

METAHEURÍSTICA ILS PARA O PROBLEMA DE COBERTURA E ROTEAMENTO EM REDES DE SENSORES SEM FIO

Wellington Gomes Ribeiro, José Elias Claudio Arroyo, André Gustavo dos Santos

Departamento de Informática - Universidade Federal de Viçosa
Campus da UFV, 36.570-000 Viçosa - MG - Brasil
{wellington.ribeiro, jarroyo, andre}@dpi.ufv.br

RESUMO

Neste artigo é abordado o problema de cobertura e roteamento de dados multi-período em Redes de Sensores Sem Fio (RSSFs). Uma RSSF é formada por uma estação base e um conjunto de nós sensores espalhados numa área a ser monitorada. O objetivo do problema é determinar, para cada período p de tempo, um subconjunto de nós sensores com baixo consumo de energia de forma a maximizar a cobertura da área e assegurar a conectividade da rede. Isto é, garantir a existência de uma rota de comunicação entre todo nó sensor ativo e a estação base. Para resolver o problema, propõe-se um método de otimização baseado na metaheurística ILS (*Iterated Local Search*) que consiste na aplicação sucessiva de uma busca local para melhorar uma solução obtida pela perturbação de uma solução ótimo local. A qualidade das soluções obtidas pelo método proposto é verificada através da comparação com soluções geradas por uma metaheurística GRASP e o software CPLEX que resolve o modelo matemático do problema.

PALAVRAS CHAVE: Metaheurísticas, ILS, GRASP, Redes de Sensores Sem Fio.

Áreas Principais: Metaheurísticas, Otimização Combinatória.

ABSTRACT

This paper deals with the coverage and routing multi-period problem in wireless sensor networks (WSNs). A WSN is composed by a base station and a set of sensor nodes scattered in an area to be monitored. The goal of the problem is to determine, for each time period p , a subset of the sensor nodes with low power consumption that maximize coverage of the area and provide network connectivity (ensure the existence of a route of communication between all active sensor node and base station). To solve the problem, we propose an optimization method based on the ILS (*Iterated Local Search*) metaheuristic that iteratively applies local search to modifications of a current solution. The quality of the solutions obtained by the proposed method is verified by comparison with solutions generated by a GRASP metaheuristic and the CPLEX software that solve the mathematical model of the problem.

KEYWORDS: Metaheuristics, ILS, GRASP, wireless sensor networks.

1. Introdução

Redes de sensores sem fio (RSSF) ganharam atenção mundial nos últimos anos, particularmente com a proliferação da tecnologia de sistemas micro-eletrônicos, o que facilitou o desenvolvimento de sensores sem fio inteligentes. Uma RSSF é uma ferramenta promissora para o monitoramento do mundo físico, utilizando um conjunto de sensores sem fio, alimentados por baterias, que cumprem a função de detectar, processar e transmitir informações para uma estação base, denominada sorvedouro ou *sink*. Uma RSSF pode ser implantada de forma rápida e barata, permitindo, a grande escala, o monitoramento e acompanhamento de um vasto leque de aplicações, tais como monitoramento ambiental (Ilan Sousa *et al.* 2011), previsão e detecção de calamidades naturais, rastreamento de veículos, monitoramento médico, suporte para

engenharia de tráfego, monitoramento de animais ou de certas características de um ambiente de difícil acesso (como temperatura, condições atmosféricas, luminosidade), monitoramento de hábitat (lugar específico habitado por um organismo ou população), etc. (Estrin, *et al.* 1999; Akyildiz *et al.* 2002), Murthy e Manoj, 2004; Romer e Mattern, 2004; Ian *et al.* 2002).

Uma RSSF é um tipo especial de rede móvel ad hoc (MANET - Mobile Ad Hoc Networks) composta por um conjunto grande de nós sensores (dispositivos autônomos) equipados com baterias de potências limitadas. Geralmente estas baterias limitam as capacidades de processamento e de comunicação sem fio. Além disso, na maioria das aplicações é impraticável ou inviável repor a energia destes dispositivos através da substituição das baterias. Em algumas aplicações pode-se exigir um monitoramento contínuo durante vários meses requerendo um maior tempo de vida operacional da rede. Assim, é comum que uma rede de sensores seja implantada com uma densidade elevada de nós a fim de prolongar o tempo de vida da rede. No entanto, se todos os nós sensores operam simultaneamente, dados redundantes de sensoriamento, colisões de comunicação sem fio e interferências causarão muito desperdício de energia. Portanto, é necessário cobrir a área a ser monitorada com um número mínimo de sensores (nós ativos) de modo que não existam pontos cegos e seja mantida a conectividade. Ativar apenas o número necessário de nós sensores em qualquer momento é uma maneira eficiente de economizar a energia global da rede.

O problema de cobertura torna-se um problema importante em redes de sensores de larga escala, onde os nós são implantados aleatoriamente. A cobertura é uma das questões mais fundamentais, o que afeta diretamente a capacidade e eficiência da rede de sensores. Geralmente, a cobertura e roteamento de dados podem ser considerados como medidas de qualidade de serviço (QoS) em uma rede de sensores (Meguerdichian, *et al.* 2003).

O principal desafio na concepção e planejamento de operações de redes de sensores sem fio (RSSF) é a otimização do consumo de energia para prolongar a vida útil da rede. Ativar apenas o número necessário de nós sensores em qualquer período, em particular, é uma maneira eficiente de economizar a energia global do sistema.

O problema de cobertura e roteamento de dados em RSSFs tem sido pouco abordado na literatura. Para resolver esse problema de forma ótima, Nakamura (2003) propôs um modelo de Programação Linear Inteira Mista (PLIM) para minimizar o consumo de energia dos sensores, para múltiplos períodos de tempo, garantindo as restrições de cobertura, conectividade e o roteamento. Utilizando o software CPLEX 9.0, o autor gera soluções ótimas para instâncias do problema com até 60 nós sensores e para 4 períodos de tempo.

Gao *et al.* (2003) propuseram um algoritmo eficiente de controle de densidade para redes estáticas, baseado em um modelo matemático e no número de vizinhos existentes para poder estimar a área coberta redundante de um nó randomicamente posicionado na área monitorada.

Quintão *et al.* (2005) abordam o problema de cobertura e conectividade utilizando uma metaheurística híbrida baseada em Algoritmos Genéticos e Busca Local. Os resultados encontrados são satisfatórios quando comparados com soluções ótimas obtidas pela resolução do modelo matemático de Nakamura (2003). Utilizando essa metaheurística, são feitos testes com instâncias de até 60 nós sensores, contendo 4 períodos de tempo.

Cardei e Wu (2006) utilizam esquemas de controle de densidade analisando áreas redundantes em redes estáticas. Os resultados obtidos por estes autores mostraram que pode-se alcançar uma economia significativa de energia para a rede utilizando critérios para a desativação dos sensores.

Siqueira *et al.* (2006) desenvolveram um método de controle de densidade integrado para resolver problemas de cobertura e roteamento de dados. A abordagem garante uma correta e eficiente operação da rede assegurando área de cobertura e disseminação dos dados. Duas soluções foram implementadas, uma integrando o algoritmo OGDC de Zhang e Hou (2005) e outra foi o algoritmo de roteamento EF-TREE de Nakamura *et al.* (2005).

Andrade e Nakamura (2009) propõem uma metaheurística GRASP para resolver o problema. Os resultados dessa metaheurística, para instâncias com até 60 sensores, foram comparados com as soluções ótimas obtidas pela resolução do modelo de PLIM de Nakamura (2003).

Martins (2009) aborda o problema de cobertura e roteamento de dados em RSSFs de duas maneiras: mono-objetivo e multi-objetivo. O autor propõe dois Algoritmos Genéticos para resolver, respectivamente, os dois tipos de problemas.

Andrade *et al.* (2010) propõem um algoritmo evolucionário com busca local para resolver problema de cobertura e roteamento de dados em RSSFs. Os resultados do algoritmo são comparados com soluções ótimas obtidas pelo modelo de PLIM de Nakamura (2003) e com as soluções obtidas pela heurística GRASP de Andrade e Nakamura (2009). O algoritmo evolucionário híbrido obteve um melhor desempenho em termos de tempo computacional e qualidade de soluções encontradas quando comparadas à heurística GRASP.

Este trabalho propõe a aplicação da metaheurística ILS para resolver o problema de cobertura, conectividade e roteamento de dados em RSSFs. O objetivo é determinar, para cada período de tempo, um subconjunto de nós sensores (nós ativos) para cobrir ou monitorar a uma determinada área. O objetivo é minimizar o consumo de energia da rede (aumentando seu tempo de vida) obtendo uma maior densidade de cobertura sobre a área em um baixo tempo computacional.

O presente artigo é estruturado da seguinte maneira: Na seção 2 apresenta-se a descrição do problema abordado. As etapas que compõem a heurística ILS proposta são detalhadas na seção 3. Na Seção 4 são apresentados resultados computacionais obtidos. As conclusões do trabalho estão na sessão 5.

2. Problema de Cobertura e Roteamento Multi-Períodos (CRMP)

O problema de Cobertura e Roteamento Multi-Períodos em RSSFs, abordado neste trabalho, é definido da seguinte forma. Considere uma área de monitoramento A , um conjunto S de n nós sensores, um nó sorvedouro b (base) e T períodos de tempo. Cada sensor possui uma energia inicial (bateria), um raio R_s de sensoriamento (limite de cobertura) e um raio de comunicação R_c que estabelece o alcance máximo de comunicação entre os sensores. O problema consiste em determinar, para cada período $t \in [1, T]$, um subconjunto de nós sensores (ativos/ligados) de forma a conseguir uma máxima cobertura sobre a área A , garantindo a conectividade entre os nós da rede. Isto é, garantindo a existência de uma rota entre todo nó sensor ativo e o nó sorvedouro b , respeitando os limites de energia dos nós sensores. Na Figura 1 mostra-se um exemplo de uma rede de sensores na qual existe uma rota entre cada sensor ativo e o nó base.

A fim de quantificar tanto a área monitorada quanto a cobertura pelos nós sensores, a área A é definida como um conjunto D de pontos de demanda que precisam ser monitorados. Para determinar esses pontos, a área de simulação foi dividida em grade (células) de tamanhos homogêneos, (Veja Figura 1). Os pontos de demanda são os centros das grades. O número de pontos de demanda depende dos requisitos da aplicação, e dessa maneira, quanto maior a quantidade de pontos, melhor será a aproximação de uma área contínua (Andrade e Nakamura, 2009). Neste problema consideram-se também áreas sem obstáculos, o que significa que um ponto de demanda será coberto, se ele é alcançado pelo raio de cobertura de um nó sensor ativo. Assim, o problema de cobertura é restrito a verificar se todo ponto de demanda é coberto por pelo menos um nó sensor ativo.

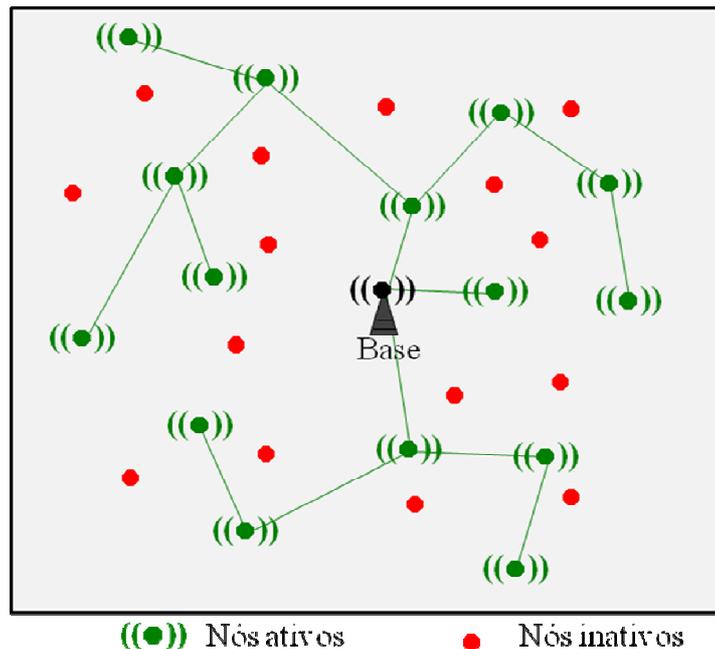


Figura 1- Exemplo de uma configuração de rede considerando conectividade e roteamento.

2.1. Função Objetivo

Para avaliar as soluções obtidas é usada uma função objetivo proposta por Andrade e Nakamura (2009). Essa função minimiza o número de nós ativos e as falhas de cobertura, implicando no aumento da cobertura sobre a área. Além de procurar atender a cobertura e roteamento, a função considera a minimização do consumo de energia para cada período de tempo (maximização do tempo de vida da rede) respeitando o limite de energia de cada sensor.

A função objetivo a ser minimizada é definida da seguinte maneira:

$$f = \sum_{i \in S} \frac{E_i^t}{A_i^t} + \left(1 - \sum_{i \in S} \frac{A_i^t}{|D|} \right) + \sum_{i \in S} y_i^t \quad (1)$$

Onde, S é o conjunto de nós sensores, D é o conjunto de pontos de demanda, $t \in [1, T]$ é o período corrente, A_i^t é área de cobertura do nó sensor $i \in S$ no período t , E_i^t é a energia consumida pelo nó sensor i no período t , e y_i^t é a variável de decisão que possui valor 1 se o sensor i não consegue garantir conectividade no período t e valor 0 caso contrário.

A função objetivo é definida baseada na premissa de, a cada período t , minimizar o consumo de energia garantindo, sempre que seja possível, a cobertura e a conectividade da rede. Note que a função objetivo (1) possui três termos. No primeiro termo priorizam-se nós que possuam baixo consumo de energia e uma grande área própria de cobertura. O segundo termo da função considera a maximização da área de cobertura. Dessa maneira, esses dois primeiros critérios tendem a maximizar a cobertura enquanto minimiza-se o consumo de energia por período. O terceiro termo da função tem por objetivo retirar nós que não consigam comportar o desconto de energia, prejudicando o roteamento da rede.

3. Metaheurística ILS para o Problema de CRMP

Neste trabalho é utilizada a metaheurística ILS (*Iterated Local Search*) para resolver o problema de CRMP em RSSFs. Esta metaheurística, introduzida por Lourenço *et al.* (2002), já foi aplicada a diferentes problemas de otimização combinatória obtendo excelentes resultados. ILS é uma heurística simples que iterativamente aplica busca local para melhorar soluções iniciais obtidas por meios de perturbações numa solução ótima local. Quatro procedimentos básicos são necessários para derivar um algoritmo ILS: Construção de uma solução inicial viável s , Perturbação

da solução s , melhoria da solução perturbada através de uma Busca Local e um Critério de Aceitação que decide qual solução será perturbada na próxima iteração. A melhor dentre todas as soluções encontradas pelo algoritmo, considerando todas as iterações (critério de parada), é retornada como resultado final do algoritmo.

Na Figura 2, é exibido o pseudocódigo do algoritmo implementado neste trabalho para resolver o problema de cobertura e roteamento multi-período em uma RSSF.

```

Algoritmo ILS-Multi-Período ( $T, \beta, niLS$ )
1.  LeituraDeParametrosDoProblema( );
2.  para  $p := 1$  até  $T$  faça //Aplicação Multi-Períodos
3.       $s :=$  HeuristicaConstrutiva( $\beta$ );
4.       $sB :=$  BuscaLocal(  $s$  );
5.      se  $f( sB ) < f( s )$  então
6.           $s := sB$ ; // atualiza a solução corrente
7.      fim
8.       $s^* := s$ ; // guarda a melhor solução
9.      enquanto  $niLS$  faça
10.          $sP :=$  Perturbação(  $s$  );
11.          $sB :=$  BuscaLocal(  $sP$  );
12.         se  $f( sB ) < f( s^* )$  então
13.              $s^* := sB$ ; //atualiza a melhor solução
14.         fim
15.         se  $f( sB ) < f( s )$  então
16.              $s := sB$ ; //atualiza a solução corrente
17.         fim
18.         senão
19.              $s :=$  CriterioAceitacao( $s, sB$  );
20.         fim
21.     fim //Fim do enquanto  $niLS$ 
22.     GuardaConfiguracaoDoPeriodo(  $s^*$  ); //melhor solução do período  $p$ 
23.     Atualiza energia dos sensores utilizados no período  $p$ ;
24.      $p := p+1$ ; //próximo período
25. fim-para
26. Exibe as configurações da rede para os  $T$  período;

```

Figura 2 - Pseudocódigo do algoritmo ILS Multi-Período.

O algoritmo possui três parâmetros de entrada: T , β e $niLS$. β é um parâmetro utilizado na construção de uma solução inicial do problema. O parâmetro $niLS$ é o número de iterações da heurística ILS que é utilizado como critério de parada. $niLS$ representa o número de vezes que cada um dos passos, que compõem o algoritmo ILS, serão executados. T é um parâmetro do problema que indica o número de períodos a serem considerados, ou seja, o número de configurações de rede a serem determinados pela heurística ILS. Observe que o algoritmo possui um laço de repetição, tendo a variável p variando de 1 a T (linha 2).

A primeira etapa do algoritmo ILS consiste em construir uma solução inicial s , através do procedimento **Construção** (passo 3). Esta solução s é melhorada por um procedimento de **BuscaLocal** obtendo a solução sB (passo 4). A melhor solução (s^*) entre s e sB será perturbada e em seguida melhorada pela Busca Local. Note que, as operações de perturbação, busca local e critério de aceitação são executadas durante $niLS$ vezes (laço formado pelos passos 9 a 21).

A melhor solução s^* , para o período p , a ser retornada pelo algoritmo é atualizada nos passos 12-14. Se a solução sB obtida pela busca local não for melhor que a solução perturbada s , então a próxima solução a ser perturbada será determinada por um critério de aceitação. O critério de aceitação é baseado no consumo de energia de toda a rede. É feita uma comparação entre os consumos de energia das soluções s e sB . A solução que obtiver o menor consumo de energia será escolhida para ser perturbada na próxima iteração.

Nas seguintes subseções descrevem-se os procedimentos de construção, busca local e perturbação utilizados no algoritmo ILS.

3.1. Construção de uma Solução Inicial

Para determinar uma configuração inicial da rede, utiliza-se a heurística construtiva gulosa proposta por Andrade *et al.* (2008). Nesta heurística, inicialmente consideram-se como nós ativos todos os sensores que possuem energia disponível na bateria (nós vivos). A heurística, passo a passo, constrói uma rede formada por um subconjunto de sensores em topologia de árvore tendo com raiz o nó base (sorvedouro).

A partir do nó sorvedouro (nó corrente), o próximo passo é analisar sua vizinhança para decidir quais serão os nós que realmente participarão do roteamento (nós ativos) e quais serão os nós a serem desativados. A vizinhança de um nó é formada pelos nós que estiverem dentro do raio de comunicação R_c do nó.

Os nós a serem desativados são determinados analisando a área de cobertura redundante dos nós vizinhos analisados. Se a área redundante é maior do que um limiar $\beta \in [0, 1]$, o nó vizinho *ao analisado* é desativado. O valor $\beta = 0$ indica que o sensor vizinho analisado não tem nenhum ponto de demanda redundante com o sensor corrente, e $\beta = 1$ indica que a cobertura do nó *vizinho ao analisado* é totalmente redundante com o nó sensor corrente. Após verificar a redundância de todos os nós sensores com o sensor corrente, os nós são marcados como “já analisados”.

Para determinar os nós que participarão do roteamento, todos os nos pertencentes à vizinhança analisada são avaliados usando uma função de avaliação gulosa g , que mede o benefício que um nó traz à solução em construção. O nó i^* que decremente mais a falha na cobertura e tiver alta energia residual (energia disponível na bateria do sensor) é nó escolhido para ser inserida na solução e ser o próximo nó corrente. A função de avaliação gulosa é definida como:

$$g = \frac{P_A^t}{P_{EB}^t} \quad (2)$$

Onde, P_A^t é a porcentagem da *área própria de cobertura* (área coberta somente pelo nó analisado e não compartilhada com mais nenhum outro) do sensor analisado no período t com relação a toda área coberta por este; e P_{EB}^t é a porcentagem de energia residual do sensor analisado no período t em relação à capacidade total da bateria do dispositivo.

A heurística construtiva finaliza quando todos os nós da rede forem examinados. Na Figura 3 mostra-se o pseudocódigo da heurística construtiva.

Heurística Construtiva (β)

1. $Sol = \emptyset$; $i^* :=$ nó sorvedouro (primeiro nó a inserido na rede);
2. **enquanto** *Existirem nós não analisados faça*
3. $i := i^*$;
4. $g(i^*) := +\infty$;
5. **para cada um dos nós j vizinhos do nó i faça**
6. $r := \text{AreaRedundante}(j)$; // determina a redundância do nó j na rede
7. **se** ($r > \beta$) **então** // ultrapassou o limite máximo de redundância
8. Desativa o nó j ;
9. **senão** //marca o j como analisado
10. **se** $g(j) < g(i^*)$ **então**
11. $i^* := j$; $g(i^*) := g(j)$;
12. **fim**
13. **fim**
14. **fim-para**
15. $Sol = Sol \cup \{ i^* \}$; // insere o nó i^* na rede para participar do roteamento
16. **fim-enquanto**
17. **retorne** Sol // configuração inicial da rede

Figura 3 – Heurística para a construção de soluções iniciais.

3.2. Busca Local

O objetivo do procedimento de Busca Local (BL) é melhorar a qualidade de uma solução (configuração de rede) s . Este procedimento consiste fazer a busca de uma solução melhor ao inserir (ativar) na rede atual diferentes nós inativos e melhorando as conexões dos nós ativos contidos vizinhança de x de tal maneira que o consumo de energia seja minimizado. Similar ao procedimento proposto por Andrade e Nakamura (2009), a BL implementado neste trabalho é constituída de três etapas: inserção, reconexão e remoção.

A etapa de **inserção** consiste em ligar (ativar) um nó x que ainda dispõe de energia. Para o nó x procura-se em sua vizinhança $Va(x)$ (formada pelos nós que estiverem dentro do raio de comunicação Rc), um nó ativo (pai) que possa rotear seus dados sem causar falha de conectividade na rede. O pai escolhido deve ser aquele onde a conexão entre ele e o nó inserido cause o menor incremento no consumo de energia da rede. Se nó x não possuir um pai que atenda a tais requisitos, ele não será considerado. Na Figura 4 mostra-se um exemplo na qual o nó x será inserido na rede. Neste exemplo, o pai do nó x é o nó base e a vizinhança do nó x é $Va(x) = \{a, b, c, d, e, base\}$.

Na etapa de reconexão determinam-se as melhores conexões entre os nós ativos contidos na vizinhança de x , de tal maneira que o consumo de energia da rede seja minimizado. Para cada nó $v \in Va(x)$ testa-se todas as possibilidades de conexões entre v e os outros nós de $Va(x)$ de tal forma que o roteamento gere uma economia de energia para a rede. Na reconexão são excluídas as possibilidades que gerem ciclos ou problemas de conectividade na rede (por exemplo, nós que não possam suportar o desconto de energia depois da reestruturação da árvore de roteamento). A reconexão deve manter pelo menos uma rota, sem ciclos, a partir de todos os nós ativos até o nó base. Na Figura 4 (b) ilustra-se um exemplo de reconexão após a inserção do nó x .

O nós que não contribuem para a cobertura (possuem áreas completamente redundante) ou não são necessários para assegurar a conectividade da rede devem ser removidos (operação de remoção). A operação de remoção é limitada pelo raio de comunicação Rc do nó x inserido, ou seja, nós ativos desnecessários dentro de $Va(x)$ são removidos. Vale lembrar que alguns nós ativos da rede são usados apenas para garantir o roteamento dos dados sem contribuir para a cobertura. Entretanto, na etapa de reconexão, os filhos desses nós podem ser adotados por outros, tornando-os desnecessários. Na Figura 4 (c) mostra-se a remoção do nó d , pois não contribui para a cobertura. O nó e não pode ser removido, pois ele é importante para o roteamento dos dados de um nó que está fora da vizinhança. Na Figura 5 apresenta-se no pseudocódigo da Busca Local implementada.

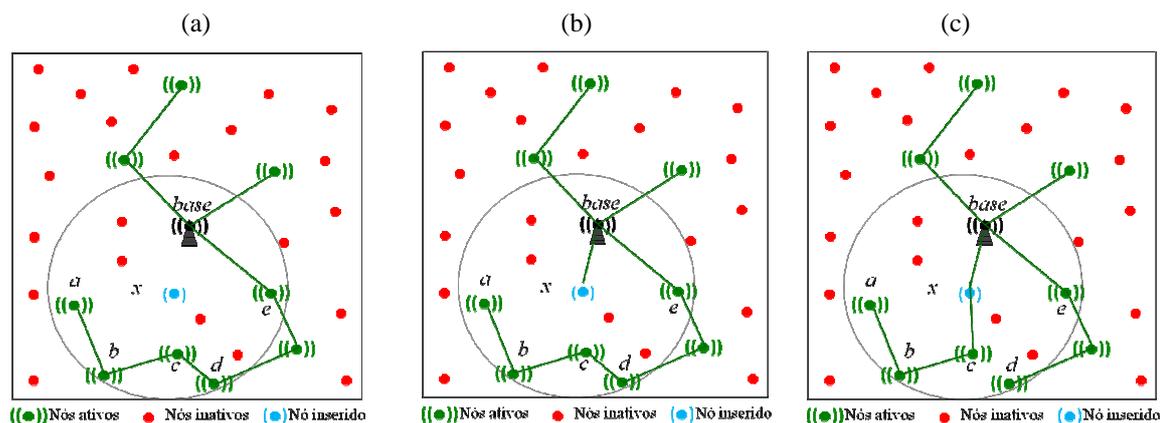


Figura 4 – Etapas da Busca Local.

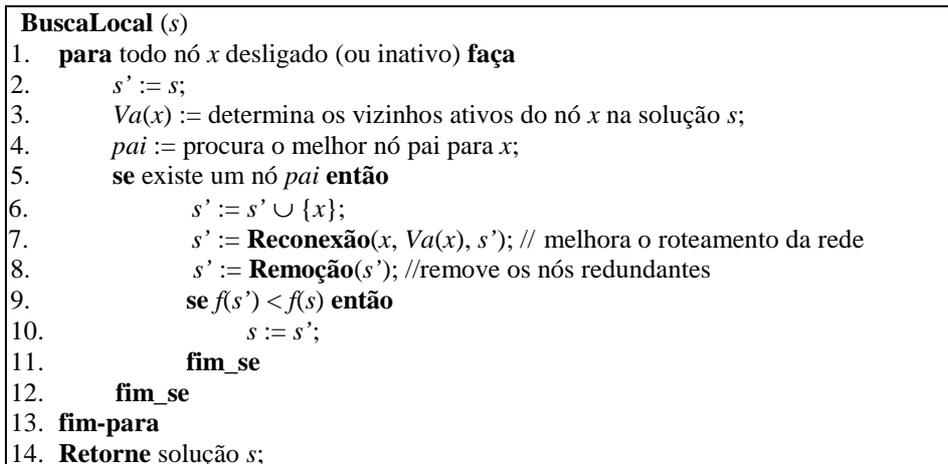


Figura 5 – Procedimento de Busca Local.

3.3. Perturbação

O procedimento de perturbação da uma solução ótima local deve permitir que a busca local explore diferentes espaços na vizinhança e deve evitar um reinício aleatório da solução (desorganizado muito a solução perturbada). Neste trabalho foram implementadas três formas de perturbação de uma solução: (1) perturbação por inserção, (2) perturbação por inserções em cascata e (3) perturbação por remoção (veja Figura 6).

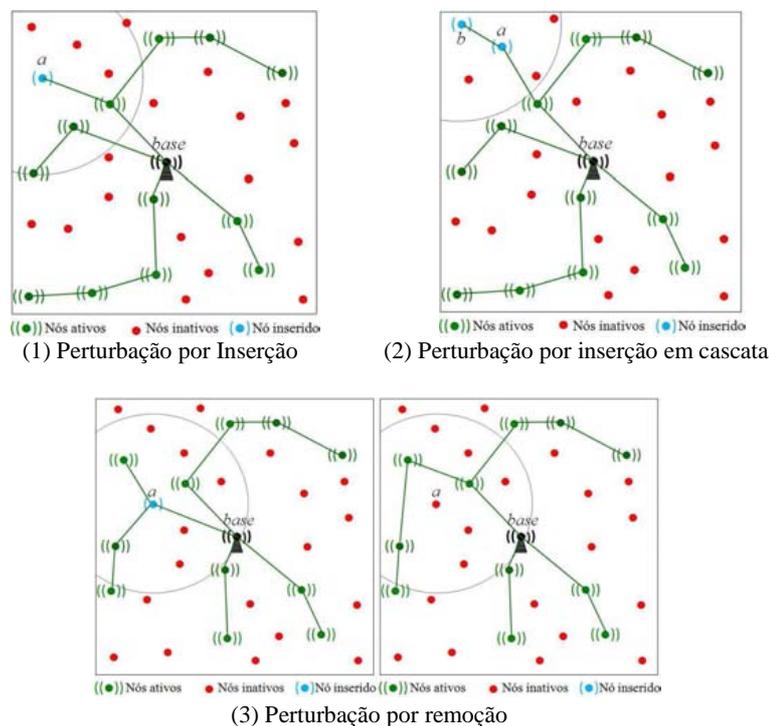


Figura 6 – Movimentos de Perturbação.

- (1) *Perturbação por inserção*: Na solução atual são inseridos simultaneamente *n* nós sensores inativos em seguida é feito a reconexão da rede de forma a obter o mínimo de acréscimo no consumo de energia total.
- (2) *Perturbação por inserção em cascata*: Na solução atual insere-se um caminho formado por dois sensores inativos *a* e *b*. O nó *a* deve estar dentro do raio de comunicação de um nó da rede

- para garantir a inserção e b deve estar dentro do raio de a , ou seja, b será um filho de a . Estas inserções são feitas respeitando a energia residual de cada sensor que forma o caminho da rede.
- (3) *Perturbação por remoção*: Na rede atual, é desligado um nó a ativo, usado apenas para garantir o roteamento dos dados, sem contribuir para a cobertura. Os outros nós ligados ao nó a deverão ser adotados por outros sensores vizinhos ativos. Caso não seja possível a reconexão e a remoção do nó a é desconsiderada, e verifica-se a remoção de outro nó.

4. Testes Computacionais e Resultados obtidos

Os testes computacionais foram feitos utilizando 6 conjuntos de problemas nos quais o número de sensores são 20, 31, 42, 53, 60 e 85, respectivamente. Para cada conjunto foram geradas 12 instâncias diferentes, resultando em um total de 72 problemas.

As instâncias foram geradas considerando informações técnicas disponíveis nos trabalhos de Nakamura (2003) e Andrade e Nakamura (2009). A configuração da área simulada foi de 100m² contendo 100 Pontos de demanda a serem cobertos (1 por m²) em $T = 4$ períodos de tempo. Os sensores foram configurados com o raio de comunicação (R_c) de 3m e o raio de sensoriamento (R_s) de 2m. A capacidade da bateria dos sensores é de 400mA, a energia de manutenção de um sensor é de 30mA por período, a energia de ativação é de 10mA por período, a energia de recepção é de 10mA por pacote e a energia de transmissão fica dependente da distância que se encontram o nó receptor (ou destino). Foi utilizado um sorvedouro no centro da área a ser monitorada, com energia ilimitada e com R_c de 3m².

Os testes computacionais foram feitos utilizando um computador Intel® Core™2 Quad CPU, 2.4GHz, com 3GB de memória RAM. O método ILS-Multi-Período foi implementado na linguagem de programação C, utilizando o compilador Dev-C++, e executado no sistema operacional Windows Vista 64 Bits. Os resultados obtidos são comparados com os resultados ótimos determinados pelo software **IBM ILOG** (CPLEX 12.1) que resolve o modelo matemático do problema e com as soluções viáveis produzidas pela heurística GRASP de Andrade e Nakamura (2009). O executável da heurística GRASP foi fornecido pelos autores e foi executado no computador descrito acima para a resolução das instâncias geradas neste trabalho.

4.1. Parâmetros do algoritmo ILS-Multi-Período

O algoritmo ILS-Multi-Período possui os seguintes parâmetros. Parâmetro β utilizado na construção de uma solução inicial e o número de iterações *niILS* do algoritmo. Como no trabalho de Andrade e Nakamura (2009), foi utilizado $\beta = 1,0$.

Inicialmente, o número de iterações *niILS* do algoritmo foi fixado em 150. Com esta condição de parada foram realizados testes experimentais para analisar o efeito das perturbações propostas: (1) perturbação por inserção, (2) perturbação por inserções em cascata e (3) perturbação por remoção. Foram testadas 15 maneiras de perturbar uma solução, elas são: (1), (1+2), (1+2+3), (1+3), (1+3+2), (2), (2+1), (2+1+3), (2+3), (2+3+1), (3), (3+1), (3+1+2), (3+2), (3+2+1). Os testes computacionais mostraram que os melhores resultados são obtidos fazendo as perturbações (3+1+2), ou seja, as soluções são perturbadas aplicando as perturbações 3, 2 e 1, nesta ordem.

Após definir a melhor maneira de perturbar uma solução, o algoritmo foi executado utilizando diferentes valores para o número de iterações. Os valores utilizados para *niILS* foram variados de 10 a 150. Os testes mostraram que, para as instâncias com 20, 31, 42, 53, 60 e 85 sensores o número de iterações necessárias para obter os melhores resultados foram 30, 70, 90, 70, 90 e 90 iterações, respectivamente. Na Tabela 1 apresentam-se os resultados contendo o melhor número de iterações (*niILS*), os valores médios do consumo de energia, da falha de cobertura, da quantidade de nós sensores ativos e do tempo gasto pelo algoritmo ILS-Multi-Período para encontrar a solução.

Na Tabela 1 observa-se que as configurações das redes tiveram em média um baixo consumo de energia, cobrindo uma maior área, desta forma minimizando a falha de cobertura em um baixo tempo computacional. Conforme observado nas três primeiras colunas da Tabela 1, quanto maior é o tamanho do problema, é necessário um número maior de iterações para conseguir uma boa solução.

Tabela 1 – Resultados para os diferentes números de iterações.

Problema	Melhor Nº de Repetições	Média - Consumo de Energia	Média - Falha de Cobertura	Média - Nós utilizados	Média do Tempo (s)
20 nós	30	2750,013	20,31	11,68	0,0232
31 nós	70	3256,813	2,25	15,43	0,1597
42 nós	90	3297,463	1	15,87	0,3195
53 nós	70	3140,588	0	15,62	0,3507
60 nós	90	2891,900	0	14,06	0,5067
85 nós	90	2696,213	0,25	13,50	0,9005

4.2. Comparação dos Resultados das Heurísticas ILS-Multi-Período e GRASP

Os desempenhos das heurísticas ILS-Multi-Período e GRASP de Andrade e Nakamura (2009) são avaliados mediante a comparação com os resultados ótimos obtidos pelo software CPLEX que resolve o modelo matemático do problema. Os métodos foram comparados utilizando 4 medidas de desempenho, quais sejam: valores médios do consumo de energia, da falha de cobertura, da quantidade de sensores utilizados e o tempo computacional.

Tabela 2. Comparação dos métodos utilizando com quatro medidas de desempenho.

Problemas	Métodos	Média Energia	Média Falha de Cobertura	Média Qtdade de Nós	Média Tempos (seg.)
20 nós	CPLEX	2577,57	24,35	11,50	30,26
	ILS-M-P	2507,07	26,71	10,87	0,04
	GRASP	2765,08	28,65	11,73	0,35
31 nós	CPLEX	3150,67	2,41	14,64	18,33
	ILS-M-P	3238,70	2,41	15,10	0,15
	GRASP	3233,58	2,41	14,60	2,50
42 nós	CPLEX	3084,17	0,75	14,20	287,56
	ILS-M-P	3213,62	0,75	14,95	0,26
	GRASP	3186,00	0,75	14,43	4,98
53 nós	CPLEX	2948,46	0,00	14,08	1.907,68
	ILS-M-P	3056,77	0,00	14,72	0,40
	GRASP	3132,55	0,00	14,35	8,43
60 nós	CPLEX	2834,46	0,16	13,50	4.745,10
	ILS-M-P	2952,55	0,16	14,18	0,48
	GRASP	3058,94	0,00	13,85	11,33
85 nós	CPLEX	2727,66	0,33	13,58	14.958,66
	ILS-M-P	2861,10	0,16	14,10	0,90
	GRASP	3037,15	0,00	13,92	32,57

Na Tabela 2, para cada grupo de problema, são apresentados os valores médios (resultantes de 12 instâncias) das 4 medidas. Observa-se que, considerando o consumo médio de energia, a heurística ILS-M-P foi melhor que o GRASP em 4 grupos de problemas, e o GRASP melhor que o ILS-M-P para dois grupos. Considerando a média da falha de cobertura, o GRASP foi melhor que o ILS-M-P para 2 grupos de problemas, e para 3 grupos de problemas, os dois algoritmos geraram em média a mesma falha. Também se observa que o GRASP gerou configurações de redes que utiliza em média um número menor sensores. Comparando com as configurações obtidas pelo método ILS-M-P, a diferença de nós utilizados é muito pequena. O algoritmo ILS-M-P proposto neste trabalho obteve resultados bastante competitivos aos resultados do GRASP. No entanto, os tempos gastos pelo ILS-M-P são bem menores quando comparados aos tempos do GRASP.

Na Tabela 2, observa-se que as soluções ótimas obtidas pelo CPLEX, não necessariamente possui os melhores valores para o consumo de energia, falha de cobertura, e quantidade de sensores

utilizados. Por exemplo, para o grupo de problemas com 20 nós, o consumo médio de energia e o número médio de sensores para o ILS-M-P são menores em comparação ao CPLEX. O tempo computacional gasto pelo CPLEX para obter a solução ótima cresce consideravelmente com o incremento do tamanho do problema.

Para as 72 instâncias testadas, na Tabela 3 mostram-se as médias do desvio relativo (*gap*) das heurísticas considerando o consumo de energia. Este desvio é calculado da seguinte maneira:

$$gap_i = \frac{f_i^{Alg} - f_i^*}{f_i^*} \quad (3)$$

Onde f_i^{Alg} é o valor do consumo de energia obtido por um algoritmo heurístico e f_i^* é o consumo de energia obtido pelo CPLEX, para um determinado problema i .

Observa-se na Tabela 3, que o método ILS-M-P apresentou os melhores *gap* em 59 instâncias, e o GRASP em 12 instâncias, ficando com o resultado de 1 instância em empate. Estes resultados mostram que, considerando o critério de consumo de energia, a heurística proposta ILS-M-P apresenta os melhores resultados.

Tabela 3. Comparação entre os métodos ILS-M-P e GRASP através do consumo de energia (*gap*).

<i>n</i>	Métodos	Instâncias											
		1	2	3	4	5	6	7	8	9	10	11	12
20	ILS-M-P	0	0,01	0,325	0,066	-0,042	0,082	-0,172	-0,216	-0,217	0,016	-0,045	-0,171
	GRASP	0,37	0,12	0,292	0,098	-0,233	0,031	0,322	-0,199	0,013	0,054	0,094	0,013
31	ILS-M-P	0	0,044	1,51E	0,047	0,013	0,008	0,006	0,071	0,069	0,035	0,015	0,001
	GRASP	0	0,066	1,51E	0,046	0,014	0,003	-0,001	0,032	2,65E	0,003	0,106	0,032
42	ILS-M-P	0,031	0,086	0,039	0,015	0,028	0,014	0,031	0,086	0,039	0,015	0,028	0,014
	GRASP	0,052	0,068	0,044	0,019	0,081	0,026	0,052	0,068	0,044	0,019	0,081	0,026
53	ILS-M-P	0,072	0,023	0,042	0,042	0,003	0,011	0,020	0,046	0,001	0,074	0,022	0,081
	GRASP	0,128	0,047	0,058	0,064	0,059	0,025	0,087	0,057	0,050	0,099	0,033	0,042
60	ILS-M-P	0,046	0,018	0,081	0,114	0,073	0,034	0,003	0,031	0,036	0,041	0,017	0,003
	GRASP	0,086	0,052	0,091	0,123	0,080	0,095	0,089	0,071	0,095	0,044	0,104	0,022
85	ILS-M-P	0,026	0,005	0,048	0,023	0,085	0,038	0,095	0,020	0,014	0,141	0,028	0,024
	GRASP	0,047	0,186	0,186	0,076	0,171	0,089	0,066	0,104	0,119	0,142	0,137	0,056

5. Conclusões

Neste trabalho foi abordado um problema bastante importante e interessante na área de Redes de Sensores: o problema de controle de densidade, cobertura e roteamento multi-período. Para tratar este problema foi proposto um método simples e eficiente baseado na heurística *Iterated Local Search*.

A heurística desenvolvida foi testada em um conjunto de 72 instâncias do problema. Os resultados obtidos foram comparados com os resultados ótimos apresentados pelo software CPLEX e com as soluções aproximadas determinadas por uma heurística GRASP proposta na literatura. A heurística ILS-Multi-Período, em geral, apresentou um bom desempenho em termos da qualidade de configurações de redes obtidas e principalmente em relação ao tempo computacional gasto para resolver os problemas.

Agradecimentos:

Os autores agradecem ao CNPq, FAPEMIG e FUNARBE pelo financiamento deste trabalho.

5. Referências

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002). *A survey on sensor networks*, IEEE Communications Magazine, pages 40, pp. 102-114.
2. Andrade, I. B. D., Januario, T. O., Papa, G. L., Mateus, G. R. (2010). An Evolutionary Algorithm to the Density Control, Coverage and Routing Multi-Period Problem. I in: *Wireless Sensor Networks*, IEEE.

3. **Andrade, I. B. D., Nakamura, F. G.** (2009) Uma heurística GRASP para o Controle de Densidade: Resolvendo o Problema de Cobertura e Roteamento Multi-Período em Redes de Sensores Sem Fio Plana. In: *XLI SBPO*.
4. **Andrade, I. B. D., Nakamura, F. G., Mateus, G. R.** (2008). Uma abordagem multi-período para a solução do problema de cobertura e conectividade em redes de sensores sem fio planas. In *XL Simpósio Brasileiro de Pesquisa Operacional*.
5. **Cardei, M., and Wu, J.** (2006). Energy-efficient coverage problems in wireless ad-hoc sensor networks. In *Computer Commun.*, volume 29, pages 413–420.
6. **Cugnasca, C. E.** Redes de Sensores Sem Fio. In: *Congresso Brasileiro de Agricultura de Precisão (ConBAP)*, Ano 2008.
7. **Estrin, D., Govindan, R., Heidemann, J., Kumar, S.** (1999). Next century challenges: Scalable coordination in sensor networks., In: *J. Heidemann (Ed.), Proc. of the 5th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking, Seattle, USA*, pp. 263-270.
8. **Ian F. A., Weilian, S., Yoges,h S. end Erdal, C.** A Survey on Sensor Networks. In: *IEEE Communications Magazine*, Ano 2002.
9. **Ilan Sousa, Lauro Américo, Lilian Freitas, Aldebaro Klautau e João Costa.** ListenU uma Ferramenta para Monitoramento Ambiental Usando Redes de Sensores Sem Fio. In: http://www.lbd.dcc.ufmg.br/colecoes/ein/2011/Artigo_13.pdf : 2011.
10. **Lourenço, H. R.,; Martin, O.,; Stützle, T.** Iterated Local Search. In: *F. GLOVER, F., KOCHENBERGER, G., editors, Handbook of Metaheuristics*, p. 321–353. Kluwer Academic Publishers, Norwell, MA, 2002.
11. **Martins, F. V. C.** Heurísticas Mono e Multiobjetivo Para o Problema de Cobertura e Conectividade de Redes de Sensores Sem Fio Planas. Dissertação de Mestrado, UFMG, Acessado em Agosto. 2009.
12. **Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.** (2003). Coverage problems in wireless ad-hoc sensor networks, In: *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage.*, pp. 1380-1387.
13. **Murthy, C. S. R., Manoj, B. S.** (2004). Ad Hoc Wireless Networks. In: *Architectures and Protocols*, 1st ed., Prentice Hall.
14. **Nakamura, F. G.** Planejamento Dinâmico para Controle de Cobertura e Conectividade em Redes de Sensores Sem Fio Plana. Dissertação de Mestrado UFMG, 2003. Acessado em Setembro 2009
15. **Nakamura, F. G., Quintão, F. P., Menezes, G. C., Mateus, G. R.** (2005). An optimal node scheduling for flat wireless sensor networks. In 4th International Conference on Networking - Lecture Notes in Computer Science, volume 3420, pages 104–483.
16. **Quintão, F., Nakamura, F. G., Mateus, G. R.** (2004). A hybrid approach to solve the coverage and connectivity problem in wireless sensor networks. In: *In Proceedings of the IV European Workshop on Metaheuristics (EU/ME04), volume 1.*
17. **Romer, K., Mattern, F.** (2004). The design space of wireless sensor networks. In: *IEEE Wireless Communications*, pages 11 (6) 54–61.
18. **Siqueira, I. G., Figueiredo, C. M. S., Nogueira, A. A. F. L. J. M. S., Ruiz, L. B.** (2006). An integrated approach for density control and routing in wireless sensor networks. In *20th IEEE International Parallel and Distributed Processing Symposium (IPDPS'06)*.
19. **Srivastava, Mm. B., Mmuntz, Rr. R., e Ppotkonjak, Mm.** Wireless networks for smart developmental problem-solving environments. In: *Mobile Computing and Networking*, pages 132–138. Smart kindergarden: sensor-based, 2001.
20. **Y. Gao, K. Wu, and F. Li.** Analysis on the redundancy of wireless sensor networks. In *Proceedings of ACM WSNA'2003*, volume 29, pages 108–114, Feb. 2003.
21. **Zhang, H. and Hou, J.** Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2). 2005.