

**ATRIBUIÇÃO DE PAPÉIS EM REDE DE
SENSORES SEM FIO BASEADA EM
COLORAÇÃO DE VÉRTICES**

LUCAS MIDDELDORF RIZZO

**ATRIBUIÇÃO DE PAPÉIS EM REDE DE
SENSORES SEM FIO BASEADA EM
COLORAÇÃO DE VÉRTICES**

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

ORIENTADOR: SEBASTIÁN ALBERTO URRUTIA

Belo Horizonte

Março de 2013

© 2013, Lucas Middeldorf Rizzo.
Todos os direitos reservados.

R627a Rizzo, Lucas Middeldorf
Atribuição de Papéis em Rede de Sensores sem Fio
Baseada em Coloração de Vértices / Lucas Middeldorf
Rizzo. — Belo Horizonte, 2013
xx, 50 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Sebastián Alberto Urrutia

1. Computação - Teses. 2. Sistemas Distribuídos -
Teses. 3. Redes de sensores sem fio - Teses. I. Título.

CDU 519.6*22 (043)



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

FOLHA DE APROVAÇÃO

Atribuição de papéis em redes de sensores sem fio baseada em coloração de vértices

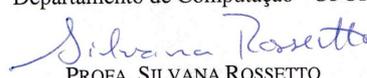
LUCAS MIDDELDORF RIZZO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. SEBASTIÁN ALBERTO URRUTIA - Orientador
Departamento de Ciência da Computação - UFMG


PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Coorientador
Departamento de Ciência da Computação - UFMG


PROF. RICARDO AUGUSTO RABELO OLIVEIRA
Departamento de Computação - UFOP


PROFA. SILVANA ROSSETTO
Departamento de Ciência da Computação - UFRJ

Belo Horizonte, 26 de março de 2013.

Agradecimentos

Agradeço à minha família pelo incentivo e oportunidade de continuar estudando. Em particular à minha mãe pelo exemplo de persistência e força de vontade para jamais desistir e à minha avó pelo carinho e atenção sempre disponíveis.

Ao meu professor e orientador, Sebastián Urrutia, pelo incentivo de realizar este curso, confiança, ensinamentos e disposição em ajudar a qualquer momento.

Ao meu coorientador, Prof. Antonio Alfredo F. Loureiro, pela paciência, compreensão, tempo disponível e por acreditar em minha capacidade.

Agradeço aos meus amigos Sara e Luiz pela amizade e companheirismo neste mundo no qual passamos a maior parte do tempo lidando e trabalhando com máquinas.

Aos demais familiares que torceram pelo meu sucesso João Carlos, Sophia Clara, Ewald, Lúcia e Lívia.

Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Universidade Federal de Minas Gerais pela oportunidade, apoio financeiro e infraestrutura fornecidos durante todo o curso.

Resumo

Devido à tecnologia disponível atualmente, viabilizou-se a utilização de amplas e densas Redes de Sensores sem Fio (RSSFs). Um dos principais problemas deste tipo de rede é a otimização na distribuição das funções atribuídas a cada sensor. Basicamente qualquer sensor em uma RSSF requer algum tipo de auto-configuração, onde são atribuídos papéis (funções) a cada um deles sem que haja alguma intervenção manual.

Este trabalho propõe e avalia uma abordagem heurística, com versões centralizadas e distribuídas, para o problema de atribuição de papéis. Tal abordagem é realizada através de uma relação com o problema de coloração de vértices. A avaliação é feita utilizando-se o simulador de redes [Sinalgo, 2007]. São considerados a geração de eventos no campo de sensoriamento, e tem-se como objetivo a economia de energia na detecção e notificação destes eventos. Para economizar energia, a rede deve ser capaz de utilizar o menor número possível de sensores na detecção de eventos além de encaminhar as informações obtidas pela melhor rota possível com o nó sorvedouro. São apresentados resultados computacionais que comprovam a eficiência do método abordado através de métricas como tempo de execução, porcentagem de eventos não sensorizados e número de mensagens enviadas por evento.

Palavras-chave: Atribuição de Papéis, Sistemas Distribuídos, Coloração de Vértices.

Abstract

The technology available nowadays has enabled the use of large and dense wireless sensor networks (WSNs). One of the main problems in these networks is the optimized assignment of functions to be performed by each sensor. Basically any sensor in a WSN requires some sort of self-configuration, in which roles (functions) are assigned to each node with no manual intervention.

This work proposes and evaluates a heuristic approach, based on centralized and distributed versions, for the role assignment problem. Such approach is based on a relationship between this problem and the vertex coloring problem. The performance evaluation is accomplished with the network simulator [Sinalgo, 2007]. It is considered the event generation in the sensor field, and the main goal is to save energy in the detection and notification of these events. In this scenario, the network should use the smallest number of sensors in the event detection and send notifications to the sink node using the best route. Simulation results confirm the effectiveness of the proposed method through metrics such as execution time, percentage of not sensed events and number of sent messages for each event.

Keywords: Role Assignment, Distributed Systems, Vertex Coloring Problem.

Lista de Figuras

3.1	Exemplo de coloração ótima encontrada pela coloração na ordem indicada pelos números em cada vértice	16
3.2	Exemplo de coloração não ótima encontrada pela coloração na ordem indicada pelos números em cada vértice	16
3.3	Sensores 1, 2 e 3 realizando a detecção de eventos de temperatura. Suponha que as prioridades dos sensores 1, 2 e 3 sejam 10, 20 e 30 respectivamente. Como os eventos T são gerados simultaneamente os vértices tentam realizar o sensoriamento ao mesmo tempo, definem seus papéis como T e informam os vizinhos. O sensor 3 recebe a informação do sensor 2 de que este já assumiu o papel T e o sensor 2 recebe a informação do sensor 1 de que este também já assumiu o papel T . Como o sensor 1 é o de menor prioridade ele é o único que manterá o papel, fazendo com que o sensoriamento do evento T_2 seja perdido	23
4.1	Distribuições dos sensores pela área de sensoriamento. (a) regular, (b) aleatória, (c) regular com reposicionamento de sensores para perto do nó sorvedouro, (d) aleatória com maior probabilidade de sensores perto do nó sorvedouro. O nó sorvedouro é representado pela cor vermelha.	31
5.1	Simulação centralizada: gráficos comparativos entre a porcentagem de eventos não sensoriados e o número de pontos de eventos.	35
5.2	Simulação distribuída: gráficos comparativos entre a porcentagem de eventos não sensoriados e o número de pontos de eventos.	36
5.3	Simulação centralizada: gráficos comparativos entre o número de <i>rounds</i> e o número de pontos de eventos.	37
5.4	Simulação distribuída: gráficos comparativos entre a o número de <i>rounds</i> da simulação e o número de pontos de eventos.	38

5.5	Simulação centralizada: gráficos comparativos entre o número de mensagens enviadas por evento e o número de pontos de eventos.	41
5.6	Simulação distribuída: gráficos comparativos entre o número de mensagens enviadas por evento e o número de pontos de eventos.	42
5.7	Simulação aleatória: comparação da porcentagem de eventos não sensoria- dos com 2 tipos de eventos	43
5.8	Simulação aleatória: comparação da porcentagem de eventos não sensoria- dos com 3 tipos de eventos	43
5.9	Simulação aleatória: comparação da porcentagem de eventos não sensoria- dos com 4 tipos de eventos	43
5.10	Simulação aleatória: comparação do tempo de vida com 2 tipos de eventos	44
5.11	Simulação aleatória: comparação do tempo de vida com 3 tipos de eventos	44
5.12	Simulação aleatória: comparação do tempo de vida com 4 tipos de eventos	45

Lista de Tabelas

3.1	Tipos de mensagens utilizados na configuração inicial da rede	21
4.1	Cálculo dos critérios de cada sensor	27
4.2	Pesos adotados	27
4.3	Cenário padrão	30
5.1	Número de eventos gerados de acordo com o número de pontos de eventos. Erros foram calculados com 95% de confiança	39

Lista de Algoritmos

1	Algoritmo Genético	10
2	Detecção de Eventos	12
3	Algoritmo Coloração de Vértices	17
4	Função de Prioridade	19
5	Atribuição de papéis centralizada	20
6	Sensoriamento	22
7	Recebe mensagem	22

Sumário

Agradecimentos	vii
Resumo	ix
Abstract	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Objetivos e Resultados desta Dissertação	2
1.2 Problema de Atribuição de Papéis em RSSF	3
1.3 Organização do Texto	4
2 Trabalhos Relacionados	7
2.1 Heurísticas e Algoritmos Distribuídos	7
2.1.1 Algoritmos distribuídos e atribuição de papéis de forma genérica	7
2.1.2 Heurísticas e atribuição de papéis reativa	9
2.1.3 Algoritmos distribuídos e atribuição de papéis reativa	11
2.2 Algoritmos Exatos	12
2.2.1 Limites para o tempo de vida da RSSF	12
2.2.2 Atribuição de papéis de forma genérica e critérios de otimização globais	13
2.3 Análise	13
3 Modelagem	15
3.1 Problema de Coloração de Vértices	15
3.1.1 Caso Particular do Problema de Atribuição de Papéis	15
3.1.2 Heurísticas Sequenciais	16

3.2	Mapeamento para o Problema de Atribuição de Papéis	17
3.2.1	Roteamento	19
3.3	Algoritmo Centralizado	19
3.4	Algoritmo Distribuído	20
3.4.1	Prioridade	20
3.4.2	Sensoriamento	22
3.4.3	Perda de sensores	23
4	Experimentos Computacionais	25
4.1	Simulação Centralizada	25
4.1.1	Iteração Básica	25
4.1.2	Definição da prioridade	26
4.2	Simulação Distribuída	28
4.2.1	Iteração Básica	28
4.2.2	Definição da prioridade	29
4.3	Cenário Padrão	29
4.4	Distribuição dos sensores	30
4.5	Métricas de comparação	30
5	Análise dos Resultados	33
5.1	Eventos Não Sensoriados	33
5.2	Tempo de Vida	34
5.3	Mensagens Enviadas por Evento	39
5.4	Coloração de Vértices Aleatória (CVA)	40
5.4.1	Comparação: eventos não sensoriados	40
5.4.2	Comparação: tempo de vida	44
5.4.3	Considerações	45
5.5	Distribuições dos sensores na região de interesse	45
6	Conclusão	47
	Referências Bibliográficas	49

Capítulo 1

Introdução

Redes de sensores sem fio (RSSFs) são projetadas para serem implantadas em ambientes físicos a fim de monitorar uma grande variedade de fenômenos do mundo real. Segundo Akyildiz et al. [2002] e Loureiro et al. [2003] algumas das áreas de aplicação para RSSFs são ambiental, saúde, militar e segurança. Na área ambiental, diferentes variáveis como temperatura, umidade relativa do ar, luminosidade e concentração de gases podem ser monitoradas continuamente em uma região urbana, por exemplo, para as pessoas saberem as condições que vão encontrar ao sair de casa. Na área da saúde pode-se realizar o monitoramento de órgãos, a fim de observar seus funcionamentos, ou a presença de substâncias indesejadas no organismo. Já a área militar apresenta aplicações como observação de forças amigas, equipamento e munição, vigilância do campo de batalha e reconhecimento do terreno. Na área de segurança, sensores podem ajudar no controle de roubos de carros (informando pela Internet ameaças ao usuário).

Devido à tecnologia disponível atualmente, viabilizou-se a utilização de amplas e densas RSSFs. Desta forma, torna-se necessária a otimização na distribuição das funções atribuídas à cada sensor. Para isto apresentam-se na literatura diversos algoritmos de atribuição de papéis, onde cada papel representa uma destas funções [Frank & Römer, 2005; Odell et al., 2003; Bhardwaj & Chandrakasan, 2002; Nakamura et al., 2009].

Os sensores em uma RSSF requerem algum tipo de auto-configuração, onde são atribuídos papéis a cada um deles sem que haja alguma intervenção manual. Um exemplo clássico seria o problema de cobertura, onde deseja-se, por exemplo, realizar a leitura de temperatura de uma determinada área de forma que dois sensores não cubram uma mesma região. Assim, de acordo com a posição de cada sensor, torna-se necessário saber quais sensores devem realizar o monitoramento em cada instante, evitando o gasto desnecessário de energia.

Citamos como aplicação para este trabalho o problema de atribuição de funções de sensoriamento [Souza, 2007], no qual a cada momento deseja-se que um tipo de coleta de informações seja realizada pelos sensores da rede de forma a atender às necessidades da aplicação naquele instante. Alguns dos tipos de dados que podem ser coletados são temperatura, imagem, áudio, vídeo e pressão. Além disso, o problema exige que uma rota entre os sensores seja construída conjuntamente com a atribuição de papéis para encaminhamento de dados a uma base de energia ilimitada.

Para resolução do problema veremos que existe uma relação entre os problemas de atribuição de papéis e o problema clássico de coloração de vértices [Wigderson, 1983]. Mapeando-se a RSSF em um grafo $G(V, E)$, onde os sensores são representados pelo conjunto de vértices V e as conexões entre os sensores pelas arestas do conjunto E , vemos que ambos os problemas objetivam a separação de seus elementos, vértices ou sensores, em conjuntos distintos, seja pela atribuição de cores ou papéis. Mais do que isso, é possível mostrar que o problema abordado é uma extensão do problema de coloração de vértices, o que implica que diversos algoritmos já estudados podem ser adaptados ao primeiro. Mais detalhes sobre esta demonstração e os algoritmos utilizados serão apresentados nos próximos capítulos.

1.1 Objetivos e Resultados desta Dissertação

Nesta dissertação estudamos o problema de atribuição de papéis e tem-se como objetivo apresentar um algoritmo para a aplicação de funções de sensoriamento. Duas versões deste algoritmo são projetadas: uma centralizada e outra distribuída. Além disso, avaliamos o algoritmo em diversos contextos, seja variando-se a quantidade de tipos de eventos a serem monitorados, posicionamento dos sensores na região de interesse ou a quantidade de eventos gerados.

A maior contribuição deste trabalho é um algoritmo que determina as funções de cada sensor em cada instante e, para onde esta informação deve ser enviada. Até onde se sabe nenhum outro trabalho realiza uma abordagem para o problema de atribuição de papéis através do problema de coloração de vértices. Além disso, diferentemente de outros trabalhos, tratamos a atribuição de funções de sensoriamento para uma grande carga de eventos em cada instante da simulação. Com isso mostramos que nossa solução é bastante robusta, mantendo a conectividade da rede, o baixo uso de energia e uma pequena taxa de eventos não sensoriados.

1.2 Problema de Atribuição de Papéis em RSSF

Redes de sensores sem fio são limitadas em termos de energia disponível e capacidade de processamento. Em contrapartida existe a necessidade de sua auto-configuração em função das condições dinâmicas que podem acontecer ao longo do tempo em uma rede, já que o elevado número de sensores impede a configuração manual dos nós sensores. Desta forma define-se o problema de atribuição de papéis como a distribuição das funções a serem realizadas por cada sensor, atendendo-se às necessidades de monitoramento, comunicação e preservação de energia, sem que haja nenhum tipo de intervenção manual.

Dois dos papéis mais básicos em redes de sensores são comunicação e sensoriamento. A comunicação de dados é a tarefa responsável por enviar dados entre nós adjacentes até que a mensagem de interesse seja entregue ao nó sorvedouro (*sink*). Neste caso, a atribuição de papéis tem como objetivo estabelecer uma topologia que permita essa comunicação. O sensoriamento é a capacidade de coletar dados de uma dada grandeza física de interesse da aplicação. Cada sensor tem um “raio de sensoriamento”, ou seja, é capaz de detectar um dado de interesse desde que ele seja observável dentro desse raio. Neste caso, a atribuição de papéis tem como objetivo estabelecer uma topologia que permita esse sensoriamento de forma adequada. Nesses dois casos (comunicação e sensoriamento), é importante observar que a atribuição de papéis deve ser feita considerando, por exemplo, a energia disponível nos possíveis nós candidatos que terão essas atribuições.

Neste trabalho será tratado o problema de atribuição de funções de sensoriamento. Desta forma, deseja-se atribuir a cada sensor um tipo de sensoriamento, de acordo com a ocorrência de determinados eventos na região de interesse, garantindo também o roteamento de informações para os nós sorvedouros. Os raios de comunicação e sensoriamento são considerados iguais. Para definição formal do problema sejam as seguintes notações:

- $S = \{s_1, s_2, \dots, s_{n_1}\}$, conjunto de nós sensores;
- $P = \{p_1, p_2, \dots, p_{n_2}\}$, conjunto de possíveis papéis a serem atribuídos;
- $B = \{b_1, b_2, \dots, b_{n_3}\}$, conjunto de nós sorvedouros;
- R = região de sensoriamento da RSSF;
- e_i^p = energia gasta pelo sensor i ao realizar o papel p ;

- m_i^j = energia para encaminhamento de informações do sensor $i \in S$ ao sensor ou nó sorvedouro $j \in S \cup B$;
- D = conjunto de todos os fluxos de dados produzidos pelos nós sensores.

Para que a atribuição de papéis possa ser realizada é necessário que tenhamos um conhecimento do estado de cada sensor. Este estado é definido de acordo com a aplicação, podendo conter informações como energia residual, localização e número de sensores adjacentes. Seja x_i o estado do sensor $s_i \in S$. Assim, o estado da rede é definido como $X = \bigcup_{s_i \in S} x_i$. Logo uma atribuição de papéis é dada formalmente como um função sobrejetora $f : S \times D \times X \rightarrow P$.

Dado que cada sensor $s \in S$ pode assumir um único papel $p \in P$ em cada instante, o problema consiste em encontrar uma atribuição que maximize o tempo de vida da rede minimizando o número de eventos não sensoriados. Segundo Bhardwaj & Chandrakasan [2002], uma possível definição para o tempo de vida da rede é: limite de tempo até que os sensores estejam inativos (sem energia) ou que o sensoriamento realizado pelos sensores ativos não garanta mais as restrições de qualidade determinadas pelo usuário. Além disso, uma rota deve ser construída conjuntamente com a atribuição de papéis para que cada sensor $s \in S$ possa encaminhar as informações obtidas para o nó sorvedouro $b \in B$ mais próximo, impactando também no gasto de energia da rede.

Diversos fatores podem ser citados na redução do gasto de energia como, por exemplo, menores rotas de encaminhamento de informações, análise de redundância entre os sensoriamentos realizados (se dois sensores não estão sensoriando o mesmo evento), sensoriamento do maior número de eventos com o menor número de sensores possível, entre outros.

1.3 Organização do Texto

Este trabalho está organizado da forma a seguir. O capítulo 2 apresenta os trabalhos relacionados. Dentre a revisão bibliográfica mostramos as principais aplicações e tipos de algoritmos para o problema, além de uma análise dos trabalhos estudados. O capítulo 3 apresenta a definição do problema de coloração de vértices e sua modelagem no problema de atribuição de papéis, levando ao projeto do algoritmo centralizado e, em seguida, sua adaptação para uma versão distribuída. O capítulo 4 detalha os experimentos computacionais realizados, apresentando a estrutura das simulações executadas, parâmetros adotados e métricas de avaliação. Já o capítulo 5 apresenta uma análise dos resultados obtidos, levando-se em conta as principais vantagens e desvan-

tagens de cada cenário. Por fim, concluímos o trabalho no capítulo 6 juntamente com propostas para trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Diversos trabalhos apresentam diferentes tipos de abordagens para resolução do problema de atribuição de papéis. Dentre os trabalhos estudados apresentamos duas categorias principais: “Heurísticas e Algoritmos Distribuídos” e “Algoritmos Exatos”. Ao fim da seção mostra-se também uma análise que relaciona as referências estudadas e as contribuições deste trabalho sobre os demais apresentados.

2.1 Heurísticas e Algoritmos Distribuídos

2.1.1 Algoritmos distribuídos e atribuição de papéis de forma genérica

Frank & Römer [2005] apresentam a atribuição de papéis de forma genérica baseada nas propriedades de cada sensor e nos nós de sua vizinhança. Para isso o problema é proposto como uma abstração de programação, de tal forma que, através de uma linguagem de configuração, regras e papéis possam ser facilmente especificados para qualquer problema de atribuição.

Para construção de um algoritmo distribuído, Frank & Römer [2005] estudam três aplicações clássicas do problema de atribuição de papéis: cobertura, clusterização e agregação. Em todas elas considera-se que os sensores possuem mesmo raio de comunicação e sensoriamento e mesma configuração de hardware. Entretanto, diferentes configurações de hardware podem ser analisadas facilmente em termos de propriedades locais de cada sensor, não influenciando no resultado final.

- Cobertura: em uma área de análise da rede se sensores sem fio espera-se cada espaço físico esteja sob a observação de pelo menos um sensor. Quando o número

de sensores é elevado um mesmo espaço pode estar coberto por mais de um sensor. O objetivo é atribuir os papéis *ligado* e *desligado* de forma que em cada momento use-se o menor número de sensores *ligados* possíveis e toda a área esteja coberta.

- Clusterização: consiste em uma técnica comum para melhora da eficiência de entrega de dados em uma rede. Nela três papéis devem ser atribuídos, *clusterhead*, *gateway* e *escravo*. Um *clusterhead* atua com um hub para escravos em sua vizinhança, de forma que escravos se comuniquem apenas com seu *clusterhead*. *Gateways* são escravos de mais de um *clusterhead*, conectando múltiplos *clusterheads* transmitindo mensagens entre eles. *Clusterheads* devem, idealmente, possuir dispositivos mais poderosos, em termo de processamento, memória, energia e etc., já que atuam como hub para seus escravos.
- Agregação: consiste em uma técnica cujo objetivo é a redução da transmissão de dados entre sensores da rede sem fio, devido à escassez e alto custo de energia na comunicação sem fio. Nela certos sensores agregam dados de várias fontes. Para isso é necessário atribuir os papéis, *fonte*, *agregador* e *consumidor*. Para reduzir o tráfego de dados, *agregadores* devem se localizar próximos de fontes cujos dados são agregados.

O algoritmo distribuído proposto por Frank & Römer [2005] consiste no agendamento constante, para cada sensor, das seguintes etapas: inicialização de dados, propagação de informações, atribuição de papel e detecção de falhas. Caso a rede alcance certo tempo de estabilidade, ou seja, caso haja um determinado tempo sem mudança de papéis, a simulação é finalizada. A seguir são detalhadas cada uma das etapas realizadas por cada sensor.

- Inicialização de dados: todos os sensores devem determinar seu papel inicial ao começo da simulação. Esse pode ser indefinido, sem papel, ou atribuído probabilisticamente. Para atribuição probabilística avaliam-se fatores como densidade da rede, características de cada papel e etc. Com isso cada sensor é inicializado com o papel mais provável. Caso um sensor seja inserido posteriormente na rede, deve ser inicializado como os demais, além de enviar uma mensagem de atualização para seus vizinhos com suas propriedades relevantes.
- Propagação de informações: são enviadas propriedades locais e que ainda não foram propagadas. Sensores que recebem essa mensagem devem mantê-la em seu diretório local. Ao receber a primeira mensagem de atualização agenda-se novamente a propagação de informações e atribuição de papel. Se uma mudança

significativa ocorrer, por exemplo no nível de bateria, envia-se uma mensagem de atualização imediatamente para todos os vizinhos.

- Atribuição de papel: avalia-se qual papel deve ser atribuído de acordo com as informações coletadas de cada vizinho e das características de cada papel. É desencadeado pelo agendamento no início da simulação ou após receber uma primeira mensagem de atualização. Se o papel for alterado após a avaliação executa-se novamente a propagação de informações.
- Detecção de falhas: nós enviam *heartbeats* a cada período pré determinado. Um *heartbeat* consiste em uma mensagem de atualização com valor nulo para o papel do sensor que envia a mensagem. Verifica-se após certa quantidade de tempo quais sensores enviaram mensagens e caso haja algum inativo suas entradas são apagadas.

Ao fim do trabalho de Frank & Römer [2005] uma análise comparativa é realizada entre as aplicações utilizando-se como métricas de avaliação o número de: mensagens enviadas, atribuições incorretas de papéis e trocas de papel atribuído a cada sensor. É realizada também uma análise de robustez em face do número de mensagens perdidas, ou seja, para cada probabilidade de perda de pacotes na rede avalia-se a razão entre o número de atribuições incorretas e o total de sensores. O trabalho conclui que a atribuição de papéis de forma genérica é praticamente viável em termos de eficiência e robustez.

2.1.2 Heurísticas e atribuição de papéis reativa

Para a minimização do consumo de energia na rede, Souza [2007] propõe diferentes abordagens a fim de reduzir o número de sensores ativos e determinar seus respectivos papéis. Considera-se o problema de atribuição de funções de sensoriamento, ou seja, para um momento específico deseja-se atribuir uma função a cada sensor da rede com base nas demandas de sensoriamento da aplicação naquele instante. Com isso um nó pode assumir os papéis de sensor de temperatura, umidade, pressão, entre outros. Para cobertura da região discretizou-se a área a ser monitorada em pontos de demanda. O trabalho é então realizado conjuntamente com o problema de atribuição de funções de sensoriamento, cobertura, controle de densidade e roteamento de informações. Ou seja, deseja-se garantir a cobertura dos pontos de demanda com os papéis necessários além da existência de uma rota entre os sensores e nós sorvedouros.

Entre as abordagens propostas por Souza [2007] encontra-se uma formulação exata, resolvida através de programação inteira mista, um algoritmo genético e um

algoritmo distribuído. A primeira mostrou-se ineficaz com o aumento do tamanho da rede, servindo entretanto como critério de comparação com as demais soluções.

O algoritmo genético é empregado tanto na atribuição de papéis estática (papéis atribuídos não podem ser mais alterados) quanto na atribuição dinâmica (papéis podem ser alterados com o decorrer do tempo). Para o caso dinâmico realiza-se uma execução periódica do algoritmo, ou seja, executa-se a metaheurística em intervalos de tempo regulares a fim de reconfigurar a rede para possíveis mudanças.

A representação da solução consiste em um vetor de tamanho n (número de sensores), um vetor binário de tamanho m representando o número de pontos de demanda e determinando se estão cobertos ou não. Um vetor de rotas indica a conectividade entre os nós. A rota é gerada conjuntamente ao procedimento, sempre buscando conectar o sensor ativo ao que esteja mais próximo a ele. O algoritmo genético é apresentado abaixo seguido de uma descrição de seus passos.

Algoritmo 1: Algoritmo Genético

```

1 População ← Geração de população inicial;
2 Melhor Solução ← avalia(População);
3 t ← 0;
4 enquanto t < NumMaxGerações faça
5   t ← t + 1;
6   População.Mutação;
7   População.Crossover;
8   se mudou(Melhor Solução) então
9     t ← 0;
10  População.Seleção;
11 retorna Melhor Solução
```

Primeiramente gera-se a população inicial do algoritmo (linha 1). Essa é construída por um método parcialmente guloso. A cada iteração escolhe-se um nó para ser ativado até que todos os pontos de demanda estejam cobertos ou não exista nenhum nó que não viole a conectividade para ser ativado. Conectividade exige que um nó ativo possa se comunicar com outro nó ativo. O papel a ser atribuído é sempre aquele que possui maior número de pontos de demanda descobertos. Para cada papel escolhe-se um dos nós dentre os 30% melhores (aqueles que consomem menor energia e cobrem mais pontos de demanda).

A melhor solução é atribuída como o melhor indivíduo da população inicial (linha 2) e o número de iterações é inicialmente zero (linha 3). Enquanto o número de iterações for menor que o número máximo de iterações permitidos realizam-se os passos de mutação, crossover e seleção.

Na mutação, linha 6, atribui-se a dois sensores aleatórios dois novos papéis, também aleatórios, podendo inclusive desativar os sensores. Já para o crossover, linha 7, são escolhidos aleatoriamente dois indivíduos que irão gerar dois filhos através da combinação de seus vetores de representação. Para isso determina-se um ponto aleatório de corte e combinam-se as partes dos vetores antes e depois desse ponto, gerando-se assim novas soluções. O crossover deve ser realizado até que a população dobre. No passo de seleção, linha 11, metade da população é selecionada para sobreviver. A seleção é baseada na aptidão dos indivíduos, aqueles mais aptos tendem a sobreviver.

A utilização do algoritmo genético proposto por Souza [2007] se torna viável se comparada a solução exata apenas em casos em que a rede apresente grande densidade de sensores. Nessa situação existe uma pequena elevação no consumo de energia que é compensada pelo tempo computacional bastante inferior.

Por fim, o algoritmo distribuído apresentado por Souza [2007] é construído de modo que cada sensor tente atribuir um papel a si mesmo, em intervalos de tempo determinados, com base nas informações locais daquele instante. Essa abordagem mostrou-se inferior a execução periódica do algoritmo genético em termos de energia e cobertura de redes esparsas.

2.1.3 Algoritmos distribuídos e atribuição de papéis reativa

Ao longo do tempo é necessário que as RSSFs se adaptem às mudanças resultantes de interferências externas, como alterações topológicas, requisições de agentes externos à rede ou eventos que acontecem na região onde a rede se encontra. Para a monitoração de eventos é necessário que os sensores sejam capazes de eficientemente transmitir dados à uma fonte. Assim como Souza [2007], Nakamura et al. [2009] apresentam um algoritmo para a atribuição de papéis reativa a fim de otimizar o roteamento de dados na rede.

Nessa solução, os papéis de coordenador (sensor que detecta um evento e notifica sua ocorrência), colaborador (sensor que detecta um evento e notifica a um coordenador) e retransmissor (sensor que transmite dados recebidos por outro sensor) são atribuídos tentando-se encontrar a menor rota até o nó sorvedouro (sensor que consume dados). Caso um evento seja detectado por mais de um sensor, é realizado um processo de clusterização, de forma que os dados sejam encaminhados ao sorvedouro apenas pelos sensores coordenadores.

O algoritmo a seguir apresenta as etapas a serem adotadas caso um evento seja detectado na rede.

Inicialmente, linha 1, o algoritmo verifica se algum evento foi detectado. Em

Algoritmo 2: Detecção de Eventos

```

1 se Evento é detectado então
2   | Construção de cluster entre os sensores que detectaram o evento;
3   | Atribui-se retransmissor aos demais sensores;
4   | Formam-se rotas entre clusters e sorvedouro;
5   | Agregação de dados para redução nos custos de comunicação;
6 senão
7   | Todos assumem papel retransmissor;

```

caso negativo todos os sensores assumem papel retransmissor (linha 8) e o algoritmo termina. Em caso positivo executa-se a instrução da linha 2. Nessa o sensor de menor id que detectou o evento terá papel coordenador. Este anuncia seu papel para sua vizinhança e os demais tornam-se colaboradores. Caso um sensor assumo o papel de colaborador mas envie pacotes do coordenador para o sorvedouro, seu papel é alterado para coordenador e sua vizinhança informada. Os sensores que não detectaram o evento possuem papel retransmissor (linha 3).

A seguir, linha 4, formam-se as rotas entre clusters e sorvedouro. Para isso escolhe-se para transmissão de dados os vizinhos que possuem menor distância até o sorvedouro. Caso haja empates escolhe-se aquele que possui menor soma de distância até os outros coordenadores. Por fim, na linha 5, realiza-se a agregação de dados. Essa é feita intra-clusters e inter-clusters. Intra-clusters: encontra árvore de menor caminho entre colaboradores e coordenadores. Inter-clusters: mesma estratégia, porém para desempate escolhe a maior soma dos caminhos aos coordenadores, havendo assim maior agregação de dados, já que mais sensores coincidirão no mesmo caminho.

Os resultados apresentados por Nakamura et al. [2009] mostraram um desempenho bastante superior em relação a algoritmos clássicos de roteamento da literatura: Krishnamachari et al. [2002] e Zhou & Krishnamachari [2003].

2.2 Algoritmos Exatos

2.2.1 Limites para o tempo de vida da RSSF

Bhardwaj & Chandrakasan [2002] propõem limites para o tempo de vida da rede que qualquer técnica de atribuição de papéis possa alcançar. São considerados três tipos de papéis: sensor (nó que observa a região de interesse e transmite informações para a base), retransmissor (simplesmente retransmite dados recebidos sem nenhum tipo de processamento) e agregador (nó que recebe dois ou mais tipos de fluxos de dados e

retransmite em um único fluxo).

Para resolução do problema define-se uma atribuição de papéis como viável se a transmissão de informações na rede para o sorvedouro é feita pelo menor número de sensores especificados. Além disso nenhum papel pode ser excluído, de tal forma que a propriedade anterior continue válida. Sabe-se que o número de atribuições viável é finito. Dessa forma procura-se determinar as proporções em que cada atribuição deve ser aplicada a fim de maximizar o tempo de vida da rede.

Através de programas lineares, Bhardwaj & Chandrakasan [2002] determinaram tais limites, mas apenas algumas classes de problema podem ser resolvidas em tempo polinomial. Para essas a atribuição de papéis é baseada em fluxo de redes.

2.2.2 Atribuição de papéis de forma genérica e critérios de otimização globais

Outro trabalho apresentado por Frank & Römer [2006] propõe a construção de um modelo de programação linear inteira (PLI) para a resolução do problema de atribuição de papéis. Tal modelo é um mapeamento da linguagem de especificação de papéis, proposta em Frank & Römer [2005], em critérios específicos de otimização.

Nesse trabalho entretanto, é necessária uma visão global da rede para resolução do PLI. São coletadas as informações de cada sensor e os papéis são então entregues a rede. Apesar de um agente global ser inviável em grande parte dos problemas de atribuição de papéis, tal abordagem apresenta vantagens como critérios de otimização global e detecção de inviabilidade do problema. O modelo apresentou resultados satisfatórios em redes de até 1000 sensores nos problemas de cobertura, clusterização e agregação.

2.3 Análise

Dois trabalhos para atribuição de papéis reativa foram apresentados. Nakamura et al. [2009] apresentam um algoritmo mais robusto tendo-se em vista o algoritmo genético formulado por Souza [2007]. Além de ser uma solução distribuída, podendo assim ser amplamente mais aplicável no contexto de RSSF, pode realizar agregação de dados, tratamento de falhas e atribuição dinâmica. Entretanto não se mostrou adequado para situações em que diferentes tipos de eventos precisem ser analisados no mesmo instante. Nesse caso o algoritmo genético, construído com essa situação em vista, possui melhor desempenho. Ambos tratam redes de sensores ôneos e mesmo raios de comunicação e sensoriamento.

Para o caso de redes sem comunicação confiável apenas Frank & Römer [2005] fazem uma análise dos possíveis resultados. Entretanto esse trabalho realiza apenas a atribuição de papéis pró-ativa (sem detecção de eventos), ou seja, deseja-se que os sensores atinjam certa configuração de modo que não haja atribuições incorretas de acordo com as especificações de cada papel. O algoritmo distribuído proposto apresentou-se também bastante robusto possuindo tratamento de falhas, inicialização de papéis probabilística e atribuição dinâmica.

Para resoluções exatas foram apresentados os trabalhos de Bhardwaj & Chandrakasan [2002] e Frank & Römer [2006]. Como principais desvantagens dessas abordagens podemos citar o alto custo computacional e algoritmo centralizado. Sua aplicação é restrita a redes de tamanho reduzido e que possuam agente global. Bhardwaj & Chandrakasan [2002] conseguem, entretanto, minimizar o uso de energia da rede, encontrando a melhor proporção possível para a aplicação das atribuições de papéis viáveis. Já Frank & Römer [2006] podem resolver problemas de atribuição com critérios de otimização global, como por exemplo minimizar o uso de determinado papel, ou atestar a inviabilidade do problema.

Capítulo 3

Modelagem

Esse capítulo apresenta a definição do problema coloração de vértices utilizado na modelagem do problema de atribuição de papéis. Os pseudo-códigos para resolução centralizada e distribuída do problema são apresentados ao final, conjuntamente com a resolução do problema de roteamento que deve ser resolvido na atribuição de papéis.

3.1 Problema de Coloração de Vértices

Dado um conjunto de vértices V , um conjunto de arestas E e um grafo não-dirigido $G(V, E)$, define-se o problema clássico de coloração de vértices como a atribuição de cores aos vértices sem que dois destes compartilhem uma mesma cor e sejam adjacentes ao mesmo tempo [Wigderson, 1983]. Além disso deseja-se utilizar o menor número possível de cores. Mostraremos que este problema pode ser considerado uma instância do problema de atribuição de papéis e algumas possíveis heurísticas para sua resolução.

3.1.1 Caso Particular do Problema de Atribuição de Papéis

Observe que o problema de atribuições de papéis definido nada mais é do que um extensão do problema de coloração. Suponhamos um caso em que todos os sensores possam assumir todos os papéis e cada papel possua um peso, ou seja, queremos que os papéis de maior peso sejam atribuídos ao maior número de sensores possível. Considera-se também que dois sensores adjacentes não podem possuir o mesmo papel pois o sensoriamento realizado por um deles seria considerado redundante. Por fim todos os sensores possuem energia ilimitada e logo o roteamento de informações pode ser tratado separadamente, não influenciando na qualidade da solução. Apesar de

Figura 3.1: Exemplo de coloração ótima encontrada pela coloração na ordem indicada pelos números em cada vértice

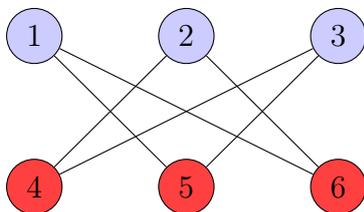
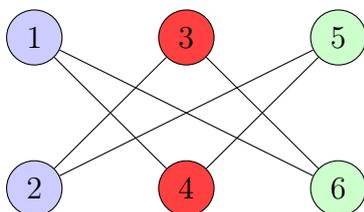


Figura 3.2: Exemplo de coloração não ótima encontrada pela coloração na ordem indicada pelos números em cada vértice



tal aplicação ser bastante improvável de existir observe que caímos exatamente no problema de coloração, no qual deseja-se minimizar as cores utilizadas.

3.1.2 Heurísticas Sequenciais

Uma das formas de se obter uma solução heurística para o problema de coloração de vértices é a utilização de uma abordagem gulosa. Para isso são atribuídas uma a uma as cores aos vértices do grafo até que todos estejam coloridos sem que nenhum vizinho possua a mesma cor. A ordem em que os vértices são coloridos impacta diretamente na solução. Veja os exemplos nas figuras 3.1 e 3.2.

Welsh & Powell [1967] propõem que os vértices de maior grau possam ser coloridos primeiro, já Brèlaz [1979] propõe que vértices com maior grau de saturação devem possuir a prioridade. O grau de saturação de um vértice é o número de cores ao qual ele é adjacente. Outra heurística é proposta por Lofti & Sarin [1986] e sugere que os vértices sejam coloridos de acordo com o conjunto independente ao qual pertencem. Vértices que pertençam ao maior conjunto independente são coloridos primeiro, os que pertencem ao segundo maior conjunto independente são coloridos em seguida e assim sucessivamente. O algoritmo 3 apresenta o pseudo-código para coloração de vértices através de uma abordagem gulosa onde os vértices são ordenados de acordo com o critério de Brèlaz [1979].

Algoritmo 3: Algoritmo Coloração de Vértices

Dados: Grafo G **Saída:** Grafo Colorido

```

1  $i \leftarrow 0$ ;
2 enquanto  $G$  contém vértices não coloridos faça
3    $i \leftarrow i + 1$ ;
4   para cada vértice  $v$  de  $G$  não colorido, por ordem de grau de saturação
5     faça
6       se Nenhum vértice adjacente a  $v$  possui cor  $i$  então
7         Atribuir cor  $i$  ao vértice  $v$ ;
         Ordenar vértices por ordem de grau de saturação;

```

3.2 Mapeamento para o Problema de Atribuição de Papéis

Neste problema podemos enxergar a distribuição dos papéis também como a construção de conjuntos independentes na RSSF, desde que consideremos que sensores adjacentes não devem assumir o mesmo papel. Esta consideração entretanto só se aplica a problemas nos quais existe pouca ou nenhuma probabilidade de eventos do mesmo tipo ocorrerem dentro da área de intersecção entre dois sensores adjacentes. Para expandir a aplicabilidade do algoritmo será considerado um histórico de eventos para cada sensor, onde sensores que possuam parte de seu histórico em comum serão considerados como um só elemento na construção dos conjuntos independentes, ou seja, apesar de serem adjacentes poderão possuir a mesma função.

No problema de atribuições de papéis, diferentemente da coloração de vértices, objetiva-se a minimização do uso de energia, monitoramento da região de interesse e roteamento das informações sensoriadas para base. Entretanto podemos adotar a abordagem gulosa proposta para o problema de coloração de vértices desde que uma ordenação dos sensores, que vise os objetivos do problema, possa ser encontrada para a atribuição de papéis.

Exceto pelo grau de cada vértice, que pode representar uma certa dificuldade de coloração do mesmo, no problema de coloração de vértices os nós são considerados homogêneos, ou seja, não existe nenhuma característica que os diferencia. Além disso as cores não acarretam nenhum atributo ao vértice além do conjunto ao qual aquele vértice pertence. Diferentemente, na atribuição de papéis o papel de cada sensor impacta diretamente na solução, pois determina o gasto de energia daquele sensor no instante em que realiza o papel atribuído. Outras características que devem ser consideradas

para se determinar a ordem em que os sensores devem receber seus papéis são:

1. Posição: Sensores posicionados em regiões de maior probabilidade de eventos podem possuir prioridade na atribuição de papéis. Isso se deve principalmente ao tipo de aplicação na qual o algoritmo está sendo usado. Aplicações nas quais se deseja realizar o sensoriamento o mais rápido possível, como por exemplo fogo, justificam esta propriedade.
2. Densidade: Regiões com maior quantidade de sensores tendem a ser atendidas mais facilmente. Dessa forma é aceitável que regiões com menor número de sensores possuam preferência na atribuição de papéis a fim de se garantir a cobertura de toda a área.
3. Nível de bateria: É fácil ver que o funcionamento do algoritmo deve visar o equilíbrio de energia gasta pelos sensores, de forma que o tempo de vida da rede possa ser estendido. Sensores sem energia podem interromper o fluxo de informações aumentando o custo de comunicação ou no pior caso isolando a base de informações da rede.
4. Nível de centralidade: Sabe-se que para cada sensor deve ser determinada uma rota de menor tamanho entre ele e a base que deverá receber suas informações. O nível de centralidade de um sensor é definido como o número de menores caminhos que passam por aquele sensor. Sensores de maior nível de centralidade devem ser preservados quanto ao gasto de energia para que o custo de comunicação não aumente e as construções de novas possam ser evitadas.
5. Grau: Assim como no problema de coloração de vértices o grau de um sensor também implica um nível de dificuldade na atribuição de papéis ao sensor. Com isso sensores de maior grau podem ser considerados primeiro para a definição de seu papel.
6. Articulação: Um sensor é de articulação se a sua retirada faz com a rede torne-se desconexa. Sensores de articulação são de extrema importância pois garantem a comunicação dele e de outros sensores com a base.
7. Possíveis papéis: Papéis podem conter pesos de acordo com sua importância no problema. Sensores que tenham detectado papéis de maior importância podem também possuir prioridade na atribuição de papéis.

Atribuindo-se pesos as características de 1-7 podemos construir um função que irá determinar a ordem em que a atribuição de papéis deve ser realizada, de forma a

fornecer uma solução que tente minimizar o uso de energia, maximizar o monitoramento da região de interesse e garantir o roteamento das informações sensoriadas para base. Os pesos adotados durante as simulações realizadas são tratados no capítulo 4.

Algoritmo 4: Função de Prioridade

Dados: S

Saída: valor referente a prioridade de S

Sejam P1, P2, P3, P4, P5, P6 e P7 os pesos associados a posição, densidade, bateria, centralidade, grau, articulação e papéis respectivamente;

- 1 $Prioridade = P1 * S.posição + P2 * S.densidade + P3 * S.bateria + P4 * S.centralidade + P5 * S.grau + P6 * S.articulação + P7 * S.papéis;$
 - 2 **retorna** *Prioridade*
-

3.2.1 Roteamento

Para o roteamento de informações na RSSF é necessário que sejam determinadas as rotas entre os sensores e os nós sorvedouros. Uma das formas é encontrar o menor caminho em saltos, ou seja, o caminho que utiliza o menor número de sensores intermediários até o nó sorvedouro, é através do algoritmo de Dijkstra [1959]. Para isso é considerado que a distância entre dois sensores adjacentes é sempre 1. Caso haja mais de um nó sorvedouro, cada sensor deverá encaminhar os dados coletados para aquele que esta mais próximo.

A definição das rotas para encaminhamento de informações deve ser realizada no início da simulação, antes que o primeiro evento ocorra, e caso algum sensor se torne inativo, por falha do mesmo ou falta de bateria por exemplo. Note que caso o sensor a se tornar inativo não participe de nenhuma rota entre outro sensor e seu nó sorvedouro, ou seja, se ele não transmite nenhuma informação recebida de outro sensor, a definição de novas rotas de encaminhamento não se faz necessária.

3.3 Algoritmo Centralizado

Podemos apresentar um algoritmo centralizado para o problema de acordo com o mapeamento feito na seção 3.2. Para isso iremos utilizar como base o algoritmo 3 onde a ordenação dos sensores é dada pelo algoritmo 4. O resultado é apresentado abaixo pelo algoritmo 5.

Algoritmo 5: Atribuição de papéis centralizada

Dados: Grafo G

Saída: Grafo Colorido

- 1 Ordenar os vértices de G em ordem decrescente de função de prioridade;
 - 2 **para** cada vértice v de G por ordem de prioridade **faça**
 - 3 **para** cada possível papel p de v por ordem de demanda **faça**
 - 4 **se** nenhum vértice adjacente a v possui papel p **então**
 - 5 Atribuir p ao vértice v ;
 - 6 **se** v permanece sem papel **então**
 - 7 Atribuir *sempapel* ao vértice v ;
 - 8 **retorna** G
-

3.4 Algoritmo Distribuído

Para implementação de uma versão distribuída três etapas precisam ser definidas: cálculo de prioridade, sensoriamento e comportamento em caso de falta de bateria. A seguir são apresentadas estas três etapas.

3.4.1 Prioridade

Assim como no problema centralizado aplica-se o algoritmo 4 para definição da prioridade. Entretanto por se tratar de uma versão distribuída o algoritmo precisa ser executado pelos próprios sensores da rede, de modo que os dados necessários também sejam coletados pelos mesmos através da troca de mensagens entre sensores vizinhos. Essa coleta de informações é realizada antes que os eventos comecem a ser gerados e utiliza os tipos de mensagens apresentados na tabela 3.1. A medida que sensores fiquem sem energia os dados são atualizados, sem que o processo precise ser repetido por completo.

- **Rotas de Encaminhamento**

Para definição de rotas de encaminhamento de mensagens, um conjunto de mensagens iniciais são propagadas pelo sorvedouro para todos seus vizinhos. Chamaremos este tipo de mensagem, para definição de rotas de encaminhamento, de tipo 1. Suponhamos que um sensor a envie uma mensagem de tipo 1 para um sensor vizinho b . Caso essa seja a primeira mensagem de tipo 1 recebida por b o mesmo atualiza seu sensor de encaminhamento de informações para o sensor a . Ao enviar a mensagem de tipo 1, o sensor a informa também a sua distância ao sorvedouro, de modo que b saiba que sua distância ao sorvedouro é de uma unidade a mais que a distância de a ao sorvedouro.

- **Centralidade**

Após definir seu sensor de encaminhamento, b envia uma mensagem de tipo 2 para o sensor a . Ao receber uma mensagem de tipo 2, a envia também uma mensagem de tipo 2 para seu sensor de encaminhamento. A centralidade de um sensor é o número de mensagens de tipo 2 recebidas pelo mesmo.

- **Densidade**

Neste mesmo processo, b envia também uma mensagem de tipo 3 para todos seus vizinhos. Seja c um vizinho de b . Ao receber a mensagem de tipo 3 enviada por b , c inclui b na sua lista de vizinhos em até 2 hops. O número de vizinhos em até 2 hops é utilizado para a definição da densidade da região em que o sensor se encontra. Por fim c propaga esta mesma mensagem para seus vizinhos, de forma que estes também possam incluir b na lista de vizinhos em até 2 hops. A mensagem de tipo 3 enviada por c inclui um parâmetro extra informando que essa não deve mais ser propagada.

- **Pontos de articulação**

O último passo para que a detecção de eventos possa ser realizada é a definição dos pontos de articulação da rede. Um algoritmo com este propósito é apresentado por Pranay [1998]. Basicamente duas etapas são realizadas: busca em profundidade distribuída e detecção de sensores de articulação. A busca em profundidade tem como objetivo a construção de uma árvore geradora. Arestas que não pertencem a árvore geradora podem identificar ciclos de modo que o segundo passo, identificar sensores de articulação, possa ser realizado. A execução do algoritmo requer $4n$ unidades de tempo, onde n é o número de sensores da rede. Este processo é bastante custoso e não será detalhado neste trabalho. Outras informações podem ser encontradas em Pranay [1998].

3.4.2 Sensoriamento

Após um período inicial de $4n$ rounds, necessário para definição da centralidade, densidade, articulação e rotas de encaminhamento inicia-se o processo de detecção de eventos. Todos os eventos são gerados simultaneamente, de modo que após a geração dos mesmos haja um tempo para troca de informações e definição de papéis pelos sensores. Em seguida mais eventos são gerados e o processo é repetido até que a simulação termine.

Mensagem	Função
Tipo 1	Propagada inicialmente pelo Sink para definição das rotas de encaminhamento.
Tipo 2	Enviada por cada sensor para seus nós de encaminhamento a fim de determinar a centralidade de cada vértice.
Tipo 3	Mensagem propagada para vizinhos de até 2 hops de modo a determinar a densidade de cada nó.

Tabela 3.1: Tipos de mensagens utilizados na configuração inicial da rede

Inicialmente todos os vértices tentam sensoriar os eventos de maior peso detectados e informam seus vizinhos, algoritmo 6. Ao receber uma mensagem informando um papel assumido os sensores devem seguir os passos do algoritmo 7.

Algoritmo 6: Sensoriamento

Dados: Sensor s
Saída: Papel de s

- 1 Detectar eventos;
- 2 $s.prioridade \leftarrow$ Prioridade (Algoritmo 4);
- 3 **para** cada possível papel p de s (iniciando-se pelos papéis de maior peso) **faça**
- 4 **se** s detectou p **então**
- 5 $s.papel \leftarrow p$;
- 6 Envia mensagem para todos os vizinhos informando $s.papel$ e $s.prioridade$;
- 7 Sai da iteração;

Algoritmo 7: Recebe mensagem

Dados: Sensor s , Mensagem m

- 1 Assincronismo $\leftarrow s.prioridade \times \text{Random}(0,1)$;
- 2 **se** $m.prioridade < s.prioridade$ ou Assincronismo $> \alpha$ **então**
- 3 $s.papel \leftarrow \emptyset$;
- 4 **para** cada possível papel p de s (iniciando-se pelos papéis de maior peso) **faça**
- 5 **se** s ainda não tentou assumir p **então**
- 6 $s.papel \leftarrow p$;
- 7 Envia mensagem para todos os vizinhos informando $s.papel$ e $s.prioridade$;
- 8 Sai da iteração;

Note que o processo de detectar um evento e informar aos vizinhos é realizado simultaneamente por todos os sensores para cada tipo de evento. Com isso o algoritmo esta sujeito a erros, já que caso um sensor não assuma um papel esta informação não é propagada a seus vizinhos, fazendo com que estes possam perder um papel sem que nenhum outro sensor cubra o evento. Um exemplo para este caso é apresentado na figura 3.3. Para isso, utilizando uma distribuição uniforme, geramos uma probabilidade que multiplicada pela prioridade de cada sensor cria uma chance que seu papel seja mantido mesmo com um vizinho de mesmo papel. Mais detalhes quanto a simulação e valores de parâmetros adotados serão apresentados na próxima seção.

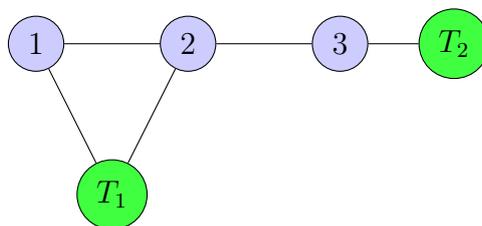


Figura 3.3: Sensores 1, 2 e 3 realizando a detecção de eventos de temperatura. Suponha que as prioridades dos sensores 1, 2 e 3 sejam 10, 20 e 30 respectivamente. Como os eventos T são gerados simultaneamente os vértices tentam realizar o sensoriamento ao mesmo tempo, definem seus papéis como T e informam os vizinhos. O sensor 3 recebe a informação do sensor 2 de que este já assumiu o papel T e o sensor 2 recebe a informação do sensor 1 de que este também já assumiu o papel T . Como o sensor 1 é o de menor prioridade ele é o único que manterá o papel, fazendo com que o sensoriamento do evento T_2 seja perdido

3.4.3 Perda de sensores

Caso um sensor tenha que ser retirado da simulação, por falta de bateria, os demais vértices devem se adaptar, atualizando os critérios para cálculo de prioridade ou também saindo da simulação caso não possuam mais rotas com o sorvedouro.

Antes que o sensor seja retirado da simulação é assumido que este possui bateria suficiente para informar sua saída aos vértices vizinhos. Ao ser informado que um de seus vizinhos não esta mais na simulação os sensores devem executar os seguintes passos:

1. Caso o sensor retirado seja o vértice de encaminhamento de mensagens, alterar o sensor de encaminhamento de mensagens para o vizinho mais próximo do sorvedouro. Este vizinho não deve possuir distância ao sorvedouro maior que a distância a ser atualizada. Caso não haja opções, sair da simulação e informar vizinhos.

2. Ao atualizar rota de encaminhamento de mensagens informar ao sensor recebedor para que este possa atualizar sua centralidade.
3. Atualizar grau.
4. Atualizar densidade.
5. Caso o sensor a ser retirado seja de articulação, todos os seus vizinhos devem acrescentar uma nova constante p_a aos futuros cálculos de prioridade.

Dentre todas as informações a serem atualizadas, informar se os sensores restantes serão de articulação é o único passo que não é definido. Isso se deve ao algoritmo distribuído para definição de pontos de articulação ser bastante demorado, fazendo com que sua execução acarretasse em uma grande perda de eventos detectados. Para contornar este problema apresentamos uma possibilidade em que os sensores vizinhos ao vértice de articulação retirados acrescentem um peso maior a suas futuras prioridades. É razoável assumir que caso os vizinhos de um ponto de articulação permaneçam na simulação mesmo após sua retirada, eles irão possuir maior importância quanto ao roteamento de informações, e por isso assumam maior prioridade.

Capítulo 4

Experimentos Computacionais

Para execução do algoritmo proposto utilizou-se o simulador Sinalgo [2007] tanto para a parte centralizada quanto para a distribuída. Este é um arcabouço de simulação para realização de testes e validações em algoritmo de redes. Todo o código é escrito em Java e compilado utilizando-se um compilador Java padrão. São apresentadas as etapas por ordem de execução de cada simulação assim como os parâmetros adotados em cada passo. Ao final é feita uma descrição do cenário padrão de simulação e das métricas de avaliação utilizadas.

4.1 Simulação Centralizada

4.1.1 Iteração Básica

A execução do algoritmo centralizado é constituída de duas partes principais, a pré execução, onde passos que devem ser executados uma única vez são processados, e o *round*, que é a iteração básica a ser repetida enquanto o critério de parada definido não for alcançado. Abaixo são descritos por ordem de execução os passos de ambas essas partes.

Pré Execução

- Geração de grafo conexo de acordo com a distribuição adotada (figura 4.1)
- Determinação das rotas de encaminhamento de dados

Round

- Geração de eventos
- Determinação da prioridade dos sensores, algoritmo 4

- Atribuição de papéis, algoritmo 5
- Envio de informações
- Avaliação critério de parada
- Se necessário, construção de novas rotas de encaminhamento

4.1.2 Definição da prioridade

Para definição dos pesos adotados pelo algoritmo 4, considerou-se a seguinte ordem de importância dos critérios de cada sensor:

$$\text{articulação} > \text{bateria} > \text{centralidade} \gg \text{densidade} > \text{papéis} > \text{posição} > \text{grau} \quad (1)$$

Tentou-se priorizar inicialmente o funcionamento da rede conjuntamente com a economia de energia, ou seja, antes de realizar a atribuição de papéis é necessário que a rede seja capaz de operar e se possível que o tempo de vida também seja prolongado. Para isso consideremos que sensores de articulação são de extrema importância, pois uma falha nos mesmos pode tornar a rede desconexa, fazendo com que não seja possível o envio de informações ao nó sorvedouro. Seguidos destes, sensores com menor nível de bateria tendem a ser poupados a fim de se prolongar o tempo de vida da rede, provendo um uso de energia mais balanceado durante a simulação. O terceiro critério é o nível de centralidade. Sensores com maior nível de centralidade também tendem a ser menos utilizados, já que devem ser sobrecarregados com o repasse de mensagens recebidas.

A seguir são tratados os critérios que objetivam melhorar a atribuição de papéis realizada. Para isso escolhemos primeiro os sensores que se encontram em regiões de menor densidade, ou seja, com menor número de sensores, e em seguida escolhemos aquele que realizaram a detecção dos papéis mais importantes. Por último estão os critérios de posição e grau. O primeiro faz com que sejam escolhidos primeiro sensores posicionados em regiões com maior ocorrência de eventos e o segundo faz com que sensores de maior grau, ou seja, mais difíceis de se atribuir um papel possuam prioridade.

Para atribuição dos valores de cada critério podemos observar que alguns são facilmente definidos, como por exemplo *articulação*. Caso um sensor seja de articulação o critério *articulação* irá possuir valor 1 e caso contrário 0. A *bateria* inicial de cada sensor é de 30 Joules e seu gasto é modelado de acordo com Mini et al. [2005]. Basicamente cada sensor possui quatro estados com seus respectivos gastos de energia: sensoriamento e comunicação desligados ($25 \mu\text{W}$), sensoriamento ligado e comunicação

Critério	Valor
Articulação	1 ou 0
Bateria	30 Joules
Centralidade	Número de rotas de encaminhamento de mensagens que passam pelo sensor
Densidade	Número de vizinhos em até 2 saltos
Papéis	Soma dos pesos dos papéis que podem ser atribuídos ao sensor naquele <i>round</i>
Posição	Somatório dos pesos de cada o papel multiplicados pelo número de detecções realizadas
Grau	Número de vizinhos

Tabela 4.1: Cálculo dos critérios de cada sensor

Peso	p_1	p_2	p_3	p_4	p_5	p_6
Valor	3000	1000	40	20	0.01	5

Tabela 4.2: Pesos adotados

desligado (38.72 mW), sensoriamento ligado e recebendo dados (52.20 mW), sensoriamento ligado e transmitindo dados (74.70 mW). Por fim o gasto no envio de cada mensagem é atribuído em média como 26.8 mJ.

O critério *posição* avalia o número de detecções realizadas por cada sensor ao longo de toda simulação e multiplica esse valor pelos pesos de cada papel. Um resumo dos valores assumidos por cada critério é apresentados na tabela 4.1. Com isso podemos definir a fórmula de prioridade do algoritmo 4 da seguinte forma:

$$\text{prioridade} = p_1 \times \text{articulação} + (p_2 - \text{bateria}) + p_3 \times \text{centralidade} + (p_4 - \text{densidade}) - \text{papéis} - p_5 \times \text{posição} - \frac{\text{grau}}{p_6} \quad (2)$$

Podemos separar os critérios da tabela 4.1 em dois grupos: *grupo 1* (articulação, bateria e centralidade) e *grupo 2* (densidade, papéis, posição e grau). Note que todos os elementos do *grupo 2* são sempre menores que os do *grupo 1*. Ou seja, o *grupo 2* é utilizado como desempate para sensores que possuem mesmos valores no *grupo 1*.

Os parâmetros p_1 a p_6 são atribuídos de forma que um único critério do *grupo*

1 seja sempre maior que os do *grupo 2*, independente de seus valores atuais. Já os pesos para critério de mesmo grupo tentam obedecer a ordenação dada por (1), porém dependendo dos valores de momento de cada sensor isso pode não ser atendido. Por exemplo, se um sensor $s \in S$ for de articulação então p_1 deve possuir um valor tal que $p_1 \times s.\text{articulação} > (p_4 - u.\text{densidade}) + u.\text{papéis} + p_5 \times u.\text{posição} + \frac{u.\text{grau}}{p_6}$ para todo $u \in S$ que não seja de articulação.

A tabela 4.2 apresenta os valores adotados aos pesos p_1 a p_6 . Tais valores não possuem grandeza e são gerados unicamente com o fim de se manter a ordenação dada por (1). Sendo assim essa escolha depende fortemente da aplicação e dos objetivos desejados. Para a aplicação estudada foi realizada uma calibragem dos valores de forma empírica utilizando-se o bom senso e o conhecimento prévio do problema. Note que o único critério que não possui peso associado é o de papéis. Isso se deve ao fatos dos papéis já possuírem pesos nesta aplicação. Os valores adotados variavam entre 3 e 12, obedecendo-se assim a ordem em (1).

4.2 Simulação Distribuída

4.2.1 Iteração Básica

Assim como na simulação centralizada esta é constituída de duas partes principais, a pré execução e o *round*. Entretanto a geração de eventos é feita somente em *rounds* múltiplos de duas vezes o número de tipo de eventos (e). Para cada tipo de evento gerado os sensores precisam de um *round* para realizar a detecção e mais um para informar os respectivos vizinhos de um possível papel assumido. Abaixo são descritos por ordem de execução os passos de ambas essas partes.

Pré Execução

- Geração de grafo conexo de acordo com a distribuição adotada (figura 4.1)
- Determinação das rotas de encaminhamento de dados
- Determinação da centralidade de cada sensor
- Determinação da densidade de cada sensor
- Identificação dos pontos de articulação da rede

Round múltiplo de $2e$

- Geração de eventos

- Detecção de eventos
- Cálculo de prioridade pelos sensores da rede
- Troca de informações para definição de papel

Round não múltiplo de 2e

- Recebimento de papéis assumidos por vizinhos
- Se necessário troca de papéis
- Mensagens para informar vizinhos de novo papel

4.2.2 Definição da prioridade

A definição da equação de prioridade para a simulação distribuída segue a mesma apresentada em (2). Entretanto um peso adicional é adotado, necessário para tratar os pesos de articulação no caso da saída de sensores da rede. Este peso, denominado p_7 , é zero se o sensor associado não tiver conhecimento de nenhum vizinho de articulação que tenha deixado a simulação e mil e quinhentos caso contrário. Assim como os demais pesos, podemos dizer que o valor adotado por p_7 tenta inserir um novo critério na ordenação dada por (1). Este novo critério faz com que sensores que possuam p_7 diferente de 0 tenham menor prioridade que sensores de articulação, porém sejam mais importantes que sensores que não são de articulação, independente de qualquer outro critério. Desse modo a fórmula de prioridade é dada como:

$$\text{prioridade} = p_7 + p_1 \times \text{articulação} + (p_2 - \text{bateria}) + p_3 \times \text{centralidade} + (p_4 - \text{densidade}) - \text{papéis} - p_5 \times \text{posição} - \frac{\text{grau}}{p_6} \quad (3)$$

4.3 Cenário Padrão

A região de sensoriamento foi discretizada em pontos com probabilidade de 10% de ocorrência de eventos por *round* para o caso centralizado e 5% para o caso distribuído. Ambas probabilidades são geradas pela distribuição uniforme. O número de pontos utilizados nesse processo é variável de acordo com a simulação, já que impacta diretamente na quantidade de eventos que ocorrem simultaneamente por *round*. O campo de sensoriamento possui área de 300 m², enquanto os raios de comunicação e sensoriamento são de 33.5 m². Foram utilizados 200 sensores e um nó sorvedouro. Por fim para o caso centralizado a simulação é finalizada se pelo menos 5% dos eventos gerados

Valores gerais	Parâmetro
Área de Simulação	$300 \times 300 \text{ m}^2$
Raio de Sensoriamento	33.5 m
Sensores	200
Nós sorvedouros	1
Algoritmo centralizado	Parâmetro
Critério de parada	5% dos eventos do <i>round</i> não sensorizados
Probabilidade de evento em cada ponto	10%
Algoritmo distribuído	Parâmetro
Critério de parada	20% dos eventos do <i>round</i> não sensorizados
Probabilidade de evento em cada ponto	5%

Tabela 4.3: Cenário padrão

em um único *round* não foram sensorizados e 20% para o caso distribuído. A tabela 4.3 apresenta um resumo destes critérios.

4.4 Distribuição dos sensores

Todas as RSSF utilizadas nos testes realizados eram conexas. Utilizaram-se quatro distribuições dos sensores na região de interesse, regular, aleatória, aleatória com maior probabilidade de sensores perto do nó sorvedouro e regular com reposicionamento de sensores para perto do nó sorvedouro. As duas últimas foram construídas a fim de se evitar que o nó sorvedouro perdesse contato com a rede, já que os nós ao seu redor possuem nível de centralidade bastante elevado. Todas as probabilidades necessárias para o reposicionamento de sensores foram geradas pela distribuição uniforme. A figura 4.1 apresenta um exemplo de cada uma das distribuições.

4.5 Métricas de comparação

Três critérios foram utilizados na comparação dos resultados: porcentagem de eventos não sensorizados, tempo de vida (medido de acordo com número de *rounds* da simulação) e total de mensagens enviadas por evento. Foram feitas análises variando-se o número de pontos de eventos (entre 300 e 900), número de possíveis tipo de eventos (entre dois

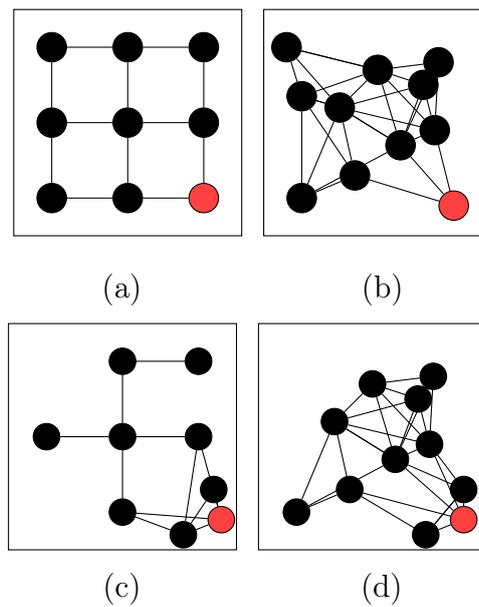


Figura 4.1: Distribuições dos sensores pela área de sensoriamento. (a) regular, (b) aleatória, (c) regular com reposicionamento de sensores para perto do nó sorvedouro, (d) aleatória com maior probabilidade de sensores perto do nó sorvedouro. O nó sorvedouro é representado pela cor vermelha.

e quatro) e tipo de distribuição do posicionamento dos sensores. O capítulo 5 apresenta os valores obtidos e faz uma análise dos mesmos.

Capítulo 5

Análise dos Resultados

As figuras 5.1 a 5.6 apresentam os resultados obtidos para porcentagem de eventos não sensoriados, tempo de vida e total de mensagens enviadas por evento, para simulações centralizadas e distribuídas. A seção 5.4 propõe um heurística aleatória distribuída e faz a comparação dos resultados obtidos com o algoritmo distribuído proposto. Foram analisadas quatro possíveis distribuições dos sensores pela rede: regular, aleatória, regular com reposicionamento (regular R) e aleatória com reposicionamento (aleatória R). Os experimentos foram realizados com dois tipos de eventos (gráficos letra *a*), três tipos tipos de eventos (gráficos letra *b*) e quatro tipos de eventos (gráficos letra *c*). Todos os gráficos apresentam valores médios, enquanto as barras de erro representam um intervalo de confiança de 95% para 10 execuções aleatórias. Por fim uma análise sobre as distribuições dos sensores na região de interesse é apresentada ao fim do capítulo.

5.1 Eventos Não Sensoriados

Para se realizar uma análise quanto ao número de eventos não sensoriados foram variados os números de pontos de eventos, impactando diretamente na quantidade de eventos por *round*, o número de tipos de eventos e a distribuição dos sensores na área de cobertura. As análises quanto as distribuições adotadas pelos sensores serão feitas na seção 5.5.

Observando-se a figuras 5.1 e 5.2 vemos que o número de pontos de eventos impactou linearmente na porcentagem de eventos não sensoriados, independente das distribuições dos sensores e da versão do algoritmo, centralizado ou distribuído. Tal resultado era esperado, já que quanto maior a quantidade de eventos maior a dificuldade do problema. Entretanto os valores obtidos se mostraram bastante robustos, princi-

palmente para as distribuições regular e regular com reposicionamento. Isto prova a eficiência do algoritmo mesmo quando o número médio de eventos por *round* é testado no pior caso. No geral nenhuma simulação obteve número de eventos não sensorizados maior que 5% e 8% para os casos centralizados e distribuídos respectivamente. Essa diferença entre os dois algoritmos é satisfatória, levando-se em conta a maior dificuldade na resolução do problema distribuído. Neste a possibilidade de erro é bastante elevada se comparado ao caso centralizado, devido a falta de informação dos sensores para que possam definir seus papéis.

O número de tipos de eventos não teve impacto nestes resultados. Podemos observar que os gráficos *a*, *b* e *c* da figura 5.1 apresentam valores proporcionalmente semelhantes entre as mesmas distribuições dos sensores, sendo o mesmo válido para a figura 5.2. Novamente isto prova que o algoritmo, centralizado e distribuído, se mantém eficiente até mesmo no pior caso (4 tipos de eventos e 900 pontos de eventos).

Por fim, devido a baixa taxa de eventos não sensorizados, podemos concluir que a função de prioridade adotada é bem sucedida na resolução do problema, já que como será mostrado adiante mantém também o tempo de vida da rede elevado. Um tempo de vida alto implica em uma boa distribuição dos papéis, de forma que o gasto de energia seja balanceado, e dessa forma atingindo todos os objetivos do algoritmo proposto.

5.2 Tempo de Vida

O tempo de vida medido em *rounds* serve como critério comparativo para os casos estudados, além de indicar uma carga média de eventos suportada pela rede até que a qualidade da solução não seja mais satisfatória. As figuras 5.3 e 5.4 apresentam os resultados encontrados para os casos centralizado e distribuído respectivamente. A grande diferença entre no número de *rounds* apresentados nestes dois casos é devido ao tempo necessário para configuração inicial da rede no caso distribuído, de forma que a maior parte do tempo seja consumida nesta parte e não no sensoriamento efetivo dos eventos. Ao contrário, no caso centralizado, todo o tempo da simulação é utilizado na detecção de eventos.

Como esperado o número de *rounds* decaiu com o aumento do número de pontos de eventos e de tipos de eventos. Obviamente mais eventos gerados na simulação criam obrigatoriamente uma demanda maior de energia pelos sensores. Um maior número de tipos de eventos faz com que mais papéis devam ser atribuídos criando-se também um maior gasto de energia.

De acordo com a figura 5.4 podemos notar que as distribuições com reposicio-

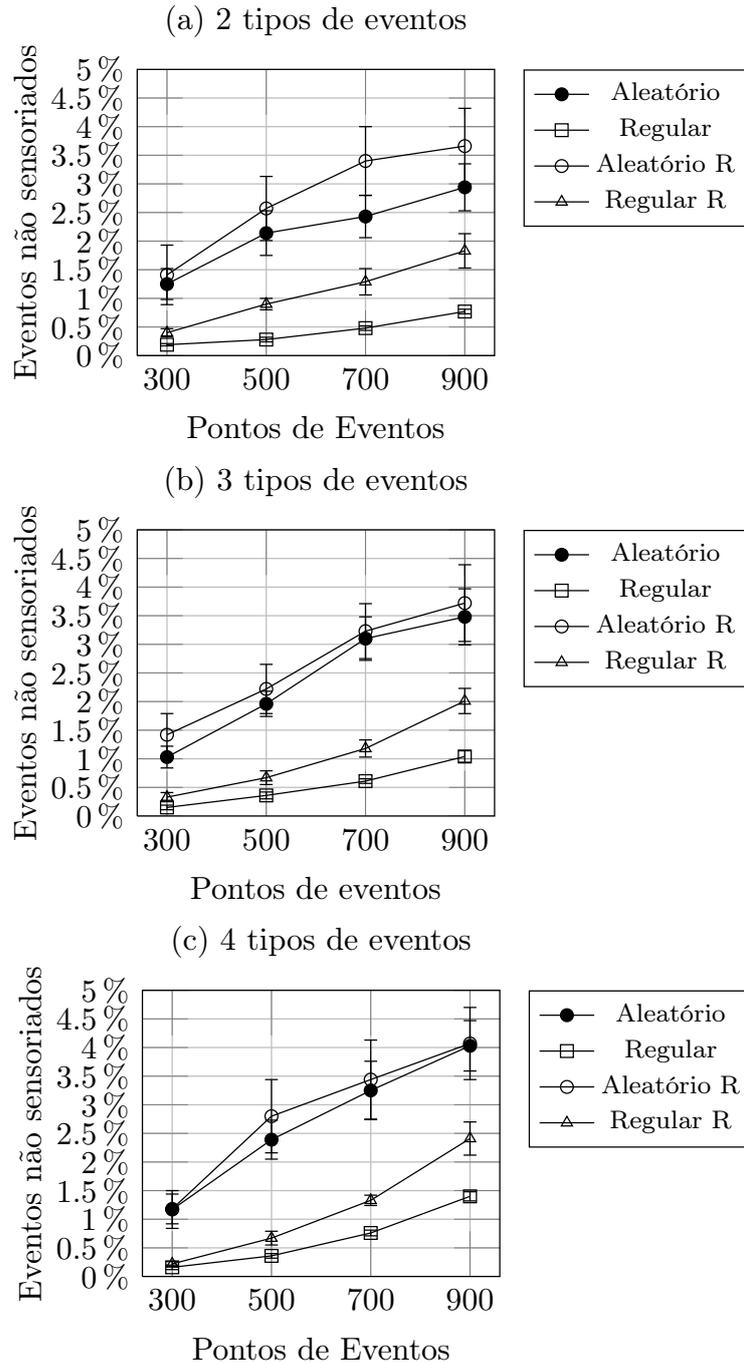


Figura 5.1: Simulação centralizada: gráficos comparativos entre a porcentagem de eventos não sensoriados e o número de pontos de eventos.

namento, ao contrário do esperado, obtiveram menor tempo de vida para simulações distribuídas. O reposicionamento de sensores ao redor do sorvedouro objetiva a garantia de comunicação da rede com o nó sorvedouro mesmo com a perda de sensores no decorrer da simulação. Entretanto podemos concluir que a retirada de tais vértices

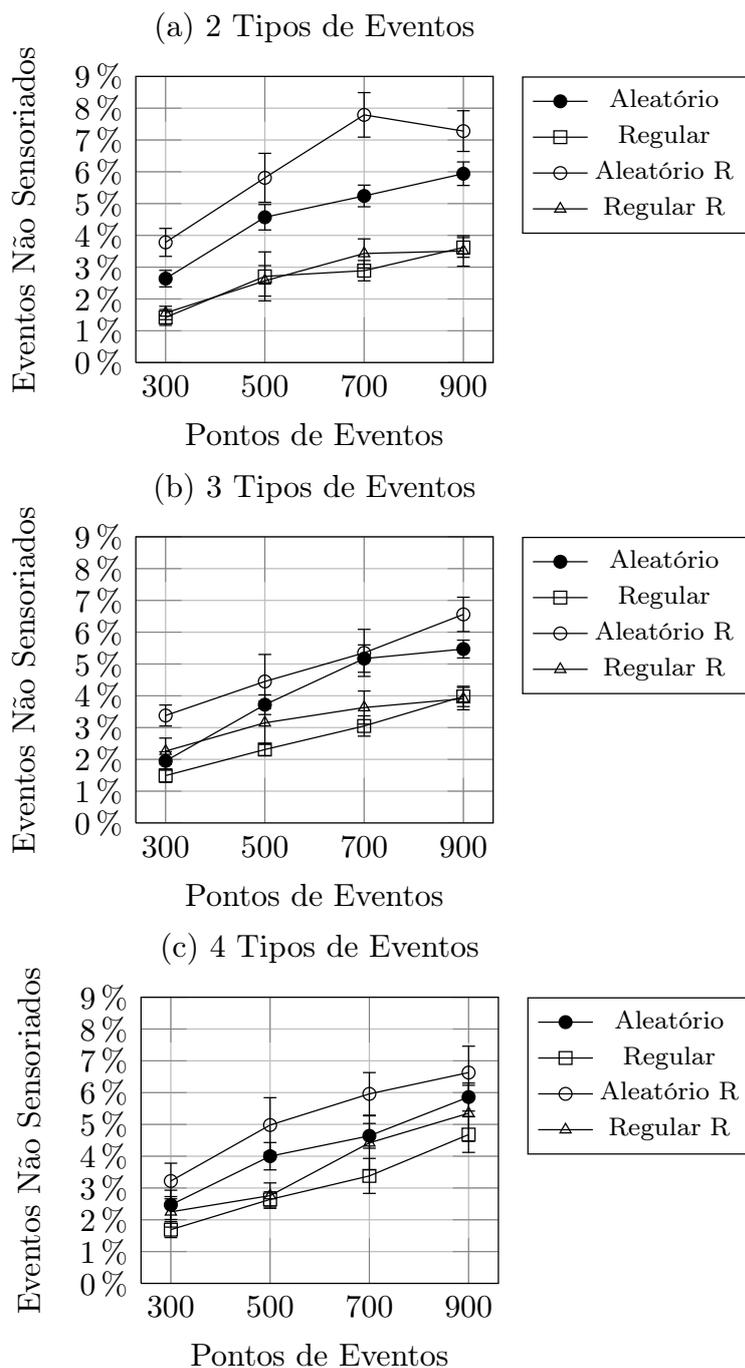


Figura 5.2: Simulação distribuída: gráficos comparativos entre a percentagem de eventos não sensorizados e o número de pontos de eventos.

faz com que haja regiões com baixa densidade de sensores e consequentemente o critério de parada seja atingido mais rapidamente. Para o caso centralizado, figura 5.3, observamos o caso inverso, onde as distribuições com reposicionamento apresentam tempo de vida maior ou igual as demais distribuições. Obviamente isto se deve ao

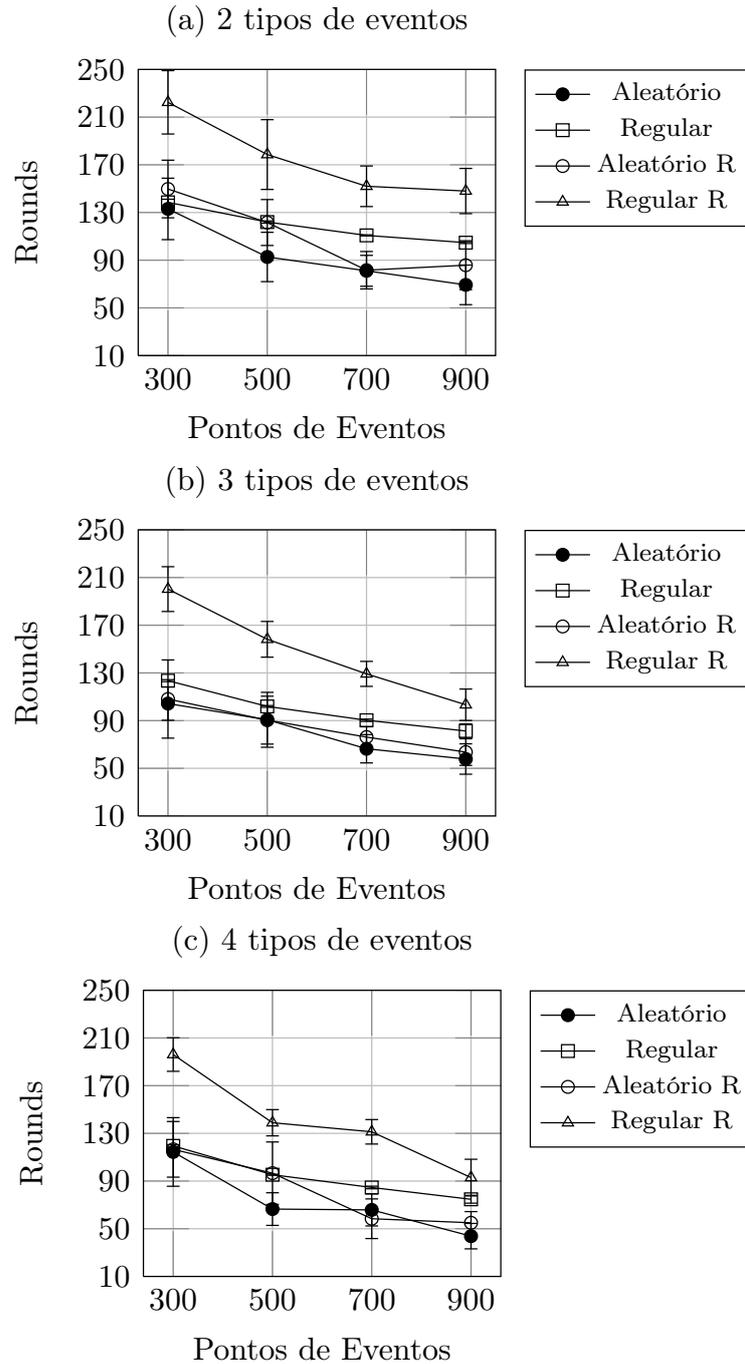


Figura 5.3: Simulação centralizada: gráficos comparativos entre o número de *rounds* e o número de pontos de eventos.

fato da simulação centralizada atingir seu limite de execução devido principalmente a falta de conexão com o nó sorvedouro, de forma que o reposicionamento de sensores possa estender seu tempo de vida. Já para o caso distribuído, devido ao alto número de mensagens enviadas entre os vértices, o critério de parada é alcançado pela falta de

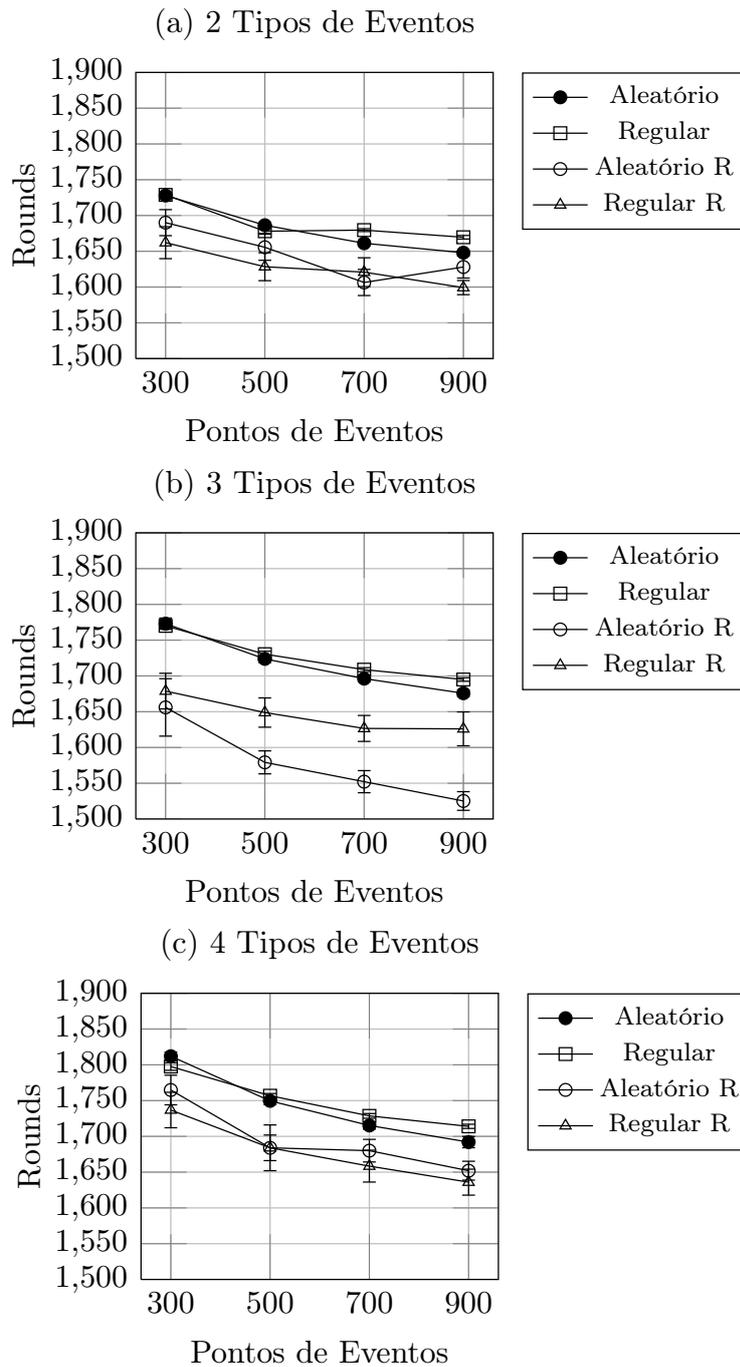


Figura 5.4: Simulação distribuída: gráficos comparativos entre a o número de *rounds* da simulação e o número de pontos de eventos.

bateria na rede, fazendo com que o reposicionamento dos sensores não contribua para melhora do resultado.

A tabela 5.1 mostra a média de eventos gerados de acordo com o número de pontos de eventos. Nesta podemos ver a grande quantidade de eventos suportados pela rede

Pontos de Eventos	Eventos Gerados	
	Simulação centralizada	Simulação distribuída
300	4329.17 ± 279.59	631.1 ± 29.04
500	5640.72 ± 381.93	863.9 ± 42.95
700	6577.78 ± 430.43	1071.79 ± 57.63
900	7357.70 ± 555.54	1293.84 ± 63.39

Tabela 5.1: Número de eventos gerados de acordo com o número de pontos de eventos. Erros foram calculados com 95% de confiança

até que o critério de parada da simulação seja alcançado. Tais valores implicam em um grande tempo de vida para uma ampla variedade de aplicações. Casos em que os número de eventos são menores podem resultar em um grande acréscimo no tempo de vida.

A menor taxa de eventos gerados para o caso distribuído se justifica devido ao alto gasto de energia com a troca de informações e definições de parâmetros iniciais da rede, como pontos de articulação, centralidade, densidade e etc. Durante a simulação distribuída grande parte ou praticamente todos os sensores da rede estão recebendo e enviando várias mensagens a cada *round*, um gasto bastante elevado que não é necessário no caso centralizado devido ao conhecimento global do estado de cada vértice.

5.3 Mensagens Enviadas por Evento

Primeiramente podemos analisar o número de mensagens enviadas por evento através da distribuição dos sensores na rede. De acordo com as figuras 5.5 e 5.6 as distribuições regular e regular com reposicionamento apresentaram valores superiores as distribuições aleatórias. Isso se deve ao fato das distribuições regulares não utilizarem ao máximo o raio de comunicação entre os sensores, ou seja, devido a forma com que foram dispostos é necessário que haja um maior número de retransmissões de dados até que a mensagem atinja o nó sorvedouro além de uma maior troca de informações para a definição de papéis no caso distribuído.

Dessa forma vemos que existe uma troca entre o número de mensagens enviadas e o número de eventos não sensoriados. Enquanto as distribuições regulares tem menor número de eventos não sensoriados e gastam mais mensagens na comunicação com o nó sorvedouro e na definição de papéis, as distribuições aleatórias requerem uma menor troca de mensagens porém podem resultar em uma maior perda de eventos sensoriados.

A melhor opção deve ser analisada de acordo com os principais objetivos da aplicação.

Por fim vemos que, como esperado, o número de tipos de eventos tem pouco impacto nestes resultados. Uma maior quantidade de tipos de eventos não implica em uma maior quantidade de eventos gerados, porém implicam em uma quantidade um pouco maior de sensores com papéis atribuídos, e por isso mais mensagens enviadas por evento.

5.4 Coloração de Vértices Aleatória (CVA)

Para comparação dos resultados implementamos um algoritmo distribuído, também baseado em coloração de vértices, e apresentamos suas vantagens e desvantagens sobre o algoritmo distribuído proposto. Denominaremos este de coloração de vértices não aleatória (CVNA). Para a CVA os critérios utilizados na definição de prioridade não precisam ser coletados, de forma que a prioridade de cada sensor seja um valor aleatório uniforme entre 0 e 1. O roteamento de informações é feito da mesma forma, definindo-se as rotas de encaminhamento de dados ao início da execução e caso algum sensor deixe a simulação.

Abaixo apresentamos uma comparação entre a porcentagem de eventos não sensorizados e o tempo de vida da rede. Não realizamos uma comparação entre o número de mensagens enviadas por evento pois a troca de mensagens para definição de papéis e o roteamento de informações é feito do mesmo modo em ambos os algoritmos. Porém, uma pequena queda no número total de mensagens enviadas pode ser observada pela CVA, devido a ausência do procedimento de configuração inicial da rede e da atualização dos pontos de articulação, centralidade e densidade de cada sensor.

5.4.1 Comparação: eventos não sensorizados

De acordo com as figuras 5.7 a 5.9 podemos observar que a CVA apresenta uma maior porcentagem de eventos não sensorizados. Para os casos com 3 tipos de eventos e 4 tipos de eventos, determinados pontos apresentam o dobro de eventos não sensorizados. Já para o caso com 2 tipos de eventos, podemos notar que não houve variação significativa entre o erro dos dois algoritmos.

O cálculo da prioridade de cada sensor tem como um dos objetivos aumentar a capacidade de monitoramento da rede. Para isso tentamos priorizar a atribuição de papéis em sensores com menor quantidade de vizinhos, com mais papéis detectados e que se encontram em regiões com maior probabilidade de eventos. Caso desconsideremos todas estas características, uma maior quantidade de eventos pode não ser

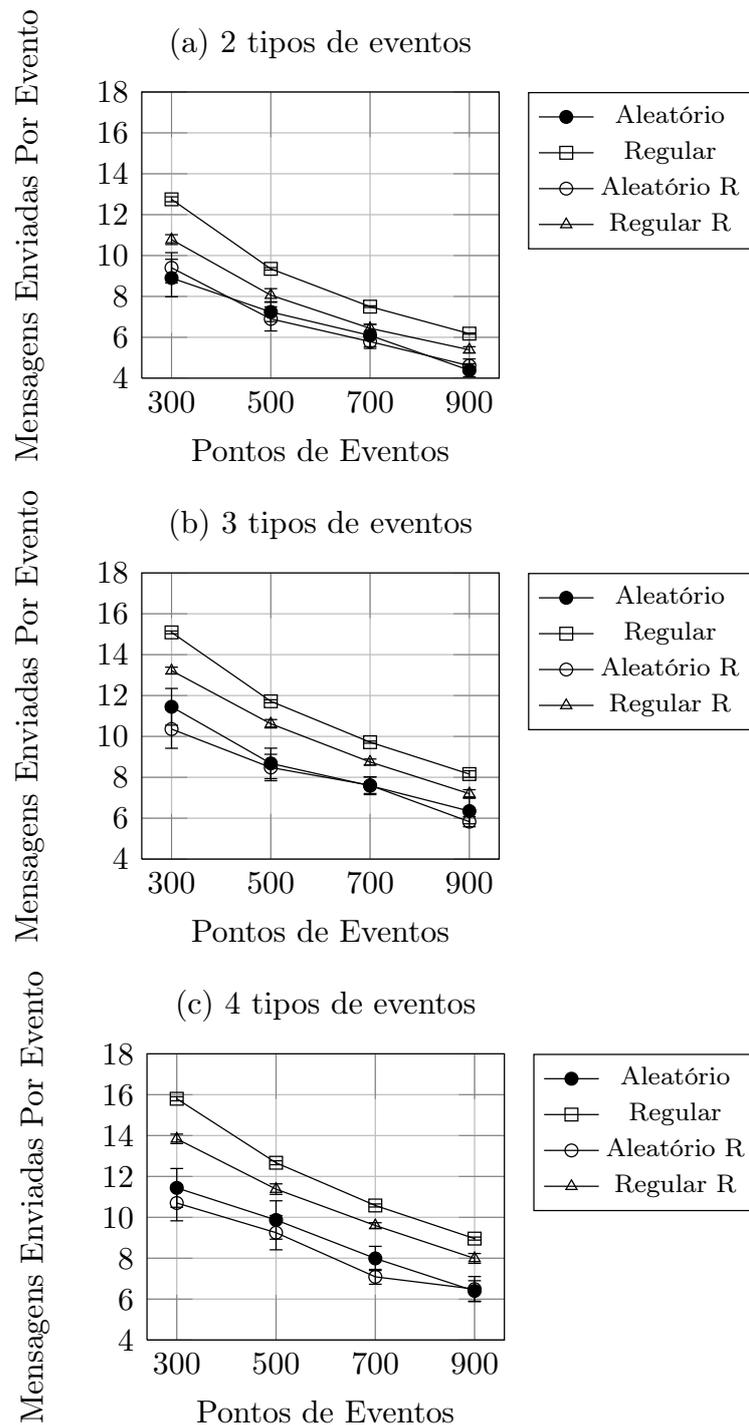


Figura 5.5: Simulação centralizada: gráficos comparativos entre o número de mensagens enviadas por evento e o número de pontos de eventos.

monitorada em detrimento de eventos isolados na região de interesse. Neste caso vemos que a porcentagem de eventos não sensoriados tende a aumentar consideravelmente, principalmente em problemas com maior número de tipos de eventos.

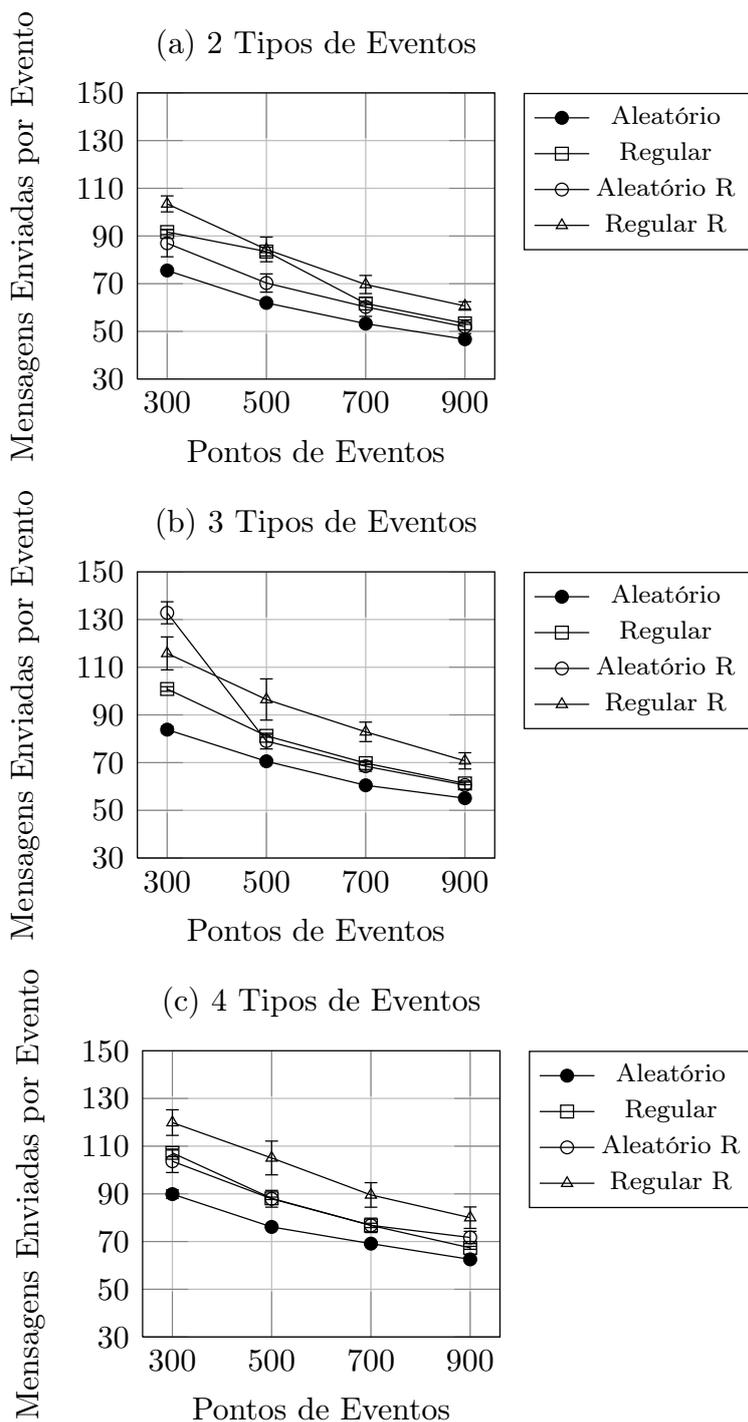


Figura 5.6: Simulação distribuída: gráficos comparativos entre o número de mensagens enviadas por evento e o número de pontos de eventos.

Por fim podemos garantir que a CVNA apresenta uma maior garantia de eventos sensorizados. Entretanto, devido a não configuração da rede pela CVA, veremos que esta apresenta maior tempo de vida em determinados casos. Esta característica não

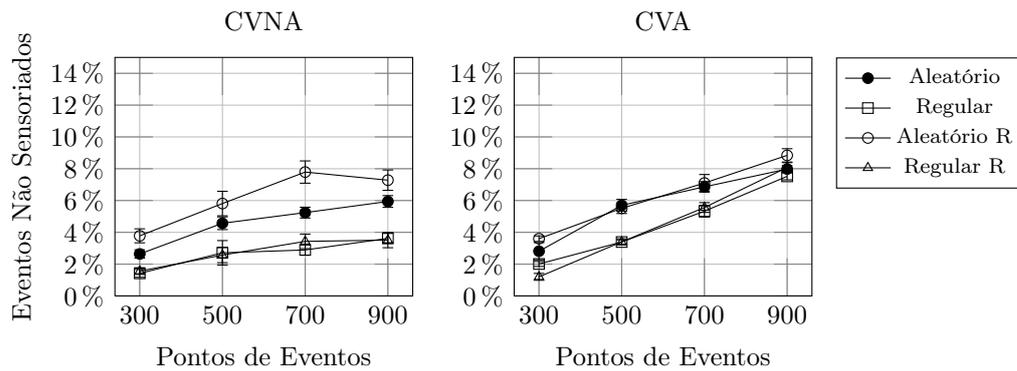


Figura 5.7: Simulação aleatória: comparação da porcentagem de eventos não sensoriados com 2 tipos de eventos

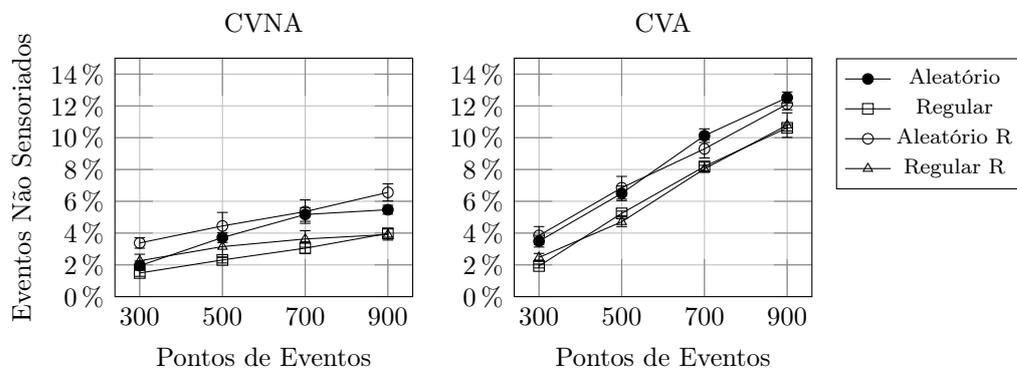


Figura 5.8: Simulação aleatória: comparação da porcentagem de eventos não sensoriados com 3 tipos de eventos

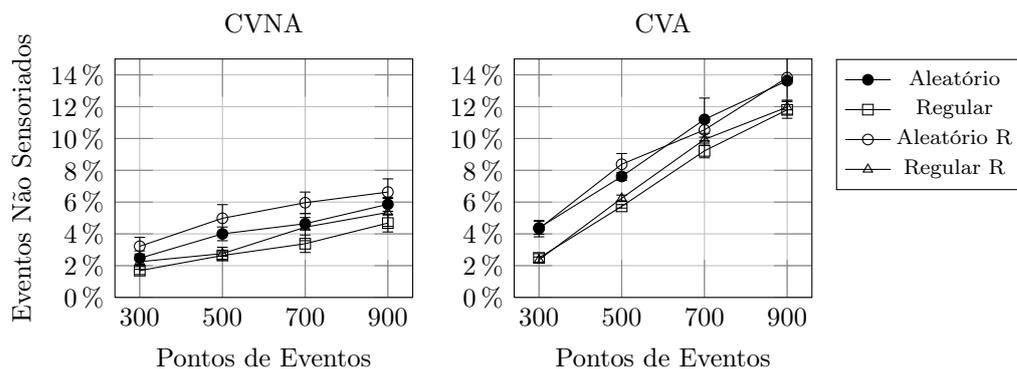


Figura 5.9: Simulação aleatória: comparação da porcentagem de eventos não sensoriados com 4 tipos de eventos

se mantém a medida que a dificuldade do problema é acrescida. A CVNA é superior também em tempo de vida nos casos com grande número de eventos ou com maior quantidade de tipos de eventos.

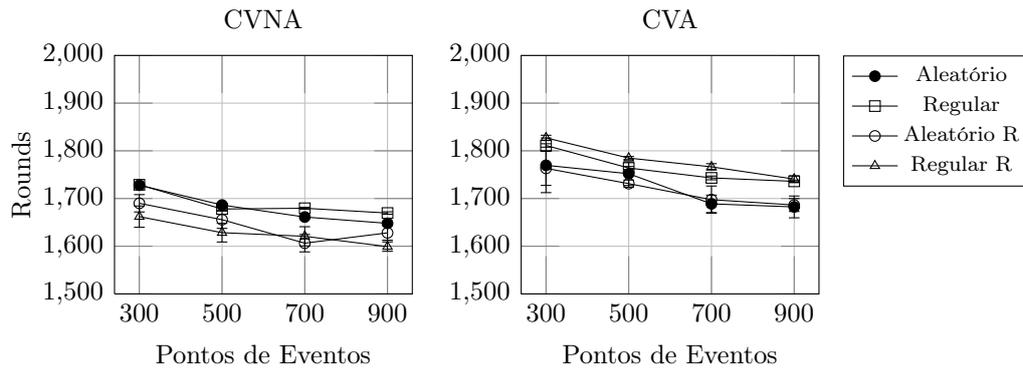


Figura 5.10: Simulação aleatória: comparação do tempo de vida com 2 tipos de eventos

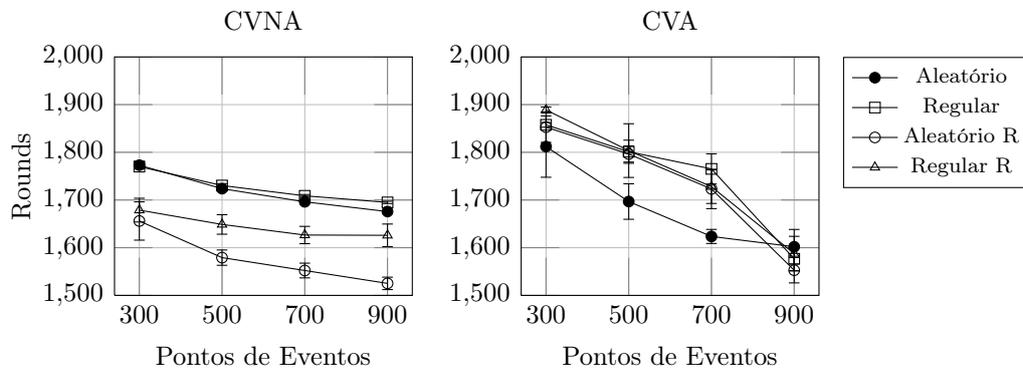


Figura 5.11: Simulação aleatória: comparação do tempo de vida com 3 tipos de eventos

5.4.2 Comparação: tempo de vida

De acordo com as figuras 5.10 a 5.12 vemos que a CVA apresenta comportamento superior exceto para casos com grande quantidade de pontos de eventos e número de tipos de eventos. Podemos atribuir este resultado ao fato da configuração inicial da rede (centralidade, densidade, articulação) e a atualização destes mesmos critérios não ser realizada pela CVA. Com isso é possível minimizar o número de mensagens enviadas e estender o tempo de vida da rede.

A economia de energia dos sensores de articulação, de alta centralidade e baixa bateria, se mostra eficiente para o CVNA nos casos em que a CVA apresenta piores resultados. Nestes casos o tempo de vida da rede depende fortemente destes sensores, já que a atribuição de papéis é mais intensa e a comunicação com o nó sorvedouro tende a ser o principal fator delimitador do número total de rounds.

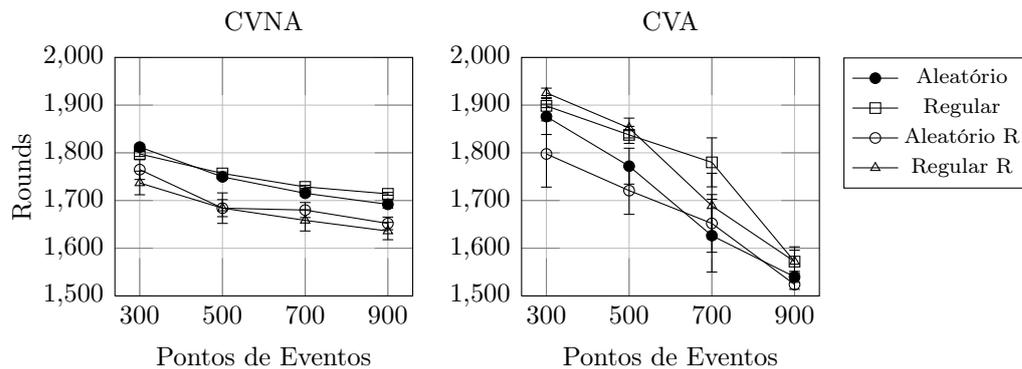


Figura 5.12: Simulação aleatória: comparação do tempo de vida com 4 tipos de eventos

5.4.3 Considerações

Até onde se sabe nenhum outro trabalho apresenta uma abordagem para resolução da atribuição de papéis com uma grande carga de eventos simultâneos. Logo, implementamos uma heurística aleatória para o caso distribuído para servir de base na comparação dos resultados.

Através desta avaliação buscamos investigar a eficiência da CVNA com relação à porcentagem de eventos não sensoriados e tempo de vida da rede. Observamos nas seções 5.4.1 e 5.4.3 que o custo para definição e atualização de pontos de articulação, centralidade e densidade é bastante alto, de modo que seu custo compensou o tempo de vida apenas nos problemas com grande número e tipo de eventos.

Já a porcentagem de eventos não sensoriados é menor em todos os casos para a CVNA. Isso comprova a eficiência da função de prioridade que tenta maximizar o monitoramento de informações. Recomendamos o uso desta função para aplicações em que se deseja alcançar um alta taxa de eventos sensoriados. Obviamente esta melhoria acarreta em um custo de energia que pode não ser compensado por uma heurística aleatória em problemas menores.

5.5 Distribuições dos sensores na região de interesse

A forma como os sensores são distribuídos na região de sensoriamento tem grande impacto na solução fornecida pelo algoritmo. De acordo com a 5.1 e 5.2 vemos que as distribuições regulares são mais eficientes no sensoriamento dos eventos. Obviamente sensores dispostos de forma regular implicam em uma cobertura homogênea dos eventos, não permitindo que existam áreas com baixa densidade de sensores. Outra

vantagem é a não existência de sensores de articulação, implicando em uma menor probabilidade de perda de comunicação com o nó sorvedouro e conseqüentemente, no caso centralizado, um maior tempo de vida, figura 5.3 . A principal desvantagem neste caso é o maior gasto no envio de mensagens. Como vemos nas figuras 5.5 e 5.6 as distribuições regulares apresentam valores superiores as distribuições aleatórias em todos os casos.

Apesar da diferença de resultados entre as distribuições implementadas, de acordo com a tabela 5.1 vemos que todas as simulações tiveram tempo de vida elevados em relação ao total de eventos da simulação. Além disso, como exemplificado na seção 5.1, o número de eventos não sensoriados também é bastante baixo independente da distribuição. Isso mostra que a escolha da distribuição adotada depende fortemente do problema ao qual o algoritmo deve ser aplicado. Caso haja prioridade por uma maior quantidade de eventos sensoriados as distribuições regulares são mais adequadas, podendo também elevar o tempo de vida no caso centralizado. Entretanto caso a comunicação com o nó sorvedouro deva ser mais rápida as distribuições aleatórias se mostraram mais eficientes, além de apresentarem menor gasto com o envio de mensagens.

Capítulo 6

Conclusão

Este trabalho apresentou um algoritmo baseado em uma heurística para o problema de atribuição de papéis em redes de sensores sem fio. Como mostrado, este problema possui uma forte relação com o problema de coloração de vértices, podendo até mesmo ser tratado como uma extensão deste. Tal característica possibilita o desenvolvimento de algoritmos de atribuição de papéis baseados em algoritmos de coloração de vértices.

O algoritmo proposto apresentou uma função de prioridade que tenta minimizar o uso de energia e maximizar a cobertura de eventos através da ordem em que os papéis são atribuídos. Com isso, foi possível poupar sensores de maior importância na comunicação da rede e, dessa forma, aumentar o tempo de vida da simulação.

Devido à impossibilidade de haver um agente global em grande parte dos problemas de atribuição de papéis, duas versões foram propostas: uma centralizada e outra distribuída. Ambas utilizam o mesmo algoritmo, mas a forma de coleta de informações se dá de forma bastante diferente. O processo de decisão na atribuição de papéis deve ser feito localmente através da troca de mensagens entre sensores vizinhos. O mesmo é válido para definição de sensores de articulação, densidade, centralidade e rotas de encaminhamento de cada vértice. Tal limitação na quantidade de informações disponíveis e dificuldade de obtenção das mesmas acarreta em uma grande dificuldade de resolução do problema.

Para realização de experimentos computacionais, diversos parâmetros foram adotados de modo que seus valores possam variar de acordo com os principais objetivos da aplicação. Foram avaliadas diferentes instâncias quanto ao número de eventos gerados e número de tipos de eventos. Além disso, foram adotadas quatro distribuições dos sensores pela área de sensoriamento. Diferentemente de outros trabalhos estudados, aplicamos uma alta taxa de eventos simultâneos por *round*, mantendo a taxa de erro abaixo de 5% e 8%, para os casos centralizado e distribuído, respectivamente. Tais

resultados tornam o algoritmo desenvolvido apto a uma maior variedade de aplicações, já que sua utilização para poucos eventos também é válida.

Trabalhos futuros incluem a análise quanto a adaptação de outras heurísticas do problema de coloração de vértices para o problema de atribuição de papéis, tanto centralizado ou distribuído. A atribuição de papéis é uma das funções básicas em redes de sensores. Assim, outra linha de investigação é o estudo da atribuição de papéis conjuntamente com outras funções de rede como fusão de dados.

Referências Bibliográficas

- Akyildiz, I.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.
- Bhardwaj, M. & Chandrakasan, A. (2002). Bounding the lifetime of sensor networks via optimal role assignments. *In Proc. of the 21st IEEE INFOCOM*.
- Brèlaz, D. (1979). New methods to color the vertices of a graph. *Communications of the ACM*, 22:251–256.
- Dijkstra, E. (1959). A note on two problems in connection with graphs. *Numerische Math*, 1:269–271.
- Frank, C. & Römer, K. (2005). Algorithms for generic role assignment in wireless sensor network. *SenSys*, pp. 230–242.
- Frank, C. & Römer, K. (2006). Solving generic role assignment exactly. *In Proc. of the 14th Int. Wkrshp. on Parallel and Distributed Real-Time Systems (WPDRTS)*.
- Krishnamachari, B.; Estrin, D. & Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. *In Int'l Workshop of Distributed Event Based Systems (DEBS'02)*, pp. 575–578.
- Lofti, V. & Sarin, S. (1986). A graph coloring algorithm for large scale scheduling problems. *Computers and Operation Research*, 13 (1):27–32.
- Loureiro, A.; Nogueira, J.; Ruiz, L.; Nakamura, E.; Serodio, C. & Mini, R. (2003). Redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores*, pp. 179–226.
- Mini, R.; Loureiro, A. & Nath, B. (2005). A state-based energy dissipation model for wireless sensor nodes. *In 10th IEEE Conference on Emerging Technologies and Factory Automation*, 2:543–550.

- Nakamura, E.; Ramos, H.; Villas, L.; Oliveira, H.; Aquino, A. & Loureiro, A. (2009). A reactive role assignment for data routing in event-based wireless sensor network. *Computational Networks*, 53:1980–1996.
- Odell, J.; Parunak, H. V. D.; Brueckner, S. & Sauter, J. (2003). Changing roles: Dynamic role assignment. *Journal of Object Technology*, 2(5):77–86.
- Pranay, C. (1998). An optimal distributed algorithm for finding articulation points in a network. *Comput. Commun.*, 21(18):1707–1715.
- Sinalgo (2007). Simulator for network algorithms. <http://www.disco.ethz.ch/projects/sinalgo/>.
- Souza, F. (2007). Algoritmos para o problema de atribuição de papéis em redes de sensores sem fio. Dissertação de mestrado, Universidade Federal de Minas Gerais.
- Welsh, A. & Powell, B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10:85–86.
- Wigderson, A. (1983). Improving the performance guarantee of approximate graph coloring. *J. Assoc. Comput. Mach.*, 30:729–735.
- Zhou, C. & Krishnamachari, B. (2003). Localized topology generation mechanisms for self-configuring sensor networks. *In Proceedings of the IEEE Globecom'03*, pp. 1269–1273.