



UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Departamento de Engenharia Elétrica

Mestrado em Engenharia Elétrica

**HEURÍSTICAS MONO E MULTIOBJETIVO
PARA O PROBLEMA DE COBERTURA E
CONECTIVIDADE DE REDES DE
SENSORES SEM FIO PLANAS**

Flávio Vinícius Cruzeiro Martins

DISSERTAÇÃO DE MESTRADO

Belo Horizonte
5 de agosto de 2009

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Departamento de Engenharia Elétrica

Flávio Vinícius Cruzeiro Martins

**HEURÍSTICAS MONO E MULTIOBJETIVO PARA O PROBLEMA
DE COBERTURA E CONECTIVIDADE DE REDES DE SENSORES
SEM FIO PLANAS**

Trabalho apresentado ao Programa de Mestrado em Engenharia Elétrica do Departamento de Engenharia Elétrica da UNIVERSIDADE FEDERAL DE MINAS GERAIS como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: *Prof. Dr. Ricardo H. C. Takahashi*

Co-orientador: *Prof. Dr. Geraldo Robson Mateus*

Belo Horizonte
5 de agosto de 2009

AGRADECIMENTOS

Tantas pessoas, tantos motivos, mas nada seria possível ou faria sentido se não dar toda honra e toda glória a quem é o Criador e dono da minha vida, Jesus Cristo. Muito obrigado Senhor por estar comigo em toda essa caminhada, pelo cuidado e pela forma na qual me conduziu. Por me acompanhar em cada etapa, sendo o pilar da minha vida, dando todo o sustento necessário e ânimo nos momentos difíceis. Tudo que tenho, tudo que sou, e o que eu vier a ser será para honrar e glorificar a Ti.

Tenho muito a agradecer a meus pais por todo apoio e incentivo para completar mais esse objetivo, por depositarem toda confiança e acreditar que chegar aqui era possível. Minha mãe com sua forma tão amorosa e carinhosa, meu pai com toda sua preocupação e cuidado. Claro que não posso deixar de falar das minhas irmãs, Patrícia e Monica que sempre acharam que eu “estudo demais”, pois então maninhas é assim que cheguei até aqui! Minha linda avozinha que sempre rogou proteção ao nosso Deus, para que Ele me guardasse nestas rodovias tão perigosas nas minhas idas e vindas à grande amada terra natal Conselheiro Pena. Ao meu avô com sua forma tão brincalhona e divertida mostrando o quão bom é viver!

Agradeço muito aos amigos da ABU (Aliança Bíblia Universitária) que estiveram sempre comigo, divertindo e compartilhando excelentes momentos. Não posso deixar de falar dos meus irmãos da Igreja Presbiteriana Floresta. Eles que com muito amor estiveram junto comigo, orando e buscando a Deus para que eu pudesse finalizar mais esta etapa. Também com muita paciência souberam entender o quão era importante terminar este mestrado e que para isso era preciso que eu abrisse mão de algumas obrigações.

Tenho muito a agradecer ao meu orientador Ricardo H. C. Takahashi, que desde o segundo semestre do meu curso de graduação na UFMG vem me acompanhando, ensinado e mostrando os caminhos de como ser um pesquisador. Agradeço também ao meu co-orientador Geraldo Robson Mateus, que também acompanhou alguns dos meus passos na graduação, tanto como orientador como também professor, ajudando-me e incentivando-me a continuar nesta caminhada. Não posso deixar de mencionar a Fabíola Guerra Nakamura, que com toda sua alegria e disposição me acompanhou durante um longo período e me fez apaixonar pelo tema abordado nesta dissertação.

Em fim agradeço muito uma pessoa em especial que tem sido uma verdadeira companheira, dando-me forças, acreditando que todo sacrifício vale a pena e depositando em mim toda confiança, e sempre tendo certeza que no fim tudo dá certo. Foi minha auxiliadora e incentivadora, dando-me seu colo e carinho, ergueu-me nos momentos de desânimo e principalmente me deu muito amor. Agradeço pela paciência e por saber entender o quão era importante finalizar esta etapa. Mesmo em meio da correria com o fim do mestrado nos casamos e agora somos uma linda família. Então, Aline Maia Cruzeiro Martins, meu muito obrigado! Te amo!!!

*“Ó profundidade das riquezas, da sabedoria e da ciência de Deus! Quão
inescrutáveis são os seus juízos e quão impenetráveis os seus caminhos!
Pois quem conheceu a mente do Senhor? Ou quem se fez o seu
conselheiro?
Ou quem lhe deu primeiro, para que lhe seja retribuído?
Pois dele, por ele e para ele são todas as coisas; a ele seja dada a glória
para sempre. Amém.”*

—BÍBLIA SAGRADA (Romanos 11:33-36)

RESUMO

Este trabalho aborda o Problema de Cobertura e Conectividade em Redes de Sensores sem Fio (RSSF), formulando-o de diferentes maneiras como problemas de otimização mono-objetivo e multiobjetivo. Em todos os casos, é considerada a questão da reconfiguração dinâmica da rede realizada à medida em que ocorram falhas na rede devidas ao esgotamento da energia dos nós sensores.

No caso da formulação mono-objetivo, a heurística proposta se baseia em um Algoritmo Genético (AG) para fazer a escolha inicial dos nós a serem ativados e para atualizar essa escolha em alguns momentos, quando a rede se afasta significativamente de uma configuração ótima. Essa heurística envolve ainda uma busca “gulosa” para realizar reconfigurações locais de forma rápida imediatamente após a ocorrência de cada falha. Tal heurística é comparada com uma formulação exata do mesmo problema, tratada por meio de Programação Linear Inteira, a qual foi resolvida utilizando um pacote comercial de otimização. Embora, conforme esperado, a heurística proposta não tenha sido capaz de determinar as soluções ótimas exatas para o problema, verificou-se que os ganhos de tempo de execução obtidos com a heurística tornam plausível sua aplicação em problemas práticos mesmo em situações envolvendo um grande número de nós sensores. No que diz respeito a desenvolvimentos realizados especificamente no âmbito desta dissertação, esta parte do trabalho apresenta alguns aperfeiçoamentos sobre algoritmos que já vinham sendo construídos anteriormente, com a participação deste autor.

A principal contribuição desta dissertação é a proposição de uma formulação multi-objetivo para o problema de cobertura e conectividade de RSSF, associada a uma correspondente heurística para sua solução. O princípio em que se fundamenta tal formulação é o de que é possível realizar uma troca entre o tempo de funcionamento da rede e o percentual da área coberta. Um Algoritmo Genético multiobjetivo, baseado no NSGA (Non-dominated Sorting Genetic Algorithm), foi desenvolvido para substituir o AG mono-objetivo, de forma a determinar, em paralelo, um conjunto de soluções não-dominadas em relação a tais objetivos. Um algoritmo simples de tomada de decisão é empregado de maneira a escolher a configuração a ser adotada pela rede a cada instante. Resultados de simulação mostram que a introdução de pequenas relaxações na área coberta permitem obter significativos aumentos no tempo de funcionamento da rede. O tempo de execução da heurística multiobjetivo proposta é similar ao da heurística mono-objetivo para redes de mesmo tamanho.

Palavras-chave: Otimização Multiobjetivo, Redes de Sensores Sem fio, Otimização Combinatória.

ABSTRACT

This work deals with the Problem of Coverage and Connectivity in Wireless Sensor Networks (WSN), formulating it in different ways as mono-objective and multiobjective optimization problems. The issue of dynamic reconfiguration of the network to be performed as soon as some failures occur in the network nodes is considered here in all the cases.

In the case of mono-objective formulation, the proposed heuristics is based on a Genetic Algorithm (GA) that performs the initial choice of the nodes to be activated and that also performs an update of this choice in some moments, when the network is significantly far from the optimal configuration. This heuristics is compared with an exact formulation of the same problem, which is stated as an Integer Linear Program and solved with a commercial optimization package. Although, as expected, the proposed heuristics has not been able to solve the problem up to the exact optimality, it has been found that the gains in the computation time obtained by the heuristics make its application to practical problems feasible, even in situations involving a large number of sensor nodes. The results presented in this part of this dissertation are related to algorithms that have been developed earlier, with the participation of this author, and that have been enhanced specifically in the context of this work.

The main contribution of this dissertation is the proposition of a multiobjective formulation for the RSSF coverage and connectivity problem, associated to a corresponding heuristics for dealing with it. This formulation is based on the observation that it is possible to perform a trade-off between the duration of working time of the network and the percentage of its coverage area. A multiobjective GA, based on the NSGA (Non-dominated Sorting Genetic Algorithm), has been developed in order to replace the mono-objective GA, allowing the parallel computation of a set of solutions which is non-dominated in relation to such objectives. A simple decision-making algorithm is employed in order to choose the configuration to be adopted in each moment by the network. Simulation results show that the introduction of rather small relaxations in the coverage allow significant gains in the network working time. The computational time of the proposed multiobjective heuristics is similar to the one of the mono-objective heuristics, for networks of similar size.

Keywords: Multiobjective Optimization, Wireless Sensor Networks, Combinatorial Optimization.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Objetivo	3
1.2 Metodologia Adotada e Resultados Obtidos	4
Capítulo 2—O Problema: Definições e Modelos	7
2.1 Definição do Problema	9
2.2 Modelo Matemático	11
Capítulo 3—Abordagem Mono-Objetivo	19
3.1 Algoritmo Genético	19
3.2 Caminho Mínimo	20
3.3 Algoritmo Híbrido Mono-Objetivo	21
3.3.1 Algoritmo Genético Implementado	23
3.3.1.1 Codificação	23
3.3.1.2 Geração da População Inicial	24
3.3.1.3 Cruzamento	26
3.3.1.4 Mutação	28
3.3.1.5 Seleção	29
3.3.1.6 Elitismo	29
3.3.1.7 Cálculo da Função Objetivo	30
3.3.2 <i>Dijkstra</i> Implementado	30
3.3.3 Algoritmo Híbrido	31
3.3.3.1 Local <i>Online</i>	32
3.4 Resultados Computacionais	35
3.4.1 Algoritmo Periódico	38
3.4.2 Algoritmo Local <i>Online</i>	41
3.4.3 Solução exata pelo modelo matemático	41
3.4.4 Resultado dos Testes Considerando Falhas Mecânicas e por Energia	41
3.4.4.1 Análise dos Resultados	41
3.4.5 Resultado dos Testes Considerando Falhas apenas por Energia . .	46
3.4.5.1 Análise dos Resultados	46

Capítulo 4—Abordagem Multiobjetivo	53
4.1 Algoritmos Genéticos Multiobjetivo	55
4.1.1 NSGA	55
4.1.2 NSGA II	56
4.2 Algoritmo <i>Online</i> Híbrido Multiobjetivo	57
4.2.1 Algoritmo Genético Multiobjetivo	57
4.2.1.1 Seleção no AG_MO	59
4.2.1.2 Elitismo no AG_MO	61
4.2.1.3 Decisão no conjunto Pareto-ótimo	61
4.3 Resultados Computacionais	63
4.3.1 Resultado dos Testes Considerando Falhas Mecânicas e por Energia	64
4.3.1.1 Análise dos Resultados	64
4.3.2 Resultado dos Testes Considerando Falhas apenas por Energia . .	68
4.3.2.1 Teste comparativo entre diferentes escolhas para ϕ . . .	70
4.3.2.2 Análise dos Resultados	71
Capítulo 5—Conclusão	79
5.1 Trabalhos Futuros	81
Apêndice A—Diferenças do NSGA e NSGA-II Aplicados na Rede de Sensores Sem Fio	83

LISTA DE FIGURAS

2.1	Principais componentes de um nó sensor.	8
2.2	Nó sensor Mica 2	8
2.3	Exemplo para identificar os raios de sensoriamento (RS) e comunicação (RC)	9
2.4	Exemplo de pontos de demanda em uma área de $100m^2$	10
2.5	Exemplo de pontos de demanda em uma área de $2500m^2$	10
2.6	Problema Dinâmico de cobertura e Conectividade em RSSFs Planas.	11
2.7	Explicação sobre a potência de recepção e transmissão.	15
3.1	Grafo para exemplificar o problema do caminho mínimo	21
3.2	Exemplo de um cromossomo de 8 bits	23
3.3	Ilustração da codificação de um cromossomo na RSSF.	24
3.4	Exemplo de cruzamento, com um ponto de corte, em um AG.	27
3.5	Exemplo de mutação em um AG (mutação do bit 5)	28
3.6	Formas de mutação utilizada na “mutação de controle”	29
3.7	Etapas do Algoritmo Local <i>Online</i>	34
3.8	Curva exponencial que modela a probabilidade de falha de um nó sensor no período de tempo t	36
3.9	Ilustração da distribuição usada para 36, 49, 64, 81 e 100 sensores em uma área de $2500m^2$	37
3.10	Exemplo da falha de um nó sensor, importante para o roteamento, em uma rede com abordagem periódica	40
3.11	Gráficos de resultado para 36 sensores na abordagem mono-objetivo com falhas mecânicas e por energia	42
3.12	Gráficos de resultado para 49 sensores na abordagem mono-objetivo com falhas mecânicas e por energia	43
3.13	Gráficos de resultado para 64 sensores na abordagem mono-objetivo com falhas mecânicas e por energia	44
3.14	Gráficos de resultado para 81 sensores na abordagem mono-objetivo com falhas mecânicas e por energia	45
3.15	Gráficos de resultado para 36 sensores na abordagem mono-objetivo com falhas apenas por energia	47
3.16	Gráficos de resultado para 49 sensores na abordagem mono-objetivo com falhas apenas por energia	48
3.17	Gráficos de resultado para 64 sensores na abordagem mono-objetivo com falhas apenas por energia	49
3.18	Gráficos de resultado para 81 sensores na abordagem mono-objetivo com falhas apenas por energia	50

3.19	Gráficos de resultado para 100 sensores na abordagem mono-objetivo com falhas apenas por energia	51
3.20	Gráfico mostrando o resumo da Tabela 3.2.	52
4.1	Exemplo de um conjunto Pareto-ótimo e de dominância	54
4.2	Cascas do Pareto para seleção por roleta	56
4.3	Roleta formada segundo os valores de aptidão do indivíduos da Tabela: 4.1	60
4.4	Gráfico da função de aptidão. Equação: 4.1	61
4.5	Curva Pareto-ótima fornecida na primeira unidade de tempo. Exemplo de escolha de uma solução para o caso de 100 sensores.	62
4.6	Gráficos de resultado para 36 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$	65
4.7	Gráficos de resultado para 49 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$	66
4.8	Gráficos de resultado para 64 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$	67
4.9	Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$	68
4.10	Gráficos de resultado para 36 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$	69
4.11	Gráficos de resultado para 49 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$	70
4.12	Gráficos de resultado para 64 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$	71
4.13	Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$	72
4.14	Gráficos de resultado para 100 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$	73
4.15	Gráfico demonstrando o resumo da Tabela 4.2.	74
4.16	Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas apenas por energia, com valores de 10, 30, 40, 70, 80 e 95, para ϕ	75
4.17	Gráficos de resultado para 100 sensores na abordagem multiobjetivo com falhas apenas por energia, com valores de 10, 30, 40, 70, 80 e 95, para ϕ	76
4.18	Gráfico traçado conforme a Equação 3.4, fazendo a variação de f_{cob} . Os valores para o raio do sensor e a área a ser monitorada foram os mesmos utilizados nos testes realizados, ou seja, $r_s = 15m$ e $a_m = 2500m^2$	78
A.1	Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas somente por energia, para $\phi = 95$	84
A.2	85
A.3	86

LISTA DE TABELAS

3.1	Consumo de corrente do nó sensor com a transmissão	39
3.2	Tabela de Tempo (em segundos) gasto por cada algoritmo Mono-Objetivo. A coluna “Q.UT” representa a quantidade de unidades de tempo em que o algoritmo manteve alguma cobertura.	46
4.1	Tabela exemplificando indivíduos em uma roleta	74
4.2	Tabela de Tempo (em segundos) gasto para todas as instâncias do CPLEX, Mono-Objetivo e Multiobjetivo. A coluna “Q.UT” representa a quantidade de unidades de tempo que o algoritmo manteve alguma cobertura.	74

CAPÍTULO 1

INTRODUÇÃO

*Comece fazendo o que é necessário, depois o que é possível, e de repente
você estará fazendo o impossível.*

—SÃO FRANCISCO DE ASSIS

*Logo que, numa inovação, nos mostram alguma coisa de antigo, ficamos
sossegados.*

—FRIEDRICH NIETZSCHE

Nos últimos anos ocorreram muitos avanços na miniaturização de componentes eletrônicos, evoluções dos processadores embutidos, rádios e de sistemas micro-mecânicos, possibilitando então a evolução da comunicação sem fio. Conseqüentemente, a partir desta evolução foi sugerido recentemente um novo tipo de rede ad hoc, as Redes de Sensores sem Fio (RSSF) [Savvides et al., 2001]. Considerada um tipo especial de rede móvel ad hoc (MANET - Mobile Ad hoc Networks), uma RSSF é composta por dispositivos autônomos e compactos que podem ter capacidades de sensoriamento, processamento e comunicação. A estes dispositivos dá-se o nome de nós sensores [Nakamura, 2003]. Na maioria das vezes uma RSSF é composta por centenas ou milhares de nós sensores, que são formados por um processador, um rádio que permite comunicação sem fio e uma placa de sensoriamento [Akyildiz et al., 2002].

Tipicamente em RSSFs, os nós sensores são posicionados em uma área e via comunicação sem fio *multihop* disseminam os dados do evento de interesse para um nó de saída denominado sorvedouro ou monitor [Tilak et al., 2002b], local em que na maioria das vezes é feito o processamento dos dados. As RSSFs podem ser utilizadas para monitoramento de vida animal, determinação de qualidade do ar em centros urbanos, monitoramento de floresta, entre outros, tendo a possibilidade de serem estabelecidas em áreas de difícil acesso e inóspitas, através do lançamento dos nós nestas regiões [Park et al., 2001, Mainwaring et al., 2002, Rabaey et al., 2000, Ye et al., 2002, Brooks & Iyengar, 1998, Vieira et al., 2003a]. Algumas características das RSSFs são predominantes, tais como: a alta redundância de nós, baixa largura de banda, dependência da aplicação e cada nó ter capacidade de energia limitada [Park et al., 2001].

Os maiores desafios na pesquisa em RSSF são inerentes da sua própria vocação, uma vez que os custos associados à obtenção de uma rede de comunicação extremamente versátil e adaptativa podem ser caros. O tamanho reduzido dos nós sensores limita a capacidade de seus componentes. Por sua vez o uso de baterias limita a autonomia dos nós sensores em termos de consumo de energia, uma vez que o estado atual de desenvolvimento desses dispositivos de armazenamento não possibilita a atuação prolongada dos nós sensores.

Para tornar as redes de sensores ainda mais populares, a garantia da qualidade de serviço tem sido fonte de intensas pesquisas na área. Os principais aspectos de interesse em qualidade de serviços em redes de sensores são a manutenção da cobertura da área monitorada, a garantia da conectividade entre os nós sensores ativos na rede e o roteamento dos dados coletados. As pesquisas tornam-se ainda mais desafiadoras visto que todas as soluções propostas devem levar em consideração os limites de energia da rede. Por isso soluções eficientes em termos de consumo de energia são essenciais em aplicações de RSSFs que necessitam operar por um longo período de tempo.

Como já mencionado as RSSF são classificadas como um tipo especial de redes ad hoc. No entanto, existem algumas diferenças entre este tipo de rede e as tradicionais ad hoc [Nakamura, 2003]. Dentre as diferenças encontradas, vale a pena destacar que nas RSSFs há um alto número e alta densidade de nós sensores que as compõem, os quais fazem uso de uma comunicação broadcast contrastando com a comunicação ponto-a-ponto realizado pelas redes tradicionais. Os nós sensores possuem capacidade limitada de energia, memória e processamento e a topologia da RSSF é considerada dinâmica. Outra característica interessante das RSSFs é sua dependência das aplicações, pois neste tipo de rede pode-se determinar o tipo de sensor a ser utilizado para se enquadrar melhor à aplicação, também a infra-estrutura e os requisitos de qualidade de serviço que podem ser considerados, como: confiabilidade, área de cobertura, conectividade e “tempo de vida” da rede.

A alta redundância de nós sensores é uma característica marcante das RSSF. Esta quantidade que em alguns casos pode parecer excessiva se faz necessária devido à circunstância dos nós sensores estarem sujeitos a falhas durante o “tempo de vida” da rede. Ao se ter mais sensores na rede, menor é a probabilidade de se perder a cobertura e a conectividade entre os sensores, quando se tem um mecanismo que faça um controle da rede. Sabendo desta característica, a referência [Tilak et al., 2002a] mostra a possibilidade de se fazer esquemas de escalonamento de nós que visam escolher periodicamente um subconjunto de nós para realizar as atividades da rede, enquanto os demais são agendados para dormir ou entrar em modo economia de energia; este método demonstrou economias significativas de energia. O desafio neste caso é minimizar o número de nós que permanecem ativos e ao mesmo tempo garantir que a rede alcance uma qualidade aceitável dos serviços para uma dada aplicação, em particular, mantendo uma área de cobertura suficiente e a conectividade entre os nós ativos, requerimentos críticos nas RSSFs.

Através de Programação Linear Inteira (PLI) pode ser modelado o Problema de Cobertura em RSSF [Megerian & Potkonjak, 2003]. Na resolução desses modelos, os autores propõem algoritmos baseados em soluções do Problema de Cobertura de Conjuntos. A referência [Chakrabarty et al., 2001] também aborda o problema de cobertura com PLI e apresenta um modelo matemático que minimiza os custos de estabelecimento de nós sensores heterogêneos na área de monitoramento sujeito a restrições de garantia de cobertura da área.

Os autores de [Quintao et al., 2004] fazem uso de algoritmos genéticos e algoritmos em grafos para tratar a cobertura e aspectos de conectividade para as RSSFs. Já [Vieira et al., 2003b] e [Meguerdichian et al., 2001] baseia-se em análise geométrica e grafos para abordar o mesmo problema de cobertura. Todos os trabalhos citados abordam

os problemas de forma estática, ou seja, dada uma configuração da rede determina-se a melhor solução naquele instante.

Na referência [Nakamura et al., 2005] os autores propõem um modelo de PLI para o problema multi-período de cobertura e conectividade em RSSFs e o resolvem com o pacote de otimização comercial CPLEX [Ilo, 2006]. O objetivo é prover uma solução que assegure a melhor cobertura e conectividade possível em todos os períodos de tempo, respeitando a capacidade da bateria dos nós.

A referência [Quintao et al., 2005] compara a solução do CPLEX para um modelo de PLI para o problema multi-período de cobertura com as obtidas por um algoritmo evolucionário, obtendo bons resultados, considerando os valores da função objetivo. Porém aspectos mais importantes que diferenciam a metodologia proposta dizem respeito aos tempos de execução do algoritmo.

Na referência [Ye et al., 2003] os autores apresentam o algoritmo PEAS (*Probing Environment and Adaptive Sleeping*). PEAS consiste de dois algoritmos, que determinam (1) quais nós sensores devem funcionar e como sensores que acabaram de acordar decidem se devem voltar a “dormir” ou não e (2) como o tempo médio de “sono” de cada sensor pode ser ajustado dinamicamente. Com estas duas características, o algoritmo garante um crescimento linear no tempo de vida da rede em função do número de sensores dispostos. Entretanto, o algoritmo não garante a cobertura da área de sensoriamento.

Em [Zhang & Hou, 2005] é apresentado o algoritmo OGDC (*Optimal Geographical Density Control*). Segundo os autores, o algoritmo é totalmente descentralizado e localizado. O OGDC é baseado no fato de que, caso o raio de comunicação seja pelo menos duas vezes maior do que o de sensoriamento, a garantia da cobertura implica na garantia da conectividade. A partir desta observação, os autores apresentam um conjunto de condições ótimas sobre as quais um conjunto de nós ativos pode ser encontrado para cobertura total da rede, e apresentam um algoritmo que mantém estas condições quando a rede possui alta densidade e cada nó sabe a sua localização.

1.1 OBJETIVO

O objetivo deste trabalho é formular heurísticas para desenvolver topologias de rede em um problema que busca manter a cobertura, a conectividade e o roteamento em uma rede de sensores sem fio plana em um ambiente *online* de monitoramento. Este monitoramento é realizado com um conjunto de sensores e pelo menos um sorvedouro presentes em uma área na qual se deseja fazer a coleta dos dados. As heurísticas desenvolvidas devem montar uma rede entre os sensores e também fazer uma “gerência da rede” ao longo do “tempo de vida” da mesma, buscando manter esta rede cobrindo a área determinada e garantindo a conectividade e o roteamento dos dados entre os sensores ativos, mesmo quando algum sensor falhar, de forma que esta rede persista monitorando por um maior tempo possível.

1.2 METODOLOGIA ADOTADA E RESULTADOS OBTIDOS

Primeiramente serão apresentados no Capítulo 2 breves conceitos, para então ser definido formalmente o problema a ser tratado. A partir desta definição é apresentado um modelo matemático que pode ser modelado como um problema de Programação Linear Inteira (PLI). A partir da definição do problema e sua modelagem, a solução seguirá em duas vertentes, sendo primeiro apresentada uma abordagem mono-objetivo e logo após sendo discutida uma abordagem multiobjetivo de uma maneira ainda não encontrada na literatura. Ambas as abordagens tratam inicialmente o problema de forma estática, ou seja, fornecendo o conjunto de sensores e de sorvedouros e a área a ser monitorada, é decidido qual a melhor configuração para a rede neste instante de tempo. Esta configuração é escolhida tendo em vista um menor consumo de energia e garantindo a cobertura desejada de forma que a conectividade e o roteamento entre os sensores ativos sejam respeitados. Este tratamento estático da rede se dividirá basicamente em dois estágios, são eles:

- Utilização de um Algoritmo Genético para tratamento da Cobertura.
- Utilização de um Algoritmo de Caminho Mínimo para tratar a conectividade e o roteamento.

Utilizado para o tratamento da cobertura da rede, o Algoritmo Genético (AG) é considerado uma classe particular de algoritmo evolutivo que faz uso de técnicas inspiradas pela biologia evolutiva, como: hereditariedade, mutação, seleção natural e recombinação (conhecida também como *crossing over*). O AG foi inicialmente desenvolvido por John Henry Holland [Holland, 1975] e hoje é uma ferramenta amplamente utilizadas para tratar problemas que buscam uma otimização global (maiores detalhes sobre AGs serão discutidos no decorrer dos Capítulos 3 e 4). Especificamente neste trabalho, estando de posse das características do conjunto de nós sensores e do sorvedouro, e a área que se deseja monitorar, os AGs serão utilizados para que se decida qual o subconjunto destes sensores devem ser ativados de forma que se obtenha um menor consumo de energia mantendo a cobertura desejada da área.

A solução fornecida pelo AG não se preocupa diretamente com a conectividade entre os sensores, é então que se faz necessária a aplicação do segundo estágio, no qual se usa um Algoritmo de Caminho Mínimo. Segundo a teoria de grafos, o Problema do Caminho Mínimo consiste na minimização do custo da travessia entre dois nós de um grafo, onde o custo é dado pela soma dos pesos de cada aresta percorrida neste grafo. Um dos métodos mais utilizados na literatura para se encontrar este menor custo é o algoritmo de Dijkstra [Dijkstra, 1959], o qual será mais bem detalhado na Seção 3.2 do Capítulo 3. Já na subseção 3.3.2 será mostrado mais especificamente como o algoritmo de Dijkstra implementado resolve o problema da conectividade entre os sensores para assim possibilitar o roteamento entre os dados de cada sensor ao nó sorvedouro.

Após decidir os nós sensores que deverão ficar ativos e a forma pela qual será realizado o roteamento dos dados coletados por cada sensor de maneira a possibilitar que estes dados cheguem ao nó sorvedouro, estará formada a rede inicial estática. De posse desta rede inicial, esta será estendida para uma abordagem dinâmica em que cada nó sensor

estará sujeito a falhas. Para fazer o controle da rede nesta ocasião serão utilizadas duas estratégias: Local e Global, como serão mostradas na subseção 3.3.3 presente no Capítulo 3. As rotinas presentes na estratégia Local possuem característica *Online*, ficando responsáveis por garantir, quando possível, a manutenção da cobertura e da conectividade da rede quando um ou mais sensores falharem. Já as rotinas da estratégia Global, por terem um custo computacional mais elevado, serão acionadas somente em alguns determinados momentos. Devido à sua característica de ter uma visão global no momento em que entra em ação, ela irá buscar uma reestruturação total da rede. A estratégia Global nada mais é que uma chamada recursiva dos próprios algoritmos utilizados para a formação da rede inicial, com os parâmetros de entrada relativos à rede no momento da chamada. Fornecidos os resultados, novamente a rede estática é estendida para uma rede dinâmica.

A junção das estratégias mencionadas acima será considerada uma abordagem Híbrida composta pelas heurísticas de Caminho Mínimo, Algoritmo Genético, *Online*(Local) e Global, que serão aplicadas nas abordagens a serem utilizadas para resolver o problema da RSSF proposto. Ambas as abordagens irão se basear na modelagem matemática que será mostrada no Capítulo 2, onde será apresentada uma função de energia a ser otimizada, juntamente com seu conjunto de restrições de cobertura e conectividade. A diferença entre as abordagens poderá ser encontrada na maneira de se tratar a cobertura, como consequência a elaboração delas irá diferir na forma de se implementar o AG, pois é este algoritmo o responsável por garantir a cobertura da rede. Todas as outras rotinas permanecerão inalteradas. Na abordagem mono-objetivo será usado um AG mono-objetivo, buscando selecionar sensores presentes na RSSF para ficarem ativos de forma que se cubra completamente, sempre que possível, a área a ser monitorada, ao mesmo tempo buscando minimizar o consumo de energia da rede (esta aplicação poderá ser vista no Capítulo 3). Já na abordagem multiobjetivo o conceito de cobertura irá sofrer uma pequena alteração. Utilizando um Algoritmo Genético Multiobjetivo, será realizada uma busca por soluções, de maneira que essas formem uma curva Pareto-ótima em dois objetivos, possibilitando buscar soluções que ao mesmo tempo correspondam a um baixo consumo de energia e a uma baixa perda de cobertura. Então um decisor externo irá definir qual a taxa mínima da área que necessariamente deverá estar coberta, para assim ser escolhida uma solução presente na curva Pareto-ótima (esta aplicação poderá ser vista no Capítulo 4).

Para cada abordagem apresentada foram realizados testes com instâncias de 36, 49, 64, 81 e 100 nós sensores sem fio do padrão MICA2 [Crossbow, 2006], distribuídos de forma aleatória em uma área de $2500m^2$. Para cada instância um nó sorvedouro está presente para coletar e processar todos os dados monitorados pelos sensores, ficando ele também responsável por garantir o controle da rede, pois devido à sua energia ser ilimitada, nele são executados grande parte dos algoritmos. Portanto, a decisão sobre quais nós sensores devem ficar ativados, e como será feita a conectividade e o roteamento entre os sensores de forma a garantir a disseminação dos dados pela rede é processada neste nó sorvedouro. Para cada instância, serão apresentados gráficos mostrando a taxa de cobertura, a energia consumida e a energia residual da rede, e serão analisados e discutidos os respectivos resultados.

A modelagem através de Programação Linear Inteira que será mostrada no Capítulo 2, possibilitou obter resultados exatos para o problema, ao ser utilizado o pacote comercial CPLEX [Ilo, 2006], um software disponível no mercado para resolver problemas de programação inteira, programação linear inteira, programação linear em grandes escalas, dentre outros. Para ambas as abordagens baseadas em AG's estes resultados serviram de limite superior para avaliar a cobertura e a energia residual da rede, e de limite inferior para avaliar a energia consumida da rede. Para cada abordagem serão apresentadas duas baterias de testes, a primeira considerando falhas mecânicas e falhas por energia, tentando mostrar o comportamento da RSSF em situações de falhas inesperadas, e a segunda considerando falhas apenas por energia, ou seja, esgotamento da bateria, possibilitando visualizar o comportamento do algoritmo até o limite do "tempo de vida" da rede. Para a segunda bateria de testes será mostrado também o tempo de processamento de cada abordagem utilizada e da solução fornecida pelo pacote comercial CPLEX. Para evidenciar as características apresentadas pela abordagem multiobjetivo, será mostrada uma análise do *trade-off* entre os objetivos, pois se tem uma tomada de decisão envolvendo um compromisso entre o tempo de vida da rede e a sua taxa de cobertura.

No final dos capítulos nos quais serão mostradas as abordagens Mono-objetivo e Multiobjetivo, serão apresentados os resultados e as análises para a cobertura, energia consumida e energia residual. Para a versão mono-objetivo será possível visualizar que os resultados se mostraram próximos, no entanto um pouco piores, quando comparados aos valores ótimos fornecidos pelo CPLEX. Também será possível ver que a grande diferença entre as aplicações foi encontrada no tempo de processamento dos algoritmos. Na Abordagem Multiobjetivo será mostrado que ao relaxar a restrição de cobertura total, mesmo em níveis pequenos como de 5%, será possível obter resultados melhores tanto em relação ao CPLEX como à Abordagem Mono-objetivo, no que diz respeito ao "tempo de vida" da rede. Através da análise do *trade-off* será possível ver que à medida em que é diminuída a exigência na cobertura, maior é o "tempo de vida" obtido pela rede.

CAPÍTULO 2

O PROBLEMA: DEFINIÇÕES E MODELOS

O maior desafio para qualquer pensador é enunciar o problema de tal modo que possa permitir uma solução.

—BERTRAND RUSSELL

A vida está cheia de desafios que, se aproveitados de forma criativa, transformam-se em oportunidades.

—MAXWELL MALTZ

Uma RSSF é composta basicamente por dois componentes, são eles:

- Um conjunto de nós sensores sem fio;
- Um conjunto de nós sorvedouros.

Os sensores são projetados para coletar informações diversas. Eles são compostos basicamente por uma unidade de comunicação sem fio, unidade de energia, unidade de sensoriamento e unidade de computação (Figura 2.1) [Vieira et al., 2003a]. O modelo do sensor empregado neste trabalho, foi o MICA2 (Figura 2.2) ([Crossbow, 2006]). Este nó sensor tem a característica de coletar a informação e enviá-la para outro nó sensor, um nó sorvedouro que poderá processar o conjunto dos dados coletados. A grande maioria dos sensores tem dimensões pequenas e um baixo custo, esta limitação faz refletir uma baixa capacidade de recursos como: energia, processador e transceptor. Apesar de essa característica individual ser limitada, um esforço colaborativo entre eles permite a realização de tarefas mais complexas, o que não seria possível através de apenas um componente. [Ruiz, 2003]

Cada nó sensor é projetado com um raio de sensoriamento e outro de comunicação. Como o próprio nome diz o raio de sensoriamento é a distância ao redor do nó sensor que ele consegue cobrir, ou seja, a região que será monitorada por ele. Já o raio de comunicação representa a distância na qual ele consegue enviar as informações coletadas. Podemos observar na Figura 2.3 as diferenças entre os tipos de raio, RC representa o raio de comunicação e RS o raio de sensoriamento. Vale a pena observar que na maioria das vezes o raio de sensoriamento é menor que o de comunicação.

Existem possibilidades de variações desses raios, tanto de sensoriamento como o de comunicação. Podem variar de forma dinâmica de acordo com a demanda e necessidade, potência ou estado de energia do nó. Sabe-se que quanto mais distante um sensor enviar uma informação, mais energia é gasta por ele.

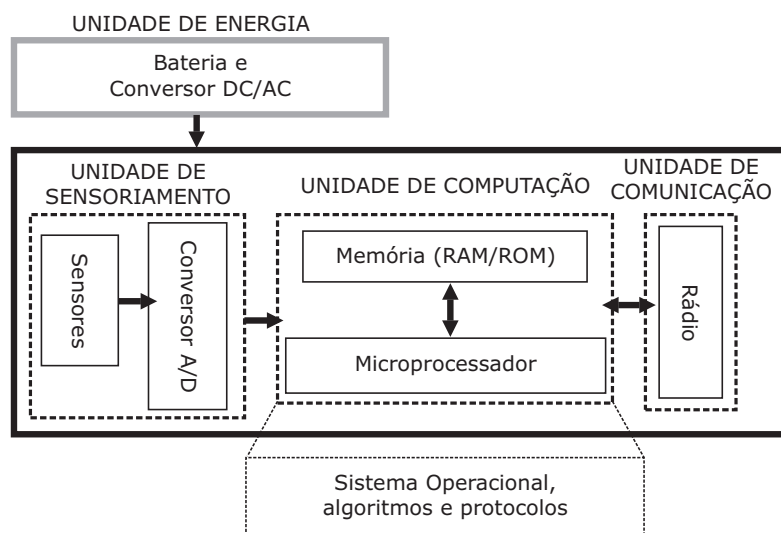


Figura 2.1 Principais componentes de um nó sensor.



Figura 2.2 Nó sensor Mica 2

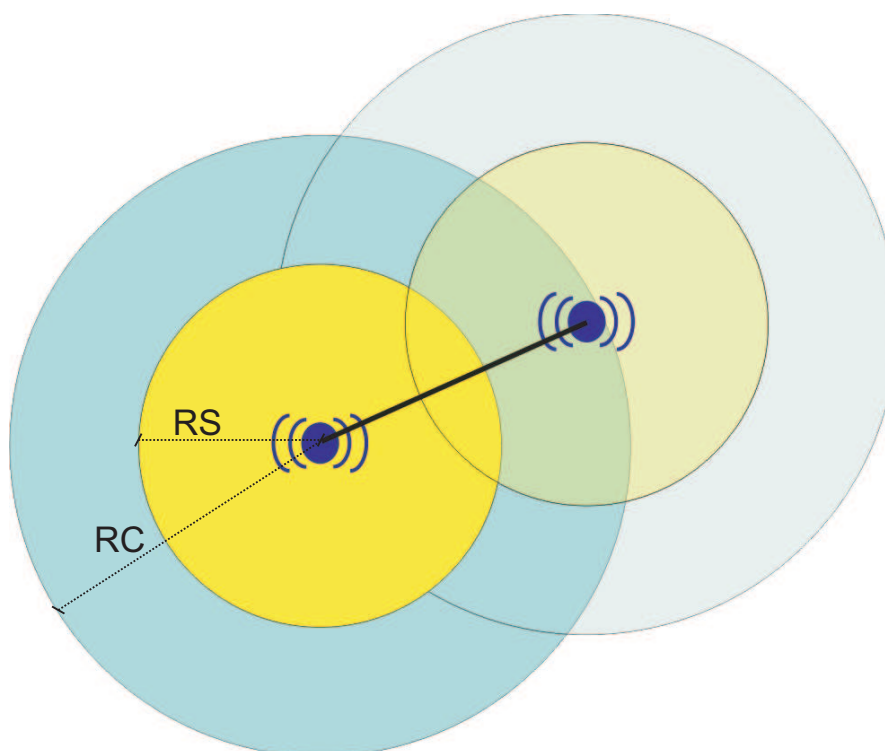


Figura 2.3 Exemplo para identificar os raios de sensoriamento (RS) e comunicação (RC)

Como já citado, além dos sensores tem-se também os nós sorvedouros. Estes têm como característica processar as informações coletadas pelos sensores presentes na rede. Existe várias formas de projetá-los, podendo ser móveis, de forma a caminhar pela rede coletando os dados fornecidos pelos sensores [Kim et al., 2003, Wang et al., 2005, Gandham et al., 2003, Aioffi et al., 2007], como também serem projetados para serem fixos, em um determinado ponto da área que será monitorada, e através de uma rede de comunicação recebem os dados dos sensores [Nakamura et al., 2007, Martins et al., 2007].

2.1 DEFINIÇÃO DO PROBLEMA

Dado uma determinada área a ser monitorada, pontos de demanda são projetados nesta área de forma a discretizar toda área desejada. Cada ponto de demanda corresponde a uma área de $1m^2$. Por exemplo, ao querer monitorar uma área de $10m \times 10m$, esta área é dividida em 100 pontos de demanda, igualmente espaçados e distribuídos em um grid, como se pode ver na Figura 2.4. Analogamente, ao se ter uma área de $2500m^2$, tem-se 2500 pontos de demanda distribuídos, Figura 2.5.

Uma primeira versão do problema tratado nesta dissertação pode ser formalmente definido como: *Dada uma área de monitoramento \mathcal{A} (Figura 2.6a), um conjunto de nós sensores \mathcal{S} (Figura 2.6b), um conjunto de nós sorvedouros \mathcal{M} (Figura 2.6c), um conjunto de pontos de demanda \mathcal{D} (Figura 2.6d), o Problema de Cobertura e Conectividade em Redes de Sensores sem Fio (PCC-RSSF) consiste em garantir para cada ponto de de-*

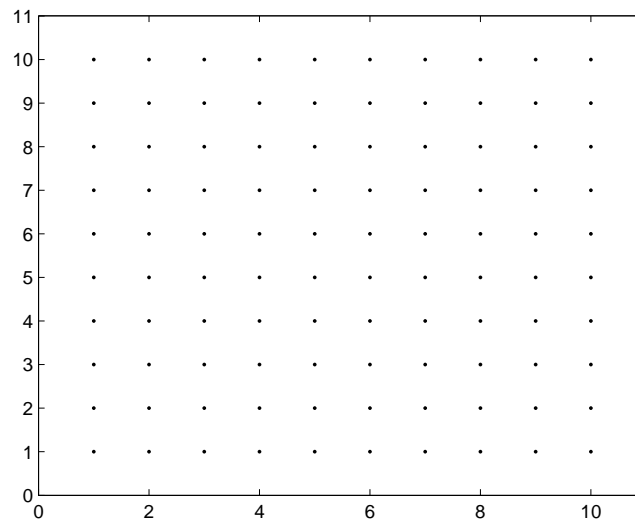


Figura 2.4 Exemplo de pontos de demanda em uma área de $100m^2$

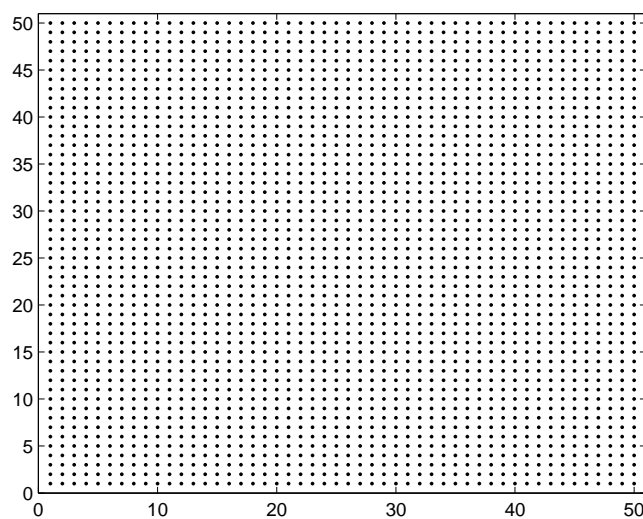


Figura 2.5 Exemplo de pontos de demanda em uma área de $2500m^2$

manda $d \in \mathcal{D}$ na área A que pelo menos 1 (um) nó sensor $s \in \mathcal{S}$ o cubra e que exista uma rota entre cada nó sensor ativo $s \in \mathcal{S}$ e um nó sorvedouro $m \in M$. A solução ótima é aquela que garanta a cobertura e conectividade da rede com o menor custo de energia consumida. Neste trabalho é simulada uma RSSF com nós sujeitos a restrições de energia e falhas, por se tratar de um cenário mais adequado à realidade proposta para RSSFs.

Além desta versão do problema, esta dissertação também aborda uma versão em que a cobertura da área a ser monitorada pode ser relaxada de forma a atender a demanda necessária da aplicação. Formalmente pode-se defini-la como: *Dada uma área de monitoramento A (Figura 2.6a), um conjunto de nós sensores \mathcal{S} (Figura 2.6b), um conjunto de nós sorvedouros M (Figura 2.6c), um conjunto de pontos de demanda \mathcal{D} (Figura 2.6d), o Problema de Cobertura e Conectividade em Redes de Sensores sem Fio (PCC-RSSF) consiste em garantir que uma taxa ϕ de pontos de demanda $\in \mathcal{D}$ na área A seja coberta por um determinado subconjunto de nós sensores $s_{MO} \in \mathcal{S}$ e que exista uma rota entre cada nó sensor ativo $s_{MO} \in \mathcal{S}$ e um nó sorvedouro $m \in M$.*

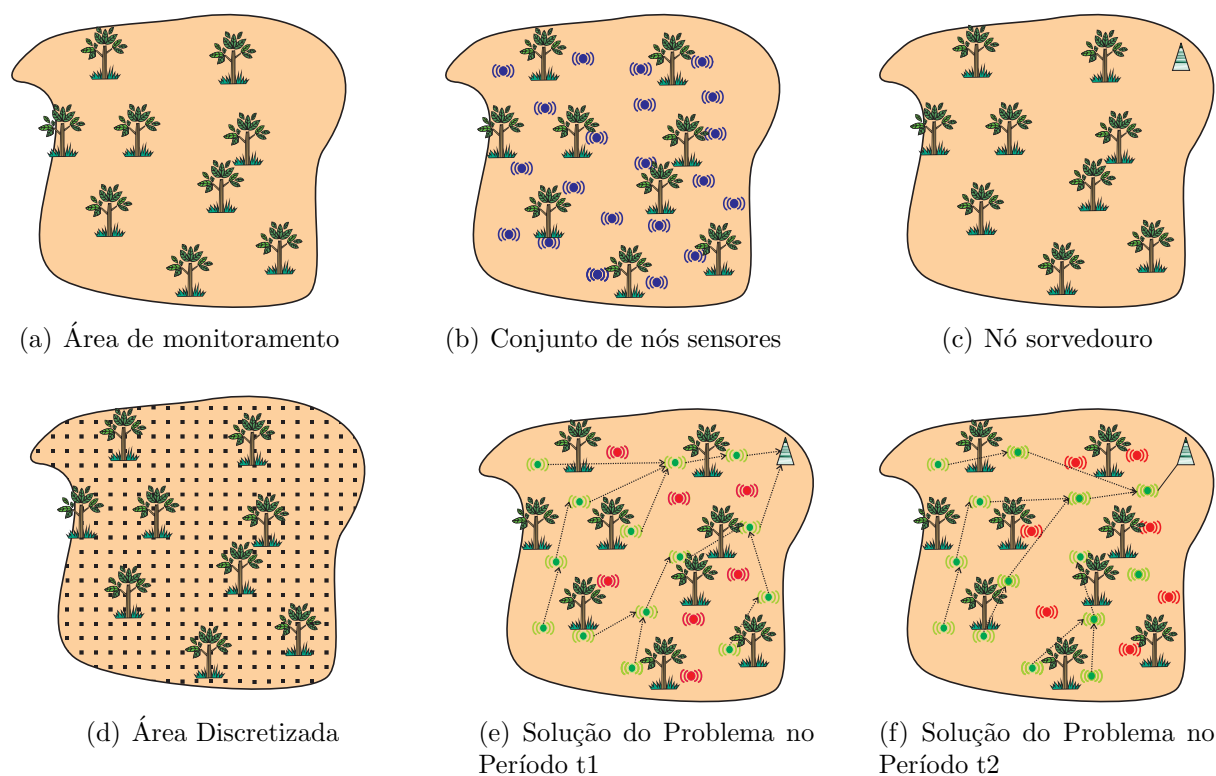


Figura 2.6 Problema Dinâmico de cobertura e Conectividade em RSSFs Planas.

2.2 MODELO MATEMÁTICO

O PCC-RSSF estático pode ser modelado como um problema de Programação Linear Inteira (PLI) [Martins et al., 2007]. Maiores detalhes sobre PLI podem ser vistos em [Kadayif et al., 2005] e [Schrijver, 1986]. O modelo matemático aqui apresentado é similar ao apresentado nos seguintes trabalhos: [Martins et al., 2007], [Nakamura et al., 2007],

[Martins et al., 2008], [Quintao et al., 2004], [Quintao et al., 2005], [Nakamura et al., 2005] entre outros. Na elaboração do modelo e algoritmos algumas considerações são feitas:

- Os nós possuem um identificador único e sabem sua localização;
- A aplicação requer monitoramento contínuo e a disseminação de dados periódica;
- O tráfego na RSSF é relativo apenas a dados da aplicação;
- A área de monitoramento é discretizada em um conjunto de pontos denominados pontos de demanda;
- A área de cobertura de um nó é considerada como um círculo de raio R_s , onde R_s é o raio de sensoriamento do nó. Se a distância entre um ponto de demanda e um nó é menor que R_s então este ponto pode ser coberto por este nó;
- A região de conectividade de um nó é considerada como um círculo de raio R_c , onde R_c é o raio de conectividade do nó. Se a distância entre um nó sensor a outro nó sensor ou sorvedouro é menor que R_c então estes podem se comunicar.

O seguintes parâmetros são utilizados na formulação.

\mathcal{S} conjunto de nós sensores;

\mathcal{M} conjunto de nós sorvedouros;

\mathcal{D} conjunto de pontos de demanda;

\mathcal{T} conjunto de períodos de tempo;

\mathcal{B}_{lj} matriz de conectividade, que tem o valor 1 na posição (l, j) se o nó l alcança o ponto de demanda j e 0 caso contrário;

\mathcal{A}^s conjunto de arcos conectando os nós sensores;

\mathcal{A}^m conjunto de arcos conectando os nós sensores aos nós sorvedouros;

$\mathcal{I}^l(\mathcal{A})$ conjunto de arcos que entram no nó sensor $l \in \mathcal{S}$ e que pertencem ao conjunto \mathcal{A} ;

$\mathcal{O}^l(\mathcal{A})$ conjunto de arcos que saem no nó sensor $l \in \mathcal{S}$ e que pertencem ao conjunto \mathcal{A} ;

q precisão na cobertura que indica o número de nós sensores que devem cobrir um ponto de demanda. Em geral é feito igual a 1;

n número mínimo de nós sensores que devem estar ativos por período;

EB_l capacidade da bateria do nó $l \in \mathcal{S}$;

EA_l energia de ativação do nó $l \in \mathcal{S}$;

PM_l^t potência de manutenção do nó $l \in \mathcal{S}$ ativo em cada período, que representa o consumo de energia do nó com o sensoriamento, processamento e escuta de canal em cada intervalo $t \in \mathcal{T}$;

PT_{li}^t potência de transmissão consumida pelo nó l com a operação de transmissão dos pacotes gerados para o nó i em cada período $t \in \mathcal{T}$;

PR_l^t potência de recepção consumida pelo nó l com a operação de recepção dos pacotes em cada período $t \in \mathcal{T}$;

EH_j penalidade de não cobertura representada através de um custo de energia por não cobrir o ponto de demana j .

As variáveis usadas no modelo são:

x_{lj}^t variável que indica se o nó l está cobrindo o ponto de demanda j no período de tempo t ;

z_{lki}^t variável de decisão que possui valor 1 se o arco ki faz parte da rota entre o nó l e um nó sorvedouro no período de tempo t , e 0 caso contrário;

w_l^t variável de decisão que possui valor 1 se o nó l é ativado no período de tempo t , e 0 caso contrário;

y_l^t variável de decisão que possui valor 1 se o nó l está ativo no período de tempo t , e 0 caso contrário;

h_j^t variável que indica a não cobertura de um ponto de demanda j no período de tempo t .

A energia consumida a cada período de tempo t para cada nó sensor é dada pela equação (2.1). Esta energia é calculada baseada na energia gasta pela a operação de ativação, manutenção, recepção e transmissão. Para um melhor entendimento, cada trecho da equação será explicado.

$$\begin{aligned}
 e_l^t &= \sum_{t \in \mathcal{T}} (EA_l \cdot w_l^t + PM_l^t \cdot y_l^t) \\
 &+ \sum_{k \in (\mathcal{S}-l)} \sum_{i \in \mathcal{I}^l(\mathcal{A}^s)} PR_l^t \cdot z_{kil}^t \\
 &+ \sum_{k \in \mathcal{S}} \sum_{li \in \mathcal{O}^l(\mathcal{A}^s \cup \mathcal{A}^m)} PT_{li}^t \cdot z_{kli}^t)
 \end{aligned} \tag{2.1}$$

A cada vez que um sensor é acionado, passando de um estado de não monitoramento para o estado de estar ativado e monitorando, há um gasto de energia pelo processo de inicialização do sensor. Para representar o gasto devido a esse processo foi adotado o nome energia de ativação (EA). Então EA_l será computada no período de tempo t em

que o nó sensor l for ativado, pois neste período a variável de decisão w_l^t , que o multiplica, terá valor 1. Caso o nó l já esteja ativo este valor não será computado.

A cada período de tempo t um sensor consome potência para se manter ativo e monitorando, a essa potência dá-se o nome de potência de manutenção (PM). Na equação 2.1, PM_l^t representa o valor consumido pelo sensor l para se manter ativo em estado de monitoramento no período de tempo t . Pode-se observar que PM_l^t é multiplicada por y_l^t , uma variável de decisão que pode assumir apenas dois valores, 1 se o nó l estiver ativo no período de tempo t , e 0 caso contrário, ou seja, só se tem o cálculo da potência de manutenção para o caso dos sensores ativos.

O próximo termo da equação é a potência consumida pela recepção de todos os pacotes de dados enviados ao sensor l . Para um melhor entendimento podemos observar a Figura 2.7. Nela tem-se uma árvore que está conectada ao sensor l (sensor pai). Cada sensor presente nesta árvore está coletando informações que são enviadas para processamento a cada período de tempo. Na RSSF, aqui apresentada, existe uma hierarquia de rede, em que cada sensor tem o seu “sensor pai” e os sensores que possuem grau de entrada maior que 0 (zero) no grafo direcional da rede, possuem “sensor(es) filho(s)”. Cada “sensor pai” precisa receber todos os dados da(s) árvore(s) de sensores ligada a ele e depois transmitir esses dados para seu “sensor pai”. Para receber toda essa informação há um consumo de potência, e a este consumo dá-se o nome de potência de recepção (PR). A parte da equação 2.1 que trata este consumo de potência é: $\dots \sum_{k \in (\mathcal{S}-l)} \sum_{il \in \mathcal{I}^l(\mathcal{A}^s)} PR_l^t \cdot z_{kil}^t + \dots$. Neste termo, o primeiro somatório é realizado para todo conjunto de sensores \mathcal{S} presentes na rede, exceto o nó sensor l , o segundo somatório, através do índice $il \in \mathcal{I}^l(\mathcal{A}^s)$, é limitado apenas aos sensores que fazem comunicação direta com o sensor l , ou seja, somente os sensores que possuem arcos que fazem parte do conjunto de arcos que entram no nó sensor l . Então nesse termo é calculada toda potência consumida para receber os pacotes de informação transmitidos pelos “sensores filhos” do sensor l . Pela Figura 2.7 tem-se o nó sensor 2 recebendo informações dos nós 4 e 5. O sensor 1 recebe os dados transmitidos pelos nós sensores 2 e 3.

Por último tem-se o termo: $\dots + \sum_{k \in \mathcal{S}} \sum_{li \in \mathcal{O}^l(\mathcal{A}^s \cup \mathcal{A}^m)} PT_{li}^t \cdot z_{kli}^t$ que trata a potência de transmissão (PT). A PT representa toda potência consumida para transmitir os pacotes de informação recebidos pelo nó sensor l mais os dados coletados por este. Como no caso da recepção, transmitir os pacotes de dados consome potência. No caso da transmissão essa potência é diretamente proporcional à distância. Para cada tipo de sensor este consumo é computado segundo uma tabela de distância ou por uma equação que rege este fenômeno. Com isso tem-se PT_{li}^t como sendo a quantidade de potência consumida para transmitir dados do nó sensor l para o nó sensor i no período de tempo t . O índice, $li \in \mathcal{O}^l(\mathcal{A}^s \cup \mathcal{A}^m)$, presente no segundo somatório, faz com que se restrinja apenas ao(s) arco(s) que saem de l em direção ao nó i . Este(s) arco(s) é(são) responsável(is) por transmitir a(s) informação(ões) de l para i . Como exemplo tem-se a Figura 2.7. Nela o nó sensor 2 transmite para o nó 1 toda informação coletada por ele mais as informações enviadas pelos seus filhos 4 e 5.

Então se tem a formulação PLI para o problema PCC-RSSF. A função objetivo tem por intenção minimizar a energia consumida da RSSF durante seu período de tempo, como podemos ver na equação 2.2

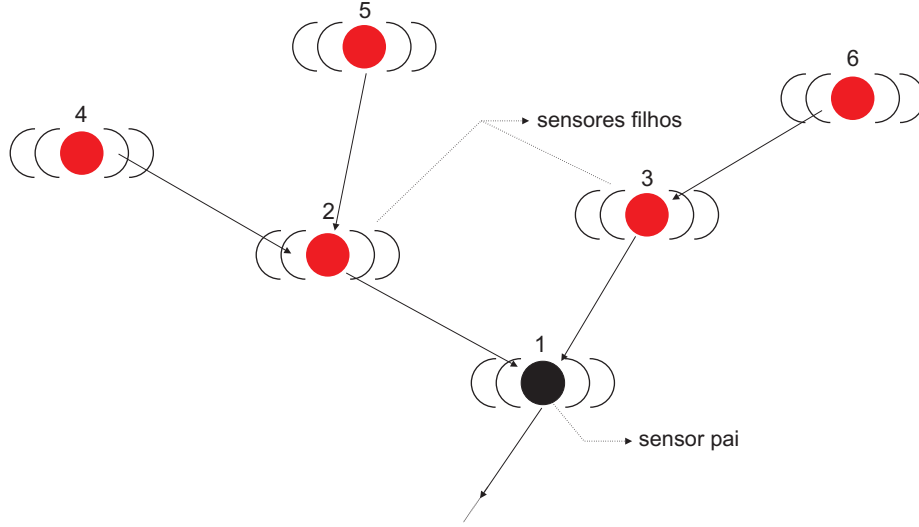


Figura 2.7 Explicação sobre a potência de recepção e transmissão.

$$Y^* = \arg \min_Y \sum_{l \in \mathcal{S}} e_l^t + \sum_{j \in \mathcal{D}} \sum_{t \in \mathcal{T}} EH_j \cdot h_j^t \quad (2.2)$$

em que Y terá solução do tipo: $Y = \{y_1, \dots, y_{|\mathcal{S}|}\}$.

Pode-se ver que além do consumo de energia dado pelo termo $\dots \sum_{l \in \mathcal{S}} e_l^t + \dots$, a função objetivo ainda traz um outro termo expressado por $\dots + \sum_{j \in \mathcal{D}} \sum_{t \in \mathcal{T}} EH_j \cdot h_j^t$ que traz uma maior flexibilização, pois abre a possibilidade de falhas de cobertura. Neste segundo termo EH_j representa um custo de energia pela não cobertura de um ponto de demanda j . Já a variável h_j^t indica a não cobertura do ponto de demanda j no período de tempo t .

O conjunto de restrições (2.3), (2.5) e (2.6) trata do problema de cobertura.

$$\sum_{l \in \mathcal{S}} \mathcal{B}_{lj} \cdot x_{lj}^t + h_j^t \geq q, \quad \forall j \in \mathcal{D} \text{ e } \forall t \in \mathcal{T} \quad (2.3)$$

$$h_j^t \geq 0, \quad \forall j \in \mathcal{D} \text{ e } \forall t \in \mathcal{T} \quad (2.4)$$

$$\mathcal{B}_{lj} \cdot x_{lj}^t \leq y_l^t, \quad \forall l \in \mathcal{S}, \forall j \in \mathcal{D} \text{ e } \forall t \in \mathcal{T} \quad (2.5)$$

$$0 \leq \mathcal{B}_{lj} \cdot x_{lj}^t \leq 1, \quad \forall l \in \mathcal{S}, \forall j \in \mathcal{D} \text{ e } \forall t \in \mathcal{T} \quad (2.6)$$

A restrição (2.3) garante que pelo menos q nós sensores cubram cada ponto de demanda. A variável x_{lj}^t assume valor 1 (um) se o sensor l cobrir o ponto de demanda j no período de tempo t . Então ao multiplicá-la por \mathcal{B}_{lj} faz garantir a cobertura se o nó sensor l alcançar o ponto de demanda j . Já no segundo termo a variável h_j^t traz a possibilidade de não cobertura de um ponto de demanda. Em seu pior caso h_j^t seria igual a q , o que significaria que nenhum nó sensor estaria cobrindo o ponto de demanda. Já para o caso em que pelo menos um nó sensor estivesse cobrindo o ponto de demanda j em um determinado período de tempo t , ter-se-ia $h_j^t = q - \sum_{l \in \mathcal{S}} \mathcal{B}_{lj} \cdot x_{lj}^t$.

A restrição (2.5) indica que um nó só pode cobrir um ponto se estiver ativo. Por exemplo, pode-se ter o caso em que um nó consiga cobrir um determinado ponto de demanda, mas por alguma eventualidade ele não está ativo. Neste caso tem-se: $x_{lj}^t = 1$ e $y_l^t = 0$ violando assim a restrição (2.5). A restrição (2.6) é usada apenas para limitar a variável x .

O conjunto de restrições (2.7), (2.8), (2.9) e (2.10) lida com o problema de conectividade.

$$\sum_{ip \in \mathcal{I}^p(\mathcal{A}^s)} z_{ip}^t - \sum_{pk \in \mathcal{O}^p(\mathcal{A}^s \cup \mathcal{A}^m)} z_{pk}^t = 0, \quad \forall p \in (\mathcal{S} - l), \quad \forall l \in \mathcal{S} \text{ e } \forall t \in \mathcal{T} \quad (2.7)$$

$$- \sum_{pk \in \mathcal{O}^p(\mathcal{A}^s \cup \mathcal{A}^m)} z_{pk}^t = -y_l^t, \quad p = l, \quad \forall l \in \mathcal{S} \text{ e } \forall t \in \mathcal{T} \quad (2.8)$$

$$z_{ip}^t \leq y_i^t, \quad \forall i \in \mathcal{S}, \quad \forall l \in (\mathcal{S} - p), \quad \forall ip \in (\mathcal{A}^s \cup \mathcal{A}^m) \text{ e } \forall t \in \mathcal{T} \quad (2.9)$$

$$z_{ip}^t \leq y_p^t, \quad \forall p \in \mathcal{S}, \quad \forall l \in (\mathcal{S} - p), \quad \forall ip \in (\mathcal{A}^s \cup \mathcal{A}^m) \text{ e } \forall t \in \mathcal{T} \quad (2.10)$$

As restrições (2.7) e (2.8) asseguram um caminho entre cada nó ativo e o nó sorvedouro. Tem-se que z_{ip}^t representa a variável de decisão a qual possui valor 1 se o arco ip faz parte da rota entre o nó l e um nó sorvedouro no período de tempo t , ou 0 caso contrário. Então em (2.7) quando atendida, todo o fluxo da rede garantidamente é canalizado a um sorvedouro. A restrição (2.8) garante que cada sensor irá se comunicar para enviar informações, até o sorvedouro, com apenas um sensor da rede na unidade de tempo t .

As restrições (2.9) e (2.10) permitem que apenas nós ativos componham este caminho, semelhante à restrição antes dada por (2.5).

A equação de restrição de energia (2.11) indica que a energia consumida por um nós sensor $l \in \mathcal{S}$ durante seu funcionamento é limitada pela capacidade de sua bateria.

$$\sum_{l \in \mathcal{S}} e_l^t \leq EB_l, \quad \forall l \in \mathcal{S} \quad (2.11)$$

A restrição (2.12) estabelece o limite inferior para a energia consumida em cada nó. Como se trata de quantidade de energia, os valores aceitos são apenas maiores ou iguais a zero. Para o caso de e_l ser igual a zero, tem-se um sensor l não ativo na rede.

$$e_l \geq 0, \quad \forall l \in \mathcal{S} \quad (2.12)$$

As restrições de ativação (2.13) e (2.14) mostram a relação entre as variáveis w , y , o período no qual o nó foi ativado, e se o nó está ou não ativo no período t . Por exemplo, caso tenha um nó sensor l ativo no período de tempo t e este não estava ativo no período $t - 1$, tem-se $w_l^t = 1$, com isso o custo EA_l é contabilizado na Equação (2.1) e conseqüentemente na função objetivo (Equação 2.2) e na restrição (2.11).

$$w_l^1 - y_l^1 \geq 0, \quad \forall l \in \mathcal{S} \quad (2.13)$$

$$w_l^t - y_l^t + y_l^{t-1} \geq 0, \quad \forall l \in \mathcal{S}, \quad \forall t \in \mathcal{T} \text{ e } t \geq 2 \quad (2.14)$$

Já a restrição 2.15 indica o número mínimo de nós sensores que necessitam estar ativos a cada período de tempo t .

$$\sum_{l \in \mathcal{S}} y_l^t \geq n, \quad \forall t \in \mathcal{T} \quad (2.15)$$

Para a modelagem matemática em questão, o cálculo do número mínimo de nós ativos por período (n) é calculado pela Equação 2.16.

$$n = \frac{A}{\pi R_s^2} \quad (2.16)$$

onde A representa a área a ser monitorada e R_s o raio de sensoriamento. E finalmente a restrição 2.17 define as variáveis de decisão como binárias.

$$y, z, w \in \{0, 1\} \quad (2.17)$$

A solução do modelo consiste do conjunto Y de nós sensores que devem ser ativos, isto é, $Y = \{i \in \mathcal{S} | y_i = 1\}$. Esta solução é gerada periodicamente (ou quando necessário) e indica quais nós devem cobrir quais pontos de demanda e garante pelo menos uma rota entre esses nós sensores e um nó sorvedouro.

CAPÍTULO 3

ABORDAGEM MONO-OBJETIVO

Na sobrevivência dos indivíduos e raças favorecidas, durante a luta constante e recorrente pela existência, vemos uma forma poderosa e incessante de seleção.

—CHARLES DARWIN

O fator decisivo para vencer o maior obstáculo é, invariavelmente, ultrapassar o obstáculo anterior.

—HENRY FORD

Neste capítulo será mostrada uma abordagem mono-objetivo para a solução do problema de cobertura e conectividade em uma RSSF, apresentando seus resultados e análises. Preliminarmente serão definidas as ferramentas utilizadas para a resolução do problema, Algoritmo Genético e Caminho Mínimo.

3.1 ALGORITMO GENÉTICO

Os Algoritmos Genéticos (AG's) vêm sendo estudados intensamente nos últimos anos, tendo se constituído numa ferramenta-padrão para tratar, de maneira sistemática, a otimização global (pelo menos aproximada) de funções genéricas, possivelmente não-diferenciáveis, descontínuas e/ou multimodais, ou seja, funções que não exibem nenhuma das propriedades usualmente necessárias para permitir a aplicabilidade de métodos convencionais de programação matemática.

Através de um conjunto de soluções-candidatas consideradas simultaneamente, denominado de população, o Algoritmo Genético trabalha em busca de uma solução ótima. A cada solução-candidata se atribui o nome de indivíduo. Por fazer uso de populações e estas evoluírem em busca de uma solução, este tipo de algoritmo é conhecido na literatura como Algoritmo Evolucionário. Pode ser considerado um AG o algoritmo que emprega, dentre outras, pelo menos alguma versão de três operações básicas correspondentes aos elementos constituintes da evolução dos seres biológicos, definidas por:

- Cruzamento: operação que combina a informação de dois indivíduos, gerando novos indivíduos;
- Mutação: operação que “perturba” um indivíduo, gerando um novo indivíduo com alguma semelhança com o indivíduo que o originou;
- Seleção: operação que gera uma nova população a partir da população corrente, com maior probabilidade de inserção, na nova população, os indivíduos de melhor valor

de função-objetivo, conseqüentemente, menor probabilidade para os indivíduos com pior função-objetivo.

Esses operadores são aplicados seqüencialmente, repetindo-se a seqüência a cada iteração, até a convergência para um ponto considerado ótimo segundo algum critério de parada. Algoritmos constituídos segundo esse esquema são genericamente denominados Algoritmos Genéticos. Algumas referências que abordam o estudo desses operadores são [Belmont-Moreno., 2001]; [Choi & Oh, 2000]; [Hasancebi & Erbatur, 2000]; [Takahashi et al., 2003]; [Vasconcelos et al., 2001]. A maioria dos AG's atualmente utilizados emprega, além desses operadores básicos, também outros operadores genéticos adicionais, que aumentam a eficiência do algoritmo. Alguns desses operadores possuem aplicabilidade genérica, mantendo o grau de generalidade do AG básico, tais como:

- Elitismo: causa a seleção determinística de parte da população corrente, usualmente os melhores indivíduos, para integrarem a nova população;
- Nicho: evita a concentração excessiva de indivíduos explorando a mesma região de um espaço de busca.

O estudo de operadores não-básicos é apresentado em diversas referências, por exemplo [Takahashi et al., 2003]; [Vasconcelos et al., 2001]; [Fan et al., 2000]; [Potts et al., 1994]; [Sareni & Krahenbuhl, 1998].

3.2 CAMINHO MÍNIMO

O problema do caminho mínimo (PCM) faz parte do conjunto de problemas pertencentes à teoria dos grafos. O PCM consiste em achar o menor caminho possível (considerando a soma dos pesos de cada aresta percorrida) entre dois nós em um determinado grafo com pesos ou dito valorado. Formalmente, tem-se:

Seja $G = (N, A)$ um grafo com pesos, N o conjunto de nós, A o de arestas e $f : A \rightarrow \Re$ a função de peso. Dado qualquer elemento $n \in N$, encontrar um caminho C , de n a cada outro nó $n' \in N$, tal que $\sum_{c \in C} f(p)$ seja mínima entre todos os caminhos conectados de n a n' .

Para um melhor entendimento é mostrado na Figura 3.1 um exemplo de um grafo em que é possível mostrar o caminho mínimo entre dois nós. Observando esta figura, pode-se perceber que existem vários caminhos que ligam o nó 1 (um) ao nó 4 (quatro). No entanto existe um caminho que é o mais curto, considerando os pesos dados pelas arestas. Temos no total 7 caminhos possíveis sem que se repitam arestas, no entanto o de menor custo é passando seqüencialmente pelos nós 1, 3 e 4.

Um dos algoritmos mais conhecidos na literatura para resolver o problema do caminho mínimo é o Algoritmo de Dijkstra [Dijkstra, 1959], assim chamado devido à sua criação em 1959 pelo cientista da computação holandês chamado Edsger Dijkstra. Esta implementação sugere um algoritmo de rotulação para caminhos em grafos para arcos positivos, utilizando indução e ajuste. Segundo [Goldbarg & Luna, 2000] sua complexidade é de $O(n^2)$ e também é de fácil implementação computacional. Nesta referência também é dado pelo pseudocódigo Algoritmo 1.

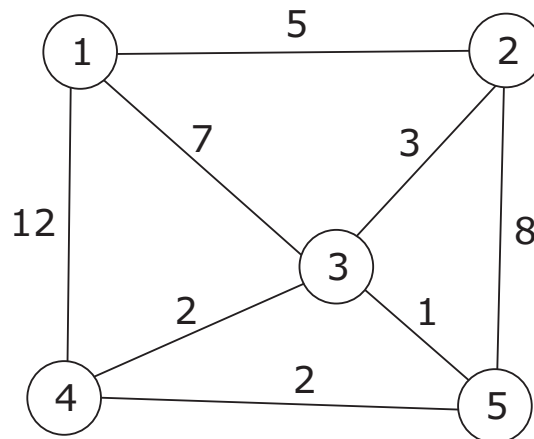


Figura 3.1 Grafo para exemplificar o problema do caminho mínimo

Chamando de:

Lista O (lista dos nós fechados) \equiv o conjunto dos vértices para o qual já se conhece um caminho mínimo.

Lista A (lista dos nós abertos) \equiv o conjunto dos nós para o qual ainda não se conhece um caminho mínimo.

$t \equiv$ contador de iterações.

$V \equiv$ representando o conjunto dos nós rotulados e abertos em G (grafo).

$f \equiv$ índice do vértice a ser fechado na iteração t .

$C = [c_{ij}] \equiv$ matriz de custos representando as distâncias entre vértices ligados diretamente.

$d_{ij} \equiv$ a distância entre os vértices x_i e x_j .

$d_{ij}^t \equiv$ a distância calculada entre os vértices x_i e x_j na iteração t .

$vet(i) \equiv$ vetor que guarda o vértice que deu origem à distância calculada para o vértice de índice i .

$\gamma(f) \equiv$ conjunto de vizinhos do vértice de índice r .

3.3 ALGORITMO HÍBRIDO MONO-OBJETIVO

No Capítulo 2 foi apresentado o problema de rede de sensores sem fio, mostrando suas aplicações e limitações. Nas seções anteriores foram apresentados conceitos básicos dos algoritmos que serão aplicados na resolução deste problema. São eles: Algoritmos Genéticos (AG's) e Caminho Mínimo (PCM). Então, juntando todos os conceitos apresentados, será

Algoritmo 1 Algoritmo Dijkstra (Caminho Mínimo)

INÍCIO

Ler os dados de $G = (N, A)$, onde d_{ij} é a distância entre os nós vizinhos de G

Inicializar variáveis $d_{11} \leftarrow 0$; $\{d_{1i} \leftarrow \infty \forall i \in N \setminus \{x_1\}\}$; $V \leftarrow \{x_1\}$; $A \leftarrow \{N\}$;
 $O \leftarrow \emptyset$; $\{vet(i) \leftarrow 0, \forall i \in N\}$

para $t = 1$ a n **faça**

$f \leftarrow x_i$ tal que $d_{1i} \leftarrow \min_{x_i \in A} \{d_{1i}\}$

$O \leftarrow O \cup \{f\}$;

$A \leftarrow A \setminus \{f\}$;

$V \leftarrow A \cap \gamma(f)$

para $i \in V$ **faça**

$r \leftarrow \min \{d_{1i}^{t-1}, (d_{1r} + d_{ri})\}$

se $r < d_{1i}^{t-1}$ **então**

$d_{1i}^t \leftarrow r$;

$vet(i) \leftarrow f$;

fim se

fim para

fim para

introduzido nesta seção o Algoritmo Híbrido Mono-Objetivo, que aplica tanto AG's como PCM para a solução do problema apresentado anteriormente.

Como já citado no Capítulo 2, os nós sensores são usados em conjuntos, formando uma cadeia de cooperação para um determinado fim, pois quando se trata apenas de um nó sensor individualmente, sabe-se que seus recursos são bastante limitados. Ao ter basicamente um conjunto de sensores, juntamente com pelo menos um sorvedouro, pode-se então montar uma rede de sensores sem fio (RSSF). Como visto na Figura 2.1, cada sensor tem um processador onde é rodado um protocolo ou um determinado algoritmo. É esse algoritmo embutido em cada sensor que faz seu controle, decidindo como ele irá se portar, ficando ativado, desativado ou “dormindo”. Juntamente com esse algoritmo, tem-se também outro algoritmo que faz o controle da rede. Este algoritmo é rodado no nó sorvedouro uma vez que o mesmo exige maior processamento. Justamente esse tipo de algoritmo de tratamento da rede é que será mostrado aqui.

Estando de posse da descrição do problema (Seção 2.1) e a modelagem matemática (Seção 2.2), foi implementada uma heurística para a solução deste problema. Pois se sabe, e também será mostrado aqui e nas próximas seções, que a solução exata, através da modelagem matemática, se torna inviável quando a quantidade de nós sensores presente na rede aumenta. E como as RSSF são usadas em grandes aplicações como: monitorar florestas e grandes centros urbanos, grandes quantidades de sensores são indispensáveis.

Então neste capítulo será apresentada uma das abordagens usadas para implementar o programa que é responsável por controlar a rede. A este foi dado o nome de Algoritmo Híbrido (AH).

O AH, em sua essência, faz uso, basicamente, de três algoritmos: Algoritmo Genético (AG), Dijkstra e Local *Online*. O AG é usado para tratar a cobertura da rede decidindo

quais nós sensores devem ficar ativos em determinado período de tempo. O algoritmo de Dijkstra é usado para determinar o caminho de comunicação, em que cada nó sensor deve enviar suas informações ao nó sorvedouro. Cada um destes algoritmos foram implementados dentro de seus conceitos genéricos, no entanto existem algumas particularidades que foram introduzidas para se adaptar ao uso destas soluções para RSSF aqui apresentada. Por fim o algoritmo *Local Online* faz uma busca local entre os sensores disponíveis na rede, no momento em que um sensor falha, parando de funcionar por algum motivo. Serão detalhadas individualmente as particularidades de cada etapa aqui mencionada e como elas foram implementadas.

3.3.1 Algoritmo Genético Implementado

Como já citado na Seção 3.1, sabe-se que para ser considerado um AG, deve-se ter pelo menos uma unidade de cada operação genética básica. São elas: cruzamento, mutação e seleção. Além destas operações elementares, que serão aqui descritas, vale a pena observar a grande importância da modelagem e adaptação do AG ao problema, ou seja, como ele foi codificado. Então para um melhor entendimento de como foram feitos os operadores, será mostrado como o algoritmo foi codificado.

3.3.1.1 Codificação A codificação de um AG é a primeira etapa de sua implementação. Uma má codificação pode afetar todo o resultado [Carrano et al., 2006]. Pode-se ter um bom algoritmo implementado com excelentes operadores, no entanto, se a codificação do AG não for de uma boa qualidade, pode comprometer todo o trabalho.

A etapa da codificação consiste na abstração do problema real. Existem várias formas de codificação de AG's, dentre elas tem-se: real, binária, por permutação, por árvore, dentre outras. Entende-se que a melhor codificação para resolver o problema da RSSF aqui proposto é a codificação binária.

Como dito na Seção 3.1, um AG é composto por uma população de indivíduos. Cada indivíduo é representado por um cromossomo. Como a codificação adotada é a binária, cada cromossomo é representado por uma cadeia de bits, podendo cada gene assumir valores “0” (zero) ou “1” (um), Figura 3.2.

$$\mathbf{g} = \begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \mathbf{g} & = & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{1} & \boxed{0} & \boxed{1} \end{array}$$

Figura 3.2 Exemplo de um cromossomo de 8 bits

Cada gene de um indivíduo no AG representa um sensor da rede, e o conjunto de todos os genes que é o cromossomo, mostra todos os sensores, informando os ativos e os não ativos (Figura 3.3), sendo $g(\cdot)$ a variável que representa o vetor cromossomo, $g(i)$ representa o gene da posição i , que analogamente representa o sensor i . Sabe-se que $g(i)$ somente pode assumir os valores “0” (zero) ou “1” (um). Então, se o valor de $g(i) = 0$ significa que o sensor i , não está ativo, e se o valor for “1” ele está ativo. Portanto, fica claro que a quantidade de genes presente em um cromossomo é igual ao número de

sensores, ou seja, se uma RSSF é composta por 81 sensores, o cromossomo no AG terá 81 bits.

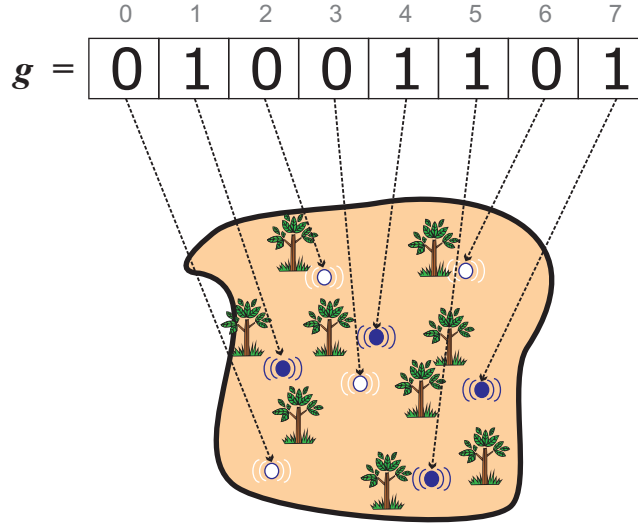


Figura 3.3 Ilustração da codificação de um cromossomo na RSSF.

Para identificar a cobertura de um dado sensor i representado no vetor $g(i)$, tem-se uma matriz binária de cobertura, denominada $matCob(i, j)$. Uma matriz $Ns \times Md$, em que Ns é o número total de sensores presente na rede e Md é o número total de pontos de demanda. Então cada linha desta matriz representa os pontos de demanda cobertos por um determinado sensor, ou seja, caso o sensor i cubra o ponto de demanda j , tem-se $matCob(i, j) = 1$, caso contrário, $matCob(i, j) = 0$.

$$matCob = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Na figura 3.3, tem-se a representação de um indivíduo. Pode-se observar que se trata de um cromossomo de 8 bits em que os bits 1, 4, 5 e 7, possuem valores igual a “1” (um) e os demais “0” (zero). Na RSSF ter-se-ia uma rede de 8 sensores em que apenas os sensores 1, 4, 5 e 7, estariam ativos. Então, para saber quais pontos de demanda seriam cobertos por estes sensores que estariam ativados, bastaria apenas verificar as colunas da matriz $matCob$ de índice 1, 4, 5 e 7. Percorrendo assim cada linha destas colunas seria possível verificar quais pontos de demanda são cobertos por estes sensores.

De posse desta codificação foram implementados os demais operadores do AG.

3.3.1.2 Geração da População Inicial Como já citado, o AG é um algoritmo evolucionário, em que uma população de indivíduos sofre alteração a cada geração (iteração)

do algoritmo. No entanto deve-se ter uma população inicial de indivíduos para que se possa aplicar os operadores a cada geração do algoritmo. Geralmente esta população inicial é gerada de forma aleatória.

Como mostrado na Seção 3.3.1.1 a codificação do AG implementado é binária, em que cada indivíduo (cromossomo) é representado por uma cadeia de bits (exemplos nas Figuras 3.3 e 3.2) que assumem valores 0 ou 1. No caso de problemas de variáveis contínuas em que cada indivíduo representa um ponto de uma determinada função contínua, a codificação binária é realizada de maneira a mapear, com uma determinada precisão, o espaço de busca desejado.

No entanto, em alguns problemas a maneira tradicional de geração inicial da população não se mostra eficiente, podendo além de gerar indivíduos inviáveis, criar soluções extremamente ruins, podendo prejudicar a eficiência do algoritmo. Normalmente este fenômeno acontece ao se resolver problemas que têm por característica serem discretos. Sabendo-se que o problema em questão, o da RSSF, é um problema discreto, foi realizado um estudo de como gerar uma população inicial de uma forma mais eficiente.

Em um problema de RSSF sabe-se inicialmente a área que se deseja monitorar, sabendo-se também as características do sensor usado, principalmente a capacidade de seu raio de cobertura. Através de considerações geométricas, consegue-se prever quantos sensores são necessários para que se consiga ter uma porcentagem determinada da área coberta. Este cálculo é feito da seguinte forma. Sendo:

r_s o raio de sensoriamento de um sensor.

a_s a área coberta por um sensor.

γ fator de correção devido a área a ser monitorada ser de forma retangular e a área de sensoriamento de cada sensor ser circular.

a_m a área a ser monitorada.

f_{cob} fator de cobertura que se deseja em a_m . Sendo que $\{f_{cob} \in \mathfrak{R} \mid 0 < f_{cob} \leq 1\}$

mSA a estimativa da quantidade mínima de sensores que devem estar ativos para que cubra uma determinada área desejada.

Temos que a área coberta por um sensor é dado por:

$$a_s = \pi r_s^2 \quad (3.1)$$

O fator de correção γ é calculado através da área de um quadrado inscrito na circunferência de raio igual ao raio de cobertura do sensor no qual está sendo utilizado na rede. Sendo assim têm-se gama como sendo:

$$\gamma = \frac{\pi}{2} \quad (3.2)$$

De posse destes dados, pode-se calcular a estimativa da quantidade mínima de sensores que se deve ter para cobrir uma determinada área desejada, que é dado por:

$$mSA = \frac{a_m f_{cob}}{a_s} \quad (3.3)$$

$$mSA_\gamma = \lceil mSA \cdot \gamma \rceil \quad (3.4)$$

Sabe-se que em algumas RSSF existem muitos nós sensores, e que para monitorar toda uma área desejada não é necessário usar todos eles em um mesmo período de tempo. Ou seja, quando se tem uma rede densa, a quantidade de nós que necessitam ficar ativos em um determinado período é relativamente pequeno comparando-se com a quantidade de nós disponíveis. Por exemplo, de posse de uma rede com 81 nós sensores (modelo usado neste trabalho) para monitorar uma área de $2500m^2$, calcula-se a equação 3.4. O resultado mostra que são necessários apenas seis sensores para que se consiga cobrir toda área desejada. Seis sensores representam menos de 8% dos 81 sensores disponíveis.

Tendo em vista esta observação foi elaborada uma forma para que seja feita a geração da população inicial, calculando o valor de mSA , a quantidade mínima de sensores que são necessários para que se cubra uma determinada área. O indivíduo é gerado inicialmente com todos os sensores desativados, então de forma aleatória seleciona-se mSA bits para mudar de estado, ou seja, passando para ativo. Para um melhor entendimento são mostradas no algoritmo 2 as etapas para se gerar a população inicial aplicada à RSSF. Vale a pena destacar a possibilidade de se melhorar a maneira que esta população inicial é gerada, podendo considerar a área coberta ou até mesmo a energia disponível em cada sensor, no entanto, neste trabalho foi abordado uma forma mais simples.

Algoritmo 2 Geração da População Inicial para o AG usado na RSSF (ENTRADA: r_s , a_m , f_{cob} , $nPop$, $nBits$)

INÍCIO

Inicializa variável $popCromossomo$;

$mSA = \text{calcular_mSA}(r_s, a_m, f_{cob})$ /* Equação: 3.4 */

para $n = 1$ a $nPop$ **faça**

$cromossomo = popCromossomo.get(n)$

para $b = 1$ a $nBits$ **faça**

$cromossomo(b) = 0$

fim para

para $t = 1$ a mSA **faça**

$id = \text{abs}(\text{rand}(1)nBits)$ /* gera uma posição aleatória no vetor Cromossomo. */

enquanto $(cromossomo(id) = 1)$ **faça**

$id = \text{abs}(\text{rand}(1)nBits)$ /* garante a não repetição da posição */

fim enquanto

$cromossomo(id) = 1$

fim para

fim para

3.3.1.3 Cruzamento A operação de cruzamento é uma etapa fundamental para um AG, pois é ela quem homogeneiza a população ao longo das gerações, podendo assim, convergir para um ótimo local, que em algumas situações pode até ser um ótimo global

da função objetivo a ser otimizada. Esta operação, feita da forma tradicional, faz uma recombinação entre dois indivíduos, gerando assim novos indivíduos filhos.

Existem várias formas de desenvolver este operador. Por exemplo, pode-se ter: cruzamento de um ponto, cruzamento multi-ponto, cruzamento uniforme, dentre outros. Aqui, será detalhada apenas a forma do cruzamento adotado neste trabalho, o cruzamento de um ponto. Neste tipo de cruzamento são formados os pares a serem cruzados, e cada par tem certa probabilidade de cruzamento (geralmente essa probabilidade varia entre 5% a 100%). Caso o casal seja selecionado para cruzamento, escolhe-se um determinado ponto de corte e faz-se uma permutação das partes dos indivíduos, gerando a partir de dois indivíduos pais, dois indivíduos filhos, que substitui os pais na população corrente. Para auxiliar o entendimento, tem-se a Figura 3.4 que mostra um exemplo deste tipo de cruzamento. Pode-se ver que foi feito um corte entre os bits 3 e 4 no “pai 1” e “pai 2”, dividindo os cromossomos em duas partes, “parte 1” e “parte 2”. Então são gerados os dois filhos permutando as partes dos pais, ficando o “filho 1” formado pela “parte 1” do “pai 1”, mais a “parte 2” do “pai 2”. Analogamente o “filho 2” é formado pela “parte 1” do “pai 2”, mais a “parte 2” do “pai 1”. Em muitos casos, o ponto de corte onde é feita a divisão do cromossomo é escolhido aleatoriamente, no entanto, na implementação aqui realizada, a escolha foi determinística, fixada sempre na metade do cromossomo.

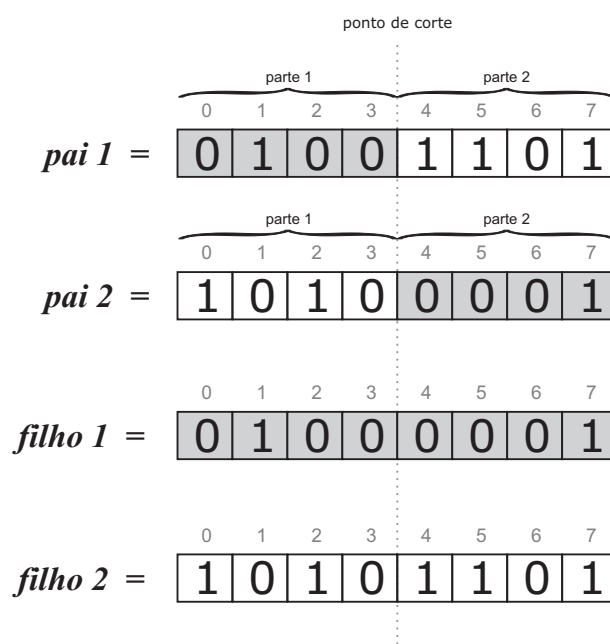


Figura 3.4 Exemplo de cruzamento, com um ponto de corte, em um AG.

Se o AG tivesse apenas a operação de cruzamento, poderiam ocorrer várias convergências prematuras (convergência para um ponto que não é o ótimo global), pois como dito, esse operador tende a homogeneizar a população em torno de uma região. Para evitar tal fenômeno, em contrapartida tem-se o operador de mutação que insere novas características na população.

3.3.1.4 Mutação A mutação em um AG é responsável por inserir mais diversidade na população, assim tornando possível fazer uma maior varredura em busca da solução no espaço de busca. Com este operador, diminui a probabilidade de convergência prematura (convergência em que o AG evolui rapidamente para uma solução não-ótima, ou os mínimos locais).

Em um AG binário, a mutação pode ser realizada fazendo a alteração do estado do gene, ou seja, alterando-se o estado do bit. Por exemplo, se o bit for “0” passa-se ele para “1”, ou vice e versa. Existem várias formas de implementar este tipo de mutação, como: mutação bit a bit, em que cada bit tem uma probabilidade, geralmente inferior a 3%, de mutar-se. Como também uma mutação, em que através de uma probabilidade muito maior, decide se o cromossomo irá sofrer mutação ou não, caso sim, escolhe-se uma determinada quantidade de bits no cromossomo, de forma aleatória, e muda seu estado. Na figura 3.5, se pode ver o caso em que o bit 5 de um determinado cromossomo sofreu alteração, mudando seu estado, de “1” para “0”.

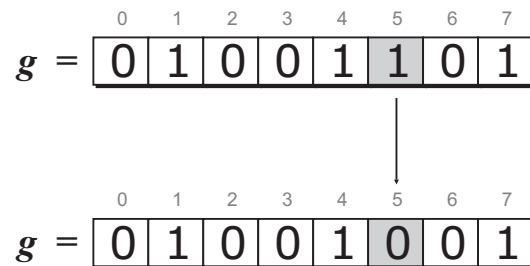


Figura 3.5 Exemplo de mutação em um AG (mutação do bit 5)

Inicialmente adotou-se neste trabalho a mutação bit-a-bit, no entanto percebeu-se que era possível adaptar esta forma de mutação para um padrão que controle a quantidade de nós sensores ativos. A esta nova mutação adaptada para o problema da RSSF foi dado o nome de “mutação de controle”. Com uma probabilidade maior de acontecer, cerca de 20% para cada indivíduo, esta mutação faz uso do valor dado pela variável mSA calculado na Equação: 3.4. Caso o indivíduo seja selecionado para sofrer uma mutação, primeiramente verificam-se quantos sensores estão ativos para este indivíduo, ou seja, em seu vetor de bits, quantos possuem valores iguais a “um”. Caso este valor ultrapasse o valor de mSA escolhe-se aleatoriamente um destes bits de valor igual a “um” para alterar seu estado, passando assim para “zero”. No entanto se o número de sensores ativos no indivíduo for inferior a mSA , executa-se o processo de maneira inversa, escolhendo um sensor dentre os não ativos, para mudar de estado. Já quando o número de bits do cromossomo que possuem valores igual a “um” for igual ao valor dado pela variável mSA , é feita uma mutação de troca, em que se escolhe aleatoriamente um bit no cromossomo que possua valor igual a “um” e também de forma aleatória escolhe-se um bit que possua valor igual a “zero”. Feita esta escolha faz-se uma permutação entre eles, ou seja trocando-os de posição. Para um melhor entendimento a Figura 3.6 mostra as formas de mutação usadas na “mutação de controle” para o caso em que mSA seja igual a 3.

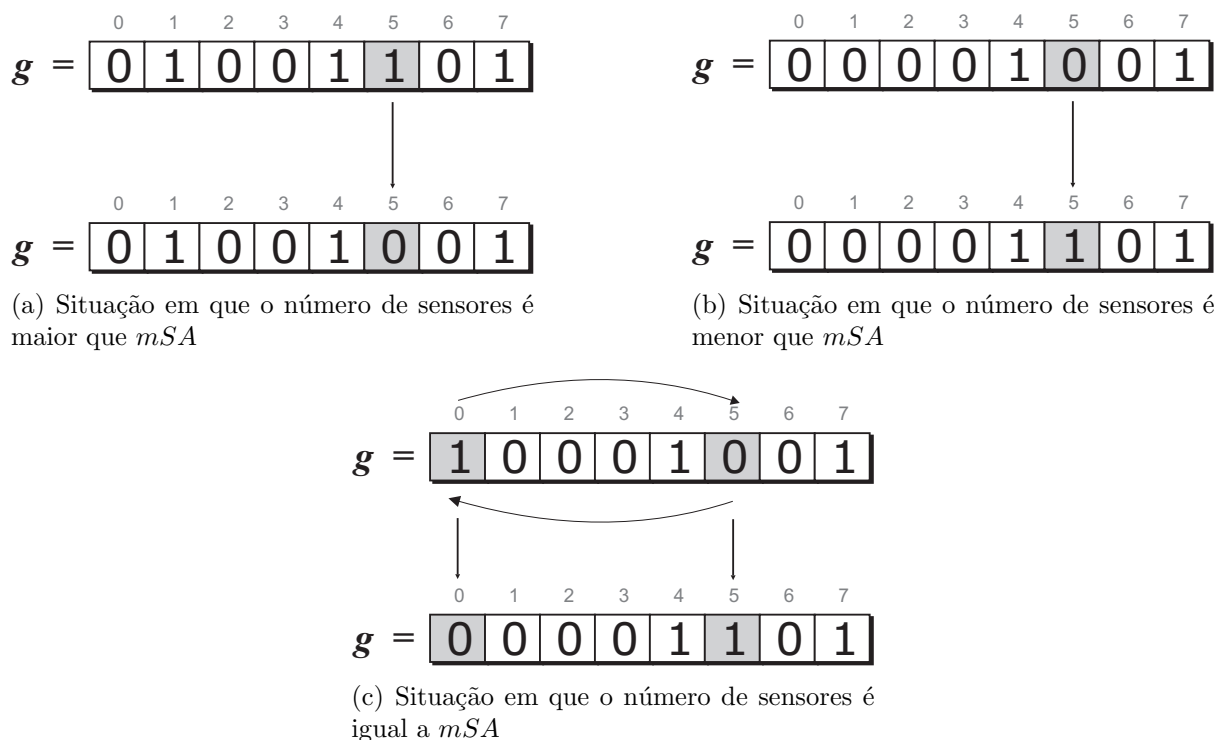


Figura 3.6 Formas de mutação utilizada na “mutação de controle”

3.3.1.5 Seleção Como em todo processo natural de evolução, aqui também ocorre a seleção natural dos indivíduos da população, idealizada inicialmente por Charles Darwin em 1859. É aqui que os melhores da população têm maiores chances de “sobreviverem” à uma nova geração. Existem basicamente três tipos: Determinística, Estocástica e Híbrida. Na seleção determinística existem apenas duas possibilidades, o indivíduo tem 0 ou 100% de chance de sobreviver na próxima geração. Na seleção estocástica, os melhores indivíduos, ou seja, os de melhores valores de função objetivo têm maior probabilidade de surgirem na nova população. Já a seleção híbrida é a combinação destas duas estratégias em etapas diferentes do processo de seleção. No AG aqui apresentado, é usada a seleção determinística, permanecendo sempre os melhores indivíduos da população corrente. A população é ordenada segundo o valor de função objetivo e são banidos os piores até que o número de indivíduos presentes seja igual ao da população inicial.

3.3.1.6 Elitismo O elitismo garante que o melhor ou um conjunto de melhores indivíduos, sempre estarão na geração seguinte. Neste operador, a cada geração é separado o melhor, ou os melhores, cromossomos. Caso os indivíduos separados não surjam na geração seguinte, eles são acrescentados deterministicamente na nova população. No AG aqui implementado foi mostrado que no operador de cruzamento são eliminados os indivíduos pais. Então, pode acontecer da melhor solução da geração corrente ser eliminada, caso ela seja um dos pais descartados. Para que isso não ocorra é implementado o elitismo, separando o melhor indivíduo no fim de cada geração. A operação de elitismo é

realizada no fim do processo de seleção, conferindo se o melhor cromossomo da geração anterior está presente na nova população. Caso não, o pior indivíduo da geração corrente é substituído pelo indivíduo separado.

3.3.1.7 Cálculo da Função Objetivo Em cada iteração do AG é necessário o cálculo da função objetivo (FO) de cada cromossomo. É através dela que é possível fazer a comparação dos indivíduos para que o AG saiba quem são os melhores.

Nesta versão mono-objetivo, a FO (Equação (3.5)) usada no algoritmo genético, inclui:

- O custo para ativar o nó.
- O custo de manutenção de cada nó.
- O custo do caminho mínimo entre o nó i e o sorvedouro mais próximo (denominado CM_i), sendo custo dos arcos dado pela potência de transmissão entre os nós. Este parâmetro é adotado visando penalizar os nós sensores em que o caminho para alcançar o sorvedouro seja muito caro. Os arcos que compõe o grafo são aqueles cuja distância entre os nós não ultrapassa o raio de comunicação máximo do nó transmissor.
- Penalidade de não cobertura.

$$FO(H) = \sum_{i \in \mathcal{S}} (EA_i + PM_i + CM_i) \times y_i + \sum_{j \in \mathcal{D}} EH_j \times h_j \quad (3.5)$$

Tem-se então na equação 3.5 a função objetivo usada para calcular o custo de cada cromossomo. Nela é feito um somatório dos custos de cada sensor presente na rede. Observa-se que o primeiro somatório é multiplicado pela variável y_i . Relembrando, essa variável possui valor “1” se o sensor i estiver ativo e “0” caso contrário, e que cada sensor é representado por um bit no vetor cromossomo “g” (Figura 3.2). Então, o custo resume-se aos sensores ativos da rede, que são identificados segundo a configuração dada pelo cromossomo “g”.

A equação 3.5 resume a modelagem matemática dada pelas equações 2.1 e 2.2. A maior diferença entre elas se encontra no cálculo do custo do caminho. Com a redução do CM_i , automaticamente, pode-se otimizar a topologia da rede ao montar os caminhos de comunicação através do algoritmo de roteamento, como também reduzir, principalmente os custos de energia na transmissão, pois ele é diretamente proporcional à distância.

3.3.2 Dijkstra Implementado

Como mostrado anteriormente o AG é usado para resolver o problema de cobertura, não se preocupando diretamente com a conectividade entre os sensores. Após ele fornecer o vetor de bits que mostra os sensores ativos na solução, é feito um caminho mínimo com cada sensor ativo em direção ao sorvedouro. As arestas para a solução do *Dijkstra* são construídas da seguinte forma:

- Uma aresta entre dois sensores se a distância entre os nós for menor que o raio de comunicação.
- O peso de cada aresta (i, j) é proporcional à corrente gasta pelo sensor i para transmitir um dado para o sensor j . Tem-se então três situações:
 - Caso os dois sensores estejam ativos, o custo da aresta é o valor da corrente gasta.
 - No caso de um sensor estar ativo e outro não, usa-se o valor da corrente multiplicado por uma penalidade.
 - Já quando ambos os sensores se encontram desligados, o custo da aresta é igual ao valor da corrente multiplicado pela penalidade ao quadrado.

Caso o caminho escolhido passe por algum sensor não ativo, automaticamente esse sensor muda de estado, passando assim para ativado. A forma com a qual é distribuído os pesos nas arestas prioriza o caminho na seguinte ordem: ambos sensores ativos, apenas um ativo e por último nenhum ativo. Isso faz com que se evite ao máximo o acionamento de sensores que se encontram desligados, melhorando o consumo de energia da rede. Percebe-se que a vantagem obtida em usar caminho mínimo individual para cada sensor, está em tentar evitar ao máximo a sobrecarga de sensores próximos ao nó sorvedouro, pois serão formadas pequenas árvores a partir do sorvedouro.

A escolha de usar como peso nas arestas o valor da corrente se dá pelo fato de entender que se pode otimizar tanto a distância, como também o consumo de energia durante a transmissão dos dados por cada sensor, pois como já dito, este consumo é diretamente proporcional à distância da transmissão.

3.3.3 Algoritmo Híbrido

O Algoritmo Híbrido (AH) combina uma estratégia de solução global, que reconstrói toda a rede, com uma estratégia local, de menor custo computacional, onde a cada falha ocorrida, uma ação para restabelecer a possível perda na cobertura e conectividade na vizinhança da falha é realizada. O uso de um algoritmo com uma visão global de toda a RSSF, tem a vantagem de poder reestruturar todo o problema de acordo com a situação atual de energia de cada sensor disponível. Entretanto, ela não é escalável, tanto em termos computacionais quanto para disseminação da solução na rede. Já a solução local, tem a vantagem de ser bem mais rápida na obtenção de um resultado, conseqüentemente, demandando um esforço computacional menor, pois sua ação é somente nos arredores de onde a falha se manifesta. A vantagem da abordagem híbrida é combinar cada uma das estratégias, adaptando-se a RSSF nesta única solução.

O AH é composto por um Algoritmo Global Sob-Demanda (AGSD) (Algoritmo 3), que faz uso de Algoritmo Genético (Seção 3.3.1) como em [Quintao et al., 2004], para resolver um Problema de Cobertura baseado no Problema de Cobertura de Conjunto. Também faz uso do algoritmo de *Dijkstra* (Seção 3.3.2) para construir a conectividade que garante a disseminação dos dados de uma forma eficiente pela rede.

A função objetivo (Equação (3.5)) usada no algoritmo genético inclui o custo para ativar o nó, mais o custo do caminho mínimo entre o nó i e o sorvedouro mais próximo (denominado CM_i), no qual o custo dos arcos é dado pela potência de transmissão entre os nós. Este parâmetro é adotado visando penalizar os nós sensores em que o caminho para alcançar o sorvedouro seja muito caro. Os arcos que compõe o grafo são aqueles cuja distância entre os nós não ultrapassa o raio de comunicação máximo do nó transmissor.

Algoritmo 3 GlobalSobDemanda(**entrada:** S, m, D, RS (raio de sensoriamento), RC (raio de comunicação))

Inicie população, aleatoriamente
 Calcule função objetivo (*aptidão* Equação 3.5) de cada cromossomo
 Ordene a população de acordo com o *fitness*
enquanto (houver melhoria do melhor indivíduo em 50 gerações) **faça**
 Faça o casamento da população e realize *crossing over* (probabilidade α)
 Realize mutação sobre a população, com probabilidade π
 Calcule função de (*aptidão*) de cada cromossomo
 Remova os piores cromossomos até que retorne ao tamanho original da população
fim enquanto
para todo ($i \in S$) **faça**
 $PC_i = Dijkstra(i, sink)$
fim para

3.3.3.1 Local Online O AH é composto também por um Algoritmo Local que é acionado a cada perda de cobertura e conectividade devido a falhas de nós. Seja $c(i, j)$ o valor da corrente de transmissão entre os nós i (candidato a substituto) e j , e F o conjunto de filhos do nó que falhou. O algoritmo seleciona um novo nó baseado no valor da Equação (3.6). A distância ao quadrado é utilizada para que a escolha dos nós priorize um candidato que esteja, ao mesmo tempo, perto do pai e dos filhos do nó que falhou, dado que a corrente de transmissão é diretamente proporcional à distância entre os nós. O procedimento Local é mostrado em alto nível no Algoritmo 4.

$$Valor(i) = c^2(i, pai\ do\ sensor\ falho) + \sum_{j \in F} c^2(i, j) \quad (3.6)$$

Após a escolha do nó sensor para substituir (SE), duas situações podem ocorrer:

- Haver possibilidade de conexão direta entre o SE e os ascendentes e descendentes diretos de SF, bastando então apenas redirecionar as arestas.
- Não haver conectividade em algum dos elementos. Usa-se então o algoritmo do caminho mínimo até ao sorvedouro para conectar o elemento à rede (Seção 3.3.2).

Na figura 3.7, pode-se ver uma ilustração das etapas do algoritmo *Local Online*. No período de tempo t , na Figura 3.7a, a rede se encontra em perfeito funcionamento, no

Algoritmo 4 *LocalOnline* (**entrada:** S, SF (sensor falho), C (filhos de SF), m, D, RS (raio de sensoriamento), RC (raio de comunicação))

se (Perda de Cobertura = TRUE) **então**
 Encontrar melhor nó para substituir SF
 Conectar *MelhorNó* na rede; /*busca por uma melhor rota do *MelhorNó* ao sorvedouro(*m*)*/
fim se
para todo ($i \in C$) **faça**
 Conecta o nó *i* na rede; /*busca por uma melhor rota do nó *i* ao sorvedouro(*m*)*/
fim para

entanto no período de tempo $t + 1$, a Figura 3.7b, mostra que um sensor na rede falha, provocando uma interrupção na cobertura e na comunicação. Por fim, na Figura 3.7c é mostrada a ilustração da correção feita pelo algoritmo *Local Online*, em que o sensor “7” foi escolhido para substituir o sensor “2” que falhou. As arestas de comunicação dos sensores “4” e “5” foram redirecionadas para o novo sensor ativo, “7”. Como o raio de comunicação do sensor “7” não é suficiente para que se consiga uma comunicação direta com o sensor “1” (antes o “sensor pai” do sensor falho), então o algoritmo de *Dijkstra* como mostrado na Seção 3.3.2 foi acionado, criando assim um novo caminho de comunicação. Percebe-se que neste novo caminho, foi necessária a ativação do sensor “8”.

Como o objetivo é otimizar a energia gasta na RSSF, deve-se introduzir a forma em que esta será rotulada e mencionada durante o trabalho. Sabe-se que cada sensor individualmente tem sua capacidade de energia dada por sua bateria. Sendo:

Eb_i^t a quantidade de energia da bateria, presente no sensor i no instante t .

S o conjunto de nós sensores.

Estando $S_a(t) \subset S$, tem-se que $S_a(t)$ é o conjunto de sensores ativos no instante t .

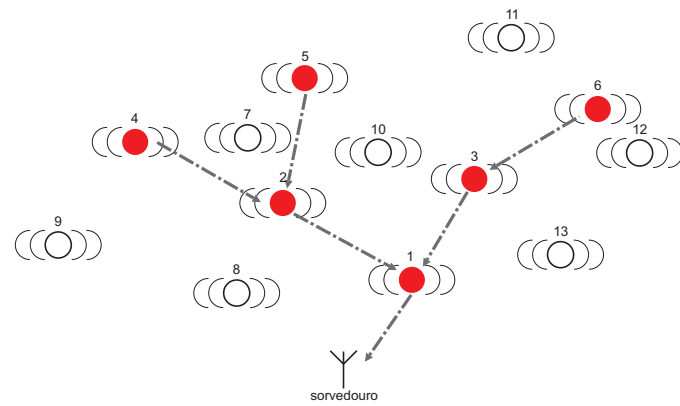
Então, tem-se:

- Energia Residual (E_{res}^t): É a soma de toda energia presente na rede no instante t , ou seja, é a soma da energia armazenada em todos os sensores, ativos ou não.

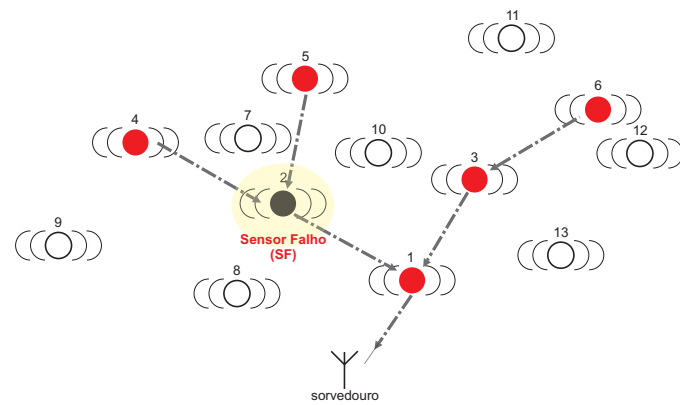
$$E_{res}^t = \sum_{i \in S} Eb_i^t \quad (3.7)$$

- Energia Consumida (E_{cons}^t): É a soma de toda energia consumida pelos sensores ativos da rede no instante t . Matematicamente, tem-se:

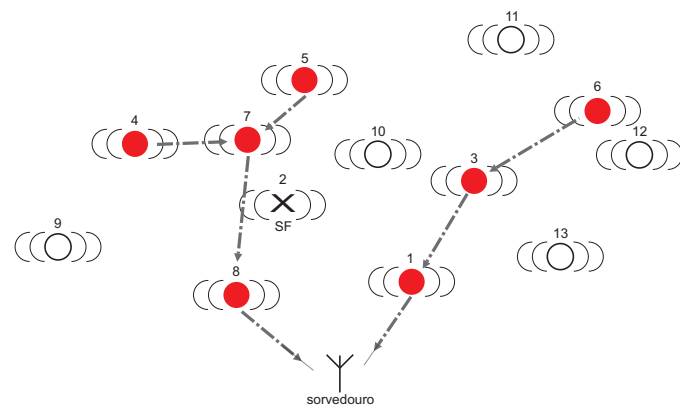
$$E_{cons}^t = \sum_{i \in S_a(t)} Eb_i^t \quad (3.8)$$



(a) Etapa 1 - Rede em funcionamento normal no período de tempo t



(b) Etapa 2 - Falha de um sensor da rede no período de tempo $t + 1$



(c) Etapa 3 - Correção dada pelo algoritmo Local no período de tempo $t + 1$

Figura 3.7 Uma ilustração das etapas do Algoritmo Local *Online*

Algoritmo 5 Híbrido(**entrada:** S , m , D , RS (raio de sensoriamento), RC (raio de comunicação))

```

GlobalSobDemanda(S,m,D,SR,CR); /*gera solução inicial*/
enquanto ( $S \neq \emptyset$ ) faça
  Falha = Verificar Falha de Nós;
  se (Falha = VERDADEIRO) então
    LocalOnline(S,f,C,m,D,SR,CR); /* Restaurar cobertura e conectividade */
     $S = S - f$ ; /*exclui nó f do conjunto S*/
    se (Condição para Executar AGSD = Verdadeiro) então
      GlobalSobDemanda(S,m,D,RS,RC);
    fim se
  fim se
fim enquanto

```

A Abordagem Híbrida, que utiliza os Algoritmos 3 e 4 definidos anteriormente, é mostrada no Algoritmo 5.

Para que o AGSD seja executado no Algoritmo 5 devem ser obedecidas as seguintes condições:

Seja:

- j o período de tempo onde houve a última chamada ao AGSD.
- $S_a(t)$ o número de sensores ativos no período t .

Caso o somatório:

$$\sum_{m=j}^t |S_a(m+1) - S_a(m)|$$

seja igual a 5% de $|S|$, o conjunto de sensores presentes, ou quando o nível de energia consumida da rede aumentar em 5%, então o Algoritmo Global é chamado.

O período t é então rotulado como o último período de tempo onde o AGSD foi chamado.

3.4 RESULTADOS COMPUTACIONAIS

Os testes apresentados se dividem em duas partes. Primeiramente, foram realizados experimentos em um ambiente onde cada sensor da rede estava sujeito a falhas mecânicas e por falta de energia. O tempo de monitoração da rede foi de 100 (u.t.). Para que se possa contabilizar o consumo de energia cada unidade de tempo representa 1h. Na segunda parte, os testes consideraram apenas falhas causadas por falta de energia, e neste caso o tempo de monitoração não foi limitado, deixando-se a rede funcionar até que não houvesse mais cobertura e/ou conectividade.

A falha mecânica mencionada acima permite em média a falha de um sensor por unidade de tempo e é simulada por um gerador de falhas. A cada unidade de tempo

o gerador determina se haverá ou não uma falha. No caso da resposta ser positiva, um sensor, podendo estar ativo ou não, é escolhido aleatoriamente com distribuição uniforme. A probabilidade de que ocorra a falha é modelada por uma função exponencial, em que no tempo inicial tem aproximadamente 12% de chance de ocorrência e no tempo final alcança patamares de 70%. Na Figura 3.8 tem-se o gráfico exemplificando o comportamento desta exponencial. Como se pode observar ocorre uma variação entre 12 a 70% de probabilidade de falhas no período de 1 a 100 unidades de tempo. Por exemplo, no gráfico é mostrado que para o período de tempo igual 70 u.t tem-se 31.63% de probabilidade de surgir uma falha em um sensor presente na rede.

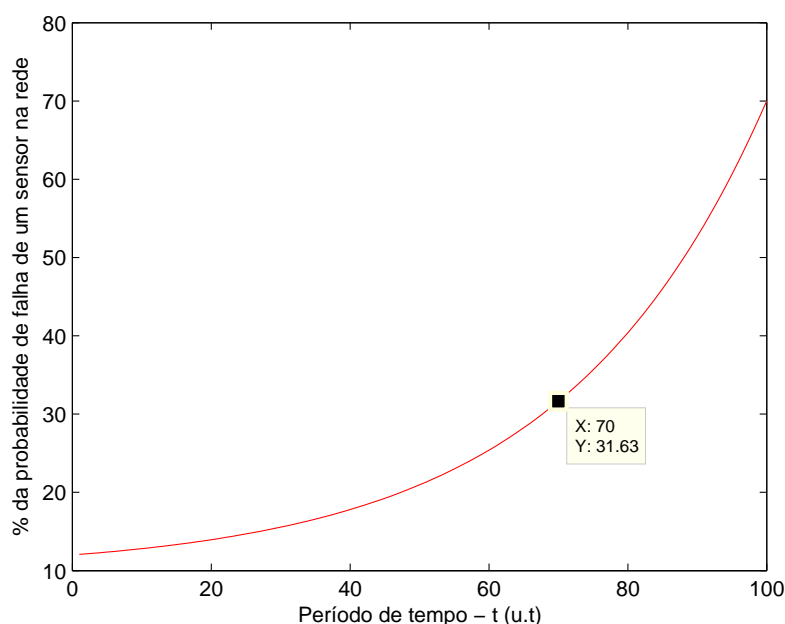
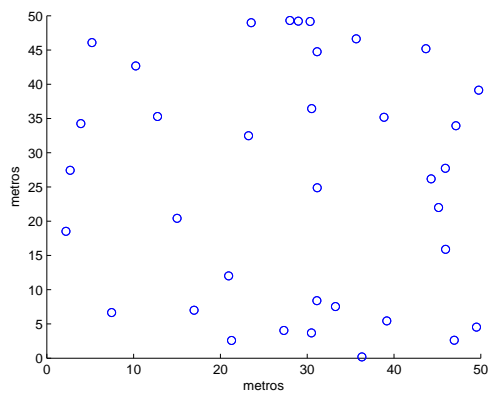


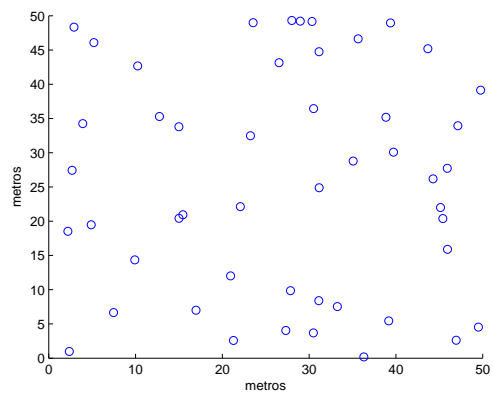
Figura 3.8 Curva exponencial que modela a probabilidade de falha de um nó sensor no período de tempo t

Dispondo da área a ser monitorada, primeiramente deve-se inserir os nós sensores nesta região. Uma das formas usuais é jogar estes sensores através de um avião, pois muitas vezes, estas áreas são de difícil acesso [Ruiz, 2003]. Para simular a posição que cada sensor obteve nesta região, foram gerados de forma aleatória, pontos nesta área. A distribuição usada foi a uniforme, ou seja, para cada ponto da área, a probabilidade de surgir um sensor é igual. Na Figura 3.9 pode-se ver a distribuição aleatória de forma uniforme de 36, 49, 64, 81 e de 100 sensores em uma área de $2500m^2$. Observa-se que a instância de 49 sensores foi gerada a partir da instância de 36 sensores, ou seja, adicionando mais 13 sensores de forma aleatória e uniforme. As demais instâncias foram geradas da mesma forma, sempre adicionando os sensores na rede já existente. Decidiu-se fazer desta maneira, para que o testes sejam comparados de forma mais precisa, sendo possível então, uma identificação de forma mais correta, da influência da inserção de novos nós sensores em uma RSSF.

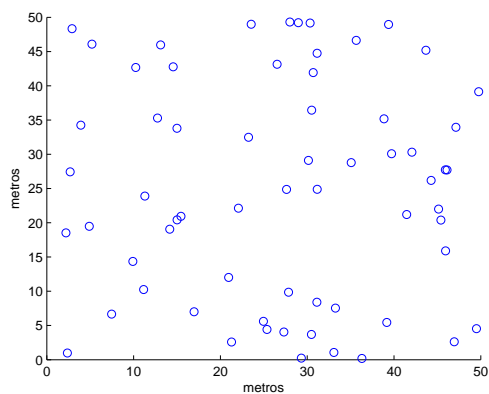
Todos os testes realizados a seguir, correspondem a monitoração nesta área de $2500m^2$.



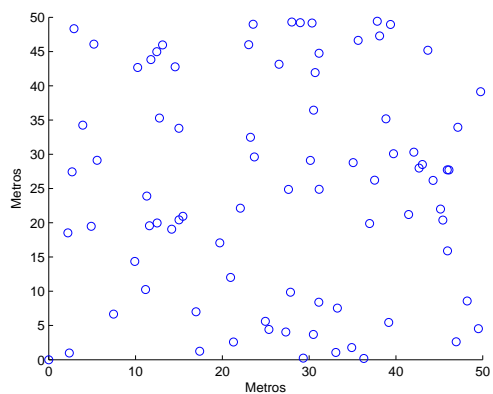
(a) Distribuição uniforme de 36 sensores



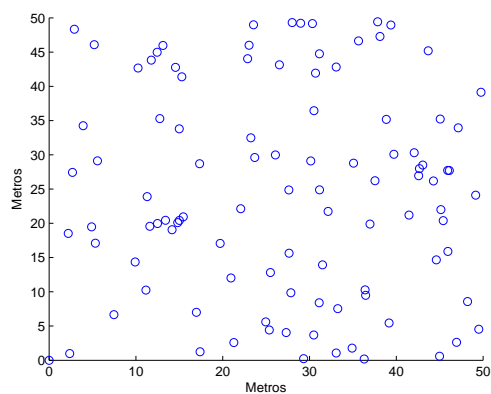
(b) Distribuição uniforme de 49 sensores



(c) Distribuição uniforme de 64 sensores



(d) Distribuição uniforme de 81 sensores



(e) Distribuição uniforme de 100 sensores

Figura 3.9 Ilustração da distribuição usada para 36, 49, 64, 81 e 100 sensores em uma área de $2500m^2$

Cada tipo de sensor possui características e parâmetros individuais, e todos os valores aqui apresentados, são baseados no nó sensor comercial Mica2 [Crossbow, 2006]. Tem-se os seguintes parâmetros para cada sensor da RSSF ¹:

$RS \leftarrow 15m$ (raio de sensoriamento).

$RC \leftarrow 25m$ (raio de comunicação).

$EA \leftarrow 5mAh$ (energia de ativação).

$PM \leftarrow 13mA$ (potência de manutenção).

$PR \leftarrow 2mA$ (potência de recepção).

$EB \leftarrow 1000mAh$. (energia da bateria).

A potência de transmissão (PT) é calculada com base na distância em que o nó sensor transmite os dados. Os valores respectivos da energia consumida de acordo com a distância pode ser vista na Tabela 3.1.

A EA é retirada de cada sensor quando ele passa do estado de não ativo para o estado de ativo. A PM é a estimativa da soma das energias consumidas com processamento e manutenção do sensor, ela é gasta a cada período de tempo. Já a PR e PT é proporcional à quantidade de dados que são recebidos e transmitidos respectivamente. Os valores para PR e PT descritos acima, correspondem respectivamente ao gasto para receber e transmitir um pacote de dados em uma unidade de tempo. Cada sensor coleta um pacote de dados a cada período de tempo e transmite o seu pacote, mais os pacotes que foram recebidos, ou seja os enviados por seus filhos (sensores conectados a ele).

Para se comparar a eficiência da abordagem proposta neste trabalho, seus resultados serão comparados com as seguintes versões de algoritmos:

- Algoritmo Periódico.
- Algoritmo Local *Online*.
- Solução exata pelo modelo matemático.

3.4.1 Algoritmo Periódico

Mais usual na literatura, o Algoritmo Periódico (AP) trata a RSSF de uma forma estática, não podendo responder a falhas em tempo real. A rede é tratada em períodos de tempo definidos, por exemplo, tendo uma área a ser monitorada por 100 unidades de tempo, divide-se todo esse período em pequenos períodos de tempo, podendo ser 10 períodos de 10 unidades de tempo cada. Então, apenas a cada dez unidades de tempo, o algoritmo

¹Supondo fixada uma tensão na bateria, a energia armazenada ou retirada da bateria é proporcional à *corrente x tempo*, podendo ser medida na unidade A.h. A potência, por sua vez, é proporcional à corrente, podendo então ser medida em A.

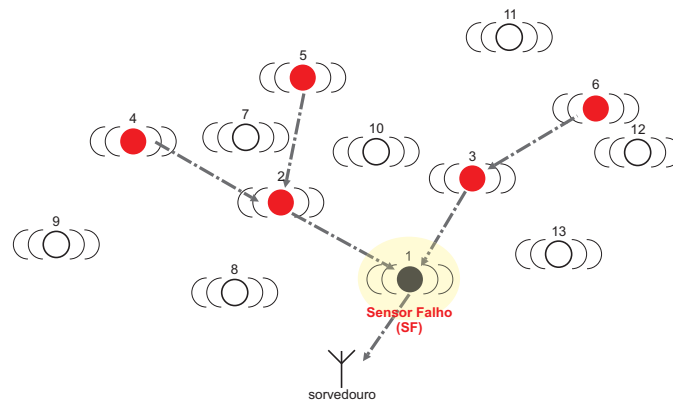
Alcance (m)	Corrente (mA)
5.142	8.6
5.769	8.8
6.473	9.0
7.263	9.0
8.150	9.1
9.144	9.3
10.260	9.3
11.512	9.5
12.916	9.7
14.492	9.9
16.261	10.1
18.245	10.4
20.471	10.6
22.969	10.8
25.771	11.1
28.916	13.8
32.444	14.5
36.403	14.5
40.845	15.1
45.829	15.8
51.420	16.8
57.695	17.2
64.735	18.5
72.633	19.2
81.496	21.3
91.440	25.4

Tabela 3.1 Consumo de corrente do nó sensor com a transmissão

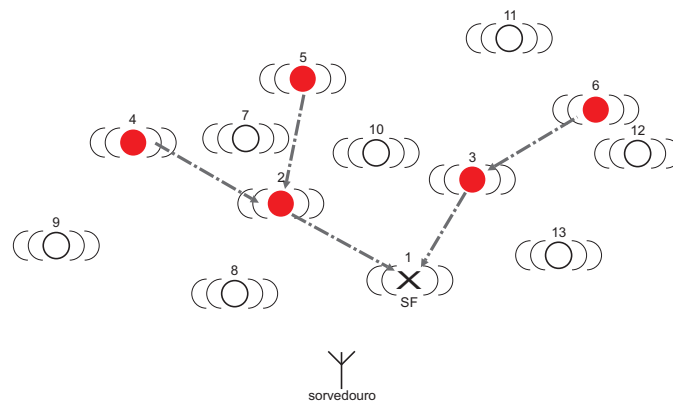
irá entrar em ação, podendo assim alterar alguma configuração na rede em vigor. Contudo, fica claro, que se durante o intervalo de cada período, ocorrer qualquer problema na cobertura ou na conectividade da rede, devido à perda de algum nó sensor ativo, nenhuma providência é tomada. Esta característica faz com que este tipo de algoritmo seja extremamente ineficiente enquanto a falha de nós é considerada.

A Figura 3.10 mostra um exemplo em que um nó sensor (*nó sensor 1* na figura), extremamente importante para o roteamento dos dados da rede, falha em uma unidade de tempo t . Suponha-se que $t + 1$ não corresponda ao próximo período de tempo do AP e considerando a ação apenas deste algoritmo, haverá uma perda significativa da conectividade, e conseqüentemente da cobertura, pois os dados que antes passavam pelo *nó sensor 1*, não poderão chegar mais ao sorvedouro, sendo assim, não processados. Além disto, os sensores ativos, que dependiam deste roteamento, continuarão ativos e consumindo energia de suas baterias.

A implementação aqui usada do AP, consiste de chamadas do AGSD (Algoritmo: 3) a cada 10 unidades de tempo, não tendo nenhuma característica Local, ou seja, ele não reage imediatamente à ocorrência de falhas.



(a) Falha do *nó sensor 1*



(b) Interrupção do roteamento, provocado pela falha do *nó sensor 1*

Figura 3.10 Exemplo da falha de um nó sensor, importante para o roteamento, em uma rede com abordagem periódica

3.4.2 Algoritmo Local Online

O Algoritmo Local *Online* consiste na aplicação do AGSD (Algoritmo: 3) para definir uma solução inicial para a RSSF, e assim, acionar o Algoritmo Local *Online* (Algoritmo: 4) para tratar a rede de uma forma dinâmica (embora sem buscar uma otimização global do problema dinâmico).

3.4.3 Solução exata pelo modelo matemático

Com a definição matemática dada no Capítulo 2, modelou-se o problema para ser resolvido de forma exata, pelo pacote comercial CPLEX [Ilo, 2006]. Dado a primeira solução pelo CPLEX, ele é novamente acionado toda vez que surgir uma falha na rede.

3.4.4 Resultado dos Testes Considerando Falhas Mecânicas e por Energia

Nesta primeira bateria de testes, serão mostrados resultados obtidos para as redes de sensores formadas por 36, 49, 64 e 81 sensores que foram dispostos em uma área de $2500m^2$ como descrito na Figura 3.9. Como já informado, cada sensor tem sua bateria limitada, além disso, nos testes aqui apresentados, cada um terá uma probabilidade de falha mecânica como descrito anteriormente, modelado pela exponencial mostrada na Figura 3.8.

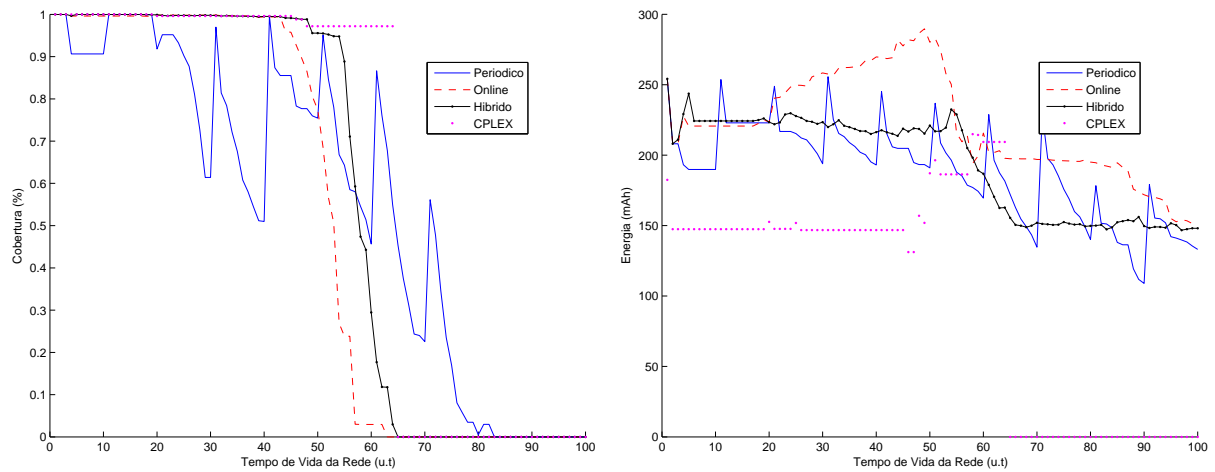
Para cada rede serão mostrados três gráficos, demonstrando assim a energia residual, energia consumida e a cobertura a cada período de tempo. Neste primeiro teste serão mostrados os resultados em 100 unidades de tempo da rede.

Os gráficos demonstrando os resultados para 36, 49, 64 e 81 sensores, podem ser visualizados nas Figuras 3.11, 3.12, 3.13 e 3.14 respectivamente. Todos os gráficos mostrados aqui são de médias em 33 execuções, devido às características aleatórias do algoritmo.

3.4.4.1 Análise dos Resultados Ao observar os gráficos das Figuras 3.11, 3.12, 3.13 e 3.14, percebe-se que o comportamento de cada algoritmo permanece o mesmo, independente da quantidade de sensores. Para todos os casos que serão citados, são mostrados os resultados exatos, dados pela modelagem matemática, e resolvidos pelo pacote comercial CPLEX. Estes resultados servem de limite inferior, no caso da medição de energia consumida, e de limite superior, para a cobertura e a energia residual.

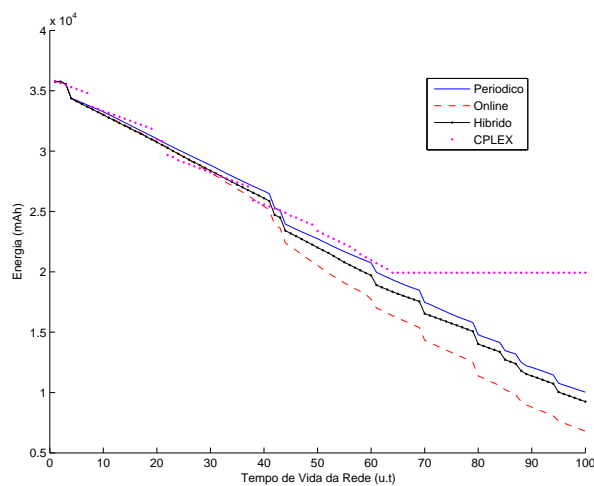
Na versão periódica ocorrem várias oscilações na cobertura, mantendo-se um comportamento bastante instável. Vale a pena citar, que como os gráficos apresentam uma média em várias execuções, não foi mostrado que para casos isolados, em apenas uma execução, o Periódico chega a patamares de praticamente “zero” de cobertura, mesmo nas unidades de tempo iniciais. Isso se dá pelo fato deste algoritmo não tomar nenhuma ação imediata quando há presença de falhas na rede.

Para todos os casos, 36, 49, 64 e 81 sensores, o comportamento da abordagem periódica segue o mesmo comportamento, tanto na cobertura como no consumo de energia. Nas Figuras 3.11a, 3.12a, 3.13a e 3.14a, pode-se ver que a cada 10 períodos de tempo (como descrito na Seção 3.4.1) há uma tentativa de recuperar a cobertura, no entanto logo após começa-se novamente uma queda. O fato de esta versão conseguir atingir um tempo



(a) Cobertura para rede de 36 nós sensores

(b) Energia Consumida rede 36 nós sensores



(c) Energia Residual rede de 36 nós sensores

Figura 3.11 Gráficos de resultado para 36 sensores na abordagem mono-objetivo com falhas mecânicas e por energia

maior de monitoramento é facilmente explicado, pois poucos nós estão ativados a cada período de tempo, devido a falhas que podem ocorrer, no entanto a consequência disto é a grande falha na cobertura. Como já mostrado, este algoritmo não tem mecanismos que façam com que ele reaja a falhas de forma instantânea, podendo tomar decisões apenas nos períodos de tempo que ele é acionado. Nos gráficos das Figuras 3.11b, 3.12b, 3.13b e 3.14b, é mostrado o comportamento da energia consumida. Nestes gráficos observa-se que há picos de energia a cada 10 períodos de tempo. Isto ocorre pelo fato de vários sensores serem ativados devido à reconfiguração da rede, com isto há um gasto maior com a energia de ativação dos sensores neste período.

A versão *Online*, devido à sua característica de responder rapidamente a falhas, consegue, de forma local, suprir a cobertura nos mesmos níveis da solução exata dada pelo CPLEX, até uma certa unidade de tempo. Para o caso de 36 e 49 sensores, mostrado

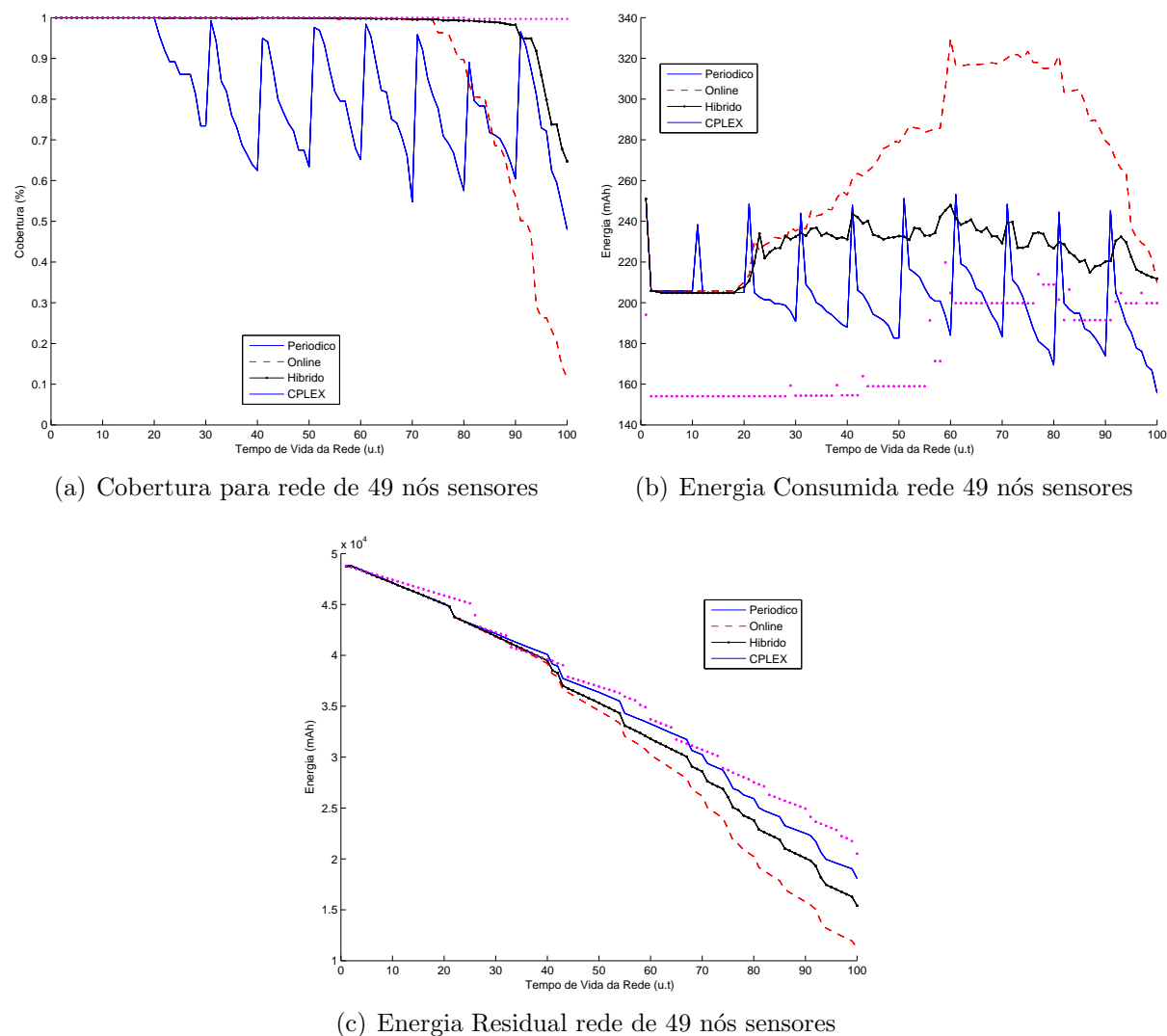
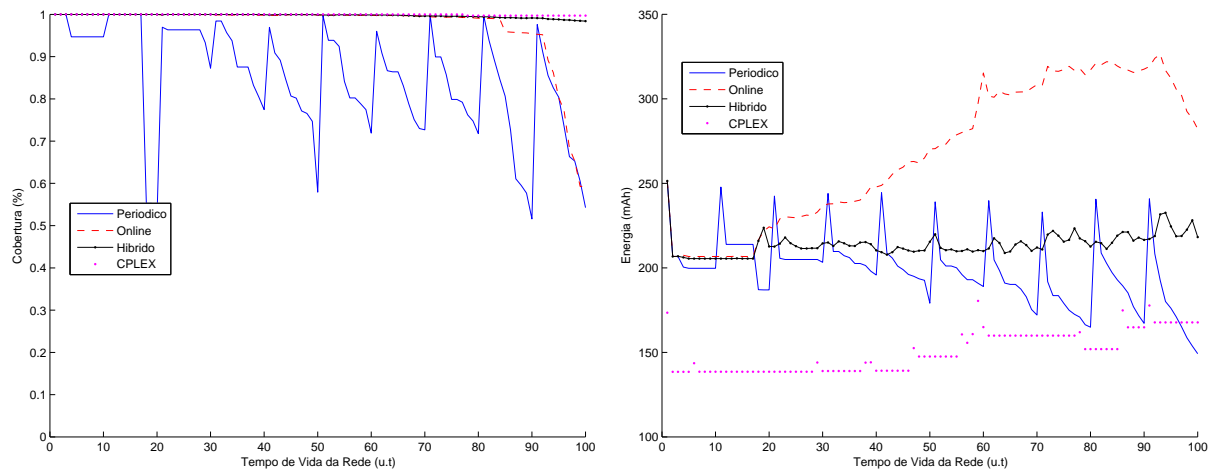


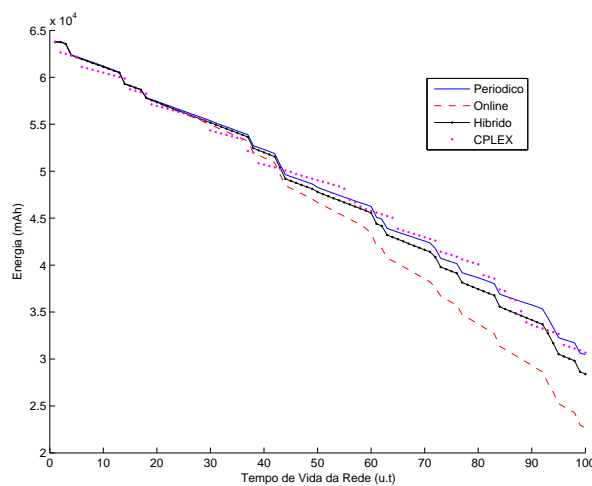
Figura 3.12 Gráficos de resultado para 49 sensores na abordagem mono-objetivo com falhas mecânicas e por energia

respectivamente nos gráficos das Figuras 3.11a, 3.12a, pode-se observar que o algoritmo *Online* é o primeiro a sofrer com a queda na cobertura. A explicação para este comportamento é dada pela característica do algoritmo, que busca sempre um nó sensor para suprir a cobertura perdida por um eventual sensor falho. Desta forma, há apenas mecanismos em que nós sensores são ativados, uma consequência é o constante aumento da energia consumida à medida em que passa o tempo, a qual pode ser vista nos gráficos mostrados nas Figuras 3.11b, 3.12b. Nestes gráficos, observa-se que a curva representada pelo *Online* mantém um crescimento até a unidade de tempo em que também mantém a cobertura. Então, quando a cobertura começa a cair, fica caracterizado que não há mais sensores disponíveis que consiga suprir os sensores que estão sofrendo falhas. Consequentemente, diminuindo a quantidade de sensores, a energia consumida começa a cair também. O fato de mesmo quando não há mais cobertura, ainda haver gastos de energia,



(a) Cobertura para rede de 64 nós sensores

(b) Energia Consumida rede 64 nós sensores



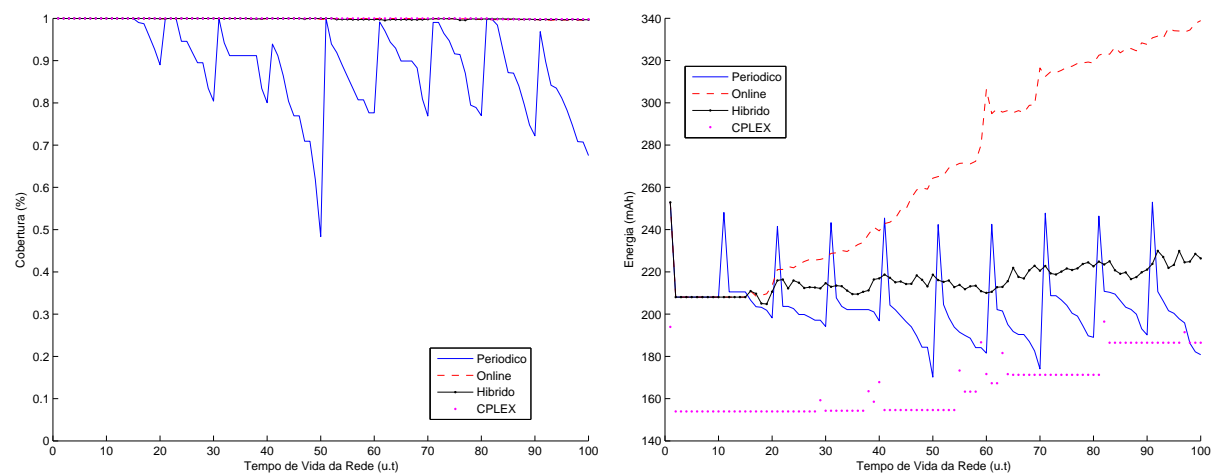
(c) Energia Residual rede de 64 nós sensores

Figura 3.13 Gráficos de resultado para 64 sensores na abordagem mono-objetivo com falhas mecânicas e por energia

é devido a ainda haver sensores monitorando, no entanto sem conectividade, ou seja, não enviando os dados ao sorvedouro.

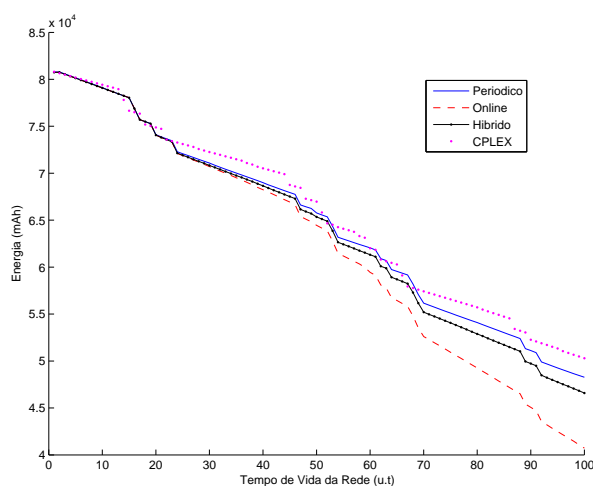
Para os demais casos, de 64 e 81 sensores, o comportamento se mantém semelhante. Como a densidade de sensores aumentou, há uma possibilidade de um maior tempo de monitoramento. Este fato pode ser confirmado através dos gráficos das Figuras 3.13a, 3.14a, em que pode ser visto que o algoritmo *Online* se mantém praticamente todo o tempo no mesmo patamar de cobertura que a solução dada pelo CPLEX. No entanto ao olharmos para os gráficos de energia consumida nas Figuras 3.13b, 3.14b, observa-se que tem-se o mesmo comportamento descrito anteriormente para o caso de 36 e 49 sensores. Mas como a medida foi feita somente até 100 unidades de tempo, não é possível ver o comportamento até o final.

Ao olhar para os resultados para versão *Online* Híbrida (AH) (Figuras 3.11, 3.12, 3.13



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores



(c) Energia Residual rede de 81 nós sensores

Figura 3.14 Gráficos de resultado para 81 sensores na abordagem mono-objetivo com falhas mecânicas e por energia

e 3.14), é visível sua superioridade em relação à versão puramente *Online* e à Periódica. Devido à sua característica de fazer um controle global na densidade de nós sensores ativos em determinados períodos de tempo, a versão AH consegue manter o consumo de energia em patamares mais equilibrados. Ao observar os gráficos relacionados à energia consumida, nas figuras acima, pode-se ver que para todos os casos a energia consumida se mantém praticamente no mesmo nível em todos os períodos de tempo.

Quando analisados os gráficos que representam a cobertura, para os casos de 36, 49 e 64 sensores, é visível a superioridade em relação à versão Periódica e *Online*. Ao ser comparado com o Periódico, o AH tem a grande vantagem de se manter praticamente todo o tempo com a cobertura estável, em cem por cento, ao contrário da versão Periódica em que há uma constante oscilação. Já quando comparado à versão *Online* é notório que o AH consegue atingir tal cobertura por mais tempo. Para a situação em que se tem

INSTÂNCIA	CPLEX		ONLINE		HÍBRIDO	
	Q.UT	TEMPO	Q.UT	TEMPO	Q.UT	TEMPO
36 Sensores	71 u.t.	38.08	73 u.t.	0.04	80 u.t.	0.2356
49 Sensores	116 u.t.	133.56	108 u.t.	0.05	123 u.t.	0.51
64 Sensores	149 u.t.	676.96	126 u.t.	0.05	154 u.t.	0.80
81 Sensores	198 u.t.	2194.90	154 u.t.	0.06	202 u.t.	1.25
100 Sensores	240 u.t.	9972.43	175 u.t.	0.08	243 u.t.	1.82

Tabela 3.2 Tabela de Tempo (em segundos) gasto por cada algoritmo Mono-Objetivo. A coluna “Q.UT” representa a quantidade de unidades de tempo em que o algoritmo manteve alguma cobertura.

81 sensores tanto o AH como a versão puramente *Online* conseguem manter a cobertura em cem por cento por todo o período de medição. Isso ocorre devido à alta densidade de nós sensores na rede. No entanto a diferença do comportamento entre os algoritmos pode ser notada ao se visualizar os gráficos de energia consumida e residual. Através destes gráficos é intuitivo que se fosse feita uma coleta de dados por um período maior, novamente a versão AH iria persistir durante um tempo maior que a *Online*, pois seu consumo de energia é bem menor, mantendo então mais energia disponível nos sensores da rede.

3.4.5 Resultado dos Testes Considerando Falhas apenas por Energia

Nesta segunda etapa dos testes, não foi considerada a falha mecânica, com isso os sensores falhos ocorrem apenas por falta de energia (bateria). Então, ao invés de fazer a medida em apenas 100 unidades de tempo, deixou-se que a rede funcionasse até o seu limite, sendo assim possível ver todo seu potencial de cobertura neste período.

Da mesma maneira, para cada rede serão mostrados três gráficos, ilustrando a energia residual, a energia consumida e a cobertura a cada período de tempo.

Os gráficos mostrando os resultados para 36, 49, 64, 81 e 100 sensores, podem ser visualizados nas Figuras 3.15, 3.16, 3.17, 3.18 e 3.19 respectivamente. Todos os gráficos mostrados aqui são médias em 33 execuções, devido às características aleatórias do algoritmo.

Além dos gráficos citados acima, nesta etapa dos testes mediu-se também o tempo de processamento gasto pelas versões *Online*, Híbrida e pelo pacote comercial CPLEX 10.2. Todos esses testes foram executados em um servidor de processos rodando o Sistema Operacional Linux Ubuntu Hardy Heron (8.04), com um processador Intel Core 2 Quad 2.5 GHz e com 4GB de memória RAM. Os tempos obtidos por cada versão podem ser vistos na Tabela 3.2 e também no gráfico da Figura 3.20.

3.4.5.1 Análise dos Resultados Da mesma maneira que nos testes mostrados na Seção 3.4.4, aqui serão comparados os resultados para as instâncias de 36, 49, 64 e 81 nós, e além destas, mais uma de 100 nós sensores. Todos estes resultados podem ser visualizados, respectivamente, nos gráficos das Figuras 3.15, 3.16, 3.17, 3.18 e 3.19.

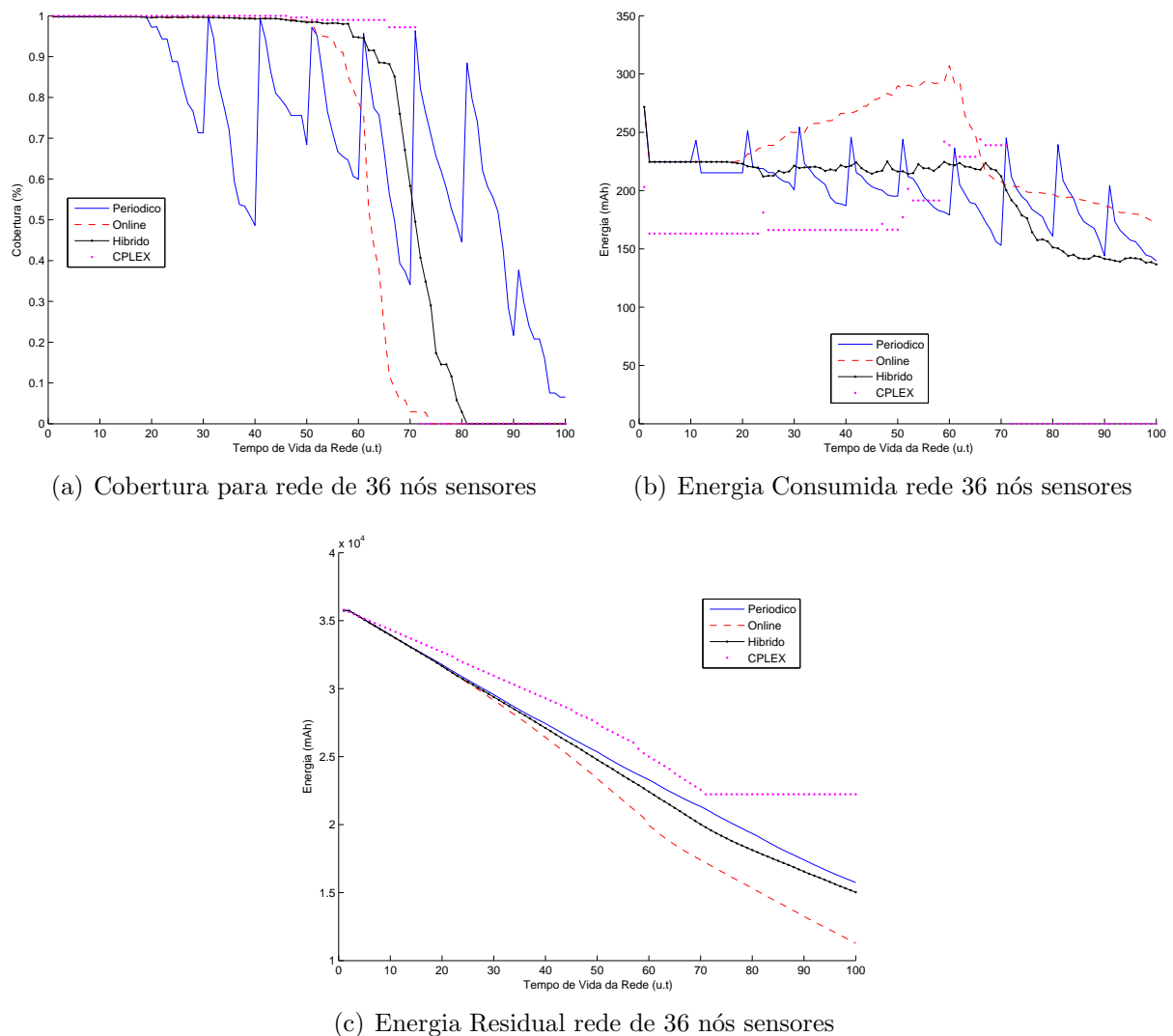
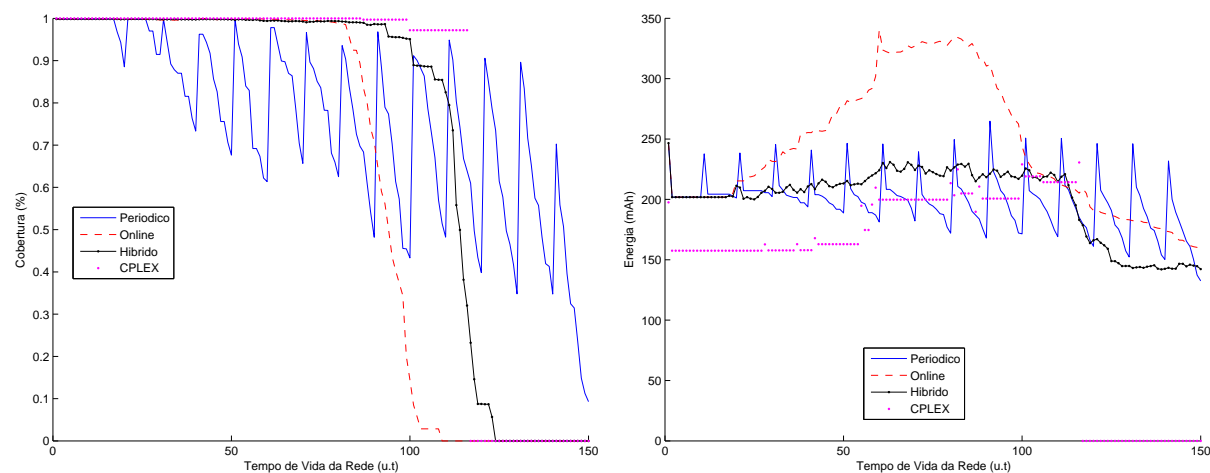


Figura 3.15 Gráficos de resultado para 36 sensores na abordagem mono-objetivo com falhas apenas por energia

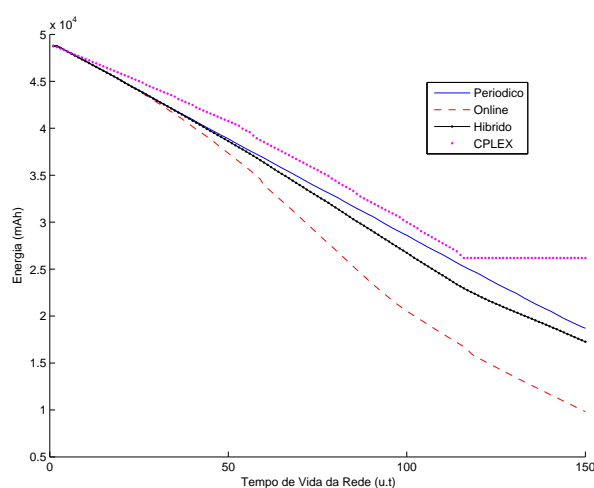
Percebe-se que da mesma maneira que nos testes anteriores, o comportamento de cada algoritmo permanece o mesmo, independente da quantidade de sensores. Para todos os casos que serão citados, novamente são mostrados os resultados exatos, dados pela modelagem matemática, e resolvidos pelo pacote comercial CPLEX. Da mesma maneira, estes resultados servem de limite inferior, no caso da medição de energia consumida, e de limite superior, para a cobertura e a energia residual.

As diferenças entre os testes realizados na Seção 3.4.4 e os apresentados aqui, são que agora apenas falhas por falta de energia, ou seja, bateria dos sensores, foram consideradas, não havendo mais falhas por fatores externos, como as falhas mecânicas citadas nos testes anteriores. Além disto, o período de monitoramento também foi modificado, não mais sendo apenas 100 unidades de tempo, e sim, a quantidade de unidades de tempo em que a rede conseguiu-se permanecer em estado de monitoração. Também foi computado



(a) Cobertura para rede de 49 nós sensores

(b) Energia Consumida rede 49 nós sensores

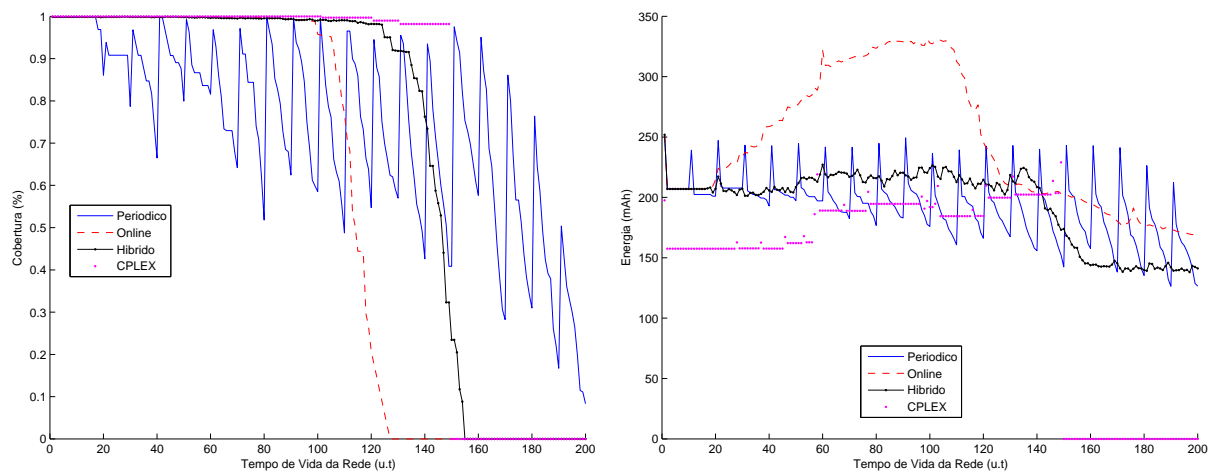


(c) Energia Residual rede de 49 nós sensores

Figura 3.16 Gráficos de resultado para 49 sensores na abordagem mono-objetivo com falhas apenas por energia

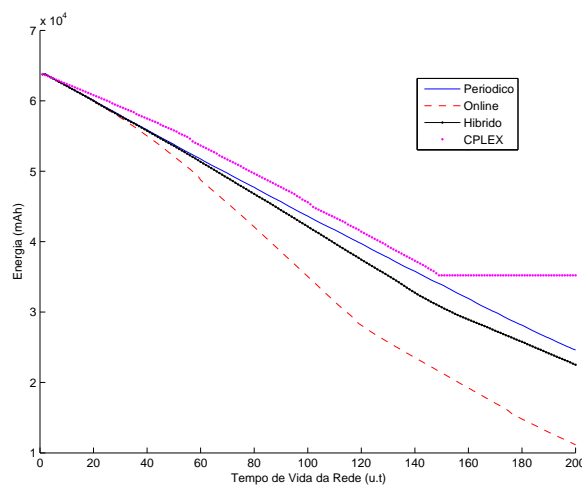
o tempo de processamento, para efeitos comparativos da eficiência entre cada aplicação dos algoritmos aqui apresentados.

Ao se comparar os resultados obtidos nesta fase dos testes com o resultados obtidos na Seção 3.4.4, percebe-se que para todos os casos, as diferenças entre cada implementação (*Periódica*, *Online*, *Online Híbrida* e *CPLEX*) permanecem as mesmas. A versão *Periódica* mantém seu comportamento oscilatório. A *Online* consegue resultados de cobertura próximos aos exatos, no entanto com alto consumo de energia, e conseqüentemente permanecendo menos tempo em monitoração. A *Online Híbrida* mantém sua superioridade em relação às anteriores e mantém seu consumo de energia mais equilibrado, fazendo com que atinja maiores tempos de monitoramento. Por fim tem-se os resultados obtidos pelo pacote comercial *CPLEX*, que novamente serviu como parâmetro para mostrar o quão próximo as soluções sugeridas estão da solução ótima do problema,



(a) Cobertura para rede de 64 nós sensores

(b) Energia Consumida rede 64 nós sensores



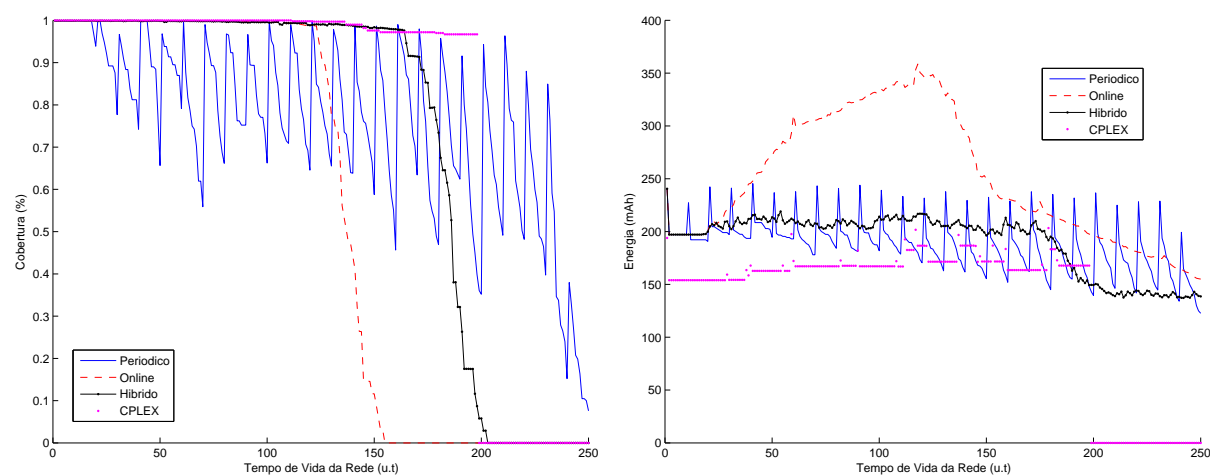
(c) Energia Residual rede de 64 nós sensores

Figura 3.17 Gráficos de resultado para 64 sensores na abordagem mono-objetivo com falhas apenas por energia

conforme a modelagem matemática introduzida no Capítulo 2.

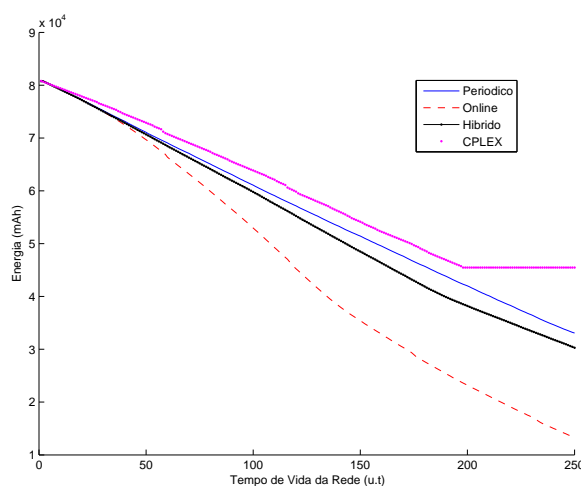
Uma pequena diferença entre os testes, que pode ser mais bem vista na instância de 36 sensores, vale a pena ser citada. Nas Figuras 3.11a e 3.15a, observando resultados obtidos na primeira e segunda bateria de testes respectivamente, percebe-se que para todos os algoritmos, a queda na cobertura ocorre em um tempo maior para o gráfico da Figura 3.15a. Isto se dá pelo fato de menos sensores estarem sofrendo falhas, então mais sensores ficam disponíveis para atender à demanda da rede.

Nesta nova bateria de testes, é possível fazer uma nova análise. Ao observar somente as soluções puramente *Online* e a versão Híbrida, percebe-se que quanto mais sensores são inseridos na rede, mais unidades de tempo essas versões conseguem atingir na cobertura. No entanto a distância da queda na cobertura entre elas tende a aumentar quanto mais densa fica a rede. Este fenômeno pode ser explicado novamente pelas diferenças entre



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores

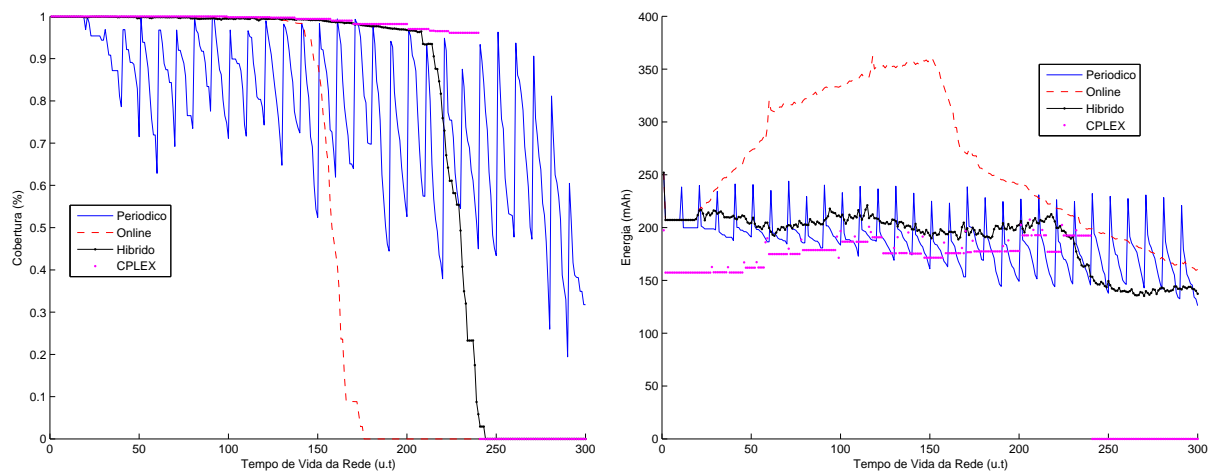


(c) Energia Residual rede de 81 nós sensores

Figura 3.18 Gráficos de resultado para 81 sensores na abordagem mono-objetivo com falhas apenas por energia

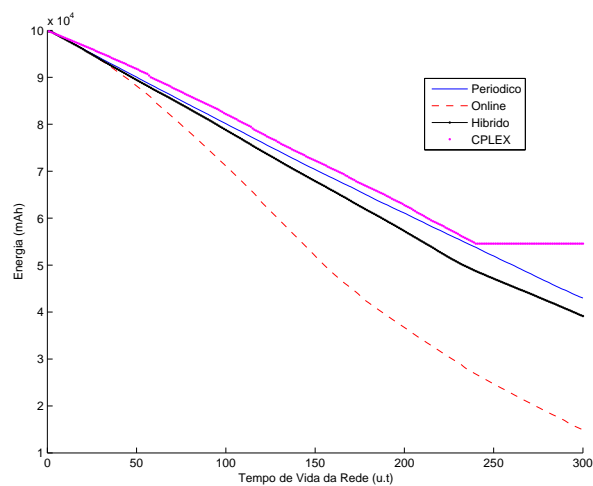
suas implementações. No *Online* não há um aproveitamento equilibrado da quantidade de sensores presente na rede, pois este não possui uma visão global do problema como no caso da versão Híbrida, e sim, somente uma visão local em que corrige as falhas de cobertura sem se preocupar com o nível de energia global consumido na rede.

Uma das grandes diferenças entre o método aqui proposto e a solução exata do modelo matemático, pelo pacote comercial CPLEX, está no esforço computacional. Esta diferença pode ser visualizada na Tabela 3.2 e também no gráfico da Figura 3.20. Ao olhar para estes resultados, pode-se inferir a grande eficiência da abordagem proposta, pois seu tempo de processamento, se mostra extremamente menor que o do CPLEX. Como exemplo, tem-se a instância de 100 sensores, em que a solução dada pelo CPLEX chega a ser mais de 5472 vezes mais lenta que a versão *Online* Híbrida. Ao analisar o gráfico da Figura 3.20, que se encontra em escala logarítmica, pode-se perceber que o crescimento do



(a) Cobertura para rede de 100 nós sensores

(b) Energia Consumida rede 100 nós sensores



(c) Energia Residual rede de 100 nós sensores

Figura 3.19 Gráficos de resultado para 100 sensores na abordagem mono-objetivo com falhas apenas por energia

esforço computacional para o CPLEX é bem próximo de um crescimento exponencial, o que permite concluir que para aplicações com grande quantidade de sensores, o uso deste método se torna inviável. No entanto, quando analisados os resultados para o *Online* Hibrido, observa-se que sua aplicação permaneceria viável para grandes quantidades de sensores.

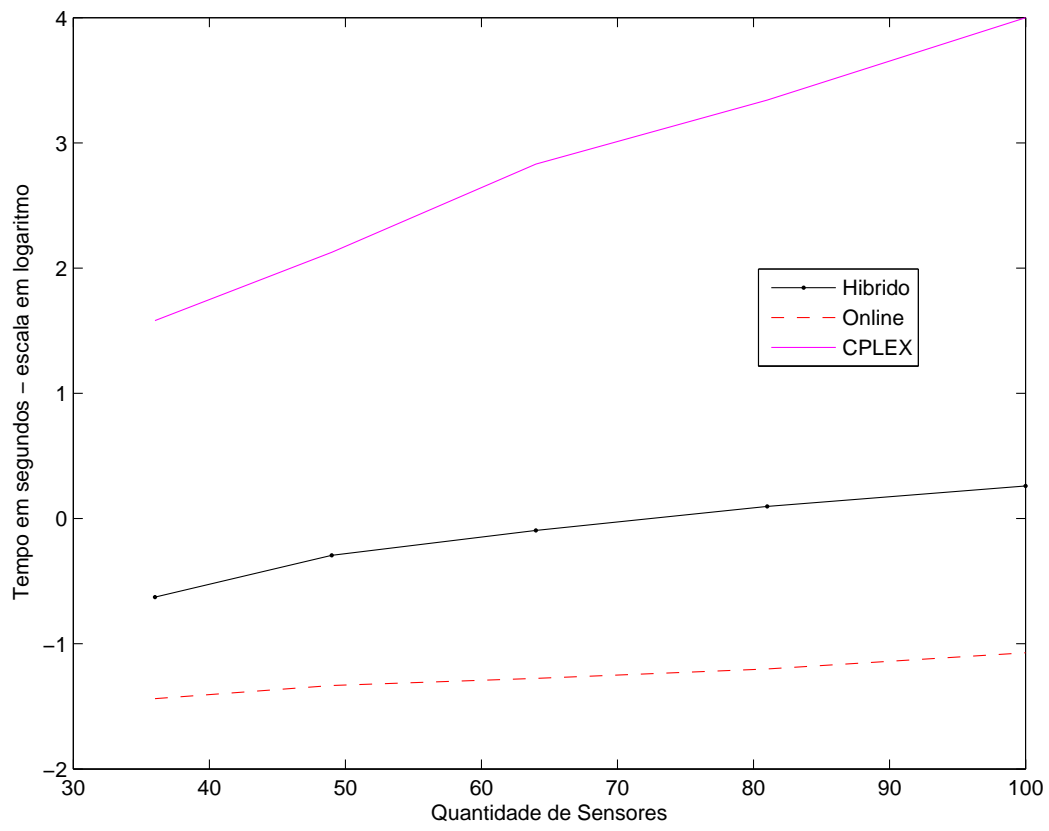


Figura 3.20 Gráfico mostrando o resumo da Tabela 3.2.

ABORDAGEM MULTIOBJETIVO

Um homem não pode fazer o certo numa área da vida, enquanto está ocupado em fazer o errado em outra. A vida é um todo indivisível.

—MAHATMA GANDHI

Tudo no universo esta em busca de equilíbrio.

—ALBERT EINSTEIN

Em problemas de projeto ótimo de sistemas, a abordagem multiobjetivo tem uma ampla vantagem ao ser comparada com a monoobjetivo tendo em vista principalmente o número de soluções. Nela temos na verdade não apenas uma solução ótima, mas sim um conjunto de soluções ótimas que são chamadas de soluções Pareto-ótimas.

Para entendermos melhor uma solução Pareto-ótima, precisamos definir quando a solução de um problema pertence a este conjunto, e também como esse conjunto é definido.

Sendo:

χ : Espaço dos parâmetros de otimização.

χ^* : Conjunto de *soluções eficientes*, ou *conjunto Pareto-ótimo* de um Problema de Otimização Vetorial (POV). O conjunto χ^* é um subconjunto do espaço de parâmetros χ .

Υ : Espaço dos “objetivos”, isto é, o espaço no qual se representa a imagem da função $f(\cdot)$.

x^* : Um ponto pertencente ao conjunto χ^* .

η_x : O conjunto dos pontos factíveis, subconjunto do conjunto χ .

Um POV pode ser formulado da seguinte maneira:

$$(POV) = \begin{cases} \chi^* = x^* \in \mathbb{R}^n \mid \\ x^* = \arg \min_x f(x) \\ \text{sujeito a : } x \in \eta_x \end{cases}$$

Sendo $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$ o vetor de objetivos do problema e $\eta_x \subset \mathbb{R}^n$ a região factível.

Segundo [Takahashi, 2004] dado um POV, ele se define a partir da análise do ordenamento das soluções levando em conta os diversos objetivos. As soluções multiobjetivo, ou soluções Pareto, são as melhores soluções dentre as quais não existe um ordenamento, ou seja, não há como definir, a partir da avaliação das funções objetivo, que uma solução é melhor que a outra. Então, o objeto da otimização multiobjetivo consiste em determinar um conjunto de soluções χ^* , denominado conjunto Pareto-ótimo, que irá conter as possíveis soluções x^* do POV.

[Takahashi, 2004] define dominância como sendo:

Definição 1. Dominância: Diz-se que o ponto $x_1 \in \chi$ domina o ponto $x_2 \in \chi$ se $f(x_1) \leq f(x_2)$ e $f(x_1) \neq f(x_2)$. Equivalentemente, diz-se que $f(x_1) \in \Upsilon$ domina $f(x_2) \in \Upsilon$, nessas mesmas condições.

Definição 2. Solução Pareto-ótima: Diz-se que $x^* \in \eta_x$ é uma solução Pareto-ótima de um POV se não existe qualquer outra solução $x \in \eta_x$ tal que $f(x) \leq f(x^*)$ e $f(x) \neq f(x^*)$, ou seja, se x^* não é dominado por nenhum outro ponto factível.

Então, um ponto pertencente ao conjunto Pareto-ótimo têm as seguintes propriedades:

- É melhor (ou pelo menos equivalente) que qualquer outro ponto em pelo menos um objetivo.
- Não é dominado por nenhum outro ponto embora não precise dominar todos os outros pontos.

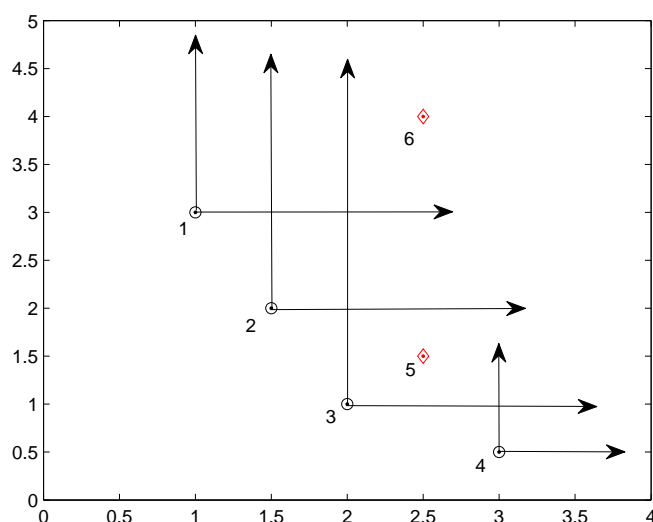


Figura 4.1 Exemplo de um conjunto Pareto-ótimo e de dominância

De posse de um conjunto Pareto-ótimo, muitas vezes é necessário escolher apenas uma solução. Para isso, é aplicado algum critério de escolha. Em determinadas vezes esta escolha pode ser feita por um decisor externo, ou também ser realizada de forma determinística pré-determinada.

Na Figura 4.1 pode-se ver um exemplo em que os pontos 1, 2, 3 e 4, fazem parte do conjunto Pareto-ótimo da solução apresentada. Têm-se também nesta figura que as soluções 1 e 2, dominam a solução 6, e a solução 3 domina a 5 e a 6.

Dados estes conceitos em relação a um POV. Nas próximas seções serão aplicados esses conceitos e também serão discutidas algumas abordagens multiobjetivo em Algoritmos Genéticos.

4.1 ALGORITMOS GENÉTICOS MULTIOBJETIVO

Como já introduzido na Seção 3.1, sabe-se que os Algoritmos Genéticos (AG's) são amplamente utilizados para resolver problemas diversos. Considerado um Algoritmo Evolutivo, este modelo de algoritmo faz uso de um conjunto de soluções candidatas conhecida como “população” de indivíduos, nela são aplicadas várias operações, fazendo com que este conjunto evolua a cada geração, buscando assim uma solução ótima, ou aproximada do problema. Para maiores detalhes ver a Seção 3.1.

Sabe-se que os AG's podem ser definidos como a aplicação sucessiva na “população” dos seguintes operadores: cruzamento, mutação, seleção e elitismo; como já mencionados em 3.1. Após algumas gerações, ou seja, aplicações destas operações, a “população” tende a convergir para soluções que de alguma forma são próximas das soluções globais do problema.

Segundo [Takahashi, 2004], qualquer AG mono-objetivo pode ser adaptado para obtenção de um AG multiobjetivo. Tais adaptações necessárias são:

- Selecionar dentro de uma população inicial \mathcal{Q}_0 , o grupo de indivíduos que forma o Pareto-ótimo, a esse é dado o nome de $\bar{\mathcal{P}}_0$.
- A cada iteração, uma nova população \mathcal{Q}_i é gerada pela aplicação dos operadores genéticos. Recalcular a estimativa do conjunto Pareto-ótimo $\bar{\mathcal{P}}_i$, eventualmente excluindo alguns indivíduos e incluindo outros. O conjunto $\bar{\mathcal{P}}_i$ é empregado como o “conjunto-elite” na operação de elitismo.
- O funcional que guia o operador de seleção deve ser composto com os funcionais individuais que compõem o vetor de objetivos.

Então, desta maneira, ao invés de procurar por apenas uma solução, como era realizado no caso mono-objetivo, o AG multiobjetivo pode encontrar um conjunto de soluções Pareto-ótimas pertencentes a $\bar{\mathcal{P}}_i$. No caso mostrado, o elitismo na solução é garantido deterministicamente, pois: $\mathcal{P}_{i+1}^- \leq \bar{\mathcal{P}}_i$ e $\mathcal{P}_{i+1}^- \supseteq \bar{\mathcal{P}}_i$.

Definido basicamente como é formado um Algoritmo Genético Multiobjetivo (AGM), em seguida, serão mostradas duas formas amplamente conhecidas na literatura que tratam de implementações de AGM's. São elas: NSGA e NSGA II, introduzidas respectivamente em [Srinivas & Deb, 1994] e [Deb et al., 2002].

4.1.1 NSGA

Inicialmente em 1967, Rosenberg sugeriu, porém não implementou, uma busca genética multiobjetivo [Rosenberg, 1967]. Uma versão de algoritmo implementado somente foi surgir em 1984 desenvolvido por Schaffer [Schaffer, 1984], e dado o nome de VEGA (*Vector Evaluated Genetic Algorithm*). Alguns anos depois, Goldberg sugeriu um método de ordenação por não dominância [Goldberg, 1989], tentando superar as deficiências apresentadas pelo VEGA. No entanto, o método sugerido por Goldberg foi ser implementado pela primeira vez por Srinivas e Deb [Srinivas & Deb, 1994], recebendo o nome de NSGA (*Nondominated Sorting Genetic Algorithm*).

A idéia por trás do NSGA é o método de seleção por *ranking*, usado para enfatizar os melhores pontos (indivíduos) e o método de nicho, o qual é usado para manter a estabilidade dos melhores indivíduos.

O NSGA é derivado de uma simples implementação de um AG, sua maior diferença se encontra em sua forma de seleção. Em cada população corrente são realizados os testes de dominância, como descrito inicialmente neste capítulo, separando assim o Pareto-ótimo na iteração atual. Para cada indivíduo deste Pareto é atribuído um valor alto de *fitness* (função de aptidão). Então, desconsiderando o conjunto para o qual já foi atribuído valor de *fitness*, é identificado um novo conjunto de pontos não-dominados em relação ao restante da população. A estes novos indivíduos se atribui então outro valor de *fitness*, menor que o anterior. Desta maneira vai-se “descascando” os conjuntos de pontos que são dominados, de acordo com um “grau de dominância” no espaço de objetivos. Resumidamente, um pequeno exemplo de como ficariam as “cascas” pode ser visualizado na Figura 4.2.

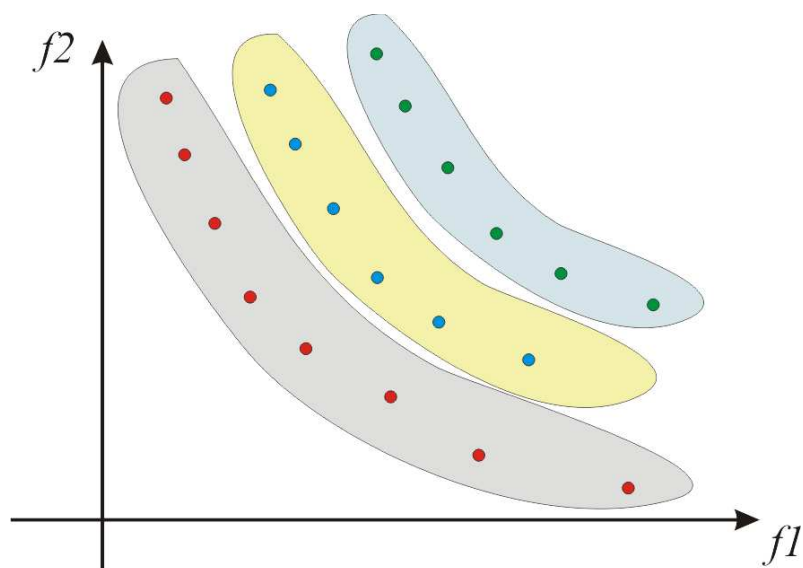


Figura 4.2 Cascas da população, por ordenação de dominância, para seleção por roleta

Então, tem-se na primeira fronteira um maior valor de *fitness*, forçando então na geração seguinte surgirem mais cópias destes indivíduos. Atribuídos os valores de *fitness* para cada indivíduo, um método de seleção estocástico é realizado.

4.1.2 NSGA II

Depois de alguns anos, surgiu na literatura uma nova proposta para o NSGA, conhecida como NSGA-II [Deb et al., 2002]. As novidades em relação a versão NSGA se encontram na melhora da operação em que se separam as soluções dominantes, diminuindo de forma considerável o esforço computacional. Também é embutida na estrutura do algoritmo, através de um arquivo de soluções, uma maneira de garantir o elitismo de forma a também diminuir o tempo de processamento. Por fim, uma nova técnica de nicho é empregada

usando *Crowding-distance*, cálculo usado para diferenciar as soluções presente em um mesmo nível de dominância, priorizando as soluções mais isoladas. Outro detalhe é a não necessidade de uma função *fitness* para diferenciar as soluções, usando uma seleção por ordem de Pareto, além da seleção por torneio usada nas iterações do AG.

4.2 ALGORITMO ONLINE HÍBRIDO MULTIOBJETIVO

Da mesma maneira que na Seção 3.3 do Capítulo 3, que tratava do Algoritmo *Online* Híbrido Mono-Objetivo, esta seção tomará como base a apresentação do problema de rede de sensores sem fio introduzido no Capítulo 2. Da mesma forma abordada no caso mono-objetivo (Seção 3.3) a versão multiobjetivo usará conceitos de Algoritmos Genéticos (AG's) e Caminho Mínimo (PCM).

A situação em que é empregado o algoritmo aqui sugerido é a mesma da apresentada na Seção 3.3, todo algoritmo responsável pelo controle da rede, no qual se tem um custo computacional relevante, é executado no nó sorvedouro, o qual possui energia ilimitada. Como novamente a versão aqui apresentada é uma proposta de controle da rede, este é implementado para ser executado também no nó sorvedouro.

Novamente de posse da descrição do problema (Seção 2.1) e da modelagem matemática (Seção 2.2), foi implementado uma nova heurística para a solução do problema das RSSF. Pois se sabe, e também já foi mostrado nos resultados computacionais apresentados para o caso mono-objetivo (Seção 3.4), que a busca pela solução exata através da modelagem matemática, se torna inviável quando a quantidade de nós sensores presentes na rede aumentam. Como as RSSF são frequentemente usadas em grandes aplicações como: monitorar florestas e grandes centros urbanos, grandes quantidades de sensores são indispensáveis.

Esta nova heurística toma como base o algoritmo apresentado na Seção 3.3 do Capítulo 3. A alteração é realizada apenas na maneira de como é tratada a cobertura da rede. Como já citado, todo AG mono-objetivo pode ser adaptado para se tornar um AG multiobjetivo, no entanto, parte dos seus operadores permanecem inalterados.

4.2.1 Algoritmo Genético Multiobjetivo

De posse das definições mencionadas no início deste capítulo (Seção 4), e das formas das soluções já propostas na literatura para AG's multiobjetivos (Seção 4.1), foi implementado um Algoritmo Genético Multiobjetivo (AG_MO) para tratar o problema de cobertura das RSSF, da mesma forma que no caso mono-objetivo. No entanto, agora com a finalidade de obter uma curva Pareto-ótima de acordo com os seguintes objetivos:

- Energia da Rede, como anteriormente (eq. 3.5) no entanto não considerando a penalidade por não cobertura.

$$- FO(H1) = \sum_{i \in S'} (EA_i + PM_i + CM_i) \times y_i$$

- Cobertura

$$- FO(H2) = pC \cdot nC + pAS \cdot \sum_{i \in S} y_i$$

– Sendo:

- pC penalidade de “não cobertura”;
- nC número de pontos de demana não cobertos;
- pAS penalidade de não ativação.

O AG_MO implementado segue as linhas do NSGA, com apenas algumas alterações. A justificativa de não ter sido usado o padrão mais atual, o NSGA-II, se dá pelo fato dos resultados obtidos com esta versão serem menos eficientes no caso específico do problema aqui resolvido. Isto ocorre basicamente devido a alta pressão seletiva presente no NSGA-II, ou seja, as soluções tendem a convergir para um determinado ponto de uma maneira mais rápida, o que faz estas em alguns casos convergirem para soluções que não são as ótimas, ou próximas destas.

Maiores detalhes sobre as diferenças encontradas na aplicação do NSGA e do NSGA-II, no problema da RSSF aqui apresentado, e também os resultados que firmam tal justificativa, podem ser visualizados no Apêndice A.

Para um melhor entendimento da implementação do AG_MO, é mostrado um pseudo-código no Algoritmo 6.

Sabe-se que a versão do AG_MO mostrada no Algoritmo 6, é uma versão estendida da introduzida na Seção 3.3.1. Para se entender as diferenças implementadas neste novo algoritmo, serão definidas as operações que permaneceram inalteradas, as que sofreram alterações e as que foram inseridas.

As operações inalteradas de uma implementação para outra, foram:

- Geração da População Inicial. (Seção 3.3.1.2)
- Cruzamento. (Seção 3.3.1.3)
- Mutação. (Seção 3.3.1.4)

As que sofreram alterações, foram:

- Seleção.
- Elitismo.

Já as novas operações inseridas, foram:

- Teste de Dominância.
- Decisão do nível de dominância.

As operações alteradas e as inseridas, serão discutidas na subseções seguintes.

Algoritmo 6 AG_MO

```

Pop = gerar população inicial. /* Seção: 3.3.1.2 */
Poparq = []
 $\phi \equiv$  porcentagem mínima desejada para cobertura da RSSF.
para todo cromossomos  $\in$  Pop faça
    Pop.AvalieFO(H1)
    Pop.AvalieFO(H2)
fim para
Faça o teste de dominância segundo os objetivos.
Determine os níveis de dominância.
Ordene a população de acordo com o nível de dominância.
Poparq = elitismo(Poparq, P1) /* P1 ← conjunto Pareto-ótimo na iteração.*/
enquanto (Gerações <  $\beta$ ) faça
    Pop.crossing over (probabilidade  $\alpha$ ) /* Cruzamento */
    Pop.Mutaçao (probabilidade  $\pi$ ).
    para todo cromossomos  $\in$  Pop faça
        Pop.AvalieFO(H1)
        Pop.AvalieFO(H2)
    fim para
Faça o teste de dominância segundo os objetivos.
Determine os níveis de dominância.
Ordene a população de acordo com o nível de dominância pertencente.
Poparq = elitismo(Poparq, P1) /* P1 representa o conjunto Pareto-ótimo na iteração.*/
    Pop = Pop.Seleção /* Método da Roleta.*/
fim enquanto
cromo = Poparq.decisor( $\phi$ ) /*Aplica um decisor na população arquivo*/
Retorno cromo

```

4.2.1.1 Seleção no AG_MO A forma de seleção adotada para o AG foi através de roleta. Esta forma de seleção consiste em criar uma roleta cujos espaços são divididos entre os indivíduos presentes na população corrente. O espaço determinado para cada indivíduo é dado por uma função de aptidão, conhecida também como *fitness*. Por exemplo, na Tabela 4.1 mostra exemplos de valores de aptidão para uma população de 8 indivíduos. Então é montada uma roleta com os espaços divididos proporcionalmente de acordo com esses valores, por exemplo, Figura 4.3. É “girada” a roleta, o indivíduo em que esta parar será selecionado para a próxima geração. Desta maneira pode-se selecionar a quantidade de indivíduos desejados. Vale a pena observar, que irão surgir cópias de indivíduos na próxima geração, e que esta chance é maior para os de maiores valores de *fitness*, pois estes têm um espaço mais representativo na roleta.

O Algoritmo 7 é um pseudo-código da implementação da roleta usada no AG_MO. Para calcular a função *cromossomo.aptidão*(*i*) (através da Equação 4.1), é necessário fazer o “descascamento” dos níveis de dominância, como é feito no NSGA. Para isso

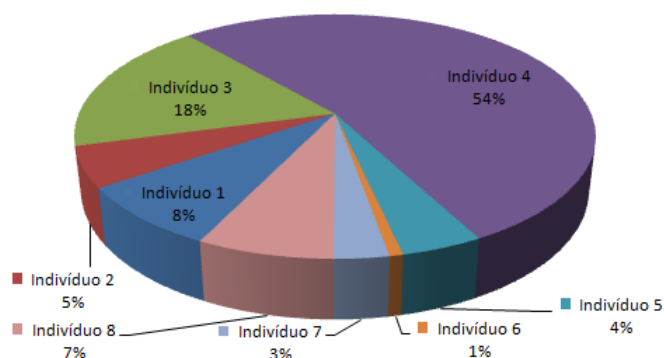


Figura 4.3 Roleta formada segundo os valores de aptidão do indivíduos da Tabela: 4.1

as soluções são ordenadas de acordo com a camada de dominância a que pertencem. Sendo $numPareto$ a variável que representa o nível de dominância em que a solução está presente, ter-se-ia $numPareto = 1$ para o conjunto Pareto-ótimo. Para os demais níveis de dominância, $numPareto$ é igual ao inteiro subsequente segundo a ordenação feita anteriormente, ou seja, para o segundo nível de dominância $numPareto = 2$ e assim por diante.

Algoritmo 7 Algoritmo de seleção por roleta

$tamPop \equiv$ tamanho da população

$somaAptidão = 0$

para $i = 1$ a $tamPop$ **faça**

$somaAptidão = somaAptidão + cromossomo.aptidão(i)$ /* segundo a Equação: 4.1 */

fim para

para $i = 1$ a $tamPop$ **faça**

$rand \equiv$ um número aleatório entre 1 e $somaAptidão$.

$S = 0$

para $i = 1$ a $tamPop$ **faça**

$S = S + cromossomo.aptidão(i)$

se $S \geq rand$ **então**

Selecione o cromossomo i

fim se

fim para

fim para

Atribuído os valores de $numPareto$ para cada solução, são então calculados os valores de aptidão segundo a Equação 4.1

$$fitnessMO = \frac{1}{numPareto} \left(\frac{1}{2^{0.8numPareto}} \right) \quad (4.1)$$

Onde:

- $numPareto$ é um número inteiro maior que 0 que representa a posição no nível de dominância.

No gráfico da Figura 4.4, pode-se observar que quanto menor o valor de $numPareto$, maior é o valor para $fitnessMO$. Também é possível notar que não há uma discrepância muito grande entre os valores atribuídos para cada camada. Esta menor “pressão” para indivíduos melhores ajuda a não ocorrer uma convergência prematura.

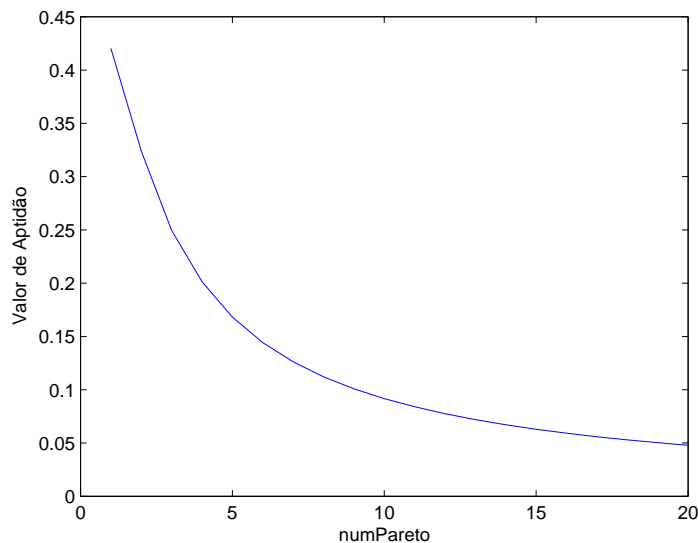


Figura 4.4 Gráfico da função de aptidão. Equação: 4.1

4.2.1.2 Elitismo no AG_MO Para garantir que nenhuma solução Pareto-ótima se perca durante a rotina de seleção, foi realizada uma operação de elitismo, conforme mostrado no Algoritmo 6, na operação $Pop_{arq} = elitismo(Pop_{arq}, P_1)$. Esta operação mantém uma “população arquivo”, este arquivo é formado da seguinte forma: A cada iteração do AG_MO é realizado um Teste de Dominância (segundo a Definição 1) entre as soluções já presentes em Pop_{arq} e as pertencentes em P_1 (soluções presentes no Pareto-ótimo da iteração atual). Então as soluções que não são dominadas por nenhuma outra solução, passam a ser a “população arquivo”, representada pela variável Pop_{arq} .

4.2.1.3 Decisão no conjunto Pareto-ótimo De acordo com o Algoritmo 6, após β gerações do AG_MO, tem-se um conjunto Pareto-ótimo segundo a Definição 2. Como mostrado no início deste capítulo, sabe-se que nas soluções presentes no conjunto Pareto-ótimo não se tem uma solução considerada “melhor” que outra, tendo em vista ambos os objetivos. Em grande parte dos problemas é necessário fazer então uma escolha dentre as soluções presentes neste nível de dominância.

A operação $cromo = Pop_{arq}.decisor(\phi)$ mostrada no Algoritmo 6, representa a operação na qual é realizada a escolha de uma solução. Esta operação tem como parâmetro

de entrada ϕ , no qual é fornecido por um decisor externo. ϕ representa a porcentagem mínima de cobertura da RSSF desejada por este decisor. A partir desta escolha, a implementação do critério de decisão passa a seguir os seguintes passos:

1. Ordene todas as soluções presentes no conjunto Pareto-ótimo segundo o objetivo $FO(H1)$. Automaticamente por ter apenas dois objetivos, ao fazer esta ordenação, o objetivo $FO(H2)$ fica ordenado de forma inversa.
2. A partir da melhor solução dada por $FO(H1)$, percorra os indivíduos ordenados verificando sua perda de cobertura.
3. Selecione o primeiro indivíduo no qual a perda de cobertura for menor ou igual a $100 - \phi$.
4. Retorne o indivíduo selecionado.

Na Figura 4.5 pode-se visualizar dois casos distintos de decisão. No primeiro, a necessidade mínima de cobertura é de 70% (Figura 4.5a) e no segundo caso 95% (Figura 4.5b).

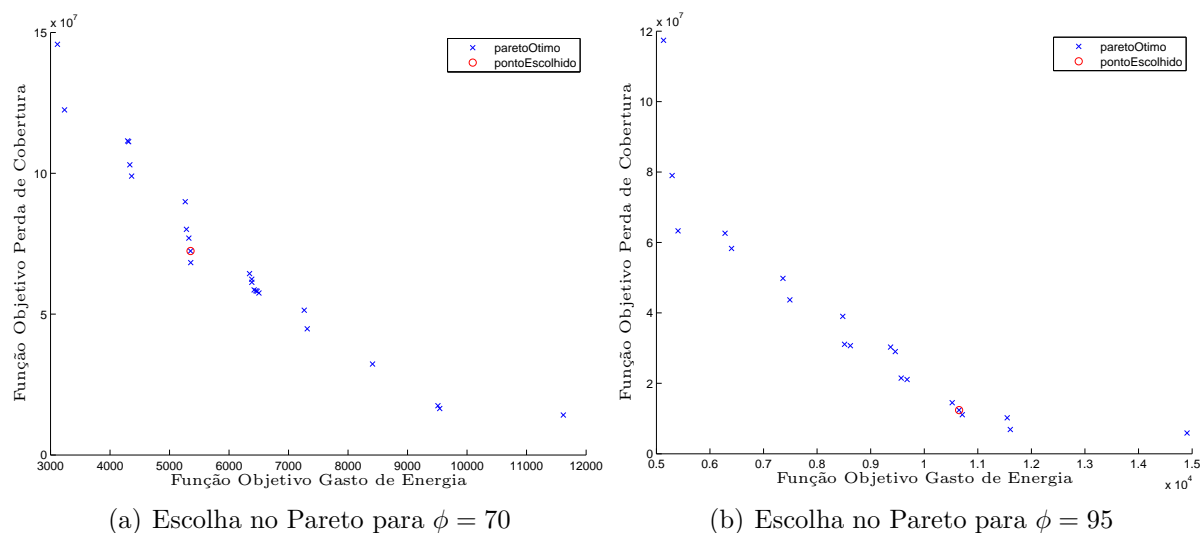


Figura 4.5 Curva Pareto-ótima fornecida na primeira unidade de tempo. Exemplo de escolha de uma solução para o caso de 100 sensores.

Depois de descritas as etapas do AG_MO, é possível formar o Algoritmo Global Sob-Demanda Multiobjetivo (AGSD_MO), que pode ser visualizado no Algoritmo: 8.

De forma semelhante à mostrada na Seção 3.3, a versão híbrida para a solução do problema da RSSF aqui apresentado, segue os mesmos passos do Algoritmo 5, alterando a rotina *GlobalSobDemanda* pela *GlobalSobDemanda_MO*. Esta nova versão híbrida para o caso multiobjetivo pode ser visualizada no Algoritmo 9.

Algoritmo 8 GlobalSobDemanda_MO(**entrada:** S, m, D, RS (raio de sensoriamento), RC (raio de comunicação))

Realize configurações gerais do ambiente
 Compute matrizes de cobertura e conectividade
 /*AG*/
 AG_MO /*Algoritmo: 6*/
 /*end AG*/
para todo ($i \in S$) **faça**
 $PC_i = Dijkstra(i, sink)$
fim para

Algoritmo 9 Híbrido_MO(**entrada:** S, m, D, RS (raio de sensoriamento), RC (raio de comunicação))

*GlobalSobDemanda_MO(S,m,D,SR,CR); /*gera solução inicial*/*
enquanto ($S \neq \emptyset$) **faça**
 Falha = Verificar Falha de Nós;
 se (Falha = VERDADEIRO) **então**
 $LocalOnline(S,f,C,m,D,SR,CR); /* Restaurar cobertura e conectividade */$
 $S = S - f; /*exclui nó f do conjunto S*/$
 se (Condição para Executar AGSD = Verdadeiro) **então**
 $GlobalSobDemanda_MO(S,m,D,RS,RC);$
 fim se
 fim se
fim enquanto

4.3 RESULTADOS COMPUTACIONAIS

De maneira semelhante ao no Capítulo 3, Seção 3.4, serão realizadas duas baterias de testes. A primeira considerando falhas por energia e falhas mecânicas, monitorando 100 unidades de tempo (u.t) e na segunda bateria de testes serão consideradas novamente apenas falhas por energia, monitorando a rede até a unidade de tempo que ela conseguir permanecer cobrindo algum ponto de demanda, permanecendo também conexa.

A forma de gerar as falhas mecânicas, as instâncias de nós sensores utilizados, o tipo de sensor e a forma de coletar os dados da rede, são exatamente os mesmos utilizados nos testes apresentados na Seção 3.4.

Os parâmetros utilizados para o AG_MO foram:

- Taxa de cruzamento (α) = 0.9
- Taxa de mutação (π) = 0.2
- População de indivíduos = 300
- Número de Gerações (β) = 150

Os Resultados aqui apresentados serão comparados novamente com a solução exata dada pelo pacote comercial CPLEX e também com os resultados apresentados para o caso mono-objetivo apresentado no Capítulo 3.

4.3.1 Resultado dos Testes Considerando Falhas Mecânicas e por Energia

Novamente como realizado no capítulo anterior, nesta primeira bateria de testes serão mostrados resultados obtidos para as redes de sensores formadas por 36, 49, 64 e 81 sensores que foram dispostos em uma área de $2500m^2$ como descrito na Figura 3.9. Como já informado, cada sensor tem sua bateria limitada, além disso, nos testes aqui apresentados, cada um terá uma probabilidade de falha mecânica como descrito anteriormente, modelado pela exponencial mostrada na Figura 3.8.

Para cada rede serão mostrados três gráficos, demonstrando assim a energia residual, consumida e a cobertura a cada período de tempo. Neste primeiro teste serão mostrados os resultados em 100 unidades de tempo da rede.

Os gráficos mostrando os resultados para 36, 49, 64 e 81 sensores, podem ser visualizados nas Figuras 4.6, 4.7, 4.8 e 4.9 respectivamente. Todos os gráficos mostrados aqui são de médias em 33 execuções, devido às características aleatórias do algoritmo. Para todos os casos a aplicação do decisor na curva Pareto-ótima foi considerado $\phi = 95$, ou seja, foi exigida uma cobertura mínima de 95% na RSSF.

4.3.1.1 Análise dos Resultados Ao observar os gráficos das Figuras 4.6, 4.7, 4.8 e 4.9, percebe-se que o comportamento de cada algoritmo permanece o mesmo, independente da quantidade de sensores. Para todos os casos citados, são mostrados os resultados exatos dados pela abordagem por programação linear inteira, e resolvidos pelo pacote comercial CPLEX. Ao serem comparados com a versão mono-objetivo, estes resultados serviram de limite inferior, no caso da medição de energia consumida, e de limite superior, para a cobertura e a energia residual. No entanto, nos resultados para abordagem multiobjetivo, introduzida neste capítulo, ao comparar com a energia residual e consumida, os resultados dados pelo CPLEX servem como parâmetro para determinar a qualidade da abordagem adotada. Esta mudança se dá pelo fato de ao relaxar a cobertura, menos nós sensores poderem ser ativados, consumindo assim menos energia na rede. Devido a isto, em alguns resultados apresentados nos gráficos que mostram os consumos de energia, pode ocorrer de a abordagem multiobjetivo apresentar resultados superiores à abordagem por programação linear inteira.

Para todos os casos aqui apresentados, o único no qual a abordagem multiobjetivo não manteve monitorando durante as 100 u.t. em seu limite mínimo de 95% para cobertura ($\phi = 95$), foi para a instância de 36 nós sensores. Neste caso como há uma baixa densidade de sensores, e estes falhando, poucas possibilidades estão disponíveis para se tentar recuperar a cobertura. Isto é visível, pois até para o caso exato dado pelo CPLEX, o objetivo de se manter as 100 u.t. monitorando, não é atingido, mantendo-se apenas até a 64^o u.t.. No entanto, observando-se o resultado apresentado para o caso multiobjetivo, este começou a ter queda na cobertura a partir da 52^o u.t., praticamente junto com a versão mono-objetivo. Quando observado o gráfico de energia consumida

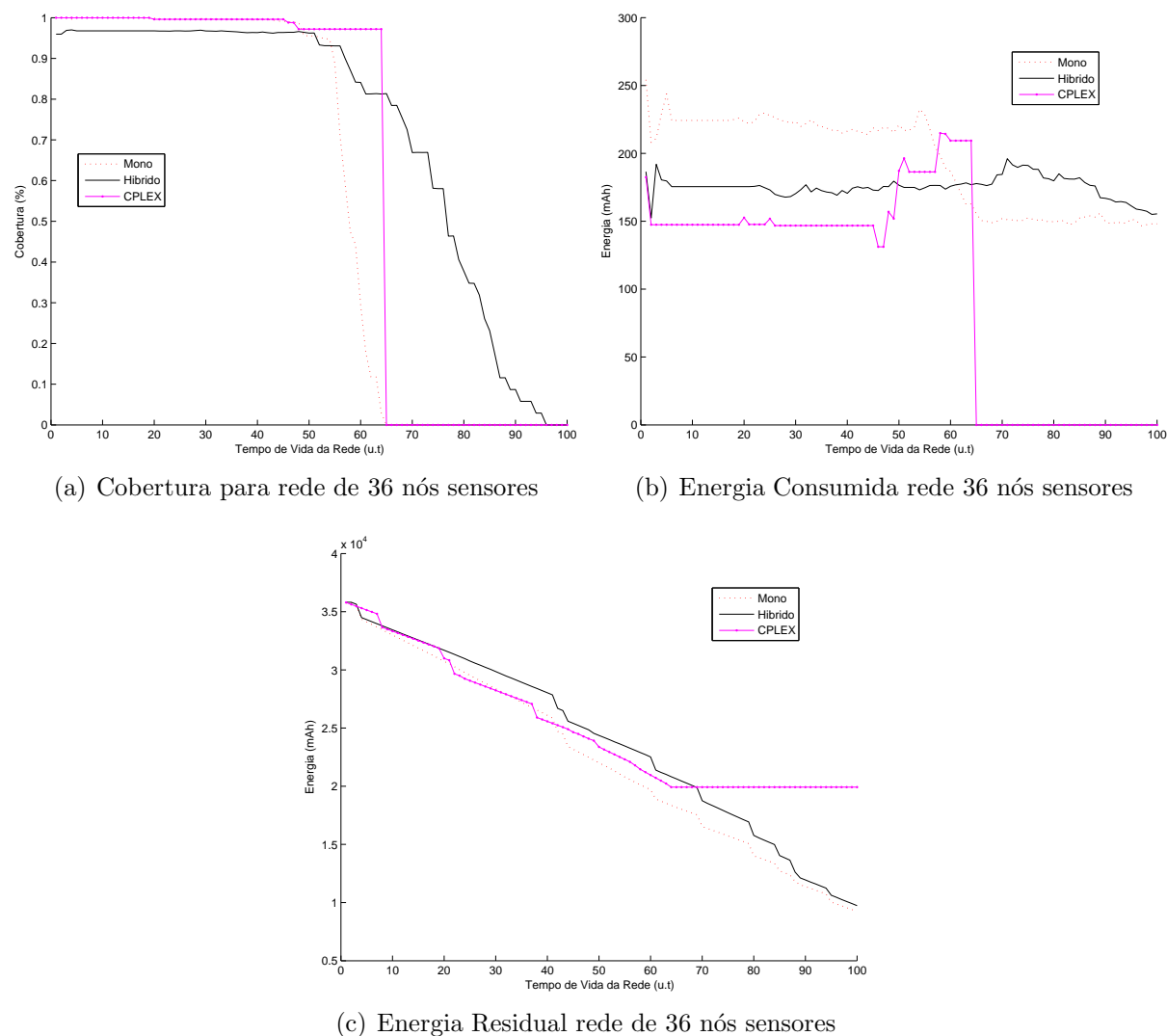
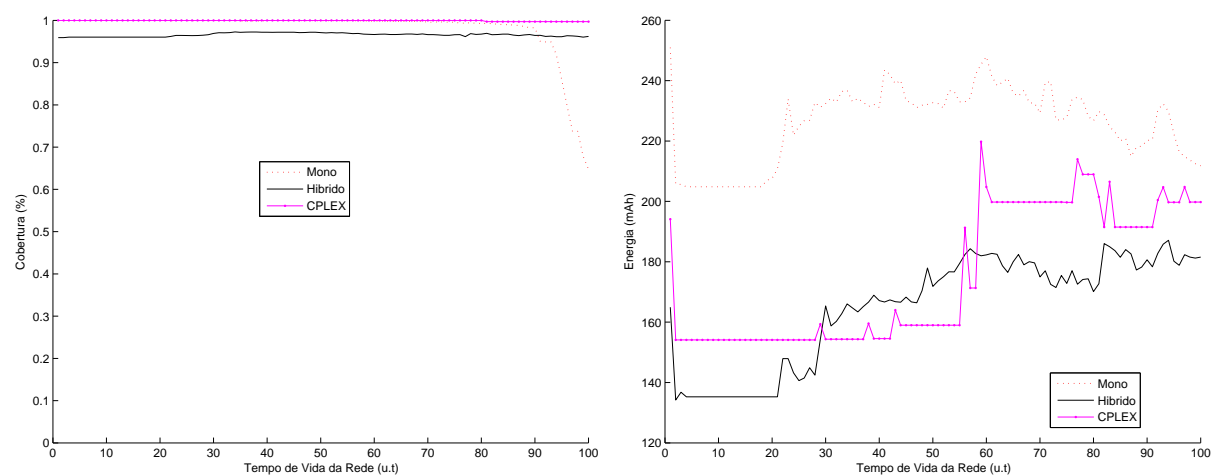


Figura 4.6 Gráficos de resultado para 36 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$.

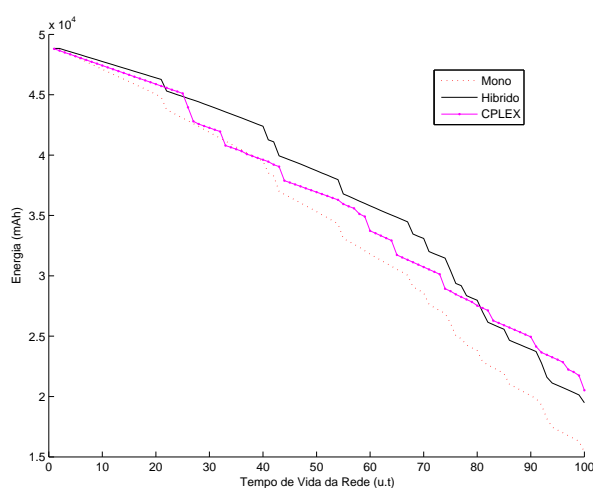
(Figura 4.6b), percebe-se que o multiobjetivo consome menos energia que a versão mono-objetivo. Devido a este menor consumo, pode-se visualizar no gráfico de cobertura que a queda da curva do multiobjetivo ocorre de maneira menos acentuada, fazendo que esta abordagem consiga manter em média, algum ponto de demanda monitorado até a 95^o u.t., superando em muito a versão mono-objetivo, a qual conseguiu chegar apenas à 64^o u.t..

Ao se observar os resultados apresentados para a instância de 49 sensores, pode se perceber que a abordagem multiobjetivo consegue manter seu objetivo de cobertura mínima de 95% por todo o período de tempo. Já a versão mono-objetivo uma queda acentuada na cobertura começa na unidade de tempo 90. Este fato é facilmente explicado quando se observam os gráficos das Figuras 4.7b e 4.7c que representam respectivamente a energia consumida e residual da rede. Percebe-se que o algoritmo multiobjetivo mantém o



(a) Cobertura para rede de 49 nós sensores

(b) Energia Consumida rede 49 nós sensores



(c) Energia Residual rede de 49 nós sensores

Figura 4.7 Gráficos de resultado para 49 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$.

consumo de energia muitas vezes até menor que o da abordagem por programação linear inteira resolvido pelo CPLEX, ao contrário do mono-objetivo, no qual há um consumo de energia bem superior às outras duas abordagens apresentadas. Conseqüentemente ao se observar a energia residual, a versão multiobjetivo preserva mais energia na rede em boa parte do tempo. Como a abordagem mono-objetivo consome mais energia, logicamente não se consegue manter monitorando por um período de tempo igual ao das outras abordagens.

Nas instâncias para 64 e 81 sensores os resultados seguem os mesmos padrões. Quanto maior a densidade de nós sensores na área a ser monitorada, mais tempo consegue-se monitorá-la. No entanto, a partir destas instâncias a diferença na cobertura pode ser notada apenas para o caso de 64 sensores, pois somente para esta houve um princípio de queda na cobertura para o caso mono-objetivo (Figura 4.8a). Com isto, faz-se pensar em

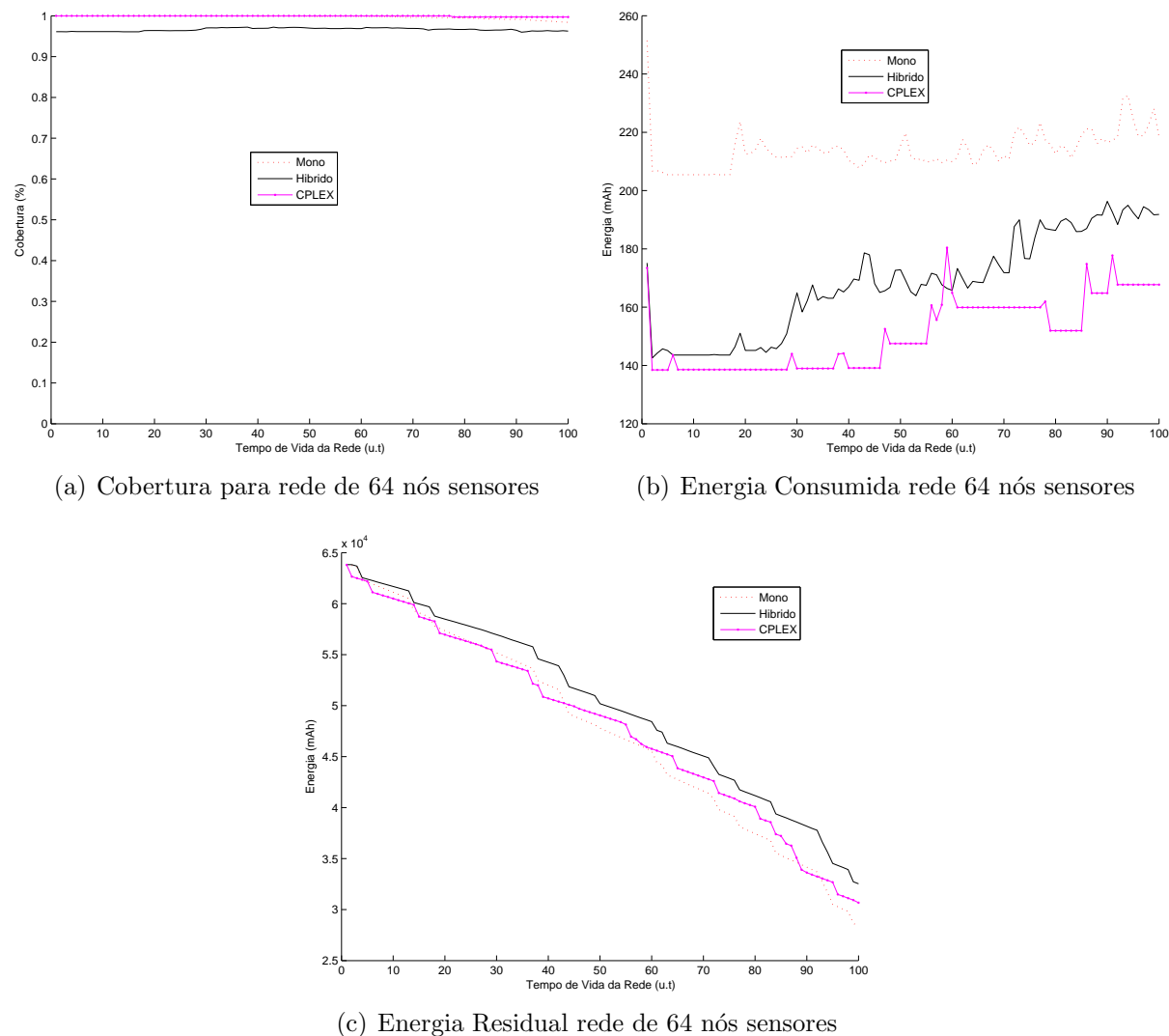
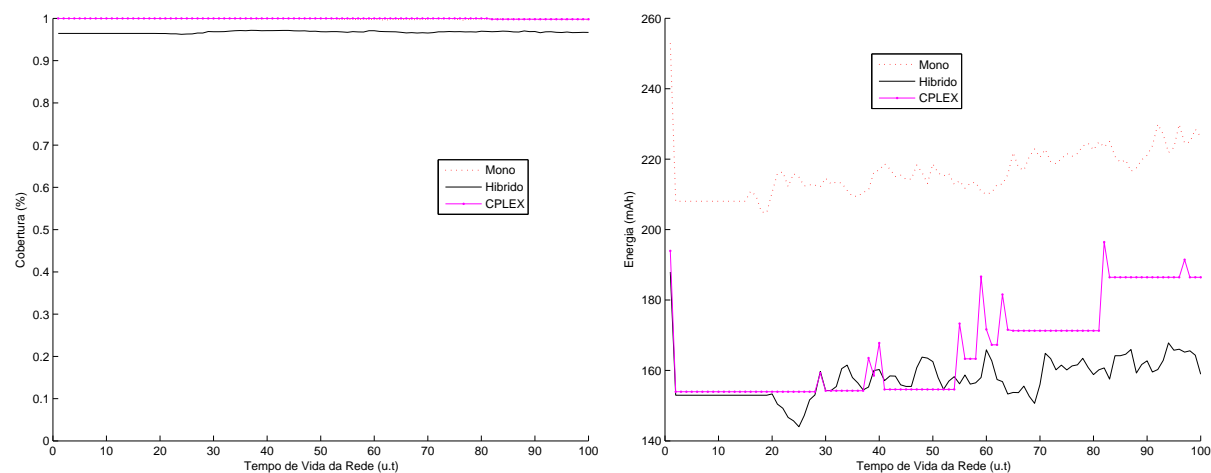


Figura 4.8 Gráficos de resultado para 64 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$.

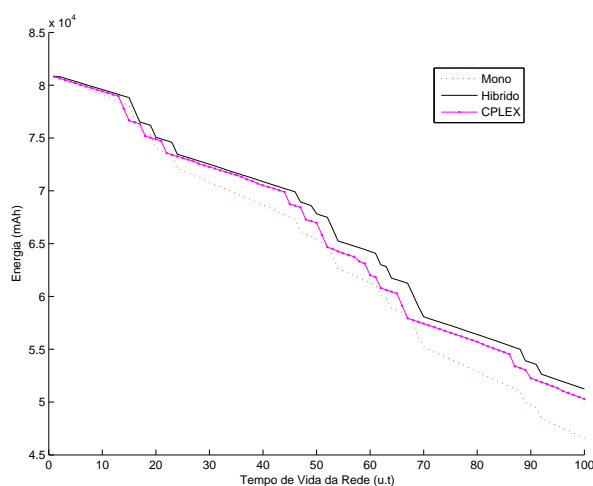
uma situação na qual houvesse a necessidade de se estender o tempo de monitoramento, neste caso espera-se que novamente esta abordagem perca mais rápido a capacidade de cobrir a área monitorada. Esta conclusão é firmada quando se observa novamente a energia consumida pelas abordagens (Figura 4.8b). Mais uma vez, a versão multiobjetivo supera significativamente a versão mono-objetivo, ficando em patamares próximos e até melhores que o dado pelo CPLEX.

Pode-se perceber que em todos os casos, a abordagem multiobjetivo trouxe o benefício de manter-se mais tempo monitorando a área de cobertura. Sabe-se que para isso ser possível foi necessário um “relaxamento” na necessidade de se manter esta área completamente coberta. Um conceito aplicado neste problema é conhecido na literatura como *trade-off*. Define-se *trade-off* como situação de escolha conflitante, ou seja, quando um ganho em um objetivo sempre vem associado a uma perda em outro objetivo. No caso do



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores



(c) Energia Residual rede de 81 nós sensores

Figura 4.9 Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas mecânicas e por energia, para $\phi = 95$.

problema das RSSF em questão, a abordagem multiobjetivo buscou permanecer monitorando por mais tempo, à custa de um relaxamento na cobertura, possibilitando em alguns casos ter menos sensores ativados em uma unidade de tempo, conseqüentemente gastando menos energia. Então, em detrimento de uma pequena área de cobertura, ganhou-se mais “tempo de vida” para a RSSF.

4.3.2 Resultado dos Testes Considerando Falhas apenas por Energia

Nesta segunda etapa dos testes, como também realizado para o caso mono-objetivo, não foi considerada a falha mecânica, com isso os sensores falhos ocorrem apenas por falta de energia (bateria). Como também já feito anteriormente, ao invés de fazer a medida em apenas 100 unidades de tempo, deixou-se que a rede funcionasse até o seu limite, sendo

assim possível ver todo seu potencial de cobertura neste período.

Como anteriormente, para cada rede serão mostrados três gráficos, ilustrando a energia residual, a energia consumida e a cobertura a cada período de tempo.

Os gráficos mostrando os resultados para 36, 49, 64, 81 e 100 sensores podem ser visualizados nas Figuras 4.10, 4.11, 4.12, 4.13 e 4.14 respectivamente. Todos os gráficos mostrados aqui são médias em 33 execuções, o que se faz necessário devido às características aleatórias do algoritmo.

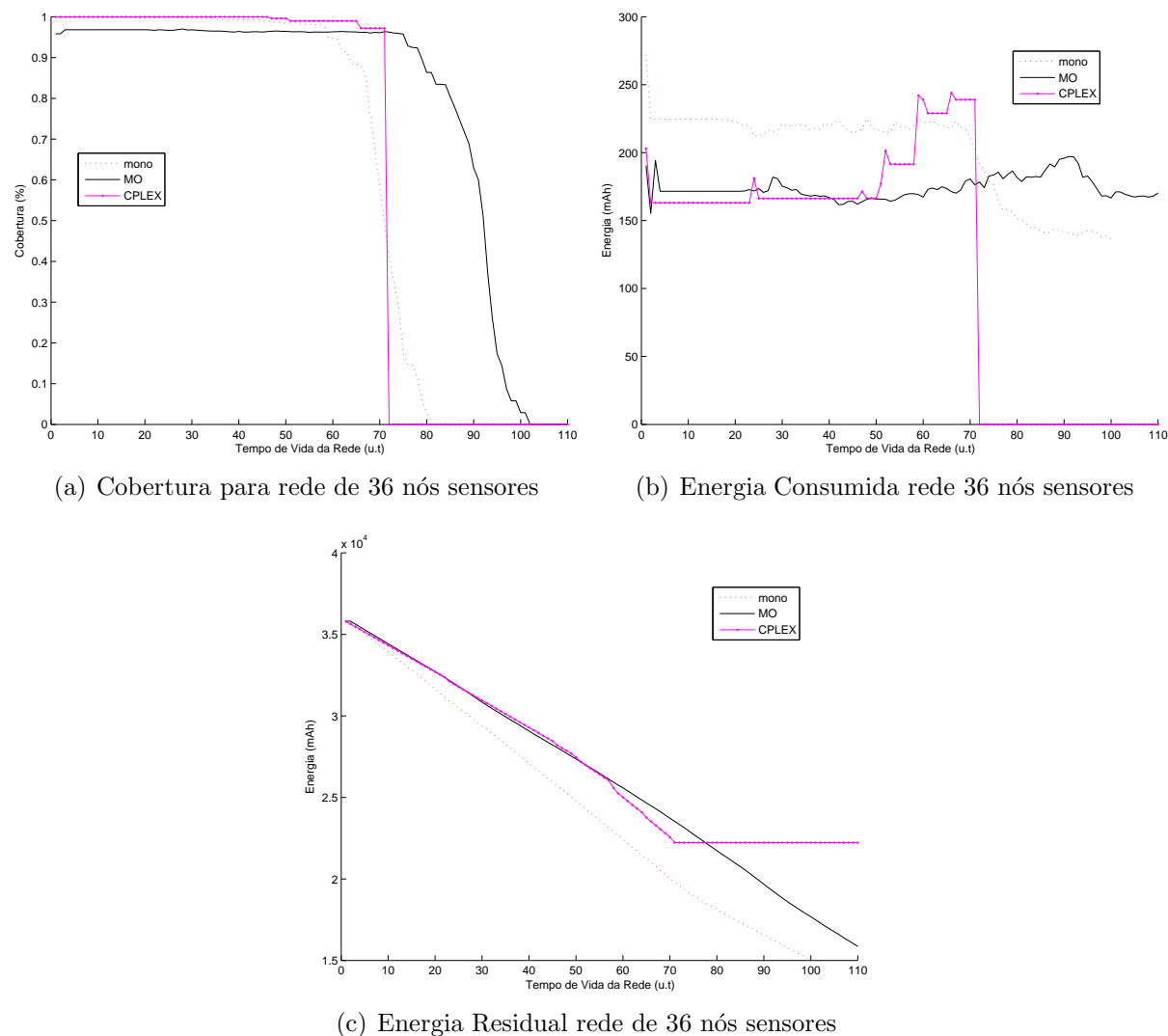
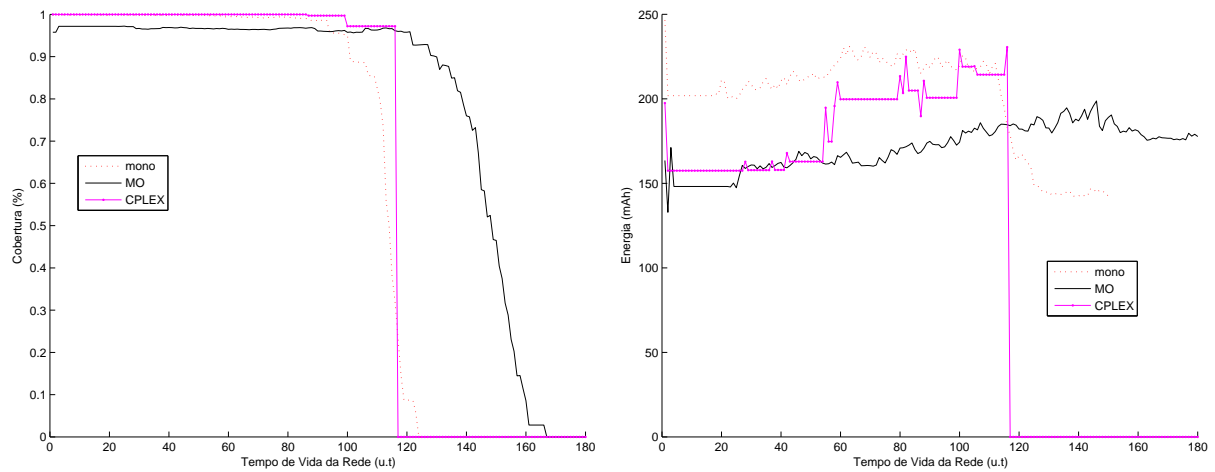


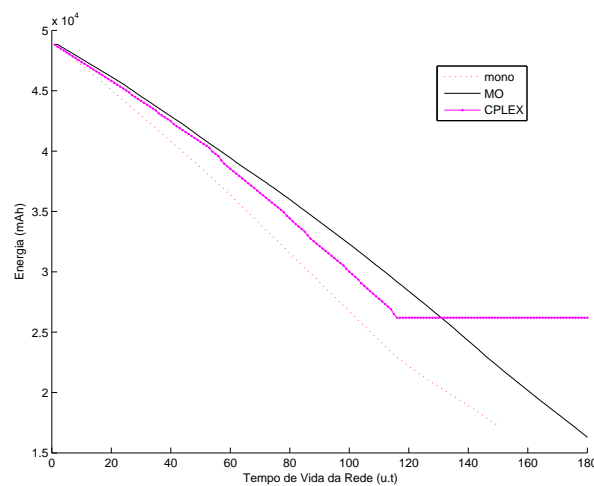
Figura 4.10 Gráficos de resultado para 36 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$.

Além dos gráficos citados acima, nesta etapa dos testes mediu-se também o tempo de processamento gasto por esta nova versão, usando uma abordagem multiobjetivo. Foi possível então, ao montar novamente uma tabela a qual mostra o tempo de processamento dos algoritmos, como realizado para os testes da abordagem mono-objetivo no capítulo anterior.



(a) Cobertura para rede de 49 nós sensores

(b) Energia Consumida rede 49 nós sensores

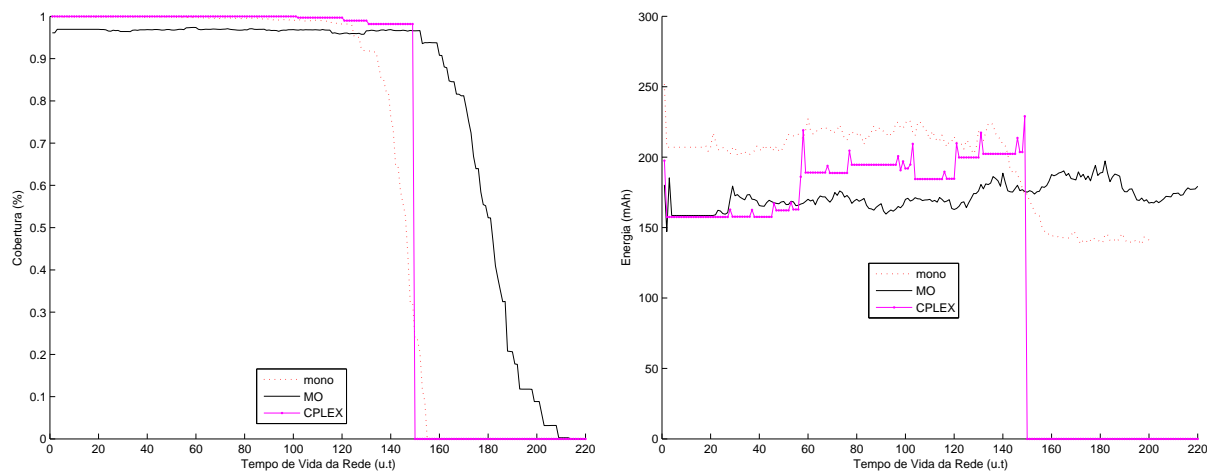


(c) Energia Residual rede de 49 nós sensores

Figura 4.11 Gráficos de resultado para 49 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$.

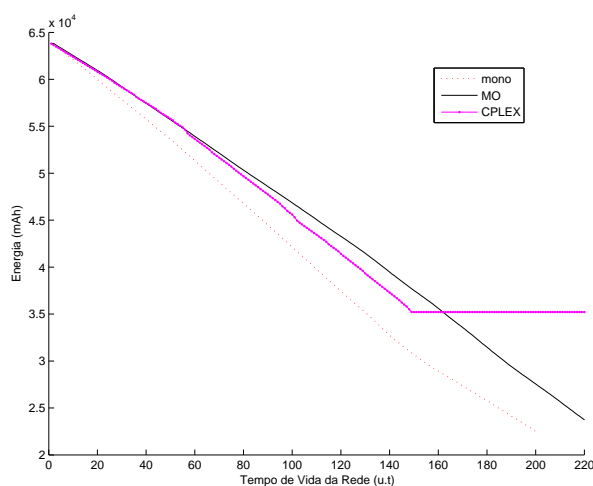
Na Tabela 4.2 ou através do gráfico da Figura 3.20 que representa os dados fornecidos por esta tabela, podem ser visualizados os tempos gastos pelas abordagens mono-objetivo, multiobjetivo e pelo pacote comercial CPLEX 10.2. Todos esses testes foram executados em um servidor de processos rodando o Sistema Operacional Linux Ubuntu Hardy Heron (8.04), com um processador Intel Core 2 Quad 2.5 GHz e com 4GB de memória RAM.

4.3.2.1 Teste comparativo entre diferentes escolhas para ϕ Para uma melhor avaliação desta versão multiobjetivo, e das suas utilidades e flexibilidades, nesta subseção serão apresentados resultados para diversos valores possíveis para escolha de ϕ . Como já informado, ϕ representa a quantidade mínima de cobertura que se deseja na RSSF a ser monitorada. Este valor deve ser fornecido por um decisor externo como parâmetro de



(a) Cobertura para rede de 64 nós sensores

(b) Energia Consumida rede 64 nós sensores



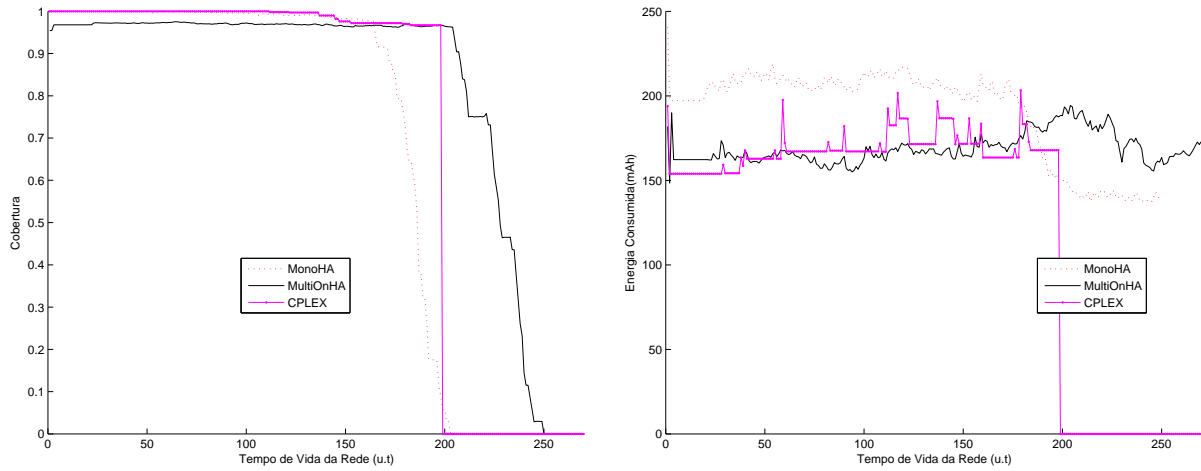
(c) Energia Residual rede de 64 nós sensores

Figura 4.12 Gráficos de resultado para 64 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$.

entrada do algoritmo que irá gerenciar a rede.

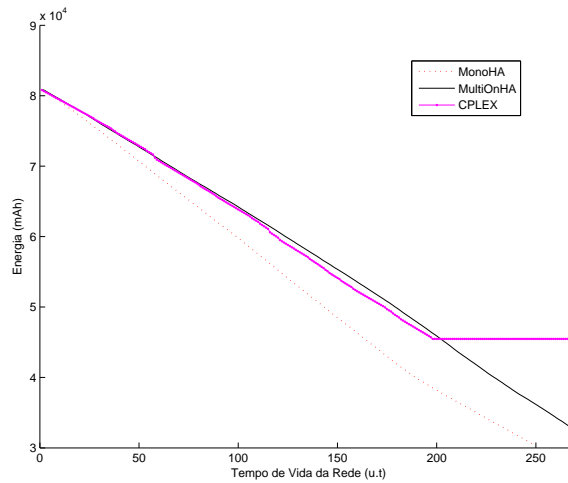
Sabe-se que a taxa de cobertura em uma RSSF está ligada à aplicação, pois podendo existir situações em que se obtendo 50% na cobertura é o suficiente, para outros casos esta necessidade pode ser de 74% de cobertura, para se considerar uma área coberta. Então para mostrar como o algoritmo consegue se adaptar à aplicação desejada, aqui serão apresentados testes variando-se valores para esta necessidade mínima de cobertura. Como este valor é representado por ϕ , os valores utilizados nos testes foram ϕ iguais a: 10, 30, 40, 70, 80 e 95, nas instâncias de 81 e 100 sensores.

4.3.2.2 Análise dos Resultados Da mesma maneira que nos testes mostrados na Seção 4.3.1, aqui serão comparados os resultados para as instâncias de 36, 49, 64 e



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores



(c) Energia Residual rede de 81 nós sensores

Figura 4.13 Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$.

81 nós, e além destas, mais uma de 100 nós sensores. Todos estes resultados podem ser visualizados, respectivamente, nos gráficos das Figuras 4.10, 4.11, 4.12, 4.13 e 4.14. Percebe-se que da mesma maneira que nos testes anteriores, o comportamento de cada algoritmo permanece o mesmo, independente da quantidade de sensores. Para todos os casos que serão citados, novamente serão mostrados os resultados exatos, dados pela abordagem por programação linear inteira, e resolvido pelo pacote comercial CPLEX. Da mesma maneira, estes resultados servirão de limite inferior, no caso da medição de energia consumida, e de limite superior, para a cobertura e a energia residual.

As diferenças entre os testes realizados na Seção 4.3.1 e os apresentados aqui, são que agora apenas falhas por falta de energia, ou seja, bateria dos sensores, foram consideradas, não havendo mais falhas por fatores externos, como as falhas mecânicas citadas nos testes anteriores. Além disto, o período de monitoramento também foi modificado, não mais

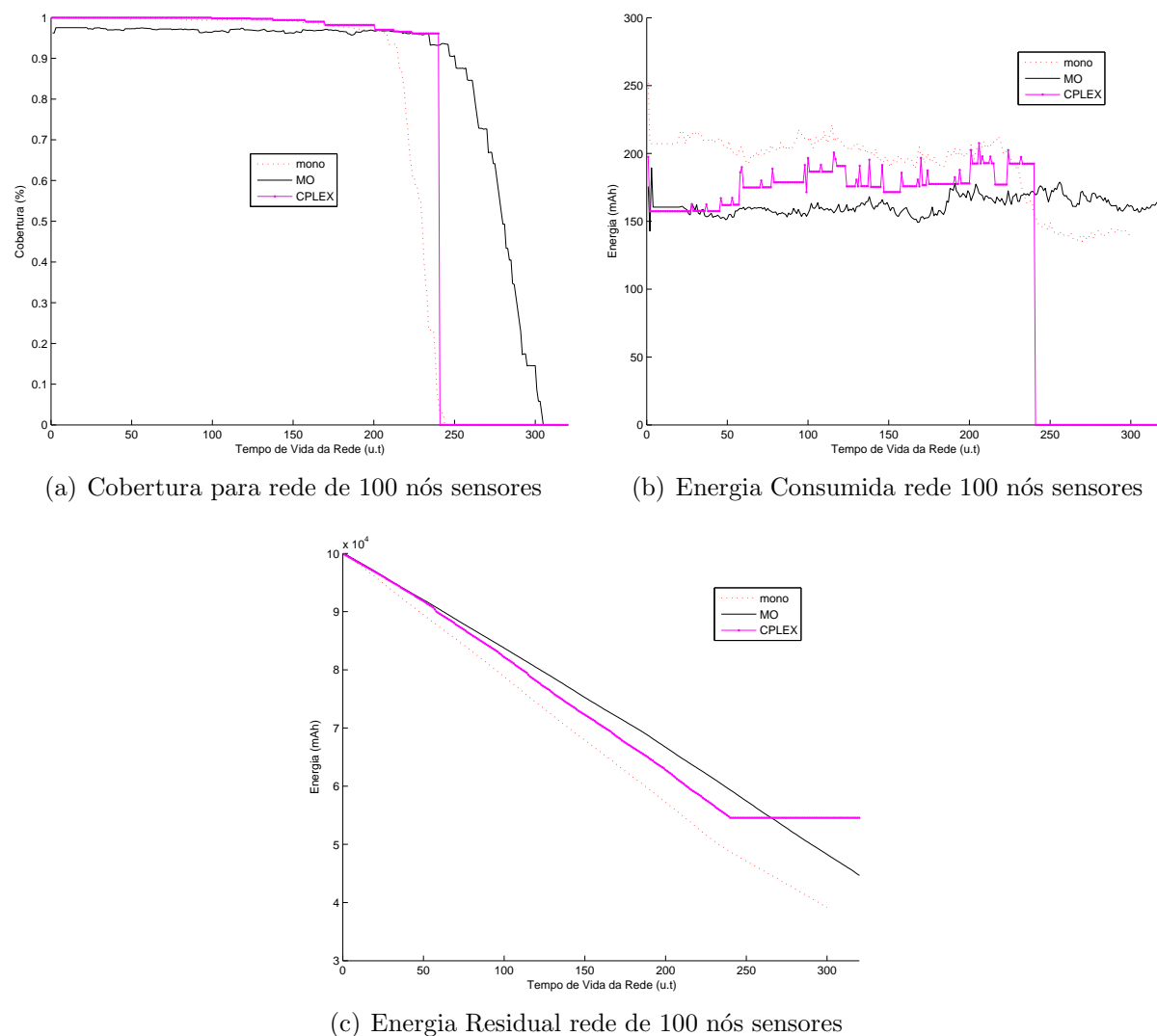


Figura 4.14 Gráficos de resultado para 100 sensores na abordagem multiobjetivo com falhas apenas por energia, para $\phi = 95$.

sendo apenas 100 unidades de tempo, e sim, a quantidade de unidades de tempo em que a rede conseguiu permanecer em estado de monitoração. Também foi computado o tempo de processamento, para efeitos comparativos da eficiência entre cada aplicação dos algoritmos aqui apresentados.

Ao se comparar os resultados obtidos nesta fase dos testes com os resultados obtidos na Seção 4.3.1, percebe-se que para todos os casos, as diferenças entre cada implementação (*Online* Híbrida Mono-objetivo, *Online* Híbrida Multiobjetivo e CPLEX) permanecem as mesmas. A *Online* Híbrida Mono-Objetivo mantendo seu consumo de energia maior que os demais, fazendo com que a rede atinja menores tempos de monitoramento em comparação com a versão multiobjetivo. Por fim têm-se os resultados obtidos pelo pacote comercial CPLEX, que novamente serviu como parâmetro para mostrar o quão próximo as soluções sugeridas estão da solução ótima do problema, conforme a modelagem ma-

Indivíduo	1	2	3	4	5	6	7	8
Aptidão	23	15	50	150	12	2	8	20

Tabela 4.1 Tabela exemplificando indivíduos em uma roleta

INSTÂNCIA	CPLEX		MONO-OBJETIVO		MULTIOBJETIVO	
	Q.UT	TEMPO	Q.UT	TEMPO	Q.UT	TEMPO
36 Sensores	71 u.t.	38.08	80 u.t.	0.24	101 u.t.	0.2560
49 Sensores	116 u.t.	133.56	123 u.t.	0.51	166 u.t.	0.53
64 Sensores	149 u.t.	676.96	154 u.t.	0.80	213 u.t.	0.73
81 Sensores	198 u.t.	2194.9	202 u.t.	1.25	254 u.t.	1.10
100 Sensores	240 u.t.	9972.43	243 u.t.	1.82	299 u.t.	1.60

Tabela 4.2 Tabela de Tempo (em segundos) gasto para todas as instâncias do CPLEX, Mono-Objetivo e Multiobjetivo. A coluna “Q.UT” representa a quantidade de unidades de tempo que o algoritmo manteve alguma cobertura.

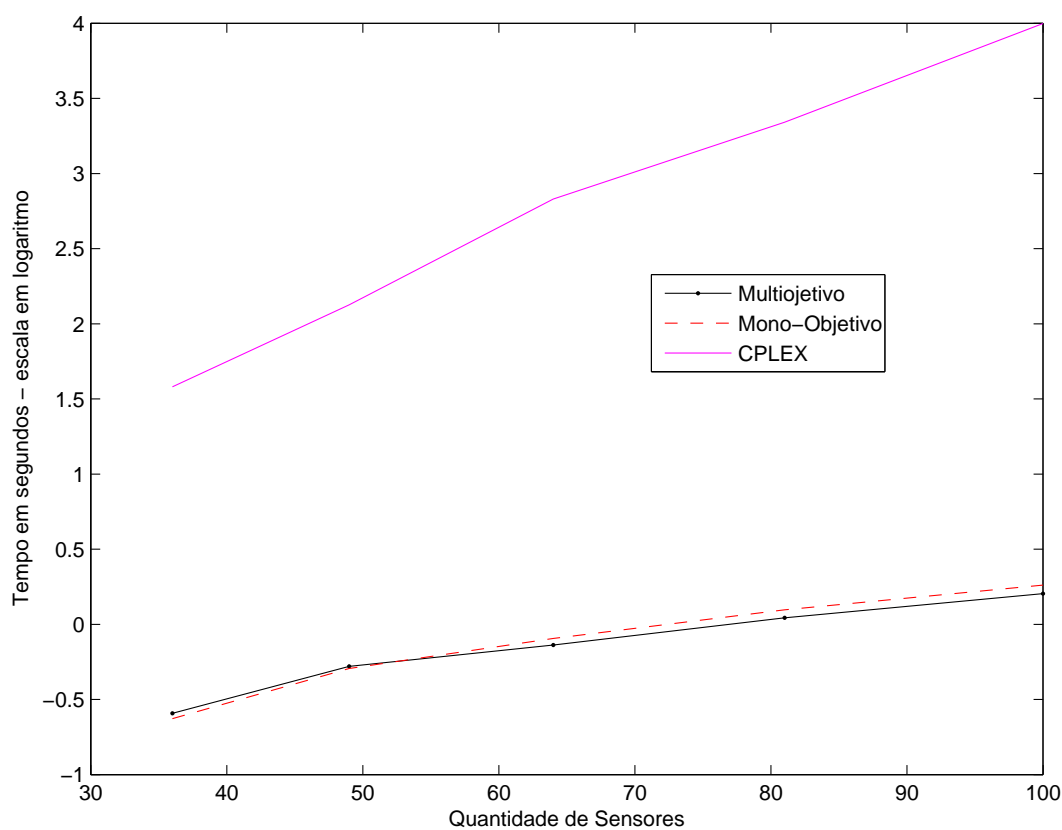
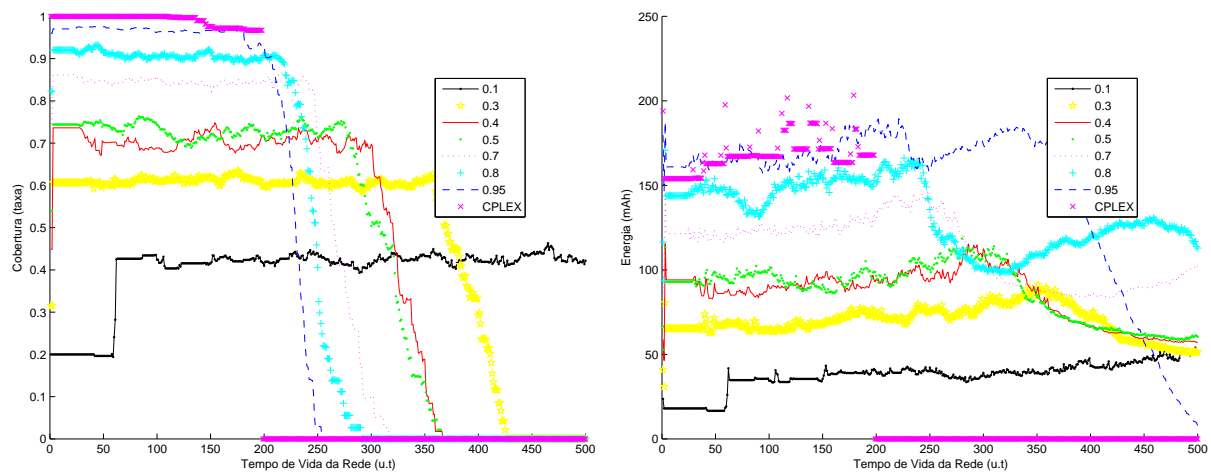
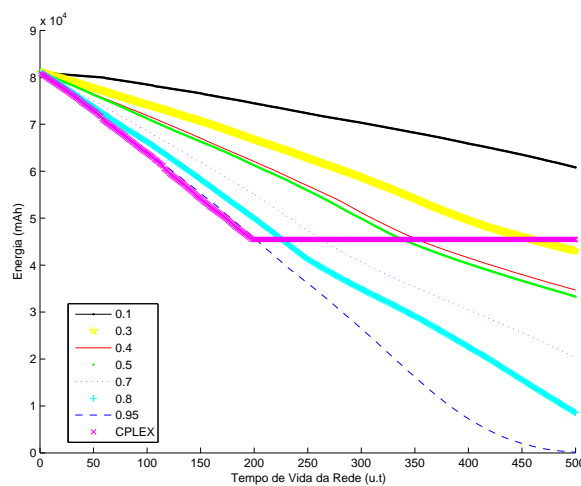


Figura 4.15 Gráfico demonstrando o resumo da Tabela 4.2.



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores

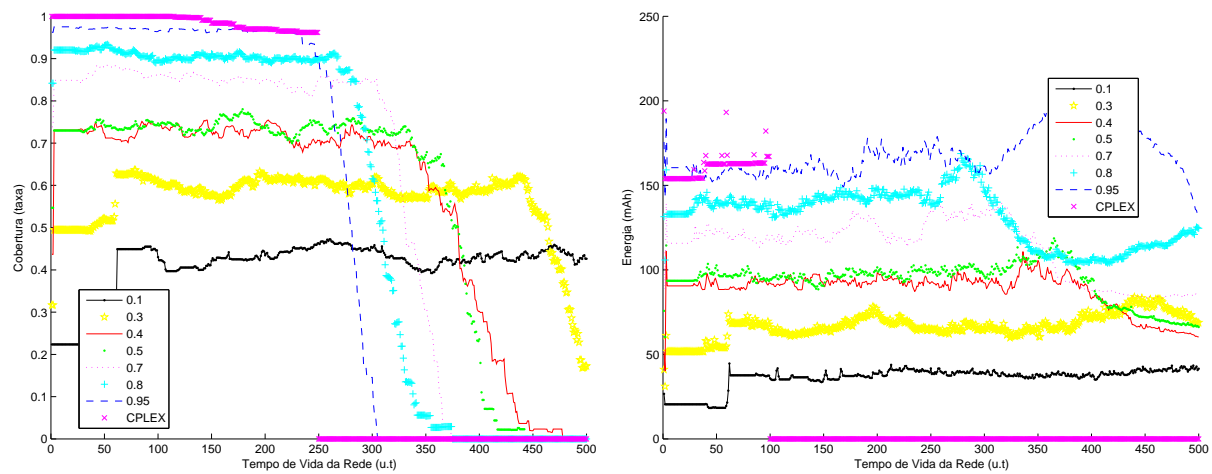


(c) Energia Residual rede de 81 nós sensores

Figura 4.16 Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas apenas por energia, com valores de 10, 30, 40, 70, 80 e 95, para ϕ .

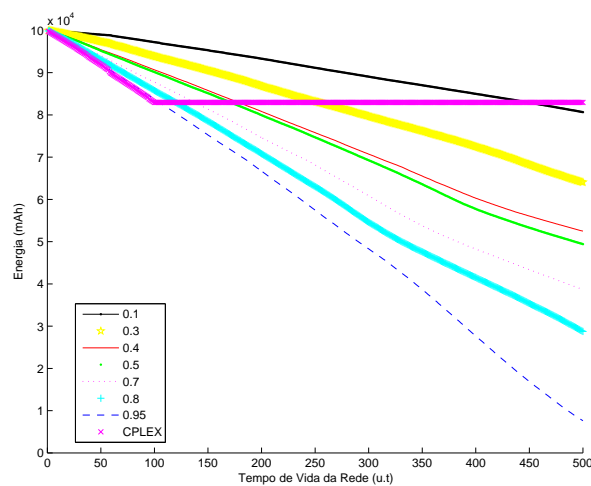
temática introduzida no Capítulo 2.

Olhando para todos os resultados, pode-se observar que a versão multiobjetivo aqui sugerida, começa a ter uma queda acentuada na cobertura em momentos próximos de quando há esta mesma queda para o CPLEX. Foi mencionado que os resultados dados pelo CPLEX para a energia consumida seriam usados como um limite inferior, no entanto, em grande parte dos resultados aqui apresentados a energia consumida para o caso multiobjetivo é inferior à obtida pelo CPLEX. Este fato pode ser entendido quando lembrado que a cobertura para o caso multiobjetivo foi fixada em no mínimo 95%, possibilitando assim um menor consumo de energia da rede. Este controle que possibilita estipular o quanto se deseja cobrir da área a ser monitorada não é possível fazer para o caso apresentado no CPLEX, então seus resultados servem apenas como parâmetro de comparação de qualidade da aplicação aqui mostrada.



(a) Cobertura para rede de 100 nós sensores

(b) Energia Consumida rede 100 nós sensores



(c) Energia Residual rede de 100 nós sensores

Figura 4.17 Gráficos de resultado para 100 sensores na abordagem multiobjetivo com falhas apenas por energia, com valores de 10, 30, 40, 70, 80 e 95, para ϕ .

Através da instância de 36 sensores, pode-se notar a diferença entre os testes dos quais foram consideradas falhas mecânicas e por energia, e os que foram apenas falhas por energia. Comparando-se as Figuras 4.6 e 4.10, as quais representam respectivamente a primeira e segunda bateria de testes, pode-se ver nos gráficos de cobertura da rede, que para o teste em que são consideradas apenas falhas por energia, houve um maior tempo de monitoramento. Este fato é facilmente explicado, pois como não foram consideradas falhas por fatores externos e sim apenas falhas por energia, mais sensores ficaram disponíveis na RSSF, possibilitando assim um monitoramento por mais tempo.

Uma das grandes diferenças entre o método aqui proposto e a solução exata do modelo matemático pelo pacote comercial CPLEX, está novamente no esforço computacional. Esta diferença pode ser visualizada na Tabela 4.2 e também no gráfico da Figura 4.15. Ao olhar para estes resultados, pode-se inferir novamente a grande eficiência da abordagem

proposta, pois seu tempo de processamento se mostra extremamente menor que o do CPLEX. Como exemplo, tem-se a instância de 100 sensores, em que a solução dada pelo CPLEX chega a ser mais de 6239 vezes mais lenta que a versão *Online* Híbrida Multiobjetivo. Ao analisar o gráfico da Figura 3.20, que se encontra em escala logarítmica, pode-se perceber que o crescimento do esforço computacional para o CPLEX é bem próximo de um crescimento exponencial como já visto no capítulo anterior, o que permite concluir que para aplicações com grande quantidade de sensores, o uso deste método se torna inviável. No entanto, quando analisados os resultados para o *Online* Híbrido Multiobjetivo e para versão Mono-objetivo, observa-se que sua aplicação permaneceria viável para grandes quantidades de sensores.

Percebe-se que o fato de usar uma abordagem multiobjetivo não degradou a velocidade na qual se resolveu o problema das RSSF, ficando o tempo de execução no mesmo patamar da versão mono-objetivo apresentada no capítulo anterior.

Sabendo-se da possibilidade de se variar a taxa de cobertura desejada para a área a ser monitorada (representada por ϕ), foram realizados testes para diversos valores de ϕ , nas instâncias de 81 e 100 sensores. Os resultados foram apresentados nas Figuras 4.16 e 4.17. Foi usada mais de uma instância, para mostrar novamente que o comportamento do algoritmo proposto não está ligado à quantidade de nós sensores presente da rede. Como se pode ver nos resultados, o comportamento para cada valor de ϕ manteve-se o mesmo, tanto para 81 como 100 sensores. A diferença ocorreu apenas para o “tempo de vida” das redes. Isso permite concluir novamente que a expectativa de “tempo de vida” é sempre maior para redes mais densas.

Como já mencionado anteriormente, a versão multiobjetivo traz a possibilidade de se utilizar o conceito de *trade-off*. Nos gráficos das Figuras 4.16 e 4.17 se pode ver claramente que nas situações das quais deseja-se um maior “tempo de vida” da rede, é preciso abrir mão da cobertura. Como o comportamento é semelhante, tanto na instância de 81 como na de 100 sensores, comentários a respeito dos resultados serão feitos apenas para o caso de 100 sensores. Por exemplo, na Figura 4.17 para o caso em que a cobertura mínima desejada (ϕ) era de 95%, a rede permaneceu monitorando acima deste patamar até 256 u.t.. Já para o caso em que ϕ foi de 30% permaneceu até 484 u.t. e neste caso a cobertura ficou em um patamar de 60% praticamente por todo este tempo.

Vale a pena observar que para $\phi = 0.4$ e $\phi = 0.5$ os resultados foram semelhantes, tanto a cobertura como o consumo de energia permaneceram em níveis similares. Para explicar o ocorrido, foi traçado um gráfico (Figura 4.18) que mostra o mínimo de sensores necessários para que se atinja uma determinada porcentagem de cobertura na área de $2500m^2$ utilizada nos testes. Neste gráfico é possível visualizar que para se atingir tanto uma porcentagem de 40 como 50% de cobertura, se faz necessária a mesma quantidade mínima de sensores, a saber 3 sensores. As quantidades mínimas de sensores serem iguais traz como consequências os mesmos comportamentos nos resultados, pois como mostrado, o algoritmo proposto faz uso desta quantidade de sensores para definir alguns parâmetros para o controle de densidade de nós ativos na RSSF.

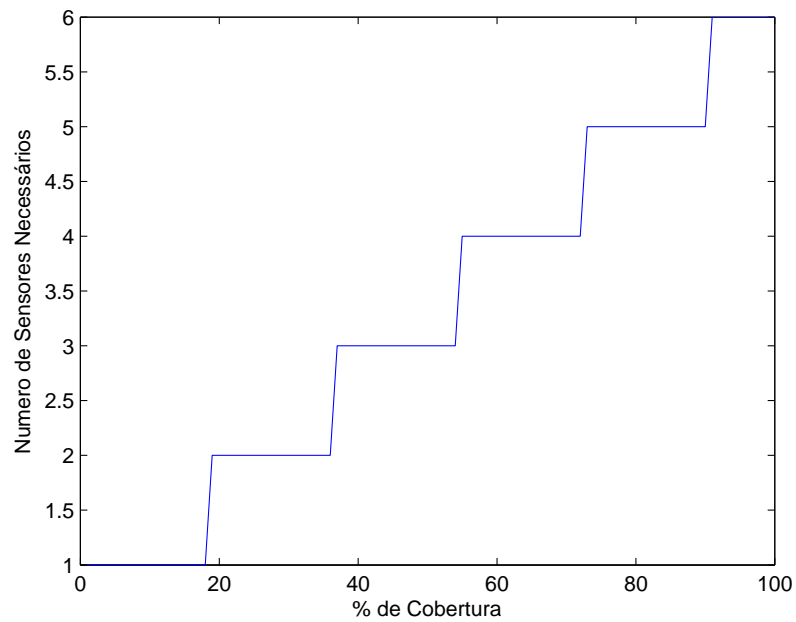


Figura 4.18 Gráfico traçado conforme a Equação 3.4, fazendo a variação de f_{cob} . Os valores para o raio do sensor e a área a ser monitorada foram os mesmos utilizados nos testes realizados, ou seja, $r_s = 15m$ e $a_m = 2500m^2$.

CONCLUSÃO

Quem conduz e arrasta o mundo não são as máquinas, mas as idéias.

—VITOR HUGO

O melhor lugar para ter sucesso é onde você está, com aquilo que você tem.

—CHARLES SCHWAB

Este trabalho trata do problema de cobertura e conectividade em uma Rede de Sensores sem Fio (RSSF). Apresentado o problema, são mostradas duas abordagens utilizadas para resolvê-lo. O problema consiste em monitorar uma determinada área ao longo de um período de tempo. Para monitorar esta área são utilizados sensores sem fio do padrão MICA2, os quais foram distribuídos de forma aleatória como se tivessem sido jogados por uma aeronave.

Ambas as abordagens fazem uso de algoritmos que buscam tratar a rede de forma estática, procurando a melhor forma de garantir a cobertura e ao mesmo tempo também garantir a conectividade entre os nós sensores em um determinado período de tempo. Este tratamento estático se dividiu basicamente em dois estágios:

1. Tratamento da cobertura utilizando Algoritmo Genético. (Seção 3.3.1)
2. Tratamento da conectividade utilizando Algoritmo de Caminho Mínimo. (Seção 3.3.2)

No primeiro estágio, estando de posse dos nós sensores disponíveis na rede, o Algoritmo Genético (AG) decide quais nós sensores devem ficar ativados, já no segundo estágio estes nós sensores são conectados de forma a garantir a existência de um caminho entre cada nó sensor da rede e o nó sorvedouro. Vale a pena lembrar que o Algoritmo de Caminho Mínimo implementado, tem autonomia para ativar nós sensores, caso for necessários para a garantia da conectividade da rede.

Após a formação da rede inicial, a mesma era estendida para uma abordagem dinâmica, em que cada nó sensor está sujeito a falhas, ora mecânica ora por esgotamento de bateria (energia). Nesta abordagem dinâmica foram utilizadas duas estratégias simultaneamente, uma Local *Online* e outra Global (Subseção 3.3.3). A Local *Online* entra em ação toda vez que um nó sensor falha, buscando suprir a cobertura e também a conectividade possivelmente perdida devido ao nó sensor falho, no entanto sua visão é apenas local, preocupando-se apenas em resolver o problema na região afetada, não se preocupando o quão isso afeta a rede como um todo. Esta abordagem à medida que o tempo se passa,

faz com que a RSSF consuma muita energia. Isso motivou a necessidade de aplicar uma estratégia Global. Esta estratégia busca reestruturar a rede de uma forma geral, para isso ela faz uso dos mesmos algoritmos utilizados no momento inicial da RSSF, em que a rede é tratada de forma estática. Desta forma foi possível ter uma visão global do problema no determinado tempo em que esta estratégia é acionada. Através da estratégia Global busca-se uma melhor solução na unidade de tempo em questão, tendo em vista a área de demanda e os nós sensores disponíveis naquele momento. De posse da nova solução, novamente esta rede estática é estendida para uma versão dinâmica. Através desta estratégia Global que é acionada segundo um determinado critério consegue-se fazer o controle da energia consumida pela rede.

As duas abordagens usadas para tratar o problema de cobertura e conectividade em uma RSSF são:

- Abordagem Mono-objetivo (Capítulo 3)
- Abordagem Multiobjetivo (Capítulo 4)

Essas abordagens baseiam-se em uma modelagem matemática que é mostrada no Capítulo 2. Essa modelagem inclui uma função de consumo de energia da rede que se busca minimizar. A modelagem também inclui as restrições de cobertura e conectividade do problema.

Suas diferenças se encontram basicamente em como tratar a cobertura da área monitorada. Na primeira abordagem, procura-se otimizar apenas um objetivo, correspondendo a um menor consumo de energia com a restrição de que toda área estivesse completamente coberta, pelo maior tempo possível. Já na segunda abordagem, divide-se a otimização em dois objetivos, otimização do consumo de energia, e também da cobertura da rede. Nesta abordagem o conceito de cobertura da rede sofre uma pequena modificação. Como a cobertura passa a ser tratada como um objetivo e não como uma restrição, esta pode ser controlada de acordo com a demanda do usuário, ou seja, não necessariamente uma área é considerada coberta quando todos seus pontos de demanda estão cobertos, mas sim, quando uma taxa ϕ destes pontos de demanda estão sendo cobertos pela RSSF.

Para se resolver o problema de cobertura na Abordagem Mono-objetivo é utilizado um AG Mono-objetivo, descrito no Capítulo 3, na Subseção 3.3.1. Este algoritmo busca fornecer a melhor solução de cobertura para a RSSF tendo em vista o consumo de energia, determinando quais sensores presentes na rede devem ficar ativados para que se obtenha o mais baixo consumo de energia, e ao mesmo tempo cobrir toda área a ser monitorada. Já na segunda abordagem, para resolver o problema de cobertura da rede, usou-se um AG Multiobjetivo conforme descrito no Capítulo 4, Subseção 4.2.1. Como não se trata mais de apenas um objetivo e sim de dois, durante a solução é apresentado uma curva Pareto-ótima que contém soluções possíveis para o problema. De posse desta curva, pode-se aplicar o conceito de *trade-off*, com a tomada de decisão envolvendo um compromisso entre o tempo de vida da rede e a sua taxa de cobertura. Nesta abordagem abre-se a possibilidade de tratar a cobertura de acordo com a demanda do usuário. Este usuário define a taxa mínima de cobertura necessária para que se possa considerar a área como coberta. A partir desta definição aplica-se um decisor na curva Pareto-ótima para fazer a

escolha de uma solução para o Algoritmo Genético Multiobjetivo, conforme se pode ver na Subseção 4.2.1.3.

Para ser possível avaliar a qualidade dos resultados apresentados, o problema de cobertura e conectividade na RSSF é modelado através de Programação Linear Inteira. Este modelo é resolvido pelo pacote comercial CPLEX, obtendo assim resultados exatos para o problema. Para ambas as abordagens baseadas em AG's estes resultados servem de limite superior para avaliar a cobertura e a energia residual da rede, e de limite inferior para avaliar a energia consumida da rede. Para cada abordagem são realizadas duas baterias de testes, na primeira consideram-se falhas mecânicas e falhas por energia, tentando mostrar o comportamento da RSSF em situações de falhas inesperadas, e na segunda são consideradas falhas apenas por energia, ou seja, esgotamento da bateria, para poder ver o comportamento do algoritmo até o limite do "tempo de vida" da rede. Como na segunda bateria de testes os nós sensores não sofrem interferência de falhas por meio externo, mede-se também o tempo de processamento de cada abordagem utilizada e também da solução fornecida pelo pacote comercial CPLEX. Nos testes são usadas instâncias de 36, 49, 64, 81 e 100 nós sensores, presentes em uma área de $2500m^2$.

Os resultados da cobertura, energia consumida e energia residual apresentados pela Abordagem Mono-objetivo (Seção 3.4), se mostram próximos, no entanto inferiores, dos valores ótimos fornecidos pelo CPLEX. No entanto seu grande ganho é apresentado ao se medir o tempo de processamento, mostrando assim sua grande eficiência em tempo computacional, resolvendo o problema em até 5472 vezes mais rápido que o CPLEX. Na Abordagem Multiobjetivo os resultados se mostram bastante interessantes (Seção 4.3). Observa-se que o relaxamento da restrição de cobertura total, mesmo em níveis pequenos como de 5%, permitiu a obtenção de resultados satisfatórios em relação ao "tempo de vida" da rede, comparados ao CPLEX e a Abordagem Mono-objetivo. Vale a pena ressaltar que esta versão Multiobjetivo atinge tempos computacionais equivalentes ao da Abordagem Mono-objetivo. Para o algoritmo multiobjetivo também são realizados testes que mostraram que à medida que a cobertura mínima desejada é menor, maior pode ser a expectativa de "tempo de vida" da rede.

Por fim é mostrada a eficiência das abordagens ao resolver os problemas propostos quando se aumenta a quantidade de sensores na rede, com seus tempos de processamento apresentando crescimento mais lento que o apresentado pela solução exata pelo pacote comercial CPLEX. Como as aplicações em problemas reais como monitoramento de florestas e grandes centros urbanos pode demandar uma grande quantidade de nós sensores, as abordagens propostas mostram-se aptas a serem utilizadas para tais aplicações.

5.1 TRABALHOS FUTUROS

Uma proposta de trabalho a ser desenvolvido no doutorado é a aplicação de técnicas de programação dinâmica ao problema de otimização em redes de sensores sem fio. Desta maneira será possível realizar uma otimização considerando uma previsão dos momentos em que ocorrerão as falhas dos sensores. Outra proposta é expandir a modelagem para uma versão não linear e juntamente implementar um controle de potência para os nós sensores, consequentemente deixando possível a variação dos raios de comunicação e

monitoramento, trazendo assim características mais reais para o problema.

APÊNDICE A

DIFERENÇAS DO NSGA E NSGA-II APLICADOS NA REDE DE SENSORES SEM FIO

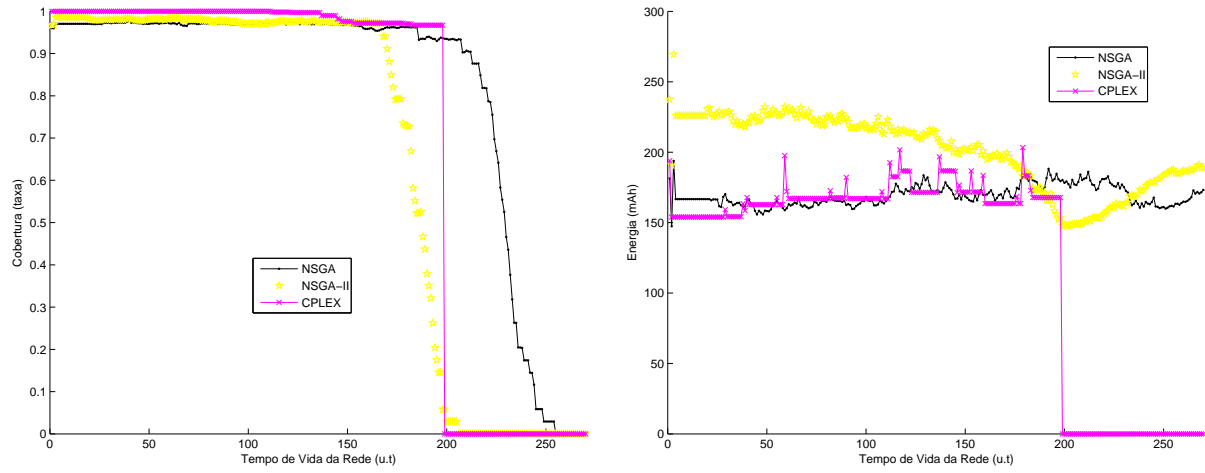
Este apêndice tem o intuito de mostrar mais detalhes das diferenças encontradas na aplicação do NSGA e NSGA-II na Rede de Sensores sem Fio (RSSF). Nas subseções 4.1.1 e 4.1.2 já foram introduzidos respectivamente os conceitos dos algoritmos NSGA e NSGA-II, então serão mostrados apenas os resultados e suas respectivas análises, para assim justificar o fato de ter-se usado o NSGA como base do Algoritmo Genético Multiobjetivo, utilizado na solução do Algoritmo *Online* Híbrido Multiobjetivo introduzido no Capítulo 4.

A justificativa de não ter sido usado o padrão mais atual, o NSGA-II, se dá pelo fato dos resultados obtidos com esta versão serem menos eficientes, no caso específico do problema aqui resolvido. Isto ocorre basicamente devido a alta pressão seletiva presente no NSGA-II, ou seja, as soluções tendem a convergir para um ponto de uma maneira mais rápida, fazendo em alguns casos convergir para soluções que não são as ótimas, ou próximas destas. Os resultados obtidos com as abordagens NSGA e NSGA-II podem ser visualizados nos gráficos da Figura A.1

Quando observado o gráfico da Figura A.1b, é visível que a abordagem baseada no NSGA-II consome mais energia do que as demais, como consequência pode-se observar claramente no gráfico da Figura A.1a, no qual representa a cobertura da rede no tempo, que a queda apresentada pelo NSGA-II ocorre aproximadamente 40 u.t. antes da apresentada pela versão baseada no NSGA. Como já falado anteriormente uma das diferenças que podem ser notadas ao se avaliar o algoritmo NSGA apresentado em [Srinivas & Deb, 1994] e o NSGA-II apresentado em [Deb et al., 2002] é a diferença na qual é feita a seleção dos indivíduos. Ao analisar ambos os algoritmos é visível que o NSGA-II faz uma maior “pressão seletiva”, ou seja, dá mais chances aos indivíduos que pertencem a níveis de dominância melhores serem selecionados nas gerações seguintes. Particularmente no problema das RSSF apresentados neste trabalho, esta “pressão seletiva” fez com que os resultados piorassem, como se pôde ver nos gráficos da Figura A.1.

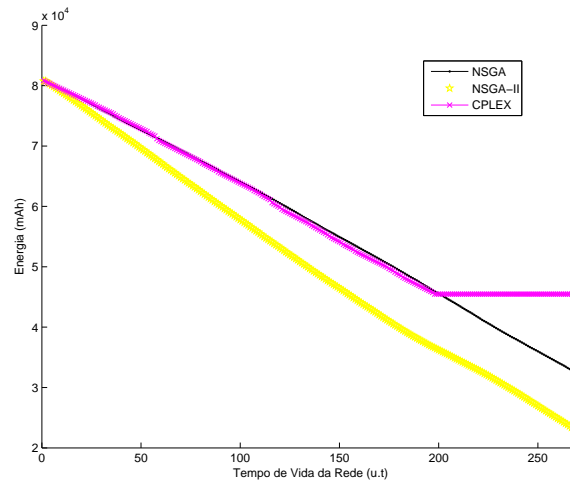
A forma de seleção do NSGA usa uma função de aptidão para avaliar cada indivíduo, através desta função é possível fazer um controle da “pressão seletiva”, ou seja, decidir o quão cada indivíduo é melhor que o outro segundo algum critério. A função de aptidão utilizada neste trabalho foi mostrada na Equação 4.1. A forma na qual foi realizado o processo de seleção no Algoritmo Genético Multiobjetivo (AG_MO) implementado, pode ser vista com mais detalhes na subseção 4.2.1.1.

Fazendo uso da possibilidade de se poder variar a “pressão seletiva” no AG_MO foram realizados testes variando esta pressão de seleção para firmar assim a sensibilidade do problema em questão à “pressão seletiva”. As funções de aptidão usadas foram:



(a) Cobertura para rede de 81 nós sensores

(b) Energia Consumida rede 81 nós sensores



(c) Energia Residual rede de 81 nós sensores

Figura A.1 Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas somente por energia, para $\phi = 95$.

1. Equação: 4.1:

$$fitnessMO = \frac{1}{numPareto} \left(\frac{1}{2^{0.8 \cdot numPareto}} \right)$$

2.

$$fitnessMO_1 = \frac{1}{numPareto} \quad (A.1)$$

3.

$$fitnessMO_2 = \frac{(numDeParetos - numPareto + 1)}{numDeParetos} \quad (A.2)$$

Para um melhor entendimento das diferenças entre as funções mostradas acima, foi traçado um gráfico (Figura A.2) mostrando o valor da função de aptidão de acordo com o nível de dominância a que a solução pertence.

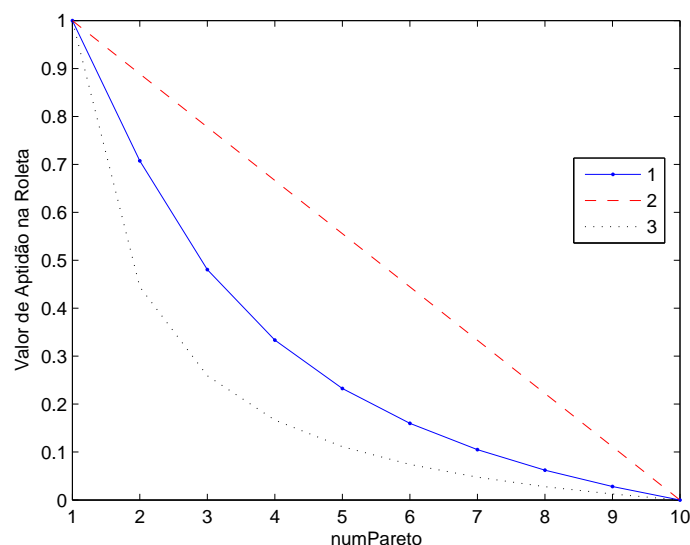


Figura A.2 Gráfico das funções de aptidão usadas para variar a “pressão seletiva”. As curvas 1, 2 e 3, representam respectivamente as funções das Equações: 4.1, A.1 e A.2, com seus valores normalizados entre 0 e 1.

Como se pode ver no gráfico da Figura A.2 cada função tem uma pressão seletiva diferente. Isto pode ser percebido na diferença de valores entre as soluções presentes em níveis de dominância subsequentes.

Para mostrar os resultados para cada função de aptidão usou-se a instância de 81 nós sensores. Os gráficos podem ser visualizados na Figura A.3.

Nos gráficos (a) e (b) da Figura A.3 é possível ver a superioridade da função de aptidão usada no trabalho (Equação: 4.1). Como mostrado na Figura A.2 esta função é a que menos faz “pressão seletiva” sobre os indivíduos pertencentes a níveis de dominância diferentes. No entanto também dá uma prioridade um pouco maior aos melhores. A função da Equação A.2 foi a que obteve resultados intermediários, sua característica é de fazer uma pressão igualitária, e não muito grande, dentre os níveis de dominância, com isso a diferença entre os valores das primeiras camadas não foi muito grande, possibilitando assim dar mais chances a indivíduos inferiores, no entanto indivíduos pertencentes a níveis de dominância muito inferiores, também acabam tendo chances relativamente altas. Já na função de Equação A.1 os resultados foram os piores, a explicação está na alta diferença dos valores de aptidão para os indivíduos presentes no primeiro nível de dominância, desta maneira a “pressão” para que eles fossem selecionados foi maior. Semelhante é o comportamento do algoritmo baseado no NSGA-II, pois ele dá maiores chances para indivíduos pertencentes ao primeiro nível de dominância, por isso esta versão não apresentou melhores resultados.

Pode-se observar que o melhor resultado foi obtido com a Equação 4.1. Quando observado o gráfico da Figura A.2 percebe-se que esta função se encontra em valores de “pressão” intermediários entre as outras duas funções apresentadas, o que faz concluir que foi buscada uma forma melhor de se fazer a “pressão seletiva”. Uma demonstração de

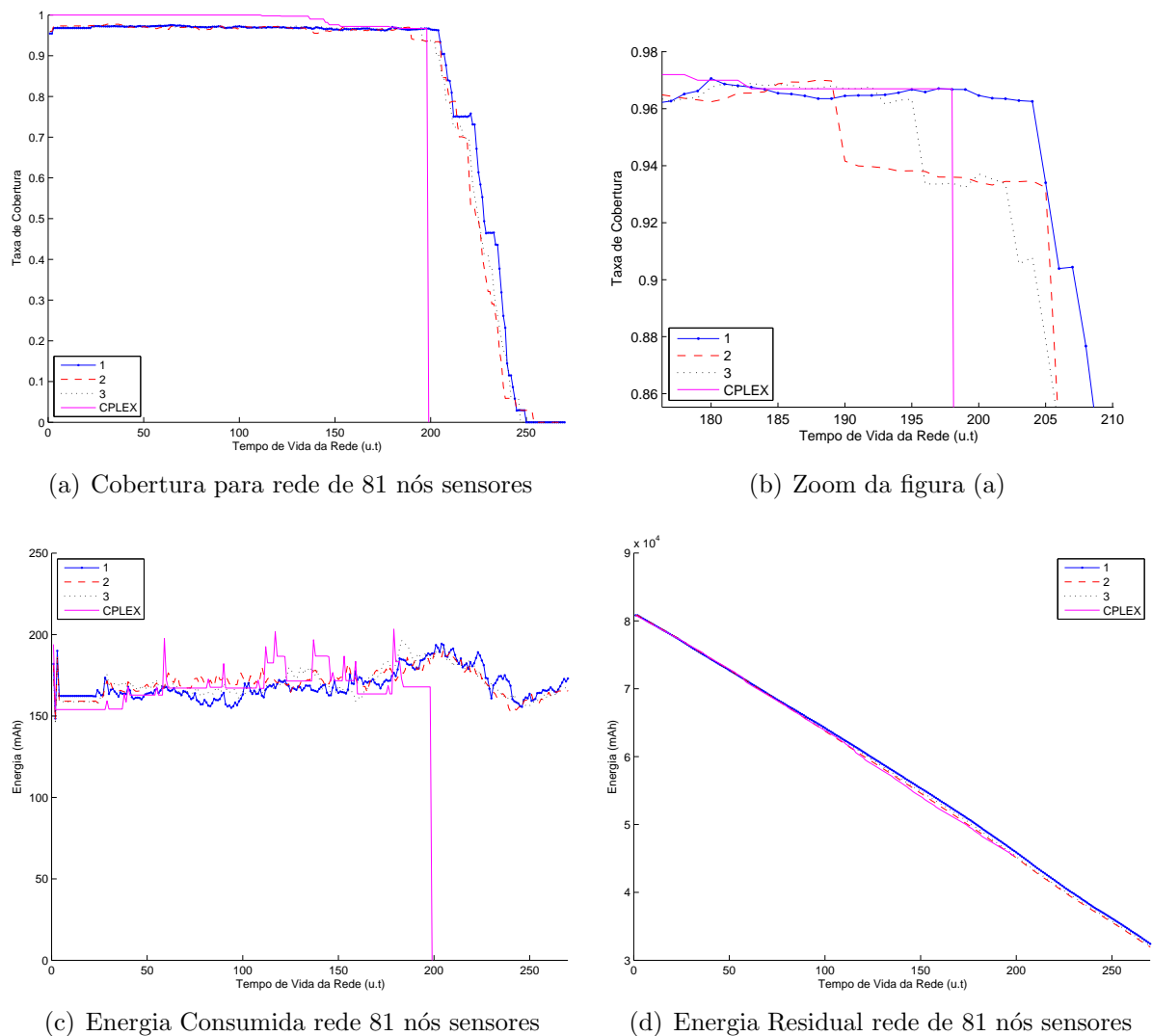


Figura A.3 Gráficos de resultado para 81 sensores na abordagem multiobjetivo com falhas somente por energia, para $\phi = 95$. As curvas 1, 2 e 3, representam respectivamente as funções das Equações: 4.1, A.1 e A.2

que realmente existe uma melhor “pressão seletiva” a ser aplicada, pode ser encontrada em [Silva et al., 2006].

REFERÊNCIAS BIBLIOGRÁFICAS

- [Ilo, 2006] (2006). Ilog cplex. source: <http://www.ilog.com/products/cplex/>.
- [Aioffi et al., 2007] Aioffi, W. M., Quintão, F. P., & Mateus, G. R. (2007). Integrated methods for organization of wireless sensor networks with mobile sink. In *International Workshop on Design and Reliable Communication Networks* La Rochelle, France.
- [Akyildiz et al., 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.
- [Belmont-Moreno., 2001] Belmont-Moreno., E. (2001). The role of mutation and population size in genetic algorithms applied to physics problems. *International Journal of Modern Physics C*, 9(12), 1345–1355.
- [Brooks & Iyengar, 1998] Brooks, R. R. & Iyengar, S. S. (1998). *Multi-sensor fusion: fundamentals and applications with software*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- [Carrano et al., 2006] Carrano, E., Soares, L., Takahashi, R., Saldanha, R., & Neto, O. (2006). Electric distribution network multiobjective design using a problem-specific genetic algorithm. *Power Delivery, IEEE Transactions on*, 21(2), 995–1005.
- [Chakrabarty et al., 2001] Chakrabarty, K., Iyengar, S. S., Qi, H., & Cho, E. (2001). Coding Theory Framework for Target Location in Distributed Sensor Networks. In *Proceedings of the International Symposium on Information Technology: Coding and Computing (ITCC01)* (pp. 130–134).
- [Choi & Oh, 2000] Choi, D. H. & Oh, S. Y. (2000). A new mutation rule for evolutionary programming motivated from backpropagation learning. *IEEE Transactions on Evolutionary Computation*, 2(4), 188–190.
- [Crossbow, 2006] Crossbow (2006). *MICA2 - Wireless Measurement System*. Source. XBOW 6020-0042-04, Crossbow Technology, <http://www.xbow.com/>.
- [Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2), 182–197.
- [Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.

- [Fan et al., 2000] Fan, H. Y., Lu, J. W. Z., & Xu., Z. B. (2000). An empirical comparison of three novel genetic algorithms. *Engineering Computations*, 8(17), 981–1001.
- [Gandham et al., 2003] Gandham, S., Dawande, M., Prakash, R., & Venkatesan, S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *IEEE GLOBECOM* San Fransisco, USA.
- [Goldbarg & Luna, 2000] Goldbarg, M. C. & Luna, H. P. L. (2000). *Otimização Combinatória e Programação Linear: modelos e algoritmos bibtex*. Editora Campus. ISBN 85-352-0541-1.
- [Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- [Hasancebi & Erbatur, 2000] Hasancebi, O. & Erbatur, F. (2000). Evaluation of crossover techniques in genetic algorithm based optimum structural design. *Computers e Structures*, 1-3(78), 435–448.
- [Holland, 1975] Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- [Kadayif et al., 2005] Kadayif, I., Kandemir, M., Vijaykrishnan, N., & Irwin, M. J. (2005). An integer linear programming-based tool for wireless sensor networks. *J. Parallel Distrib. Comput.*, 65(3), 247–260.
- [Kim et al., 2003] Kim, H. S., Abdelzaher, T. F., & Kwon, W. H. (2003). Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks.
- [Mainwaring et al., 2002] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 88–97). New York, NY, USA: ACM.
- [Martins et al., 2007] Martins, F. V. C., Nakamura, F., Quintão, F. P., & Mateus, G. R. (2007). Model and algorithms for the density, coverage and connectivity control problem in flatwsns. In *INOC - International Network Optimization Conference*.
- [Martins et al., 2008] Martins, F. V. C., Nakamura, F. G., & Takahashi, R. H. C. (2008). Uma análise multiobjetivo para o problema de cobertura e conectividade em uma rede de sensores sem fio plana. In *A Pesquisa Operacional e o Uso Racional de Recursos Hídricos: Simpósio Brasileiro de Pesquisa Operacional*.
- [Megerian & Potkonjak, 2003] Megerian, S. & Potkonjak, M. (2003). *Low power 0/1 Coverage and Scheduling Techniques in Sensor Networks*. Technical Report 030001, University of California, Los Angeles.

- [Meguerdichian et al., 2001] Meguerdichian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2001). Coverage Problems in Wireless ad hoc Sensor Networks. In *Proceedings of the IEE Conference on Computer Communications (INFOCOM01)* (pp. 1380–1387).
- [Nakamura et al., 2007] Nakamura, F., Martins, F. V. C., & Quintão, F. P. (2007). Controle de densidade em redes de sensores: Modelos e algoritmos. In *Anais do XXXIX SBPO - Simpósio Brasileiro de Pesquisa Operacional*.
- [Nakamura et al., 2005] Nakamura, F., Quintao, F., Menezes, G., & Mateus, G. (2005). An Optimal Node Scheduling for flat Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Networking (ICN05)*, volume 3420 (pp. 475–483).
- [Nakamura, 2003] Nakamura, F. G. (2003). Planejamento dinâmico para controle de cobertura e conectividade em redes de sensores sem fio planas. Master's thesis, Universidade Federal de Minas Gerais.
- [Park et al., 2001] Park, S., Savvides, A., & Srivastava, M. B. (2001). Simulating networks of wireless sensors. In *WSC '01: Proceedings of the 33rd conference on Winter simulation* (pp. 1330–1338). Washington, DC, USA: IEEE Computer Society.
- [Potts et al., 1994] Potts, J. C., Giddens, T. D., & Yadav, S. B. (1994). The development and evaluation of an improved genetic algorithm based on migration and artificial selection. *IEEE Transactions on Systems, Man and Cybernetics*, 1(24), 73–86.
- [Quintao et al., 2004] Quintao, F., Nakamura, F., & Mateus, G. R. (2004). A Hybrid Approach to solve the Coverage and Connectivity Problem in Wireless Sensor Networks. In *Proceedings of the IV European Workshop on Meta-heuristics (EU/ME04)*, volume 1.
- [Quintao et al., 2005] Quintao, F., Nakamura, F., & Mateus, G. R. (2005). Evolutionary Algorithm for the Dynamic Coverage Problem Applied to Wireless Sensor Networks Design. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC05)*, volume 2 (pp. 1589–1596).
- [Rabaey et al., 2000] Rabaey, J., Ammer, M., da Silva, J.L., J., Patel, D., & Roundy, S. (2000). Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7), 42–48.
- [Rosenberg, 1967] Rosenberg, R. (1967). *Simulation of genetic populations with biochemical properties*. PhD thesis, Univ. Michigan.
- [Ruiz, 2003] Ruiz, L. B. (2003). *MANÁ: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio*. PhD thesis, Universidade Federal de Minas Gerais.
- [Sareni & Krahenbuhl, 1998] Sareni, B. & Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 3(2), 97–106.

- [Savvides et al., 2001] Savvides, A., Park, S., & Srivastava, M. B. (2001). On modeling networks of wireless microsensors. In *SIGMETRICS '01: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (pp. 318–319). New York, NY, USA: ACM.
- [Schaffer, 1984] Schaffer, J. D. (1984). *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. PhD thesis, Nashville, TN, USA.
- [Schrijver, 1986] Schrijver, A. (1986). *Theory of linear and integer programming*. New York, NY, USA: John Wiley & Sons, Inc.
- [Silva et al., 2006] Silva, V., da Cruz, A., Carrano, E., Guimaraes, F., & Takahashi, R. (2006). On nonlinear fitness functions for ranking-based selection. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 305–311).
- [Srinivas & Deb, 1994] Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
- [Takahashi, 2004] Takahashi, R. H. C. (2004). Otimização escalar e vetorial. Notas de Aula.
- [Takahashi et al., 2003] Takahashi, R. H. C., Vasconcelos, J. A., Ramirez, J. A., & Krahenbuhl, L. (2003). A multiobjective methodology for evaluating genetic operators. *IEEE Transactions on Magnetics*, 3(39), 1321–1324.
- [Tilak et al., 2002a] Tilak, S., Abu-Ghazaleh, N. B., & Heinzelman, W. (2002a). Infrastructure tradeoffs for sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02)* (pp. 49–58).
- [Tilak et al., 2002b] Tilak, S., Abu-Ghazaleh, N. B., & Heinzelman, W. (2002b). A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(2), 28–36.
- [Vasconcelos et al., 2001] Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C., & Saldanha, R. R. (2001). Improvements in genetic algorithms. *IEEE Transactions on Magnetics*, 5(37), 3414–3417.
- [Vieira et al., 2003a] Vieira, M., Coelho, C.N., J., da Silva, D.C., J., & da Mata, J. (2003a). Survey on wireless sensor network devices. *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, 1, 537–544 vol.1.
- [Vieira et al., 2003b] Vieira, M., Vieira, L., Ruiz, L., Loureiro, A., Fernandes, A., & Nogueira, J. (2003b). Scheduling Nodes in Wireless Sensor Networks: A Voronoi Approach. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN03)* (pp. 423–429).

- [Wang et al., 2005] Wang, W., Srinivasan, V., & Chua, K. (2005). Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05* (pp. 270–283). New York, NY, USA: ACM Press.
- [Ye et al., 2003] Ye, F., Zhong, G., & e L. Zhang, J. C. (2003). Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS03)* (pp. 28–37).
- [Ye et al., 2002] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3, 1567–1576 vol.3.
- [Zhang & Hou, 2005] Zhang, H. & Hou, J. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Wireless ad hoc and Sensor Networks*, 1, 89–123.