que o GRASP com representação por demanda obteve desempenho melhor em relação ao valor da função objetivo e do tempo para cada caso de teste. Isto se dá pelo fato da representação forçar a escolha, para cada ponto de demanda descoberto, de um nó sensor com o custo baixo. Além disto, o uso de uma busca local na heurística incrementa a qualidade da solução. Apesar dos bons resultados, é notado que há espaço para melhoras, ainda é possível conseguir reduzir a distância entre a heurística e o modelo matemático.

O objetivo do Problema de Agrupamento é encontrar o menor número de grupos de nós sensores que estão, relativamente, próximos e contenha pelo menos um nó que consiga se comunicar com todos os outros nós do grupo, considerando que estes grupos não tenham sobreposição. Foi criada uma abordagem de grades virtuais, para o Problema de Agrupamento, onde divide-se a área em grades e os grupos foram compostos por um conjunto de grades adjacentes (no máximo 5 grades no grupo) formando um esquema de cruz. O objetivo do problema é minimizar o número de grupos.

A partir desta abordagem, pode-se modelar o Problema de Agrupamento como um Problema de Cobertura de Conjuntos sem sobreposição (um elemento não pertence a mais de um conjunto), e, ainda, uma heurística gulosa (*Greedy Clustering Algorithm*, GCA). O GCA encontrou bons resultados apesar de ainda ter espaço para melhoras. Apesar disto, a utilização das grades virtuais provou ser uma boa solução para o agrupamento por ser bastante simples para um nó, sabendo suas coordenadas, identificar a qual grade ele pertence.

O objetivo do Problema de Roteamento é encontrar uma rota para o nó sorvedouro móvel percorrer a área de monitoramento garantindo que o nó alcance todos os grupos. Modelado como o Problema do Caixeiro Viajante, o nó sorvedouro móvel parte de um canto da área de monitoramento, percorre a área e retorna ao ponto inicial. Para isto, propomos duas abordagens gulosas, baseadas no algoritmo do Vizinho Mais Próximo. Implementamos também, uma heurística baseada no algoritmo *Centralized Spatial Partitioning* (CSP). Na abordagem CSP, a rota é fixa e lembra o movimento de uma cobra. Os resultados mostram que a rota fixa gera um percurso com tamanho menor em comparação as heurísticas.

Analizamos, ainda, a união dos problemas de cobertura, roteamento e agrupamento, criando estratégias e heurísticas.

A união dos Problemas de Agrupamento e Roteamento foi feita com a criação de duas heurísticas: (1) o *Greedy Clustering Algorithm - Border* (GCA-B) para o agrupamento; e (2) o *Centralized Spatial Partitioning Grid - Border* (CSPG-B) para roteamento. As heurísticas, juntas, conseguiram reduzir a rota que o nó sorvedouro terá de fazer. Mostrando que resolver os problemas em conjunto permite que algumas mudanças nos algoritmos levem a melhores resultados.

Já união do Problema de Cobertura com o Problema de Agrupamento só mostrou ser benéfica quando o raio de comunicação era aproximadamente 3,9 vezes maior que o raio de nó sensoriamento. Apesar disto, nós sensores com raios de comunicação e nó sensoriamento próximos são mais utilizados pela literatura. Nestes casos é melhor optar por resolver a cobertura e formar grupos com as grades que contém nós sensores.

7.2 Trabalhos Publicados

Nesta seção são listados em ordem cronológica do mais recente ao mais antigo, os trabalhos publicados durante o mestrado.

Araújo, A. R. M., Peñaranda, A. G. and Nakamura, F. G. Uma Abordagem de Grids Virtual para Agrupamento e Roteamento em Redes de Sensores Sem Fio com Sorvedouro Móvel. XVI CLAIO (Congreso Latino-Iberoamericano de Investigación Operativa) / XLIV SBPO (Simpósio Brasileiro de Pesquisa Operacional). Rio de Janeiro, RJ, 2012.

Araújo, A. R. M., Peñaranda, A. G. and Nakamura, F. G. Avaliando o Impacto da Representação da Solução no Problema de Cobertura em Redes de Sensores Sem Fio. XVI CLAIO (Congreso Latino-Iberoamericano de Investigación Operativa) / XLIV SBPO (Simpósio Brasileiro de Pesquisa Operacional). Rio de Janeiro, RJ, 2012.

Peñaranda, A. G., Araújo, A. R. M. and Nakamura, F. G. Derivando Limites Inferiores para o Consumo de Energia em Redes de Sensores Sem Fio Planas. XVI CLAIO (Congreso Latino-Iberoamericano de Investigación Operativa) / XLIV SBPO (Simpósio Brasileiro de Pesquisa Operacional). Rio de Janeiro, RJ, 2012.

Araújo, A. R. M., Peñaranda, A. G., Nakamura, F. G., & Rodrigues, R. D. F. Heurística para o problema de cobertura em redes de nós sensores sem fio voltado ao rastreamento de animais. XLIII SBPO (Simpósio Brasileiro de Pesquisa Operacional). Ubatuba, São Paulo, 2011.

Peñaranda, A. G., Araújo, A., Nakamura, F. G., and Rodrigues, R. D. F. Um algoritmo grasp para o problema de k-cobertura em redes de nós sensores sem fio. XLIII SBPO (Simpósio Brasileiro de Pesquisa Operacional). Ubatuba, São Paulo, 2011.

7.3 Trabalhos Futuros

Como trabalho futuro para o problema de cobertura, pretende-se implementar outras heurísticas e ainda combiná-las com a heurística GRASP com representação por demanda descrita neste trabalho.

Para o problema de agrupamento, apesar de que o esquema de grades tenha tido bons resultados, deseja-se desenvolver outros esquemas para configurações dos grupos, por exemplo, o esquema de hexágonos, onde os nós sensores tem correlação espacial maior.

Para o problema de roteamento, deseja-se executar experimentos com outros cenários, considerando área com obstáculos ou com buracos (partes da área que não precisam ser visitadas). E, ainda, investigar se com estes cenários, é melhor usar uma rota fixa ou uma rota que varia de acordo com a disposição dos grupos.

Aplicar novas estratégias na união dos problemas de cobertura e agrupamento para facilitar a formação dos grupos e com isso diminuir o tamanho da rota do nó sorvedouro e número de nós sensores necessários para serem usados na rede.

Por fim, planeja-se implementar uma versão de distribuída dos algoritmos já implementados para melhor atender às redes de nós sensores sem fio.

Referências

- Abraham, A. & Jain, L. (2005). Evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization*, pp. 1--6.
- Aioffi, W. & Mateus, G. (2007). Otimização do controle de densidade com servidores móveis em redes de sensores. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 323--336.
- Aioffi, W. M.; Quintão, F. P. & Mateus, G. R. (2007). Integrated methods for organization of wireless sensor networks with mobile sink. *International Workshop on Design and Reliable Communication Networks*, pp. 1--8.
- Aioffi, W. M.; Valle, C. A.; Mateus, G. R. & da Cunha, A. S. (2011). Balancing message delivery latency and network lifetime through an integrated model for clustering and routing in wireless sensor networks. *Computer Networks*, pp. 2803--2820.
- Andrade, I. B. D. & Mateus, G. R. (2008). Uma abordagem multi-período para a solução do problema de cobertura e conectividade em redes de sensores sem fio planas. *CTIC da Sociedade Brasileira de Computação*.
- 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. Em *Symposium on Computers and Communications*, pp. 493--499.
- Araújo, A. R. M.; Peñaranda, A. G.; Nakamura, F. G. & Rodrigues, R. D. F. (2011). Heurística para o problema de cobertura em redes de sensores sem

- fio voltado ao rastreamento de animais. *XLIII SBPO (Simpósio Brasileiro de Pesquisa Operacional)*.
- Bagnall, A.; Rayward-Smith, V. & Whittle, I. (2001). The next release problem. *Information and Software Technology*, pp. 883--890.
- Balas, E. & Ho, A. (1980). Set covering algorithms using cutting planes, heuristics, and subgradient optimization: A computational study. Em *Combinatorial Optimization*, volume 12 of *Mathematical Programming Studies*, pp. 37--60.
- Beasley, J. E. (1990). A lagrangian heuristic for set-covering problems. *Naval Research Logistics (NRL)*, pp. 151--164.
- Boukerche, A.; Oliveira, H.; Nakamura, E. & Loureiro, A. A. F. (2009). Dv-loc: a scalable localization protocol using voronoi diagrams for wireless sensor networks. *IEEE Wireless Communications*, pp. 50--55.
- Caprara, A.; Fischetti, M. & Toth, P. (1998). Algorithms for the set covering problem. *Annals of Operations Research*, p. 2000.
- Chatzigiannakis, I.; Kinalis, A. & Nikolettseas, S. (2008). Efficient data propagation strategies in wireless sensor networks using a single mobile sink. *Computer Communications*, pp. 896--914.
- Chatzigiannakis, I.; Kinalis, A.; Nikolettseas, S. & Rolim, J. (2007). Fast and energy efficient sensor data collection by multiple mobile sinks. Em *Proceedings of the 5th ACM international workshop on Mobility management and wireless access*, pp. 25--32.
- Chen, J.-L.; Chen, M.-C.; Huang, P.-Y. & Chang, Y.-C. (2007). Cluster-grid structure routing protocol for sensor mobility management. Em *Sarnoff Symposium, 2007 IEEE*, pp. 1--5.
- Cormen, T. H.; Stein, C.; Rivest, R. L. & Leiserson, C. E. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edição.
- Dantzig, G.; Fulkerson, R. & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Operations Research*, pp. 393--410.

- Dias-Neto, A.; de Freitas Rodrigues, R. & Travassos, G. (2011). Porantim-opt: Optimizing the combined selection of model-based testing techniques. Em *Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on*, pp. 174--183.
- Documentation, I. (2012). ILOG CPLEX optimization studio information center. Disponível em: <http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r4/index.jsp>. Acessado em: Maio de 2012.
- Feo, T. A. & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67--71.
- Garey, M. R. & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman and Company, New York.
- Guanzhou Lu e Bahsoon, R. e. X. Y. (2010). Applying elementary landscape analysis to search-based software engineering. Em *Search Based Software Engineering (SSBSE), 2010 Second International Symposium on*, pp. 3--8.
- Gutin, G.; Yeo, A. & Zverovich, A. (2002). Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the tsp. *Discrete Applied Mathematics*, pp. 81--86.
- Harman, M. (2007). The current state and future of search based software engineering. Em *Future of Software Engineering, 2007. FOSE '07*, pp. 342--357.
- Heinzelman, W. R.; Chandrakasan, A. & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. Em *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8, HICSS '00*, pp. 8020--8030.
- Holland, J. (1975). Adaptation in artificial and natural systems. *Ann Arbor: The University of Michigan Press*.
- Huang, C.-F. & Tseng, Y.-C. (2003). The coverage problem in a wireless sensor network. Em *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pp. 115--121.

- Johnson, D. S. (1973). Approximation algorithms for combinatorial problems. Em *Proceedings of the fifth annual ACM symposium on Theory of computing*, pp. 38--49. ACM.
- Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pp. 85--103.
- Loureiro, A.; Nogueira, J.; Ruiz, L.; Mini, R.; Nakamura, E. & Figueiredo, C. (2002). Rede de sensores sem fio. *Simpósio Brasileiro de Computação - Jornada de Atualização da Informática*, pp. 179--226.
- Nakamura, F.; Quintão, F.; Menezes, G. & Mateus, G. (2005). An optimal node scheduling for flat wireless sensor networks. Em *Networking - ICN 2005*, pp. 475--482.
- Nakayama, H.; Ansari, N.; Jamalipour, A. & Kato, N. (2007). Fault-resilient sensing in wireless sensor networks. *Computer Communications*, pp. 2375--2384.
- Niculescu, D. & Nath, B. (2001). Ad hoc positioning system (aps). Em *In Proceedings of the IEEE Global Telecommunications Conference*.
- Noronha, T.; Silva, M. & Aloise, D. (2001). Uma abordagem sobre estratégias metaheurísticas. *Revista Eletrônica de Iniciação Científica (REIC) Ano I*.
- Reeves, C. R., editor (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., New York, NY, USA.
- Resende, M. & Ribeiro, C. (2003). Grasp with path-relinking. *Networks*, pp. 104--114.
- Rodrigues, R.; Pessoa, A.; Uchoa, E. & Poggi de Aragão, M. (2008). Heuristic algorithm for the parallel machine total weighted tardiness scheduling problem. Disponível em: www.producao.uff.br/conteudo/rpep/volume82008/RelPesq_V8_2008_10.pdf Acessado em: Novembro de 2012.

- Turgut, D.; Das, S.; Elmasri, R. & B., T. (2002). Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach. *Global Telecommunications Conference (GLOBECOM)*, pp. 62--66.
- Villas, L. A.; Guidoni, D. L.; Araujo, R. B.; Boukerche, A. & Loureiro, A. A. F. (2011). Explorando a correlação espacial na coleta de dados em redes de sensores sem fio. *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 425--438.
- Xu, Y.; Heidemann, J. & Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing. Em *International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 70--84.