

K-means-net: Clusterização e escolha de gateways em redes de sensores sem fio*K-means-net: Clustering and choosing gateways on wireless sensor networks*Larissa Maria Santiago Correia¹, Alexandre Celestino Leite Almeida², Marcos Tomio Kakitani³

RESUMO

O posicionamento de nós *gateways* em uma rede de sensores sem fio com múltiplos saltos tem impacto direto na qualidade dos serviços da rede. Uma boa escolha dos *gateways* resulta em um menor número de saltos entre um nó e seu *gateway* (GW) correspondente na rede, causando menos perdas, menor tempo de comunicação e melhorando a qualidade de serviço. O tamanho da rede pode tornar o problema de encontrar soluções ótimas para a implantação dos *gateways* bastante complexo. Este trabalho propõe um algoritmo baseado no método *K-means* em conjunto com teoremas de substituição com a finalidade de dividir a rede em *K* sub-redes e na escolha do GW de cada uma delas. Os teoremas de substituição são usados para encontrar nós não redundantes, possíveis *gateways*, em cada *cluster* dividido pelo *K-means*. A adaptação da utilização simultânea destes dois métodos fornece boas soluções para a implantação dos GWs na rede, bem como já fornece o conjunto de nós de cada GW.

Palavras-chave: *T-step*. *K-means*. RSSF. Rede de sensores sem fio. Teoria de grafos.

ABSTRACT

The positioning of gateway nodes in a multi-hop wireless sensor network has a direct impact on the quality of network services. A good choice of gateways results in fewer hops between a node and its corresponding gateway (GW) in the network, causing fewer losses, shorter communication time, and improved quality of service. The size of the network can make the problem of finding optimal solutions for the deployment of gateways quite complex. This work proposes an algorithm based on the *K-means* method in conjunction with substitution theorems, with the purpose of dividing the network into *K* subnets and choosing the GW of each of them. Substitution theorems are used to find non-redundant nodes, possible gateways, in each cluster divided by *K-means*. The adaptation of the simultaneous use of these two methods provides good solutions for the deployment of the GWs in the network, as well as already provides the set of nodes of each GW.

Keywords: *T-step*. *K-means*. WSN. Wireless sensor networks. Graph theory.

¹ Graduanda do curso de Engenharia de Telecomunicações da UFSJ - Campus Alto Paraopeba, Ouro Branco-MG.

E-mail:

larissa.mscorreia@gmail.com

² Docente do departamento de Física e Matemática da UFSJ - Campus Alto Paraopeba, Ouro Branco-MG.

³ Docente do departamento das Engenharias de Telecomunicações e Mecatrônica da UFSJ - Campus Alto Paraopeba, Ouro Branco-MG.

1. INTRODUÇÃO

As Redes de Sensores Sem Fio (RSSF) são um tipo de rede móvel com a finalidade de monitorar eventos e transmitir as informações coletadas através de comunicação sem fio. O monitoramento é feito por meio de pequenos componentes, chamados nós sensores. Estes nós podem se comunicar entre si ou diretamente com um *Gateway* (GW), o qual é responsável pela coleta de dados e envio para algum centro de dados externo à rede. Um GW pode ser externo à rede ou um nó da mesma. Neste trabalho, estamos interessados no problema de escolher como GW um nó pertencente à rede. Tipicamente, os nós sensores são compostos por elementos que possuem limitações quanto às suas fontes de energia, podendo apresentar restrições ou até mesmo inviabilidade em relação à recarga de suas baterias ou substituição dos nós. Desta forma, minimizar o número de saltos entre os nós e os GWs é muito importante em uma RSSF.

Uma rede de sensores sem fio (RSSF) de grande volume representa um problema para o melhor posicionamento dos sensores que vão transmitir ou recolher informações desta rede. Tendo em vista o diminuto tamanho dos nós, a duração da bateria dos mesmos depende do volume de dados que passa por ele.

Um dos principais problemas para a implantação de *gateways* (GWs) em uma RSSF é escolher um nó que diminua o maior número de saltos das informações para transmissão de dados. Desta maneira, o fluxo de dados em um mesmo nó é diminuído e a bateria do mesmo dura por mais tempo, possibilitando maior tempo de coleta de dados.

Dividir uma rede grandiosa em sub-redes menores, cada uma com um GW responsável pela coleta ou transmissão de dados de todos os nós associados àquela sub-rede é um modo estudado para a melhor implantação destes GWs. Tais sub-redes são chamadas de *clusters*.

Empregar métodos inteligentes, tais como otimização por enxame de partículas (PSO), computação evolucionária (EC), otimização de colônias de formigas (AIA), *K-means*, algoritmos genéticos (GA) com diferentes abordagens e outros exemplificados em (REBAI et al., 2015), (MÁGAN-CARRIÓN et al., 2016), (LIN, 2013), (LIN et al., 2016), (YOON e KIM, 2013) e (ABDELKHALEK et al., 2015) são estratégias bastante utilizadas para a otimização de redes sem fio.

O presente trabalho propõe para a resolução do problema de implementação dos *gateways* um método inteligente, o *K-means* adaptado, em conjunto com alguns teoremas, apresentados em (HUANG et al., 2017). O método é aplicado para o *clustering* da rede. O

nó centro do *cluster* é definido como aquele que resolve o problema de otimização da Equação 1 dentro do *cluster* e é obtido baseado nos teoremas, chamados de teorema da substituição e teorema do *t-step* substituição.

Estes teoremas permitem eliminar os nós redundantes dos *clusters*, ou seja, eliminam nós que possuem escolhas melhores de GW na rede, diminuindo o espaço de busca pelo GW dentro de cada *cluster* e aumentando a probabilidade de chegar a soluções ótimas.

Neste trabalho consideraremos que as distâncias em saltos entre quaisquer dois nós são sempre distâncias mínimas. Para otimizar a implementação de *gateways* nosso objetivo é minimizar os saltos da rede. Deste modo, nosso problema pode ser formulado como:

$$\min \max_{1 \leq j \leq k} \left\{ \max_{n_i \in C_j} d(u_j, n_i) \right\}, \quad (1)$$

sujeito a: $\{u_1, u_2, \dots, u_k\} \subset N$

sendo n_i um nó da rede, u_j um GW, N o conjunto de nós da rede, C_j um dos *clusters* existentes e $d(u_j, n_i)$ a distância em saltos entre u_j e n_i .

2. ALGORITMOS RELACIONADOS

Para a realização deste trabalho foi considerado um algoritmo de otimização em conjunto com alguns teoremas propostos em (HUANG et al., 2017).

2.1. *K-means*

O *K-means* é um bem conhecido método de agrupamento que usa técnicas estatísticas sendo um algoritmo de clusterização. Tal algoritmo encontra K centros em um conjunto de nós que minimizam a distância entre cada nó e o centro mais próximo. (KANUNGO et al., 2002).

O *K-means* tradicional calcula centroides imaginários para cada *cluster*, que são a média das posições dos nós que pertencem ao mesmo. O *clustering* (agrupamento) é então reestabelecido e novos centroides são criados. Uma resposta satisfatória é obtida quando a posição dos centroides para de variar bruscamente.

A utilização do *K-means* em uma RSSF onde o GW deve ser um dos nós já existentes não é direta, uma vez que o *K-means*, em nosso problema, não terá liberdade de escolha do centro em qualquer posição do espaço. Além disso, o *K-means* usa a distância

Euclidiana dos nós em seus cálculos e, em nosso problema, estamos interessados no número de saltos entre cada nó e seu GW.

Aplicado ao nosso problema, desenvolvemos um algoritmo baseado no *K-means*, que divide a RSSF em *K clusters*, de modo que nós com menos saltos até seu respectivo GW tendem a se conservar em um mesmo *cluster*. Os centroides deixam de ser médias das posições dos nós do *cluster* e se tornam nós existentes na RSSF. Os centroides finais serão os melhores nós para serem considerados *gateways*.

2.2. Teorema da substituição – *T-step*

Em seu trabalho recente, (HUANG et al., 2017) propuseram os teoremas da substituição. Através destes teoremas, a complexidade para encontrar boas soluções para uma RSSF composta por uma grande quantidade de nós diminui notavelmente. O espaço de busca é reduzido significativamente com a aplicação dos teoremas, que resultam na melhora do desempenho dos algoritmos que fazem o uso de tais artifícios.

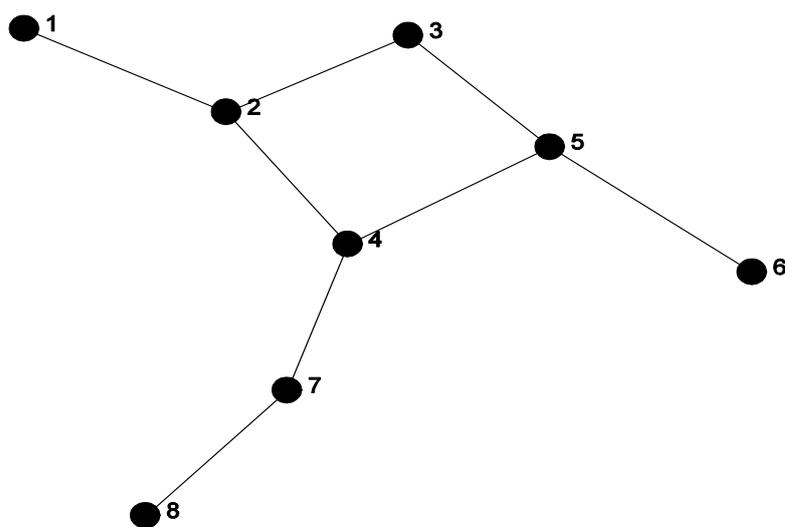


Figura 1. Exemplo de GD para uma rede sem fio com 8 nós.

Modelamos uma rede de sensores sem fio através do Grafo de Demonstração (*GD*), ilustrado na Figura 1. Definimos $GD = \{N, A\}$, de maneira que N é o conjunto de vértices, que representam os nós presentes no grafo e que são possíveis GWs e A é o conjunto de arestas do grafo que representam a distância de um salto entre os dois nós conectados. Definimos $d(n_i, n_j)$ como a distância em saltos entre dois vértices. Esta distância é o número mínimo de arestas que separam dois nós.

Para a explicação dos teoremas, são considerados alguns conceitos, explanados a seguir.

Nó folha: são considerados nós folhas aqueles sensores que possuem conexão com apenas um nó de toda rede. Claramente, um nó folha não é um bom candidato a *gateway* e, portanto, pode ser excluído do espaço de busca. Considerando a Figura 1, os nós n_1 , n_6 e n_8 possuem conexão apenas com os nós n_2 , n_5 e n_7 , respectivamente. Desta maneira, são considerados nós folha.

Nós adjacentes: Considere um nó n_i . Os nós adjacentes ao mesmo são aqueles separados por apenas uma aresta, ou seja, estão a distância de apenas um salto. Exemplificando, na Figura 1, o conjunto de nós adjacentes ao nó n_4 , $A(n_4)$, é composto pelos vértices $\{n_2, n_5, n_7\}$, pois $d(n_4, n_2) = d(n_4, n_5) = d(n_4, n_7) = 1$.

T-hop nós adjacentes: O conjunto $A_t(n_i)$ de um nó inclui todos os nós que estão a uma distância menor ou igual a t saltos de n_i . A título de exemplo, considerando $t = 2$, o vértice n_3 possui $A_2(n_3) = \{n_1, n_2, n_4, n_5, n_6\}$ uma vez que $d(n_3, n_2) = d(n_3, n_5) = 1 < t$ e $d(n_3, n_1) = d(n_3, n_4) = d(n_3, n_6) = 2 = t$.

Teorema da substituição: Quando o conjunto de nós adjacentes para o nó n_i é $A(n_i) \subset (n_j \cup A(n_j))$, n_i é considerado redundante e pode ser substituído por n_j .

Teorema da t-step substituição: Quando o conjunto de nós t – hop adjacentes para o nó n_i é $A_t(n_i) \subset (n_j \cup A_t(n_j))$ então n_i é um nó redundante e pode ser substituído por n_j .

Estes teoremas serão aplicados na função *t-step* a qual será utilizada no pseudocódigo apresentado no Algoritmo 1. Os nós não redundantes encontrados serão os nossos candidatos a *gateways*.

3. METODOLOGIA

Neste trabalho propomos um algoritmo, denominado *K-means-net*, que escolhe, em uma determinada rede, os K melhores nós para serem utilizados como *gateways* e atribui todos os nós para o GW mais próximo, formando assim, K *clusters*.

A ordem de complexidade para se encontrar a melhor configuração de *gateways* (e seus *clusters*) é relativamente alta. Para tentar mensurar, quando a rede tem tamanho n e o número de GWs é K , temos C_n^K combinações possíveis. Isso significa que uma rede com 100 nós a ser dividida em 5 *clusters* possui na ordem de 10^{148} possíveis combinações. Desta forma, a busca por heurísticas eficientes é de suma importância.

Nossa proposta é aplicar os teoremas apresentados em (HUANG et al., 2017) em uma modificação do algoritmo *K-means* para trabalhar em redes onde o GW necessita ser um dos nós. O valor de K é determinado como a quantidade de *gateways* desejada. Quando $K = 1$, não há divisão de *clusters*, sendo apenas aplicado o *t-step*, de modo a achar o melhor nó para ser GW. Um paralelo entre os algoritmos *K-means* e *K-means-net* é realizado na Tabela 1.

Tabela 1. Paralelo entre os algoritmos *K-means* e *K-means-net*

Passos do <i>K-means</i>	Passos do <i>K-means-net</i>
1 – Escolhe centros aleatórios	1 – Escolhe nós aleatórios (nós centro)
2 – Divide os <i>clusters</i> pela distância euclidiana aos centros	2 – Divide os <i>clusters</i> pela distância em saltos aos nós centro
3 – Escolhe novos centros (distância média de todos os nós atrelados ao centroide)	3 – Escolhe novos centros (usa o <i>t-step</i> para encontrar os novos nós centro)
4 – Se não atingiu o critério de parada, volta ao passo 2.	4 – Se não atingiu o critério de parada, volta ao passo 2.

O algoritmo proposto *K-means-net* é descrito no Algoritmo 1. O mesmo considera, em sua inicialização, nós aleatórios como *gateways*. O conjunto N é então submetido a uma função definida como *Cluster_Kmeans()* que divide os nós em K *clusters*, em relação aos GWs escolhidos. Esta divisão é realizada a partir da distância em saltos. Calcula-se a distância em saltos de cada nó para cada GW. O *gateway* associado a menor distância inclui o nó em seu *cluster*.

Algoritmo 1: ALGORITMO *K*-MEANS-NET

```

Entrada:  $N, K, Adj$ 
Saída:  $GW$ 
1 início
2    $Centros = centros\_aleatorio(N, K)$ 
3    $parada \leftarrow false$ 
4    $GW \leftarrow []$ 
5   Enquanto( $parada = false$ )
6      $Clusters = Cluster\_Kmeans(Adj, Centros)$ 
7     para cada  $Cluster_i \in Clusters$  faça
8        $N_i = folha(cluster_i)$ 
9        $GW_i = t\_step(N_i)$ 
10    fim
11     $parada = mudanca\_centros(Centros, GW)$ 
12     $Centros \leftarrow GW$ 
13  fim Enquanto
14 fim
15 retorna  $GW$ 
    
```

Algoritmo 1. Pseudocódigo do algoritmo proposto *K-means-net*

Aplicamos então as funções *folha()* e *t_step()* a cada *cluster*. A função *folha()* elimina do espaço de busca de *gateways* os nós folhas, uma vez que os mesmos não podem ser um nó GW. A função *t_step()* elimina do espaço de busca os nós redundantes do *cluster*. Os nós restantes são possíveis GWs. O nó não redundante com menor máxima distância em relação ao *cluster* é então escolhido como novo centroide.

A função *t_step()* retorna os nós não redundantes da rede, ou seja, aqueles que são candidatos a GWs. A mesma recebe a matriz de adjacência $A_t()$ de toda a rede. Nesta matriz quadrada, cada linha e cada coluna indicam um nó. Por exemplo, se existe uma marcação na posição (i, j) da matriz, isso significa que os vértices n_i e n_j estão conectados por alguma distância em saltos menor ou igual a t . As linhas são então comparadas. Se ocorre o descrito no teorema, ou seja, se $A_t(n_i)$ está contido em $n_j \cup A_t(n_j)$ então o nó n_i é redundante e pode ser excluído do espaço de busca pelo GW. A função começa com $t = 1$ e incrementa t para reduzir cada vez mais o espaço de busca pelos GWs. O valor de t para de ser incrementado, quando t se torna igual ao máximo salto da rede ou quando os candidatos a GWs se reduzem a apenas um.

Com novos centroides definidos, os *clusters* são recalculados e os passos são repetidos até que os centroides parem de se alterar significativamente. A função *mudança_centros()* avalia essa alteração.

4. RESULTADOS

Foram criadas redes cujas posições dos nós foram geradas aleatoriamente seguindo uma distribuição uniforme em cada coordenada e cujas arestas foram criadas apenas entre os nós que satisfazem uma distância euclidiana máxima.

A Figura 2, ilustrada a seguir, mostra o GD de uma rede aleatória com 25 nós distribuída uniformemente no plano coordenado.

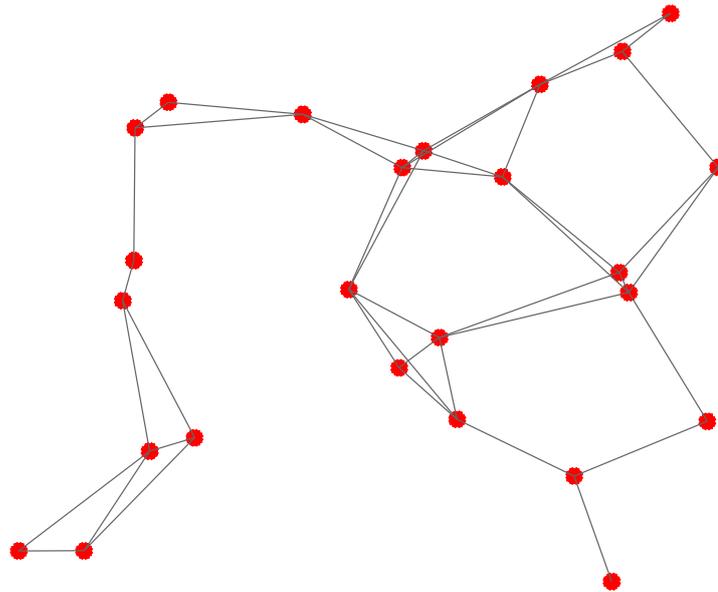


Figura 2. GD de uma rede aleatória com 25 nós.

Esta rede foi submetida ao algoritmo proposto, *K-means-net*, considerando $K = 3$. A Figura 3 mostra a mesma rede após ser submetida ao *K-means-net* e a distribuição dos *clusters*. Os nós representados por ★, são os nós apontados como melhores candidatos a *gateways* para a rede. Os nós de um mesmo *cluster* possuem cores iguais. A maior distância em saltos de um GW desta rede, após passar pelo processo de *clustering* é 2.

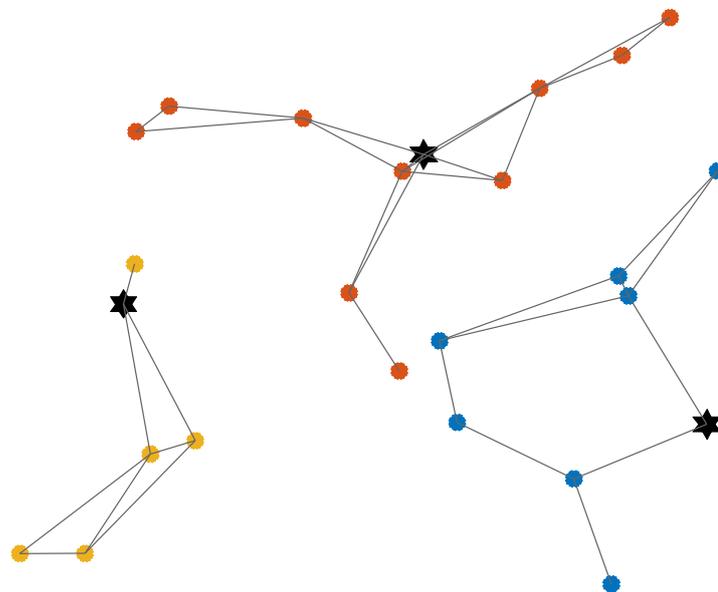


Figura 3. GD da rede aleatória de 25 nós com a divisão dos *clusters* e respectivos GWs.

Considerando agora uma rede aleatória e uniformemente distribuída com 50 nós, a Figura 4 representa seu respectivo GD, considerando $K = 5$ e mesmo raio de cobertura que a rede da Figura 2.

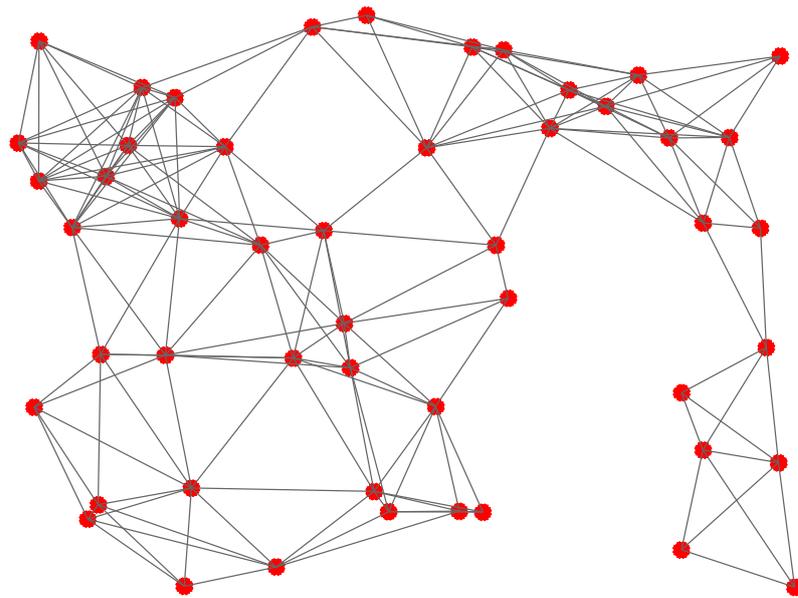


Figura 4. GD de uma rede aleatória com 50 nós.

Obtemos, após o processo de *clustering*, a rede ilustrada da Figura 5. A maior distância em saltos de um GW desta rede é, também, 2.

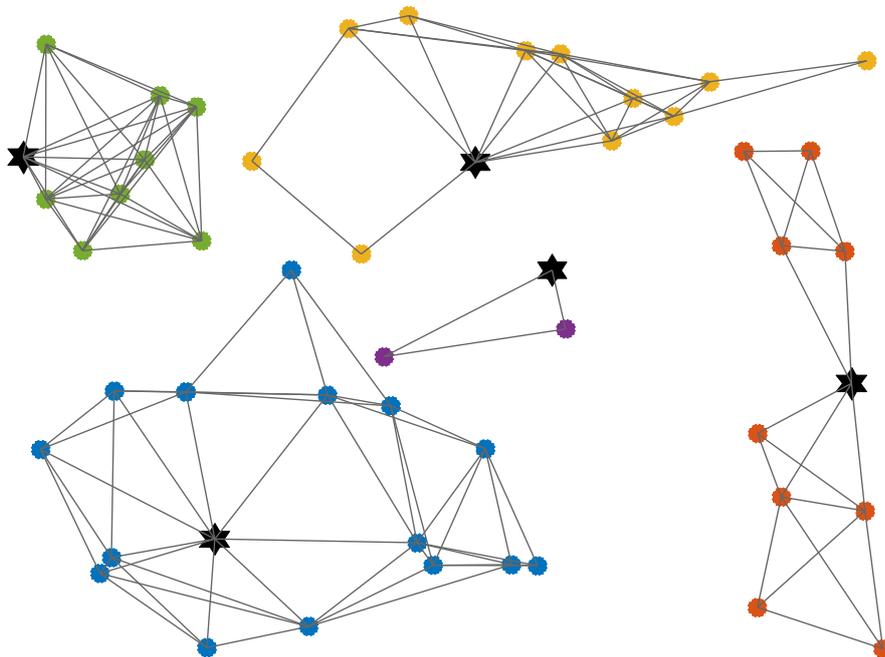


Figura 5. GD da rede aleatória de 50 nós com a divisão dos *clusters* e respectivos GWs.

Analisamos redes pequenas, com 25 e 50 nós, considerando 2, 3 e 5 *gateways* para a rede de 25 nós e 5, 7 e 10 *gateways* para a rede de 50 nós. Também analisamos redes grandes, com 100 e 200 nós, considerando, agora, 10, 15 e 20 *gateways* para a rede composta por 100 nós e 20, 35 e 50 *gateways* para a rede com 200 nós.

A Tabela 2 mostra a média dos parâmetros número de iterações, tempo de execução e máxima distância em saltos para 50 execuções do algoritmo *K-means-net* para as redes citadas anteriormente.

Tabela 2. Média das variações do tempo de execução, iterações e máximo salto com o aumento do número de *gateways*

Nós	Gateways	Número médio de iterações	Tempo médio de execução (s)	Máxima distância média (saltos)
25	2	5,72	0,096212	3,56
25	3	5,90	0,105469	2,56
25	5	6,36	0,111409	2,10
50	5	8,78	0,132006	2,18
50	7	9,04	0,141691	2,06
50	10	9,22	0,144339	2,04
100	10	14,98	0,201001	2,00
100	15	8,08	0,173505	1,94
100	20	7,24	0,168442	1,68
200	20	10,24	0,280234	1,76
200	35	3,18	0,186043	1,06
200	50	9,04	0,168149	1,04

5. DISCUSSÃO

Através da Tabela 2 é possível observar que a variação do valor de K acarreta variações no máximo salto entre GW e o nó, no tempo de execução do algoritmo e na quantidade de iterações necessárias para se chegar a uma resposta satisfatória. O aumento do valor de K e, conseqüentemente, o aumento do número de *gateways* espalhados pela rede diminui o tempo de execução e o número de iterações para redes grandes. Para redes pequenas, estes dois parâmetros aumentam.

Para redes menores, a quantidade de nós não redundantes (possíveis GWs) é maior. Por isso, quanto menor o valor de K desejado mais iterações são necessárias para se determinar os *gateways* da rede e, conseqüentemente, o tempo de execução aumenta. Para redes grandes ocorre o oposto. A quantidade de nós não redundantes é menor e, conseqüentemente, achar os melhores candidatos a GWs na rede demanda menos iterações e menor tempo de execução do algoritmo.

O parâmetro que mantém o mesmo padrão de variação é o máximo salto. Em todos os casos, o aumento do valor de K minimiza ou torna aproximadamente constante o máximo salto da rede. Isso é esperado, uma vez que aumentar o número de K , aumenta o número de *clusters* e diminui a quantidade de nós em um mesmo *cluster*, o que acarreta a diminuição do maior salto. Essa diminuição do maior salto é exatamente o objetivo do nosso problema de otimização, que como demonstrado através da Tabela 2 e ilustrado pelas Figuras 3 e 5, foi satisfatoriamente alcançado.

6. CONSIDERAÇÕES FINAIS

Este trabalho propõe a aplicação do algoritmo *K-means-net*, que trata da aplicação de um algoritmo inteligente (*K-means* modificado) em conjunto com teoremas de substituição a uma rede de sensores sem fio de múltiplos saltos, a fim de encontrar os K candidatos a *gateways* que minimizem a maior distância dos nós aos GWs.

O objetivo foi alcançado através das funções *folha()* e *t_step()*, que encontram os GWs reduzindo o espaço de busca de cada *cluster* através do aumento de t e, resultando apenas em nós não redundantes, ou seja, nós que são bons candidatos a serem GWs.

Os resultados mostram que o algoritmo *K-means-net* é eficiente para dividir *clusters* e encontrar os K bons candidatos a *gateways* dentro da RSSF. Infelizmente não é possível garantir GWs ótimos, contudo o algoritmo se mostrou eficiente, retornando boas respostas.

AGRADECIMENTOS

Os autores deste artigo agradecem à FAPEMIG (APQ-01366-16) e ao CNPq pelo apoio financeiro.

REFERÊNCIAS

- ABDELKHALEK, Ons; KRICHEN, Saoussen; GUITOUNI, Adel. A genetic algorithm based decision support system for the multi-objective node placement problem in next wireless generation network. **Applied Soft Computing**. vol.33. 2015. p. 278-291.
- HUANG, Shu-Qiang; ZHANG, Zhen; LI, Yang; LIU, Zhu-Song; LI, Young-Hui. Deployment optimization of multi-hop wireless networks based on substitution graph. **Information Sciences**. vol. 400-401. 2017. p. 129-141.
- KANUNGO, Tapas; MOUNT, David M.; NETANYAHU, Nathan S.; PIATKO, Christine D.; SILVERMAN, Ruth; WU, Angela Y. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. **IEEE Trans. Pattern Anal. Mach. Intell.** vol. 24. n. 7. 2002. p. 881-892.
- LIN, Chun-Cheng. Dynamic router node placement in wireless mesh networks: A PSO

approach with constriction coefficient and its convergence analysis. **Information Sciences**. vol. 232. 2013. p. 297-308.

LIN, Chun-Cheng; CHEN, Teng-Huei; CHIN, Hui-Hsin. Adaptive router node placement with gateway positions and QoS constraints in dynamic wireless mesh networks. **Journal of Network and Computer Applications**. vol. 74. 2016. p. 149-164.

MAGÁN-CARRIÓN, Roberto; RODRÍGUES-GÓMEZ, Rafael A.; CAMACHO, José; GARCÍA-TEODORO, Pedro. Optimal relay placement in multi-hop wireless networks. **Ad Hoc Networks**. vol. 23. 2016. p.23-36.

REBAI, Maher; BERRE, Matthieu Le; SNOUSSI, Hichem; HNAIEN, Faicel; KHOUKHI, Lyes. Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. **Computers & Operations Research**. vol. 59. 2015. p. 11-21.

YOON, Y; KIM, Y. H. An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks. **IEEE Transactions on Cybernetics**. vol. 43. n. 5. 2013. p. 1473-1483.