

# Um Algoritmo Coevolutivo Cooperativo para Configuração de uma Rede de Sensores Sem Fio

**André S. Ruela**

Departamento de Computação – Universidade Federal de Ouro Preto  
Campus Morro do Cruzeiro, Bauxita, Ouro Preto, MG  
andre.siqueira.ruela@gmail.com

**Andre L. L. Aquino**

Instituto de Computação – Universidade Federal de Alagoas  
Campus A.C. Simões, sn, Tabuleiro do Martins, Maceió, AL  
alla@ic.ufal.br

**Frederico G. Guimarães, Alan R. R. de Freitas**

Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais  
Av. Antônio Carlos, 6627, Pampulha, Belo Horizonte, MG  
fredericoguimaraes@ufmg.br, alandefreitas@gmail.com

## RESUMO

Este trabalho propõe um algoritmo coevolutivo cooperativo para a configuração de uma rede de sensores sem fio, considerando métricas de redes complexas. Propõe-se uma heurística para encontrar uma configuração de rede de forma que sua estrutura de comunicação apresente um pequeno comprimento para o caminho médio mínimo e um elevado coeficiente de agrupamento. Esta configuração considera uma rede baseada em agrupamentos, onde os líderes dos agrupamentos têm dois raios de comunicação. Nós descrevemos como o problema pode ser particionado e como o cálculo de *fitness* pode ser dividido de tal forma que o modelo de coevolução cooperativa seja viável. Os resultados revelam que a nossa metodologia permite a configuração de redes com mais de uma centena de nós, com duas características de redes complexas, permitindo assim a redução do consumo de energia e o atraso na transmissão de dados.

**PALAVRAS CHAVE.** Redes complexas, algoritmos genéticos coevolutivos, redes de sensores sem fios. Área de classificação principal. OC - Otimização Combinatória.

## ABSTRACT

This work proposes a cooperative coevolutionary algorithm for design of a wireless sensor network considering complex network metrics. It is proposed a heuristic to find a network configuration such that its communication structure presents a small value for the average shortest path length and a high cluster coefficient. This configuration considers a cluster based network, where the cluster heads have two communication radii. We describe how the problem can be partitioned and how the fitness computation can be divided such that the cooperative coevolution model is feasible. The results reveal that our methodology allows the configuration of networks with more than a hundred nodes with two specific complex network measurements allowing the reduction of energy consumption and the data transmission delay.

**KEYWORDS.** Complex networks, coevolutionary genetic algorithms, wireless sensor networks. Main area. OC - Combinatorial Optimization.

## 1. Introdução

Redes de Sensores Sem Fio (RSSFs) (Akyildiz et al., 2002) representam uma tecnologia emergente que permite o monitoramento e o controle de variáveis e condições físicas e ambientais, como temperatura, som, luz, vibração, pressão, movimento e poluição. Uma RSSF consiste em um grande número de dispositivos autônomos sem fio, distribuídos espacialmente, chamados nós sensores, que trabalham de forma cooperativa para desempenhar diferentes funções. Os sensores normalmente são equipados com um pequeno microcontrolador, um transmissor a rádio ou outro dispositivo de comunicação sem fio e uma fonte de energia. Em aplicações de RSSFs, sensores normalmente não são alocados em locais pré-determinados. Ao invés disso, uma alocação aleatória é preferível.

A informação sobre o fenômeno monitorado é reportada por meio da rede até o nó *sink* (Royer and Toh, 1999). Aplicações de RSSFs, geralmente, têm a arquitetura projetada em  $n$ -camadas ( $n \geq 1$ ), onde a mais usada é a camada dupla (Baker and Ephremides, 1981; Ryu et al., 2002). Uma arquitetura em duas camadas consiste em vários aglomerados e um ou mais *sinks*. Cada aglomerado compreende um número de nós membros, responsáveis pela tarefa de monitoramento sobre a área correspondente, e um nó *hub* designado a coletar os dados dos sensores locais e encaminhá-los até o *sink*. Tal arquitetura baseada em agrupamentos oferece algumas vantagens intrínsecas contra a arquitetura plana em termos de consumo de energia: (i) apenas os nós *hubs* são envolvidos na tarefa de roteamento e os nós sensores apenas transmitem os dados monitorados para um *hub* próximo. Então, o consumo de energia na transmissão de dados pode ser substancialmente reduzido; e (ii) considerando que apenas o *hub* transmite o dado para fora do agrupamento, isso ajuda a poupar energia, evitando colisão entre os sensores locais (Akkaya and Younis, 2005; Heinzelman et al., 2002).

Neste trabalho, nós propomos um algoritmo coevolutivo (Liu et al., 2001; Potter and Jong, 1994) para projetar uma RSSF em duas camadas, considerando métricas de redes complexas (Costa et al., 2007). As principais contribuições de nosso trabalho são: (i) o problema foi particionado em subproblemas, permitindo o uso de um algoritmo coevolutivo cooperativo; (ii) baseando-se nas características dos problemas, um novo gerador de população inicial foi implementado. Este gerador é baseado no coeficiente de centralidade dos sensores; e (iii) o método proposto tem uma convergência rápida, quando comparado a outras abordagens. Isso ocorre devido ao conjunto inicial de indivíduos melhores, gerados pelo gerador de população.

Este trabalho é organizado da seguinte forma. Na Seção 2, nós apresentamos o projeto da rede e a definição do problema. Na Seção 2.1 nós discutimos sobre a formulação do problema. Em seguida, na Seção 3, nós apresentamos o algoritmo coevolutivo cooperativo proposto. Os resultados simulados são apresentados na Seção 4, e na Seção 4 conclui-se este estudo e apresentam-se os trabalhos futuros.

## 2. Definição do Problema

Nesse trabalho, apresentamos uma estratégia de agrupamento combinado com o roteamento baseada em métricas de redes complexas (Figura 1). Essa estratégia, consiste na utilização de nós sensores com um alcance de comunicação maior do que o raio usado pelos nós normais. Nós normais propagam seus dados para um dado *hub* usando a frequência de conexão normal, e o *hub* propaga os seus dados até o *sink*, usando a sua frequência de conexão especial, eventualmente com a utilização de um segundo rádio. Em ambos os casos é usada uma comunicação *multi-hop*. O uso deste *hub* leva a importantes características de redes complexas: um baixo caminho médio mínimo entre todos os sensores e o *sink*; e

um elevado coeficiente de agrupamento, conforme discutido em (Sharma and Mazumdar, 2005). Como hipótese, assumimos que estas características de redes complexas ajudam a poupar recursos, evitando comunicação excessiva e reduzindo o tempo de envio dos dados.

Com base nestes aspectos, o problema abordado neste trabalho pode ser enunciado como segue: *Considere um grafo geométrico,  $G = (V, E)$ , onde  $V$  representa o conjunto de nós sensores e  $E$  o conjunto de arestas, representando as ligações lógicas entre os nós. Estes links são determinados considerando a geometria de comunicação do nó, ou seja, todos os vizinhos acessíveis pelo nó  $v \in V$ . O problema é encontrar os nós  $v \in V$  que deverão permitir os raios auxiliares, e, portanto, serem configurados como hubs, gerando um novo conjunto de arestas  $E'$  e, portanto, uma nova rede  $G' = (V, E \cup E')$  tal que  $G'$  pode ser caracterizado como uma rede complexa com um caminho médio mínimo pequeno e um alto coeficiente de agrupamento.*

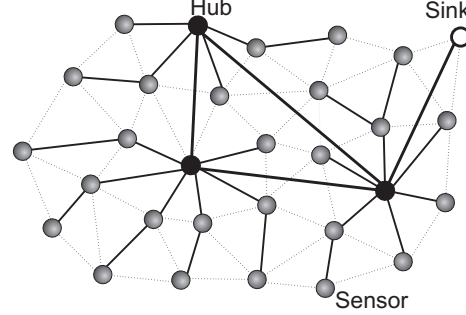


Figura 1: Propagação em Rede Complexa.

Portanto, a principal hipótese considerada sobre o problema é: *A caracterização da camada física RSSF como uma rede complexa nos permite construir uma infraestrutura que minimize o consumo de energia e o atraso na rede.* Neste trabalho, as métricas de redes complexas serão verificadas e avaliadas na camada física.

## 2.1. Formulação do Problema

Dada uma rede com um conjunto de nós, o problema consiste em encontrar os nós que serão reconfigurados como *hubs* e também as conexões que devem ser estabelecidas de forma a minimizar o custo total. Seja  $N$  o conjunto de nós normais e  $H$  o conjunto de *hubs*, de modo que  $N \cup H = V$  e  $N \cap H = \emptyset$ .

Os parâmetros do nosso modelo matemático são:  $\phi_i$  - é a demanda de comunicação, ou seja, a quantidade total de dados que o nó  $i$  deve enviar para o *sink*;  $r$  - é o raio de comunicação básico;  $d_{ij}$  - é a distância entre o nó  $i$  e o nó  $j$ ;  $c_{ij}$  - é o custo fixo de comunicação por dado a partir do nó  $i$  para o nó  $j$ ;  $a_j$  - é o custo de instalação do nó  $j$  com *hub*. Este é inversamente proporcional a distância entre  $j$  e o *sink*, ou seja, quanto maior a distância de  $j$  até o *sink*, menor será o custo de instalação.

As variáveis de decisão do nosso modelo matemático são  $z_i \in \{0, 1\}$ ,  $z_i = 1$  se o nó  $i$  é definido como um *hub*, e  $z_i = 0$ , caso contrário; e  $q_{ij} \in \{0, 1\}$ ,  $q_{ij} = 1$  se existe uma conexão lógica entre os nós  $i$  e  $j$ , e  $q_{ij} = 0$ , caso contrário.

Uma formulação em programação não-linear do problema definido acima é dada por:

$$z^* = \arg \min \sum_{i \in V} a_i z_i + \sum_{i \in H} \left( \sum_{j \in N} \phi_j c_{ji} q_{ji} \right) + \left( \sum_{k \in H} (c_{ik} q_{ik} + c_{k0}) \right) \times \sum_{i \in H} \left( \sum_{j \in N} \phi_j q_{ji} + \phi_i \right) \quad (1)$$

A função objetivo (1) fornece o custo total estabelecido pela rede em duas camadas. Este custo total inclui o custo de instalação, primeiro termo em (1), e o custo de propagação.

O segundo termo em (1) expressa o custo de propagação do dado a partir de todos os sensores até os seus respectivos *hubs*, enquanto o terceiro termo representa a quantidade total de dados que o *hub*  $i \in H$  deve enviar ao *sink*, que é multiplicado pelo custo de enviar o dado para o *sink*, por meio do *hub*  $k \in H$ , onde  $k$  pode ser igual a  $i$  ou pode ser um *hub* diferente. Isto resulta em dois ou três *hops*, a partir de qualquer nó na rede até o *sink*.

A função objetivo está sujeita às seguintes restrições:

$$\sum_{j \in V} q_{ij} = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{j \in H} q_{ij} \leq 1, \quad \forall i \in H \quad (3)$$

$$q_{ij} \leq z_j, \quad \forall i \in N, \forall j \in H \quad (4)$$

$$d_{ij} q_{ij} \leq 2r, \quad \forall i \in N, \forall j \in H \quad (5)$$

$$d_{ij} \leq 3r, \quad \forall i, j \in H \quad (6)$$

$$z, q \in \{0, 1\} \quad (7)$$

que são detalhadas como segue: a restrição (2) garante que o nó  $i \in N$  está conectado a apenas um único *hub*; a restrição (3) garante que o dado do *hub*  $i$  é roteado apenas por meio de um nó *hub* ou diretamente para o *sink*; (4) certifica que o dado do nó  $i \in N$  é roteado apenas por meio de um *hub*; (5) certifica que a distância entre o nó  $i$  e o *hub*  $j$  é menor ou igual a duas vezes o tamanho do raio de comunicação; (6) assegura que todos os *hubs* se conectam entre si, usando um raio de comunicação três vezes maior que o raio normal; e finalmente, a restrição (7) restringe os valores das variáveis inteiras  $z_i$  e  $q_{ij}$  para assumir entre 0 ou 1. É importante destacar que, no nosso caso, o dado propagado até o *sink* atravessa, no máximo, por dois *hubs*.

### 3. Algoritmo Coevolutivo

Neste trabalho desenvolvemos uma heurística baseada em algoritmos coevolutivos cooperativos (ACC) (Liu et al., 2001; Moriarty, 1997; Potter and Jong, 1994) para resolver o problema apresentado na seção anterior. Dois passos importantes na definição de qualquer modelo de coevolução são: (i) uma divisão adequada do problema e (ii) uma codificação adequada dos indivíduos.

A Figura 2 ilustra o ACC proposto nesse trabalho. O primeiro passo do algoritmo é carregar o grafo geométrico aleatório e gerar a população inicial. Então, o problema é dividido em  $L$  células, sendo que cada célula executa em um *thread* e tem sua própria subpopulação. As células são posicionadas em uma fila de execução. A primeira célula da fila executa seus operadores evolutivos locais até que o intervalo de cooperação seja alcançado. Durante este intervalo, a célula atualiza seus dados locais para cooperar com as outras, produzindo a solução global. Em seguida, a célula em execução é posicionada no final da fila e a célula seguinte na fila de inicia a sua execução. Este processo se repete até que um critério de parada seja atingido. Finalmente, a melhor solução já encontrada é retornada e as métricas complexas são calculadas sobre ela. As

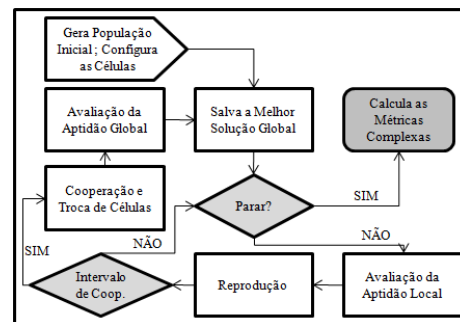


Figura 2: Fluxograma do ACC

subseções seguintes descrevem detalhadamente a divisão do problema para a cooperação, os operadores básicos utilizados e o gerador população inicial desenvolvido para o problema.

### 3.1. Divisão do Problema Para a Cooperação

A função objetivo global (1) foi explorada por algoritmo memético básico. Observando as soluções obtidas pelo algoritmo memético (Ruela et al., 2010) para todas as instâncias do problema, podemos observar que não há um grande número de *hubs*. O algoritmo mantém um equilíbrio entre o elevado custo de instalação dos nós como *hubs* e os custos de propagação. A Figura 3 é um exemplo de uma solução retornada pelo algoritmo memético, ilustrando a camada física avaliada pela função objetivo global.

O custo de instalação de um *hub* é relativamente caro para a rede. A seleção de um nó como um *hub* deve conduzir a uma redução significativa no custo de propagação, a fim de ser vantajoso. Assim, a seleção de dois ou mais *hubs*, na mesma região, em geral, não é uma boa escolha. Isso ocorre porque a diminuição do custo de propagação não cobre o aumento do custo de instalação. Assim, a instalação de apenas um *hub* por região é, geralmente, suficiente para manter o fluxo de informação na rede.

Apesar da geração aleatória de todas as instâncias do problema, é possível observar alguns padrões nos resultados obtidos após a otimização. Algumas regiões são mais propensas a ter *hubs* do que outras. Essas observações nos motivaram a aplicar o paradigma da divisão e conquista através de um algoritmo coevolutivo para reduzir o problema principal em pequenos subproblemas. Se, em geral, existem dois, um, ou nenhum *hub* por região nas melhores soluções, podemos dividir a rede em várias áreas quadradas e dividir o problema principal em subproblemas, de modo que cada subproblema consista em buscar *hubs* para cada área, se for necessário. Portanto, a dificuldade de resolver o problema original é reduzida a buscar *hubs* locais em cada célula. Cada célula possui uma população de indivíduos que codificam uma solução parcial para a respectiva célula, ou seja, o *hub* naquela célula. Os indivíduos em cada célula cooperam para produzir uma solução completa, que é a topologia da rede inteira.

É relevante observar que o algoritmo genético básico não é capaz de encontrar boas soluções para este problema da mesma forma que o algoritmo memético (Ruela et al., 2010). No entanto, o tempo de execução do algoritmo memético é muito alto. Aplicando o paradigma da divisão e conquista, e o algoritmo coevolutivo cooperativo, é possível chegar a soluções tão boas quanto às soluções retornadas pelo algoritmo memético, com menor esforço computacional.

### 3.2. Cálculo do Fitness

Na formulação do novo problema, a rede é dividida em uma grade de homogênea de células, onde cada uma tem uma função objetivo local. Os indivíduos em cada célula utilizam esta função objetivo local para calcular a seu valor de aptidão. A função objetivo local considera o custo de instalação do *hub* na célula e a avaliação do custo de propagação

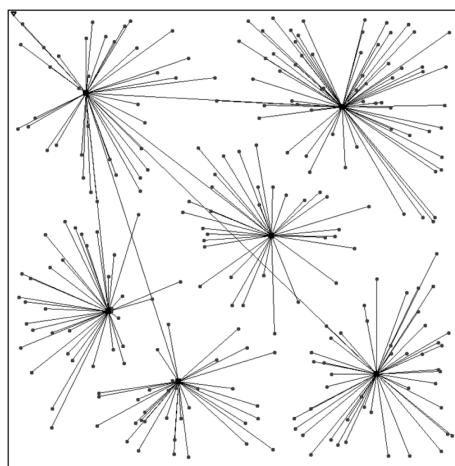


Figura 3: Solução com 256 nós.

dos dados analisa apenas os nós que podem ser cobertos por nós da célula, enquanto que na função objetivo global, todos os nós são analisados.

A Figura 4 ilustra a idéia da cobertura celular para uma determinada célula  $C_i$ ,  $1 \leq i \leq L$ . O quadrado cinza representa a área da célula. O conjunto de nós abrangidos pela célula  $i$  é denotado por  $K_i$  e é dado por todos nós em  $N$  que estão dentro do raio de pelo menos um nó em  $C_i$ . Os pontos na Figura 4 representam os nós em  $K_i$ . Observe que o conjunto de nós abrangidos pelo  $C_i$  contém alguns nós de células adjacentes. Nós podemos ver que existem sensores dentro da célula que não estão conectados ao *hub* da célula, enquanto há sensores externos que estão conectados ao *hub*. Portanto, o *hub* aceita conexões de qualquer nó na  $K_i$ , incluindo nós em células adjacentes. Assim, durante a avaliação do *fitness*, cada célula calcula apenas o fluxo dos nós que estão conectados a um *hub* localizado em seu interior, mas todos os nós abrangidos são verificados.

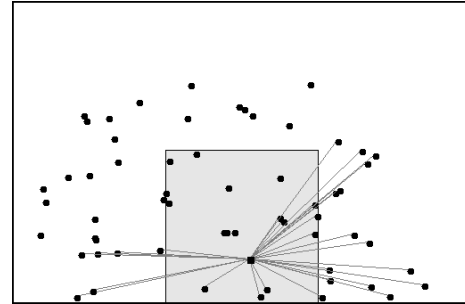


Figura 4: Cobertura da célula.

Seja  $Q_i$  o conjunto de *hubs* no interior da célula  $C_i$ ,  $K_i$  o conjunto de nós abrangidos pela célula  $C_i$ , de modo que  $|Q_i| \geq 0$ ,  $Q_i \subset H$ ,  $K_i \subset N$ ,  $Q_i \cap K_i = \emptyset$ .

A configuração da rede em cada célula pode ser representada por uma sequência de inteiros de tamanho variável, que representam os índices dos nós que estão configurados como *hubs*. Cada indivíduo  $p_{i,t}^{(k)}$  na subpopulação  $P_{i,t}$  é representado por um vetor inteiro indicando o *hub* na célula  $C_i$  na geração  $t$ .

Podemos definir a heurística a seguir a fim de eliminar as variáveis de decisão  $q_{ij}$  no modelo (1) ao resolver o problema com o ACC:

- Dado um indivíduo  $p_{i,t}^{(k)} \in P_{i,t}$ , nós obtemos um conjunto de nós em  $K_i$  e o conjunto de *hubs* em  $Q_i$ : se  $s \notin p_{i,t}^{(k)}$ , então o nó  $s$  está em  $K_i$ , caso contrário o nó  $s$  está em  $Q_i$ ;
- Como uma regra geral, cada nó  $i \in N$  envia o seu dado para um *hub*  $j \in H$  com o menor custo  $c_{ij}$ , que usualmente é o *hub* mais próximo. Dessa forma, nós atribuímos automaticamente o valor 1 ao  $q_{ij}$  correspondente.

O cálculo do custo de propagação da célula  $C_i$  considera apenas o custo da propagação de dados a partir dos nós em  $K_i$  que estão conectados a um dado *hub* em  $Q_i$  e os custos de roteamento de dados do *hub* na  $Q_i$  para a *sink*.

Com estas simplificações, as variáveis  $q_{ij}$  são implicitamente calculadas a partir da alocação de *hubs* fornecido por um indivíduo  $p_{i,t}^{(k)} \in P_{i,t}$ , simplificando assim o cálculo do custo de propagação. O cálculo do custo de instalação é simplesmente calculado, ao considerar  $z_s = 1$  se  $s$  está em  $p_{i,t}^{(k)}$ ,  $s \in N$  ou, caso contrário  $z_s = 0$ . Além disso, as restrições (2) - (4) pode ser negligenciadas por completo no modelo. Nós consideramos apenas as seguintes restrições: (i) a distância entre um nó  $i \in N$  e seu *hub* associado deve ser menor do que  $2r$ , em outras palavras, cada nó  $i \in N$  deve ter um *hub* dentro de seu raio de comunicação, e (ii) a distância de cada *hub* para outro deve ser menor do que  $3r$ .

Quando a violação de qualquer destas restrições ocorre em um determinado indivíduo, o seu valor da função objetivo é penalizado. Com os pressupostos discutidos nesta seção,

podemos empregar um modelo simplificado e não-linear para o problema. Isso mostra a flexibilidade dos algoritmos evolutivos para resolver problemas de otimização inteira e não linear.

### 3.3. Operadores Básicos

O primeiro passo para o ACC é definir as células  $C_i$ ,  $i = 1, \dots, L$ , e seus conjuntos  $K_i$  de nós coberto, com base em suas coordenadas e raio de comunicação. Então, uma população inicial para cada célula é gerada utilizando o novo gerador de população inicial, descrito na próxima subseção (3.4). Cada célula tem sua própria subpopulação com  $\mu$  indivíduos, cada um com codificações candidatas para as configurações da célula. A partir da população inicial, cada célula define o indivíduo representante da subpopulação. O representante é o melhor indivíduo que cada célula usa para executar a cooperação entre as outras células. A combinação de cada indivíduo e os representantes de outras subpopulações gera a solução global para o problema principal.

Cada célula se desenvolve e avalia a sua subpopulação local de forma independente uma da outra, considerando a função objetivo local, como descrito anteriormente, até que o intervalo de cooperação seja alcançado. O intervalo de cooperação  $\gamma$  foi definido em 5 gerações. A cada  $\gamma = 5$  gerações, as células paralisam a sua execução para cooperarem entre si, gerando uma solução global, com base em informações atuais sobre os representantes de outras células.

Durante o intervalo de cooperação, cada célula atualiza suas informações sobre os representantes de outras células. A célula que contém o *hub* mais próximo ao *sink* é eleito como a célula-mestre. Esta é responsável pela avaliação da solução global durante o intervalo de cooperação, armazenando o melhor indivíduo.

A evolução de cada subpopulação foi implementada usando a seleção por torneio binário para a reprodução, em que dois indivíduos são selecionados aleatoriamente na população atual e competem um contra o outro, e operadores de mutação projetados especificamente para o esquema de codificação adotado. Não existem operadores de cruzamento.

As novas soluções candidatas são produzidas apenas por operadores de mutação. Existem três operadores de mutação com a mesma taxa de mutação  $\rho_m = 0,1$ . A primeira mutação acrescenta ao acaso um novo *hub* de  $C_i$  no vetor  $p_{i,t}^{(k)}$ . A segunda mutação aleatória troca um *hub* em  $p_{i,t}^{(k)}$  por um novo *hub* de  $C_i$ . O terceiro operador de mutação remove aleatoriamente um *hub* de  $p_{i,t}^{(k)}$ . Essas operações possibilitam que um indivíduo tenha o vetor de *hubs* vazio e um vetor contendo vários *hubs*.

### 3.4. Geradores de População

Neste trabalho, propomos um gerador de população apropriado para o projeto de RS-SFs, com base no valor do coeficiente de centralidade (Freeman, 1977) dos nós. O coeficiente de centralidade pode ser interpretado como uma medida da influência que um nó tem sobre a disseminação de informações pela da rede. Pode ser medida como uma fração dos caminhos mais curtos entre pares de vértices em uma rede, que passam pelo nó em questão. Em outras palavras, a centralidade pode quantificar a importância de um vértice para a rede, e é definida como

$$B_u = \sum_i \sum_j \frac{\sigma(i, u, j)}{\sigma(i, j)},$$

onde,  $\sigma(i, u, j)$  é o número de caminhos mínimos entre os vértices  $i$  e  $j$  que passam pelo vértice  $u$ ,  $\sigma(i, j)$  é o total de caminhos mínimos entre os vértices  $i$  e  $j$ , e o somatório se dá sobre todos os pares  $i, j$  de vértices distintos (Costa et al., 2007).

Primeiro, o coeficiente de centralidade dos nós em  $V$  é medido. Em seguida, um operador probabilístico baseado tais valores é aplicada para selecionar cada *hub* em cada célula. Este gerador de população é chamado de *L Cells Betweenness Centrality Based* ('L'CBCB). A probabilidade  $\rho_{Q_i}$  de um nó  $u \in C_i$  a ser selecionado como um *hub* é

$$\rho_{Q_i}(u) = \frac{B_u}{\sum_{k \in K_i} B_k}.$$

Ou seja, quanto maior a importância do vértice  $u$  para o conjunto coberto  $K_i$ , maior o número de caminhos em que  $u$  está inserido, e maior a probabilidade  $\rho_{Q_i}(u)$ .

Além desse gerador, utilizamos os seguintes geradores aleatórios: 30PR - onde os indivíduos são gerados com 30% dos seus nós codificado como *hubs*; 'L'OR - onde os indivíduos são gerados com  $L$  nós selecionados como *hubs*; e 'L'CR - onde os indivíduos são gerados com  $L$  nós selecionados como *hubs*, mas assegurando apenas um *hub* em cada célula.

#### 4. Resultados da Simulação

A Tabela 1 compara a eficiência de todos os quatro geradores da população, considerando que a região é dividida em uma grade  $3 \times 3$ , tal que  $L = 9$ . A Tabela contém a média geral, a média dos melhores e a média dos piores, do total de 3300 avaliações de *fitness*, de modo que 100 indivíduos foram avaliados para cada um das 33 instâncias aleatórias, para cada tamanho de rede. Estes valores foram normalizados, dividido por  $10^{\log_2|V|}$ .

Escolher o número de *hubs* com base em um percentual fixo de, digamos, 30% produz os piores resultados para todos os tamanhos de rede. A estratégia 9CBCB disponibiliza, em média, os melhores indivíduos, o que mostra a existência de uma relação entre as medidas do coeficiente de centralidade dos *hubs* e da qualidade das soluções iniciais. Para todos os tamanhos de rede, o gerador 9OR foi melhor do que 9CR considerando os valores médios e os valores dos melhores indivíduos, mas novamente o gerador 9CBCB proporciona melhores indivíduos, em média. Embora os melhores indivíduos produzidos pelo 9OR são, em média, ligeiramente melhores do que os produzidos pelo 9CBCB, o menor desvio padrão mostra a robustez do gerador 9CBCB.

Tabela 1: Desempenho dos geradores

64 nós			
	Pior	Média	Melhor
30PR	4728.4 ± 628.5	2761.7 ± 411.0	1041.416 ± 408.1
9OR	1382.1 ± 210.7	589.2 ± 88.0	59.875 ± 48.1
9CR	1146.2 ± 142.9	632.8 ± 87.9	202.716 ± 64.5
9CBCB	572.1 ± 81.1	252.7 ± 70.1	80.701 ± 53.7
128 nós			
	Pior	Média	Melhor
30PR	4896.5 ± 339.6	3716.8 ± 217.0	2573.6 ± 291.9
9OR	362.5 ± 33.6	192.7 ± 10.3	54.2 ± 22.0
9CR	314.0 ± 22.4	210.8 ± 9.3	104.5 ± 21.4
9CBCB	208.4 ± 14.6	123.7 ± 9.1	59.6 ± 11.7
256 nós			
	Pior	Média	Melhor
30PR	3859.2 ± 161.7	3320.6 ± 142.6	2795.0 ± 180.1
9OR	78.7 ± 8.8	43.6 ± 1.9	18.0 ± 4.9
9CR	66.3 ± 3.7	47.3 ± 1.2	29.7 ± 3.9
9CBCB	50.6 ± 4.2	33.8 ± 1.4	20.7 ± 1.8
512 nós			
	Pior	Média	Melhor
30PR	2725.7 ± 56.3	2505.4 ± 55.4	2290.8 ± 71.1
9OR	25.1 ± 3.7	8.5 ± 0.2	3.9 ± 0.7
9CR	18.4 ± 1.6	9.1 ± 0.2	6.3 ± 0.4
9CBCB	16.3 ± 1.4	7.1 ± 0.2	4.8 ± 0.3



Nos casos com muitos nós,  $N = 256$  e  $N = 512$ , as diferenças entre os resultados fornecidos pelos geradores *9OR*, *9CR* e *9CBCB* diminuem, mas todos eles são muito melhores do que o gerador *30PR*. Isso indica que a divisão da região em 9 células foi muito útil para a geração de bons indivíduos. Por outro lado, a atenuação da diferença para instâncias maiores provavelmente foi causada pela diminuição na probabilidade de seleção dos melhores *hubs*. No entanto, para todos os casos, a estratégia *9CBCB* mostra-se consideravelmente melhor do que os anteriores.

#### 4.1. Resultados do Algoritmo Coevolutivo

Para esse estudo consideramos uma simulação realizada com o algoritmo implementado em Java. O algoritmo coevolutivo foi executado considerando uma população de  $\mu = 50$  indivíduos, por 500 gerações, com a tolerância máxima de 50 gerações sem qualquer melhoria na melhor solução atual. O número de simulações necessárias é dado por (Jain, 1991)

$$\#rounds = \left( \frac{100\xi\sigma}{\rho\bar{X}} \right)^2,$$

onde  $\xi$  é uma constante de valor 1,96,  $\sigma$  é o desvio padrão encontrado nas primeiras simulações,  $\bar{X}$  é a média dos valores obtidos e  $\rho$  é o percentual da média que nós queremos conseguir como desvio padrão, que neste caso foi de 5%. Consideramos 30 execuções com topologias aleatórias, e para cada topologia nós executamos o algoritmo genético 33 vezes. Os resultados são apresentados com intervalo de confiança simétrico assintótico de 95%. Os testes são executados em uma máquina Intel Core i5 2.4GHz com 4GB de RAM.

A densidade da rede é mantida constante, a área é de  $A = \pi r^2 |V| / \delta$ , onde  $r$  é o raio de comunicação,  $|V|$  é o número de nós e  $\delta$  é a densidade da rede (escolhido arbitrariamente com o valor de 8,4791). Os nós iniciam a execução com a mesma configuração de *hardware*. No final os *hubs* reconfiguram a sua frequência de rádio com base na infraestrutura dada como solução. Assim, a solução final tem uma RSSF heterogênea. Por fim, considere  $G = (V, E)$  o grafo inicial e  $G^* = (V, E^*)$ , o grafo retornado pelo algoritmo coevolutivo. A configuração resultante, utilizado em uma rede real será o grafo  $G' = (V, E')$ , onde  $E' = E \cup E^*$ . Assim, o coeficiente de agrupamento e o caminho médio mínimo são calculados considerando este grafo.

A Tabela 2 mostra os resultados comparando o Algoritmo Genético Básico (AGB), Algoritmo Genético Híbrido (AGH) (Ruela et al., 2010) e o algoritmo Coevolutivo Cooperativo (ACC), considerando a variação do número de nós como  $\{64, 128, 256, 512\}$ . Os parâmetros avaliados são: (i) o número de *hubs* instalado, (ii) o *fitness* das soluções, cujos valores foram normalizados, dividido por  $10^{\log_2 |V|}$ , onde  $|V|$  é o número de nós, (iii) o tempo de execução necessário para chegar à melhor solução, e (iv) a convergência das gerações.

Tabela 2: Comparação do desempenho das versões anteriores e do novo algoritmo.

$N$	Hubs			Fitness( $\cdot 10^{\log_2 N}$ )			Tempo do melhor			Convergência (g)		
	AGB	AGH	ACC	AGB	AGH	ACC	AGB	AGH	ACC	AGB	AGH	ACC
64	8	4	4	23.3	20.9	44.2	1.39s	7.63s	0.29s	109.6	11.6	28.3
128	11	6	6	67.3	43.5	51.9	5.15s	3m 53s	2.41s	104.5	16.3	40.4
256	27	6	8	222.9	31.9	36.6	46.73s	1h 13m 19s	20.98s	131.1	39.4	60.2
512	96	9	8	832.1	13.6	14.1	7m 6s	22h 35m 17s	1m 59s	118.8	31.2	46.6

A Tabela 3 compara os resultados do grafo aleatório geométrico inicial  $G$  e o grafo complexo  $G'$  utilizando duas métricas complexa rede: (i) o coeficiente de agrupamento, e (ii)

o caminho médio mínimo. Essas métricas não foram diretamente levadas em consideração durante o processo evolutivo, representando uma consequência da otimização, em vez de um objetivo explícito da formulação do problema.

Tabela 3: Comparação entre as métricas de redes complexas consideradas.

$N$	Coeficiente de Agrupamento				Caminho Médio Mínimo			
	$G$	AGB	AGH	ACC	$G$	BGA	HGA	CCA
64	0.73	0.73	0.73	0.73	1.96	1.95	1.94	1.93
128	0.69	0.69	0.70	0.69	2.67	2.61	2.54	2.57
256	0.66	0.65	0.66	0.66	3.67	3.51	3.60	3.40
512	0.63	0.66	0.63	0.65	5.09	4.22	4.92	4.57

Os resultados apresentados na Tabela 2 e na Tabela 3 mostram que, com o algoritmo coevolutivo, é possível construir uma topologia física para a RSSF com duas características específicas de redes complexas, o elevado coeficiente de agrupamento e o baixo caminho médio mínimo. Como podemos ver, o coeficiente de agrupamento do grafo  $G'$  é aproximadamente o mesmo valor do grafo geométrico original  $G$ , e o comprimento do caminho médio mínimo do grafo  $G'$  para as redes foi reduzido, quando comparado com o grafo geométrico original  $G$ . Baseado nesta topologia física, um algoritmo de roteamento pode ser usado para construir a melhor topologia lógica baseada em árvore. Considerando o consumo de energia e atraso, esta nova topologia lógica baseada em medidas de redes complexas será sempre melhor do que uma topologia lógica baseada no grafo geométrico original. Isso ocorre porque o número de retransmissões é consideravelmente reduzido quando os *hubs* são utilizados.

A qualidade das soluções retornada pela ACC são tão boas quanto às soluções retornadas pelo AGH, exceto para os casos com 64 nós. Para tamanhos pequenos de rede, as soluções retornadas pelo algoritmo híbrido não têm o mesmo comportamento de posicionamento dos *hubs*, como observado em outras instâncias. Portanto, a idéia de dividir a rede em um grid não foi muito útil para casos de pequeno porte. No entanto, para os casos com maior número de nós, o ACC é, em certo sentido, melhor do que os outros algoritmos, pois pode proporcionar boas topologias físicas de rede com o melhor tempo de execução. O tempo de execução decorrido para alcançar a melhor solução do ACC é inferior aos dos outros algoritmos, para todos os tamanhos de rede. Isso mostra que a abordagem da divisão e conquista foi realmente uma aplicação vantajosa: o principal problema poderia ser reduzido em problemas menores que são mais fáceis de resolver, fazendo a abordagem coevolutiva cooperativa muito útil para este problema.

Por outro lado, o desempenho ruim do AGB pode ser justificado pelo uso inadequado dos operadores genéticos e uma exploração superficial da natureza do problema. Normalmente, as topologias fornecidas pelo AGB têm um elevado número de *hubs* desnecessários, que são caros para toda a rede. Além disso, a codificação binária dos indivíduos em  $|V|$  bits, usada no AGB, é ineficiente devido ao baixo número de *hubs* que são necessários.

Considerando os aspectos do roteamento baseado em árvore sobre a topologia física, um caminho médio mínimo mais curto evita, principalmente, o atraso na entrega dos dados. O inconveniente, neste caso, é que quando o rádio extra é ativado mais energia é consumida, mas considerando o consumo global de energia, esta abordagem pode realmente economizar energia. A rede complexa tendo um valor menor para o caminho médio mínimo fornece uma topologia lógica com baixo consumo de energia, porque será necessário um número menor de saltos para enviar dados para o *sink*. Portanto, parece haver alguma relação entre

baixa aptidão na AGB e pequeno valor do caminho médio mínimo. Porém, os resultados do AGH e ACC mostram que a redução dos valores de fitness e os poucos *hubs* provocam um ligeiro aumento do caminho médio mínimo. A razão para esse comportamento é que quando menos *hubs* são alocados na rede, a rede complexa se assemelha a uma rede regular, ou seja, haverá menos arestas ligando pontos distantes, levando a um pequeno aumento no caminho médio mínimo. Portanto, o caminho médio mínimo de uma rede complexa com poucos *hubs*, correspondente às soluções encontradas por ambos AGH e ACC, tende a se assemelhar ao caminho médio mínimo da rede regular. Em contrapartida, o excesso de *hubs* examinados nas soluções retornadas pelo AGB causou uma redução, em média, do caminho médio mínimo, pois existem mais caminhos alternativos. No entanto, o alto custo de instalação dos *hubs* o torna ineficiente.

Para as instâncias com 64 nós, na qual os *hubs* das soluções do AGB e AGH são mais espacialmente implantados, o caminho médio mínimo mais baixo é encontrado nas soluções do ACC. Isso ocorre porque os *hubs* do ACC estão concentrados dentro de sua grade de células, decorrente da geração de indivíduos baseados no coeficiente de centralidade. Para as instâncias com 128 nós, os *hubs* das soluções do AGH tem uma localização mais centralizada. Isso explica porque as soluções têm o menor caminho médio mínimo. No entanto, para as instâncias com 256 e 512 nós, apesar da boa qualidade das soluções do AGH, o menor número de *hubs* resulta em um menor número de arestas do conjunto  $E^*$ , que não são suficientes para proporcionar uma significativa redução do caminho médio mínimo. As soluções AGB têm o menor caminho médio mínimo para as instâncias com 512 nós, uma vez que a inserção excessiva e de *hubs* desnecessários fornece um conjunto maior  $E^*$  de arestas. A instalação de 96 *hubs* produz um alto custo de instalação, que é muito caro para a rede e não pode ser implementado na prática. Assim, mais uma vez, o ACC apresenta-se como a melhor opção.

## 5. Conclusão e Trabalhos Futuros

Este trabalho apresentou um algoritmo coevolutivo cooperativo para o rápido projeto de uma topologia física em RSSF. O objetivo era produzir uma topologia física que apresenta um alto coeficiente de agrupamento e um caminho médio mínimo menor. Esta topologia física pode ser utilizada como infraestrutura inicial da rede minimizando assim o consumo de energia e os atrasos, elementos presentes no nosso modelo. Para redes com centenas de nós, por exemplo, a rede com 512 nós, o algoritmo de coevolutivo cooperativo foi satisfatório, obtendo-se uma topologia física que satisfaça as características de redes complexas.

Os resultados mostraram que o ACC pode encontrar um projeto de RSSF com duas características específicas de redes complexas. Isso foi evidenciado em nossos resultados que mostraram que o coeficiente de agrupamento do grafo resultante é o mesmo ou ligeiramente mais elevado quando comparado com o grafo geométrico original, e o comprimento do caminho médio mínimo do grafo resultante, em nosso cenário específico, foi reduzido quando comparado com o grafo geométrico original. O algoritmo coevolutivo foi capaz de alcançar soluções de alta qualidade com o menor tempo de execução, o que representa configurações com poucos nós instalado como *hubs*. Isso significa que, na prática, o ACC é o algoritmo mais viável, comparando com os anteriores.

O modelo de coevolução desenvolvido neste trabalho é um passo importante rumo a uma aplicação paralela, e até mesmo distribuída, para o projeto de RSSFs, que representam importantes avanços para os trabalhos futuros.

## Referências

- K. Akkaya and M. Younis**, A survey of routing protocols in wireless sensor networks, *Ad Hoc Network*, 3(3):325–349, May 2005.
- Ian F. Akyildiz, WellJan Su, Yogesh Sankarasubramaniam, and Erdal Cayirci**, A survey on sensor networks, *IEEE Communications Magazine*, 40(8):102–114, August 2002.
- D. J. Baker and A. Ephremides**, The architecture organization of a mobile radio network via a distributed algorithm, *IEEE Trans. on Communications*, 29(11):1694–1701, November 1981.
- L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas**, Characterization of complex networks: A survey of measurements, *Advances in Physics*, volume 56, pages 167–242, August 2007.
- L. C. Freeman**, A set of measures of centrality based on betweenness, *Sociometry*, volume 40, pages 35–41, 1977.
- W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan**, Energy-efficient communication protocol for wireless sensor networks, *Proc. IEEE Hawaii Int’l Conf. System Science*, pages 1–10, 2002.
- R. K. Jain**, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, April 1991.
- Y. Liu, X. Yao, Q. Zhao, , and T. Higuchi**, Scaling up fast evolutionary programming with cooperative coevolution, *Parallel Problem Solving from Nature VIII Conference*, pages 882–891, 2001.
- D. E. Moriarty**, Symbiotic evolution of neural networks in sequential decision tasks, *Ph.D. dissertation*, Univ. of Texas at Austin, 1997.
- M. A. Potter and K. A. De Jong**, A cooperative coevolutionary approach to function optimization, *Parallel Problem Solving from Nature III Conference*, pages 249–257, 1994.
- Elizabeth M. Royer and Chai-Keong Toh**, A review of current routing protocols for ad-hoc mobile wireless networks, *IEEE Personal Communications*, 6(2):46–55, April 1999.
- André S. Ruela, Raquel S. Cabral, André L. L. Aquino, and Frederico G. Guimarães**, Memetic and evolutionary design of wireless sensor networks based on complex network characteristics, *International Journal of Natural Computing Research*, 1(2):33–53, April 2010.
- J. H. Ryu, S. Song, and D. H. Cho**, New clustering schemes for energy conservation in two-tiered mobile ad hoc networks, *IEEE Trans. on Vehicular Technology*, 51(6), November 2002.
- Gaurav Sharma and Ravi Mazumdar**, Hybrid sensor networks: A small world, *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobHoc’05)*, pages 366–377, Urbana-Champaign, Illinois, USA, May 2005, ACM.