

HEURÍSTICA ILS PARA MINIMIZAÇÃO DO CONSUMO DE ENERGIA EM REDES DE SENSORES SEM FIO

Brener Joabe Baião da Silva
Universidade Federal de Viçosa
Avenida Peter Henry Rolfs s/n - VIÇOSA, MG.
brener.silva@ufv.br

Luciana Brugiolo Gonçalves, José Elias Claudio Arroyo
Universidade Federal de Viçosa - MG
Avenida Peter Henry Rolfs s/n - VIÇOSA, MG.
lbrugiolo@ufv.br, jarroyo@dpi.ufv.br

RESUMO

Devido a avanços recentes relacionados às tecnologias associadas às redes de sensores sem fio (RSSF), este tipo de estrutura tem recebido especial atenção por parte de muitos pesquisadores. Características particulares destas redes, como restrição de energia dos nós e dificuldade de acesso aos sensores, tornam desafiador o gerenciamento destas estruturas. Uma das técnicas utilizadas para minimizar o consumo de energia é a clusterização. Neste trabalho é apresentada uma heurística fundamentada no *Iterated Local Search* (ILS) para minimizar o consumo de energia com objetivo de prolongar o tempo de vida da rede. Foi desenvolvido um algoritmo para simular o comportamento e transmissões em uma RSSF. Através de experimento, foi possível verificar que os resultados obtidos pelas heurísticas propostas são relevantes se comparados com heurísticas da literatura, como LEACH e LEACH-C. Quando comparado com heurísticas mais recentes, como o GRASP, a heurística proposta foi competitiva, mostrando um tempo médio de execução inferior. Os resultados mostram que a heurística desenvolvida determinou soluções que prolongam o tempo de vida da rede.

PALAVRAS CHAVE. Otimização Combinatória, Rede de sensores sem fio, Clusterização, Metaheurísticas.

Pesquisa Operacional - Ciência da Computação

ABSTRACT

Due the recent progress related to technology associated to wireless sensor networks (WSN), that kind of structure has received special attention by many researchers. Particular characteristics of these networks, such as energy restriction of nodes and difficulty of accessing the sensors, turn challenging the management of these structures. A technique used to minimize the energy is clustering. It was showed on this paper a heuristic based on *Iterated Local Search* (ILS) with different local searches and perturbation. An algorithm was developed to simulate the behavior and transmissions of WSN. It was possible throughout the experiment to verify that the results obtained by the proposed heuristics are compared to relevant literature heuristics, like LEACH and LEACH-C. If compared to more recent heuristics, such as GRASP, the proposed heuristic was competitive showing a medium time of inferior executions. The results show that the developed heuristic could increase network's lifetime.

KEYWORDS. Combinatorial Optimization, Wireless Sensor Network, Clustering, Metaheuristics.

Operational Research - Computer Science

1. Introdução

Devido a avanços recentes relacionados às tecnologias associadas às redes de sensores sem fio (RSSF), este tipo de estrutura tem recebido especial atenção por parte de muitos pesquisadores, Loureiro *et al.* (2003). As redes de sensores sem fio podem ser vistas como uma classe especial da rede *ad hoc* onde os nodos comunicam-se diretamente, sem a necessidade de dispositivos intermediários, chamados de ponto de acesso, Pinheiro (2005). Os sensores, que são os nodos na rede de sensores sem fio, têm como objetivo monitorar o ambiente e transmitir os dados coletados através de mensagens ao *sink*, também conhecido como sorvedouro. O *sink* é um nodo especial, normalmente possui energia ilimitada sendo o responsável pelo processamento dos dados. As RSSF possuem um grande número de sensores, tem restrição de energia e devem ser capazes de autoconfiguração em relação à perda de sensores e falhas de comunicações.

Geralmente, os sensores são distribuídos em áreas remotas e de difícil acesso, o que inviabiliza a manutenção. Sendo assim, como mencionado por Ruiz (2012), os sensores devem ser capazes de se readaptarem a mudanças de topologia e roteamento na rede.

Em muitas aplicações, para minimizar o custo, os sensores não são colocados de forma ordenada, são simplesmente lançados no local a ser monitorado. Como por exemplo, o monitoramento de um vulcão ativo, por ser uma área de grande risco, uma opção pode ser lançar os sensores por um avião. Espera-se que os sensores sejam capazes de se auto-organizarem para formarem uma rede, Loureiro *et al.* (2002).

As redes de sensores sem fio possuem muitas aplicações, geralmente empregados para monitorar algum fenômeno específico. Como exemplo dessa ampla área de pesquisa pode-se citar aplicações na biomedicina, associado à coleta de informações sobre o organismo humano; para fins militares, na detecção de radiação, ataques biológicos e químicos; nas áreas de desastres naturais, para monitoramento das queimadas; entre outras (Ribeiro e Silva, 2007).

Os sensores sem fio geralmente possuem, além da capacidade de sensoriamento, capacidade para processamento e comunicação. Numa rede de sensor sem fio, além de coletar dados do ambiente, os sensores podem exercer outras funções, como a de processar dados, receber mensagens, fazer o roteamento do dado para outro sensor ou enviá-los ao *sink*.

Um sensor, além de possuir um ou mais dispositivos de sensoriamento, possui um módulo para comunicação sem fio (transceptor), um processador, memória e uma bateria responsável pela alimentação dos componentes de hardware.

Os sensores podem se diferenciar em relação a características como tamanho, peso, alcance do sinal para monitoramento, alcance do sinal de transmissão de mensagens, capacidade da bateria, modelo de consumo de energia, poder de processamento, frequência de coleta de informações, volume de dados, quantidade de memória, acoplamento de hardwares e protocolos de comunicação. Diferentes eventos podem ser monitorados através de sensores acústicos, infravermelho, câmera, temperatura, luz, calor e sísmico, entre outros (Barros, 2002).

Uma abordagem muito utilizada nas RSSF para minimização do consumo de energia é a clusterização (agrupamento). Por se tratar de um recurso escasso, a economia de energia visa prolongar o tempo de funcionamento da rede. Na clusterização, os sensores são organizados em grupos. Para cada grupo de sensores é eleito um sensor líder, denominado *cluster head*. Considerando os sensores organizados desta forma, após receberem a informação para qual sensor enviar (*sink* ou líder eleito), a mensagem de cada sensor é direcionada para o líder de seu grupo que irá encaminhá-la para o *sink*. Assim, em vez de cada sensor enviar uma mensagem ao *sink*, o que consome muita energia, apenas o *cluster head* se comunica diretamente com o mesmo. Cada *cluster head*, após receber as mensagens dos sensores de seu grupo, faz a agregação dessas informações, considerando também a informação que este sensorizou, gerando uma única mensagem que é enviada ao *sink* (ver Figura 1). Desta forma, reduz-se o número de mensagens que trafegam pela rede, reduzindo o consumo de energia de uma forma geral. A forma como os sensores são agrupados e a definição dos sensores que assumirão o papel de *cluster heads* são aspectos muito importantes para a minimização da energia, Heinzelman *et al.* (2000). Desta forma, neste trabalho são propostas abordagens heurísticas para a minimização do consumo de energia baseadas em clusterização usando *Iterated Local Search* (ILS).

Na literatura já existem algumas abordagens que tratam o problema de RSSF através da clusterização. Um dos primeiros protocolos que abordou o problema sobre este ponto de vista foi o LEACH, Heinzelman *et al.* (2000). O LEACH trabalha com rotatividade aleatória de *cluster heads*, dando uma probabilidade maior de escolha do *cluster heads* aos nós com mais energia.

A partir do LEACH, novas propostas surgiram com alguns aperfeiçoamentos a partir da ideia inicial, como o LEACH-C (LEACH-Centralized), Heinzelman *et al.* (2002). O LEACH-C tem os mesmos princípios de funcionamento do LEACH, porém o nodo principal do grupo é centralizado e eleito uma só vez. A escolha do nodo principal se dá com base nos que tem mais energia disponível. Desta forma, garante-se uma dissipação de energia mais uniforme pela área de distribuição dos nós.

Entre as abordagens mais recentes, o GRASP (*Greedy Randomized Adaptive Search Procedure*) de Matos *et al.* (2012) apresenta uma proposta que obteve resultados melhores, quando comparado com LEACH e LEACH-C. O GRASP é um método iterativo que consiste basicamente de duas fases, a primeira fase é a construtiva, que cria a solução inicial passo a passo, de modo guloso e aleatório. A segunda fase é a busca local que aperfeiçoa a solução corrente. Nesta fase é utilizado a técnica do *Path Relinking* que combina duas soluções com objetivo de encontrar soluções melhores.

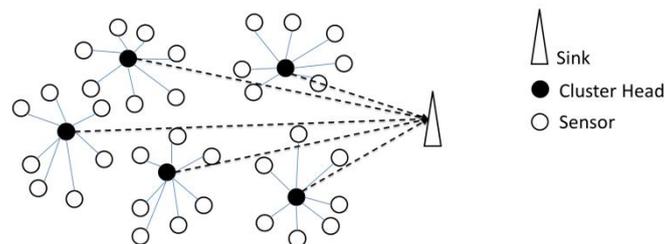


Figura 1 - Exemplo de Clusterização em RSSF

A heurística ILS (Lourenço *et al.*, 2002) proposta neste trabalho tem como característica o uso de vizinhança para melhorar a solução, usando busca local e perturbação.

Os resultados encontrados neste trabalho são comparados com a heurística GRASP de Matos *et al.* (2012).

O restante deste trabalho está organizado da seguinte forma. Na Seção 2 é apresentada uma descrição detalhada do Problema de Minimização do Consumo de Energia em RSSF. Na Seção 3 é descrita a heurística ILS proposta. Os resultados são apresentados na Seção 4 e conclusões e trabalhos futuros na Seção 5.

2. Problema de Minimização do Consumo de Energia em RSSF

Neste trabalho, considera-se como objetivo a maximização do tempo de funcionamento de uma RSSF. Para tal, como a energia é um recurso limitado, a economia no consumo deste recurso é um fator importante. O gasto de energia de um sensor depende de vários fatores como tipo do rádio, processador, dispositivo de sensoriamento, tipo de bateria, características do ambiente a ser monitorado e tamanho da mensagem que trafega na rede. Nesta seção é definido o modelo de comunicação a ser considerado pelos algoritmos desenvolvidos.

Entre as atividades desempenhadas pelos sensores, o rádio transmissor é o dispositivo que consome mais energia, se comparada com o consumo dos demais dispositivos, Loureiro *et al.* (2003). Como pode ser observado na Figura 2, existe um custo associado ao envio e também à recepção de mensagens. As características do ambiente, o tamanho da mensagem e a distância entre os sensores que se comunicam são fatores que afetam diretamente a quantidade de energia gasta em uma troca de mensagens.

Em geral, o custo da comunicação entre estes sensores depende da distância euclidiana entre eles, do tamanho da mensagem (número de *bits* - k) e de um fator associado à atenuação do sinal, que varia conforme as características do ambiente. O fator que está associado à presença de obstáculos, atribui-se valores entre 2 e 6 (na Figura 2 está representado pela variável n).

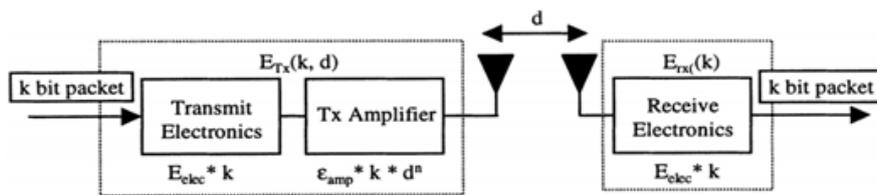


Figura 2- Modelo de dissipação de energia, Heinzelman *et al.* (2002).

De forma análoga ao apresentado em Heinzelman *et al.* (2002), o modelo de dissipação de energia considerado neste trabalho avalia o consumo para ativar o rádio transmissor/receptor (E_{elec}), o consumo associado ao circuito amplificador (ϵ_{amp}) e também o consumo para agregar mensagens (E_{BF}). As constantes que definem estes consumos dependem do sensor utilizado na rede. As Equações (1), (2) e (3) detalham o cálculo destes custos.

$$E_{Tih} = \ell E_{elec} + \ell \epsilon_{amp} \times d_{ih}^2 \quad (1)$$

$$E_R = \ell E_{elec} \quad (2)$$

$$E_g = \ell E_{BF} \quad (3)$$

Sendo E_{Tih} é a energia necessária para transmitir uma mensagem de ℓ bits entre um sensor i e um nodo h (sensor, *cluster head* ou *sink*) que está distantes d_{ih} unidades. E_R é a energia necessária para qualquer sensor receber uma mensagem de ℓ bits. Já E_g é a energia necessária para o *cluster head* agregar as mensagens de ℓ bits recebidas dos sensores. A Equação 3 mostra que o número de mensagens recebidas dos sensores é irrelevante para o gasto de energia na agregação. Assume-se que a comunicação é simétrica, isto é, a energia necessária para transmitir do sensor i para h é a mesma de h para i .

Neste trabalho, para viabilizar a comparação entre a abordagem proposta e a da literatura, são considerados os mesmos parâmetros utilizados em Matos *et al.* (2012). Estas taxas, usadas nas equações (1), (2) e (3), são apresentadas na Tabela 1.

Tabela 1 - Taxas

Operação	Taxa
Transmitir Receber (E_{elec})	50nJ/bit
Agregação (E_{BF})	5nJ/bit
Amplificar (ϵ_{amp})	10pJ/bit/m ²

O modelo de comunicação afeta diretamente o resultado dos algoritmos. Para diferentes valores, resultados variados são esperados. Para esse modelo de constantes, espera-se que o *sink* esteja longe dos sensores (Heinzelman *et al.*, 2000).

Como o sensor que assume o papel de *cluster head* terá um gasto maior de energia, devido às tarefas de agregação e envio de mensagem ao *sink*, para evitar o esgotamento prematuro da energia destes sensores é importante viabilizar o rodízio deste papel entre todos os nós da rede. Para definir o instante de reconfiguração da rede, a comunicação é dividida em sessões.

A cada intervalo de tempo, os sensores enviam ao *cluster heads* os dados obtidos a partir de seus dispositivos de sensoriamento. Ao receber as mensagens dos sensores de seu grupo, o *cluster head* realiza a agregação, gerando uma única mensagem que é então enviada ao *sink*. O envio deste conjunto de mensagens será referenciado como uma transmissão. Um conjunto de transmissões compõe uma sessão de comunicação. Para identificar o fim de uma sessão de comunicação, o *sink* deve receber um total de 1MB de dados, da mesma forma como definido em Matos *et al.* (2012), sendo que cada mensagem que trafega na rede possui 512 bytes.

Um ciclo completo, algoritmo de configuração + sessão de comunicação, será referenciado neste trabalho como sendo um *round*.

Como objetivo desse trabalho é, além de minimizar o custo de energia dos sensores, maximizar o número de transmissões, deve-se também maximizar e homogeneizar o tempo de

vida de todos os sensores, isto é, os sensores devem ficar sem energia ao mesmo tempo, ou num intervalo pequeno de tempo. Deste modo, quando alguns sensores ficarem sem energia, significa que a rede estará próxima de perder a conectividade, tornando-se inoperante.

3. O Algoritmo proposto

A heurística proposta neste trabalho é fundamentada no ILS, uma metaheurística baseada em busca local e perturbação, que se destaca por características como simplicidade de implementação, eficácia e robustez. No trabalho de Lourenço *et al.* (2002) são descritas implementações bem-sucedidas do ILS para vários problemas de otimização, como TSP, *Job Shop*, *Flow Shop* e MAX-SAT, entre outros.

Para determinar a melhor configuração da rede, considera-se a quantidade de energia residual de cada sensor, *Nuvem_Energia*, para definir o papel que cada nó irá exercer durante as transmissões. Como se espera que o *cluster head* irá consumir mais energia, devido ao consumo relacionado ao recebimento das mensagens dos membros do grupo e também devido a agregação, esse papel deve ser atribuído aos sensores com maior disponibilidade de energia. Pelo fato de o *cluster head* gastar mais energia que os outros sensores, a rotatividade do *cluster head* foi favorecida entre os sensores com mais energia residual.

É possível que um sensor envie a mensagem diretamente para o *sink*, caso seja mais vantajoso. Porém, esse evento é raro e só acontece quando a rede possui poucos sensores e o possível *cluster head* esteja distante, porém essa possibilidade não é excluída.

Além da busca local e perturbação, outra característica importante no ILS é a heurística de obtenção da solução inicial. No algoritmo 1, que descreve o funcionamento geral do ILS, é possível observar na linha 1 o uso de um procedimento construtivo que determina uma solução inicial. Na linha 2, é feita o refinamento dessa solução com o método *Busca_Local*(), que gera como resultado um ótimo local. Entre as linhas 3 e 9 tem-se as etapas iterativas da metaheurística ILS. A fim de escapar do ótimo local, na linha 4 é executado o método *Pertubação*() que consiste em alterar (perturbar) a solução corrente obtendo uma nova solução que é submetida a uma melhoria pelo procedimento de busca local (linha 5). Na linha 6, verifica-se o quanto de energia a nova solução gasta a menos que a atual, através da função *custo*(). Caso seja menor, salva-se a nova solução como a solução atual, linha 7.

Algoritmo 1 - ILS (*k*, *Nuvem_Energia*, *nPERT*, *nMAX*)

```

1: Sol ← Construtivo (k, Nuvem_Energia);
2: Sol ← Busca_Local_Interna (Sol);
3: enquanto (número de iterações sem melhora < nMAX) faça
4:     novaSol ← Pertubação (s, nPERT);
5:     novaSol ← Busca_Local_Interna (novaSol);
6:     se (novaSol.custo () < Sol.custo ()) então
7:         Sol ← novaSol;
8:     fim se
9: fim enquanto

```

Nas próximas subseções os métodos da heurística ILS são detalhados.

3.1. Método Construtivo

O objetivo da heurística de construção é gerar uma solução viável, ou seja, uma configuração válida para a rede. Assim, busca-se eleger os sensores que atuarão como líderes (*cluster head*) e também definir a que grupo é associado cada um dos demais sensores.

Considere que se conhece a priori o número de *grupos* a serem formados, indicado pela constante *k*. Do ponto de vista da eficiência energética, estudos apontam que selecionar 5% dos sensores para *cluster heads* leva a resultados de boa qualidade, Heinzelman *et al.* (2002). Este será o valor de *k* utilizado neste trabalho.

A estratégia de seleção dos k *cluster heads* foi baseada no algoritmo da literatura *k-means*, MacQueen (1967), com o auxílio da heurística Mapa Grupo.

No Mapa Grupo, a região onde se encontram os sensores é dividida em subespaços, onde o objetivo é identificar regiões mais densas, ou seja, regiões com a presença de um maior número de sensores. Para o problema de otimização tratado, o número de subespaços a serem criados é definido considerando-se o número de grupos a serem formados. Definida a forma de divisão da área e o formato das regiões monitoradas (áreas quadradas), para determinar o número de partições será considerada a Equação 4.

$$divisaoLateral = \lceil \sqrt{numClusterHeads} \rceil \quad (4)$$

Esta equação indica que cada lateral da área (lateral do quadrado) deve ser dividida em $divisaoLateral$ partes, o que vai gerar $(divisaoLateral)^2$ subespaços, o menor quadrado perfeito maior que o número de *cluster heads*.

Com o Mapa Grupo é possível identificar quais dos subespaços gerados possuem maior densidade (número de sensores). Assim, os centroides iniciais para o *k-means* são selecionados de forma aleatória entre aqueles sensores pertencentes aos subespaços de maior densidade. Para identificar as regiões mais densas são considerados apenas aqueles sensores que possuem energia residual maior que a média da energia dos sensores da rede.

Considerando que os sensores também consomem energia para receber uma mensagem, dependendo da constante E_{elec} (ver Tabela 1 e Equação 2), pode ser mais vantajosa a comunicação direta entre os sensores e seus líderes. Na comunicação indireta (ou multissaltos), os sensores não enviam as mensagens diretamente para os *Cluster Heads*, mas para outros sensores, que direcionam a mensagem para o *Cluster Head* posteriormente. Assim, a comunicação direta é outro método utilizado neste trabalho para a comunicação *IntraCluster*. Para cada sensor, sua mensagem é enviada diretamente para o *cluster head* mais próximo.

Do mesmo modo que a comunicação *intraCluster*, ou seja, dos sensores para os líderes, a comunicação dos líderes para o *sink* é feita de forma direta, isto é, os líderes transmitem diretamente para o *sink*.

O *k-means* é um algoritmo fácil de implementar e muito utilizado no pré-processamento para obtenção de soluções iniciais (Bandyopadhyay e Maulik, 2002). É um método de particionamento que tem como entrada um conjunto de pontos e o parâmetro k , que indica o número de *clusters* (grupos) a serem formados. Este método busca, iterativamente, agrupar o conjunto de pontos em k *clusters* de forma a otimizar uma função de custo, normalmente chamada de função de similaridade. Neste método, cada *cluster* é representado pelo seu centroide, sendo que o centroide de um *cluster* é o ponto onde suas coordenadas são obtidas pela média das coordenadas de todos os pontos pertencentes ao grupo. No *k-means*, a função de custo a ser minimizada considera a média das distâncias entre cada ponto e seu centroide.

A heurística construtiva proposta neste trabalho baseia-se na ideia central do *k-means*. A diferença principal é na etapa de seleção dos centroides iniciais, que no *k-means* é realizada de forma aleatória e na heurística proposta utiliza-se o método Mapa Grupo. O Algoritmo 2 descreve as principais etapas desta abordagem.

A solução obtida pelo algoritmo construtivo depende da seleção dos centroides iniciais. Escolhas inadequadas podem guiar o algoritmo para ótimos locais distantes da melhor solução. Com objetivo de indicar centroides iniciais que levem a soluções de melhor qualidade, foi utilizado o algoritmo Mapa Grupo, um método baseado em grades e também em densidade (Ferreira *et al.*, 2010).

Tendo como entrada a quantidade de grupos a serem formados (k) e a energia residual dos sensores (Nuvem_Energia), a primeira etapa do algoritmo é, através da função Mapa_Grupo(), ordenar as regiões de acordo com o número de sensores com energia residual acima de média, linha 1. Em seguida, seleciona-se aleatoriamente um sensor de cada região para ser o centroide inicial, sendo CT os centroides escolhidos, linha 2. Os demais sensores são associados ao centroide que estiver mais próximo, formando assim os k grupos. Após a definição dos centroides iniciais, o centroide é recalculado, linha 4, e os demais sensores são novamente

redistribuídos de acordo com a nova configuração. Este processo continua até que, após a redistribuição dos sensores, não haja alteração na definição dos *cluster heads*, linhas 5 a 11.

Algoritmo 2 - Construtivo (k , $Nuvem_Energia$)

```

1:  $Regiões \leftarrow$  Mapa_Grupo ( $k$ ,  $Nuvem\_Energia$ );
2:  $CT \leftarrow$  Sensores_Aleatórios ( $Regiões$ );
3:  $Sol \leftarrow$  Comunicação_IntraCluster ( $CT$ );
4:  $CT' \leftarrow$  Centroides ( $Sol$ );
5: enquanto (houve mudança) faça
6:      $Sol' \leftarrow$  Comunicação_IntraCluster ( $CT'$ );
7:     se ( $CT \neq CT'$ ) então
8:          $CT \leftarrow CT'$ ;
9:          $CT' \leftarrow$  Centroides ( $Sol'$ );
10:    fim se
11: fim enquanto
12:  $Sol \leftarrow$  CH_Centroides ( $Sol'$ );

```

Quando, após uma iteração, não houver mais atualização nos centroides, os sensores com energia suficiente para atuarem como *cluster heads* que estiverem mais próximos aos centroides serão escolhidos como *cluster heads*, ver linha 12.

3.2. Busca Local

A busca local é utilizada para aperfeiçoar uma solução. De forma iterativa, a cada etapa a solução corrente é substituída por uma solução de melhor custo de sua vizinhança. Na heurística de busca local proposta neste trabalho verifica-se, em cada *cluster*, se há algum sensor que, assumindo o papel de *cluster head*, reduza o consumo de energia em uma transmissão. O Algoritmo 3 ilustra o funcionamento desta abordagem.

Algoritmo 3 - Busca_Local_Interna (Sol)

```

1:  $Sol' \leftarrow Sol$ ;
2: enquanto (melhorar) faça
3:     para (cada Cluster Head  $ch \in Sol.CH$ ) faça
4:         para (cada Sensor  $s \in S \mid s.ch = ch$ ) faça
5:              $CH' \leftarrow$  SwapInterno ( $ch, s$ );
6:              $novaSol \leftarrow$  Comunicação_IntraCluster ( $CH'$ );
7:             se ( $novaSol.custo () < Sol'.custo ()$ ) então
8:                  $Sol' \leftarrow novaSol$ ;
9:         fim se
10:    fim para
11:     $Sol \leftarrow Sol'$ ;
12: fim para
13: fim enquanto

```

A heurística recebe uma solução como parâmetro de entrada, assim tenta-se alterar o líder de cada um dos grupos, buscando soluções onde o consumo de energia será menor. Considere $Sol.CH$ o conjunto dos *cluster heads* da solução Sol e $s.ch$ o *cluster head* do grupo ao qual o sensor s pertence.

Para cada *cluster*, tenta-se trocar seu líder ch por um sensor s pertencente ao conjunto de sensores S e que tenha ch como líder. A função *SwapInterno* (ch, s) faz a substituição do líder atual ch por um sensor s pertencente ao grupo, linha 5. O conjunto dos *cluster heads* $Sol.CH$ com a substituição do *cluster head* ch pelo líder s é armazenado em CH' . Na linha 6 é refeita a comunicação interna no *cluster*, onde cada sensor direciona sua mensagem para o *cluster head* mais próximo. Na linha 7 verifica-se se esta alteração gerou um vizinho de melhora. Caso

afirmativo, a solução encontrada é salva numa variável auxiliar Sol' e, somente após todas as soluções vizinhas serem analisadas, efetua-se a mudança para a melhor solução encontrada, como pode ser observado na linha 11. Quando verificados todos os sensores de cada *cluster* e não houver melhora, a Busca Local Interna termina.

3.3. Perturbação

Uma característica fundamental para o bom funcionamento do ILS é a estratégia de perturbação. Caso esta seja pequena, a busca local pode retornar à solução anterior à perturbação, desfazendo a operação e fazendo com que poucas soluções sejam exploradas. Agora, se a alteração for muito grande, perde-se a relação com a solução de partida e o algoritmo poderá se comportar como uma abordagem do tipo *multistart*.

Na implementação avaliada neste trabalho, optou-se por utilizar um número $nPERT$ de movimentos de troca, ou seja, $nPERT$ grupos escolhidos aleatoriamente terão seu líder alterado para um outro sensor do grupo com energia acima da média, escolhido aleatoriamente, gerando uma nova configuração para a solução.

4. Resultados Computacionais

Os algoritmos apresentados neste trabalho foram implementados em C++ e executados num computador Intel® Core™ i5-3210M de 2.50GHz com 6GB de memória RAM no sistema operacional Windows 8 de 64 bits.

Para avaliação das heurísticas desenvolvidas, a simulação foi dividida em duas etapas: definição da estrutura da rede, que consiste da solução obtida pelo ILS, e transmissão de mensagens. Nestas abordagens, considerando o status atual da rede (energia residual em cada sensor), definem-se os sensores que assumem o papel de *cluster head* e os sensores membros de cada grupo. Além disso, é necessário determinar o caminho a ser percorrido pelas mensagens de cada sensor para o *cluster head* e também dos *cluster heads* para o *sink*.

Antes de cada transmissão, é verificado se os *cluster heads* terão energia suficiente para processar as mensagens e enviar a mensagem resultante da agregação ao *sink*. Deste modo, se garante que todas as mensagens que chegarem aos *cluster heads* serão enviados ao *sink*.

Após a definição dos grupos e respectivos caminhos das mensagens, inicia-se a etapa de troca de mensagens. Nesta fase, de acordo com a configuração da rede, simula-se o seu funcionamento. São realizadas várias transmissões com a configuração obtida até que o *sink* receba um total de 1MB de dados.

Durante a simulação, um sensor será considerado como morto/inoperante quando não for capaz de participar em 3 sessões de comunicação (ou *rounds*) consecutivas. Desta forma, mesmo que um sensor não tenha energia suficiente para transmitir sua mensagem na iteração atual, este sensor terá a chance de participar na sessão seguinte. Dependendo dos *cluster heads* selecionados na próxima iteração, ainda será possível que sua mensagem seja transmitida para o *sink*. Quando o número de sensores operantes da rede ficar abaixo de 5% do número total de sensores, esta será considerada morta, o critério de parada da simulação.

Para testar as heurísticas propostas foram criadas 10 instâncias, definidas numa área quadrada de 100m x 100m onde os sensores foram espalhados de forma aleatória, similar ao realizado por Matos *et al.* (2012). O *sink* foi posicionado a 100m do limite da área sensorizada, sendo que qualquer sensor consegue alcançá-lo. Entre as 10 instâncias, 5 possuem 100 sensores e 5 possuem 200 sensores. Todos os sensores tem energia inicial de 1J e devem participar de cada sessão de comunicação, ou seja, os sensores sempre têm algum dado para transmitir e cada mensagem possui sempre um tamanho padrão de 512 bytes.

Os parâmetros do algoritmo ILS foram definidos através de testes empíricos. O número de iterações ($nMAX$) e o número de movimentos realizados na perturbação ($nPERT$) foram fixados em 5 e 2, respectivamente. Vale ressaltar que o algoritmo ILS é executado (para determinar uma configuração da rede) a cada *round* da simulação.

Inicialmente avalia-se o desempenho do algoritmo ILS em relação à solução inicial determinada pelo método construtivo. Na Tabela 2 apresenta-se o percentual de melhoria (ganho)

do ILS em relação à heurística construtiva. Nesta comparação, para uma execução, foi determinada a média aritmética do consumo de energia das configurações utilizadas em cada um dos *rounds*. Então, nessa tabela observa-se a redução percentual no consumo de energia que o ILS obtém em relação à solução inicial, isto é, o quanto o ILS economiza de energia. Na última linha da tabela verifica-se que, em média, o ILS reduz o consumo em 2,75%, esse ganho em cada *round* pode representar um número maior de transmissões, prolongando a vida da rede.

Tabela 2 – Ganho do ILS em relação à solução inicial

Instância	Ganho do ILS
100-1	2,35%
100-2	2,45%
100-3	2,04%
100-4	2,99%
100-5	2,14%
200-1	2,20%
200-2	3,25%
200-3	3,27%
200-4	3,54%
200-5	3,26%
Média	2,75%

A Tabela 3 apresenta a média aritmética do total de transmissões em 10 execuções do algoritmo para cada uma das instâncias. Nesta tabela é possível observar que, quando se executa apenas a heurística de construção, o número de transmissões fica, em média, a 1,17% das transmissões obtidas pelo ILS, mostrando que com o ILS, o consumo de energia fica menor.

Tabela 3 - Total de Transmissões

Instância	Construtivo	ILS
100-1	214.606	216.247
100-2	215.003	216.699
100-3	213.245	215.810
100-4	208.497	212.246
100-5	216.945	217.747
200-1	458.758	463.016
200-2	433.811	439.302
200-3	433.868	439.237
200-4	432.754	437.388
200-5	431.597	439.508
Média	325.908	329.720

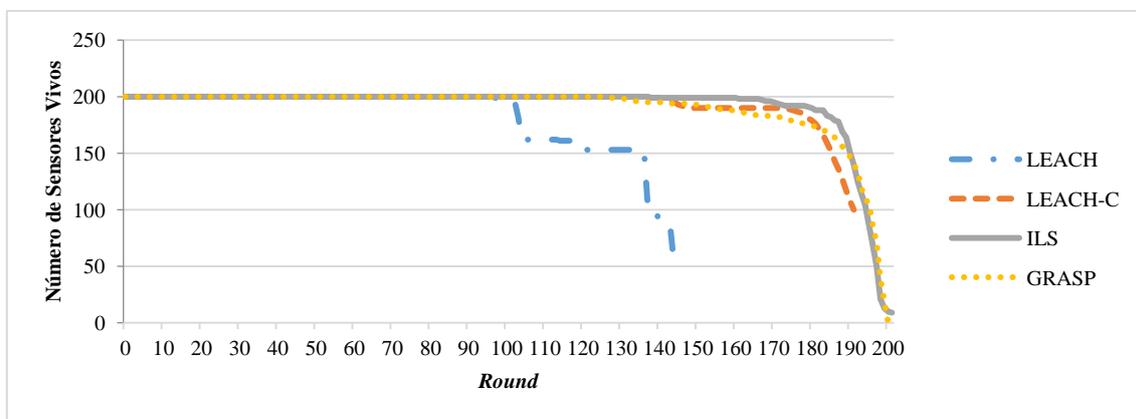
Em relação ao tempo de processamento, na Tabela 4 é apresentado o tempo médio de execução de cada heurística. Considerando o tempo médio, verifica-se que o tempo de processamento do método construtivo corresponde a 1,16% do tempo total consumido pelo ILS. Para uma comparação com as abordagens da literatura, será considerada apenas a heurística ILS.

Considerando o registro do número de sensores ativos em cada um dos *rounds* para uma execução dos algoritmos para a instância 200-5, no Figura 3 nota-se que a heurística proposta consegue manter um número maior de sensores vivos por mais tempo, o mesmo acontece com o GRASP. Já o LEACH e o LEACH-C, principalmente o LEACH, tem este número reduzido precocemente. Antes do centésimo *round* é possível observar que, utilizando o LEACH, alguns sensores já não estão mais ativos.

Tabela 4 - Tempo de processamento (em segundos)

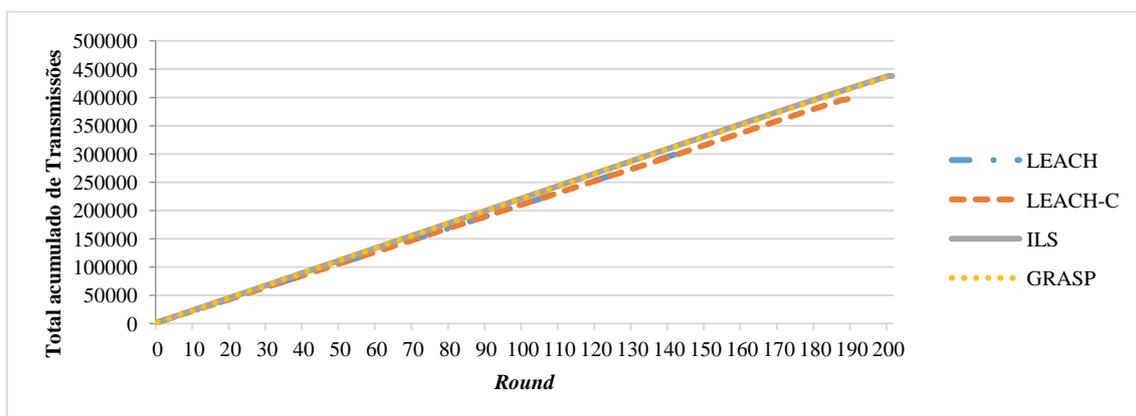
Instância	Construtivo	ILS
100-1	0,0019	0,0999
100-2	0,0019	0,0917
100-3	0,0019	0,1172
100-4	0,0019	0,1126
100-5	0,0015	0,0854
200-1	0,0086	0,8078
200-2	0,0078	0,7313
200-3	0,0078	0,7212
200-4	0,0081	0,7134
200-5	0,0081	0,7275
Média	0,0049	0,4208

Figura 3 - Número de Sensores Vivos por Round



A Figura 4 mostra, para cada heurística, o total de transmissões acumulado em cada *round* para a instância 200-5. Pode-se observar que o número de transmissões, quando comparados o ILS e o GRASP, se mantém equivalentes por praticamente todos os *rounds*, sendo que nos últimos *rounds* há uma diferença, onde a heurística proposta consegue transmitir um pouco mais de mensagens ao *sink*.

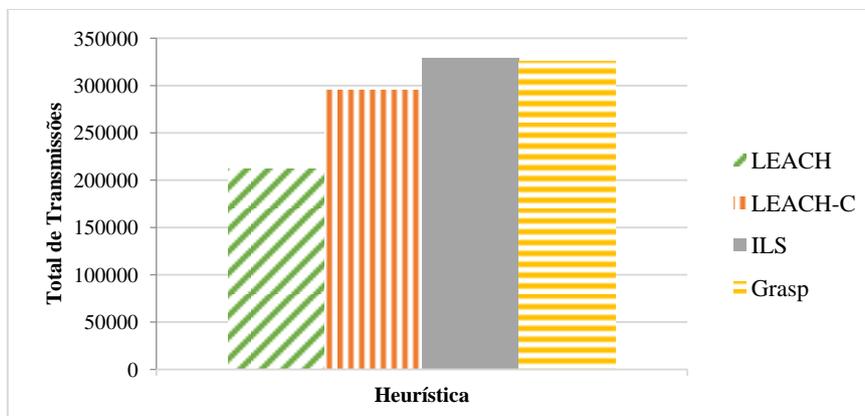
Figura 4 - Total de Transmissões por Round



Na Figura 5 é possível observar, para cada heurística, a média aritmética do total de transmissão considerando todas as instâncias. Dito isto, pode-se observar que a heurística ILS e o GRASP têm uma média muito próxima, e ao mesmo tempo, superior ao LEACH-C e principalmente ao LEACH. A diferença percentual do ILS em relação ao LEACH fica próximo

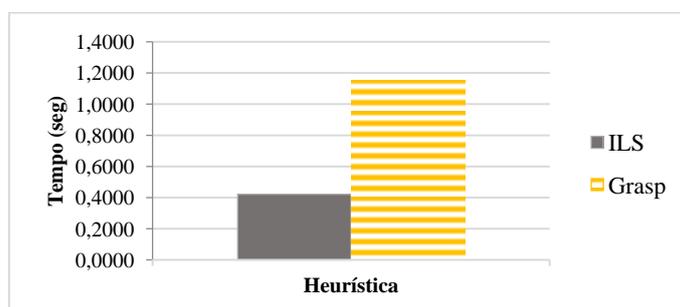
de 36%, já em relação ao LEACH-C próximo de 11% e em relação ao GRASP um pouco maior que 1% superior.

Figura 5 - Média das Transmissões das Instâncias por Heurística



Quanto ao tempo médio de execução, a Figura 6 apresenta uma comparação do ILS e do GRASP. Nesta figura é possível observar a média obtida considerando-se todas as instâncias. É possível perceber que o algoritmo proposto neste trabalho teve tempo menor que o GRASP. Os tempos do LEACH e do LEACH-C não foram comparados, pois estas abordagens não apresentaram competitividade em relação às avaliações anteriores.

Figura 6 - Tempo Médio Total das Heurísticas



5. Conclusões

Um grande desafio na área de redes de sensores está associado ao gerenciamento da energia dos sensores. Várias pesquisas vêm sendo desenvolvidas de forma a otimizar esse consumo, visando maximizar o tempo de vida útil da rede.

Neste trabalho foi proposto uma heurística baseada no ILS, que busca a melhor forma de disposição dos sensores em *clusters*, uma abordagem muito utilizada neste tipo de rede, para minimizar o gasto de energia em cada sessão de comunicação.

Para definir os sensores líderes de cada grupo, foi apresentada uma heurística de construção baseada no algoritmo *k-means*, incorporando estratégias como uso de grades considerando a distribuição dos sensores na área monitorada e sua energia residual. Como forma de aprimoramento, foi apresentado uma busca local e perturbação, caracterizando o ILS.

Em relação às abordagens da literatura, ao se comparar a abordagem proposta com os algoritmos LEACH e LEACH-C, foi possível observar o ganho da heurística proposta, consumindo menos energia por *round* e mesmo assim tendo um número superior de transmissões e mantendo um número superior de sensores vivos nos *rounds*. Isso se deve ao procedimento construtivo bem elaborado, que leva a busca local a obter bons resultados, tornando a heurística adequada ao problema de minimizar o gasto de energia e prolongar o tempo de vida da rede.

Ao se comparar com heurísticas mais recentes, o GRASP, a heurística desenvolvida foi competitiva, mostrando resultados similares, porém, com um tempo médio de execução inferior.

Para futuros trabalhos é interessante destacar que guiar o construtivo para custos menores podem levar o algoritmo proposto a resultados ainda melhores. Logo, elaborar uma estratégia que não utilize sensores aleatórios no construtivo e sim sensores que levem ao menor custo pode ser vantajoso. Sendo possível, desta forma, obter soluções ainda de melhor qualidade.

Agradecimentos

Ao CNPq e FAPEMIG pelo suporte financeiro que possibilitou a realização desse trabalho de pesquisa.

Referências

- Bandyopadhyay, S. e Maulik, U.**, An evolutionary technique based on K-means algorithm for optimal clustering, RN. Inf. Sci. Appl. 146, 1-4 (October 2002), 221-237.
- Barros, F. A. R.**, Rede de Sensores Sem Fio, GTA/UFRJ, http://www.gta.ufrj.br/seminarios/semin2002_1/flavio/, 2002.
- Defence Industry**, <http://www.defence-industries.com/contractors/army/base-camp-protection/genet/>, Acessado em 04/08/2012.
- Ferreira, F., Campos, G., e Silva, J.**, Clusterização com algoritmo genético baseado em densidade e grade, SBPO 2010.
- Heinzelman, R. W., Chandrakasan A. e Balakrishnan H.**, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, Vol. 1, No. 4, October 2002, pp. 660-670.
- Heinzelman, R. W., Chandrakasan A. e Balakrishnan H.**, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, Proceedings of the 33rd International Conference on System Sciences (HICSS '00), January 2000.
- Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., Mini, R. A. F.**, Redes de Sensores Sem Fio, 21º Simpósio Brasileiro de Redes de Computadores, 2003.
- Loureiro, A. A., Nogueira, J. M. S., Ruiz, L. B., e Mini, R. A.**, Rede de sensores sem fio, Pág. 193–234, Capítulo 5 do Livro texto da XXI Jornada de Atualização em Informática do XXII Congresso da Sociedade Brasileira de Computação, ISBN: 85-88442-23-X, 2002.
- Lourenço, H. R., Martin, O. e Stutzle, T.**, Iterated Local Search, In Handbook of Metaheuristics. Vol. 57 of Operations Research and Management Science. Kluwer Academic Publishers, 2002, pp. 321–353.
- MacQueen, J. B.**, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297, 1967.
- Matos, V. O., Arroyo, J. E. C., Santos, A. G., Gonçalves, L. B.**, An Energy-Efficient Clustering Algorithm for Wireless Sensor Networks, International Journal of Computer Science and Network Security, v. 12, p. 6-15, 2012.
- Pinheiro, J. M. S.**, Redes Móveis Ad Hoc, Última Atualização: 05/04/2005. (Acessado em 02/08/2012).
- Ribeiro, C. E. C. e Silva E. C. L.**, Redes de Sensores Sem Fio, UFRJ - Última Atualização: 05/06/2007, http://www.gta.ufrj.br/grad/07_1/rssf/, (Acessado em 02/08/2012).
- Ruiz, L. B.**, Redes de Sensores sem Fio, 2º Colóquio de Micro e NanoEletrônica – inctNAMITEC, 2012.
- Soares, S. S. R. F.**, Otimização do consumo de energia em múltiplas sessões multicast em redes de sensores, Tese de Doutorado, Instituto de Computação, UFF, 2010, Niterói.