Formulação de programação linear inteira para um problema de k-cobertura e m-conectividade em redes de sensores

José F. R. Fonseca
Universidade Federal de Minas Gerais
Av. Antonio Carlos, 6627 - Pampulha, Belo Horizonte - MG, Brasil, 31270-901
jose.fonseca@dcc.ufmg.br

Thiago F. Noronha
Universidade Federal de Minas Gerais
Av. Antonio Carlos, 6627 - Pampulha, Belo Horizonte - MG, Brasil, 31270-901
tfn@dcc.ufmg.br

Iago A. Carvalho
Universidade Federal de Alfenas
Av. Jovino Fernandes de Sales, 2600 - Santa Clara, Alfenas - MG, Brasil, 37133-840
iago.carvalho@unifal-mg.edu.br

RESUMO

Neste trabalho apresentamos um programa linear de inteiros (Integer-Linear program, ILP) para o problema K-Cobertura e M-Conectividade (K-Coverage M-Connectivity, KCMC). O problema de otimização KCMC pertence a categoria NP-Difícil e aparece em aplicações de redes de sensores sem fio (Wireless Sensor Networks, WSN) onde se deseja minimizar o número de sensores da rede enquanto garantindo dadas expectativas de qualidade de serviço e resistência à falhas de sensores. Nossa principal contribuição consiste numa formulação ILP para o problema KCMC que aprimora descrições encontradas em trabalhos relacionados ao garantir resistência a falhas em quaisquer sensores na rede. Apresentamos também testes de desempenho de uma implementação de nossa formulação ILP.

PALAVRAS CHAVE. Redes de sensores sem fio. Problema K-Cobertura M-Conectividade. Programas lineares de inteiros.

Otimização Combinatória. Pesquisa Operacional em Telecomunicações. Programação Matemática.

ABSTRACT

In this paper, we present a linear integer program (ILP) for the K-Coverage M-Connectivity (KCMC) problem. This NP-Hard optimization problem is recurrent in applications of wireless sensor networks (WSN) where it is desirable to minimize the number of sensors while guaranteeing expectations of quality of service and resistance to sensor failures. Our main contribution consists of an ILP formulation for the KCMC problem that improves upon the descriptions of the problem found on related works by guaranteeing resistance to failures in any of the sensors in the WSN. We also present performance tests of our ILP implementation.

https://proceedings.science/p/157559?lang=pt-br

KEYWORDS. Wireless sensor networks. K-Coverage M-Connectivity problem. Integer Linear Programs.

Combinatorial Optimization. Operations Research in Telecommunications. Mathematical Programming.

1. Introdução

Um sensor sem fio é um dispositivo compacto e de baixo custo utilizado para coletar, processar e transmitir dados sem a necessidade de cabos ou fios para comunicação. Ele é projetado para monitorar (ou *cobrir*) uma área circular ao seu redor chamada *área de cobertura*. O raio dessa área circular é chamado *raio de cobertura*. Dois sensores posicionados a uma distância menor do que um determinado *raio de comunicação* um do outro são ditos *comunicantes*.

Sensores são normalmente usados em grupos chamados *redes de sensores sem fio*. Uma rede de sensores sem fio (do inglês *Wireless Sensor Networks* - WSNs) consiste num grupo de sensores interconectados. Cada sensor em uma WSN (também chamado de *nodo* ou *nó*) coleta dados em sua área de cobertura e os transmite para outros sensores na WSN. Um sensor em uma WSN pode também retransmitir dados que ele mesmo não coletou, mas que recebeu de outros sensores. Combinando a cobertura de vários sensores, WSNs permitem a coleta de dados em áreas maiores que a cobertura de um único sensor. Transmitindo dados de sensor em sensor, WSNs permitem comunicação por distancias maiores que o raio de comunicação de um único sensor e possibilitam contornar obstáculos e acidentes geográficos [Petrioli et al., 2014]. WSNs são usadas em várias indústrias (Akyildiz et al. [2002]; Kandris et al. [2020]) como agricultura de precisão (Thakur et al. [2019]), monitoramento ambiental (Olatinwo e Joubert [2020]), e detecção de ameaças (Al-Mousawi e AL-Hassani [2018]). O objetivo de uma WSN é que os dados coletados por todos os seus sensores sejam transmitidos para certos sensores dotados de características especiais, chamados de *bases* (em inglês *sinks*) (Ghosh e Das [2008]; Zhu et al. [2012]; Tripathi et al. [2018]; Farsi et al. [2019]).

Um sensor base é capaz de processar, transferir ou armazenar os dados coletados ou recebidos de forma diferente dos demais sensores na WSN. Um sensor base pode ter acesso a recursos de processamento ou armazenamento que não estão disponíveis para os demais sensores, ou mesmo uma conexão com alguma outra rede e representam toda a informação obtida a partir da área monitorada pela WSN.

Algumas WSNs podem ser construídas e utilizadas para monitorar um conjunto de *pontos de interesse* (do inglês *points of interest* - POIs). Quando um sensor de uma WSN cobre um POI, dizemos que o POI é *servido* pelo sensor. Um mesmo sensor pode cobrir vários POIs. Apenas sensores que cobrem um determinado POI podem *coletar* os dados do POI. WSNs utilizadas para monitorar um conjunto de POIs tem por objetivo comunicar os dados coletados para uma base. Essas WSNs podem ter sensores que não cobrem nenhum POI, mas são usados para comunicar dados. Quando uma WSN é capaz de coletar dados de todos os POIs e transmitir os dados para alguma base, a WSN é dita *conexa* (Mahapatro e Khilar [2013]). Em certas situações, desejamos que vários sensores de uma WSN cubram um mesmo POI. Se cada um dos POIs de uma WSN conexa é coberto por ao menos K sensores diferentes, dizemos que a rede tem a propriedade de *K-qualidade de serviço* ou *K-cobertura*. Redes com requisitos de qualidade de serviço representados por maiores valores de K frequentemente requerem também maiores quantidades de sensores, o que aumenta os custos de implantação e manutenção da rede.

Sensores estão sujeitos a *situações de falha*, nas quais estão incapazes de coletar ou transmitir dados. Sensores nessas situações são ditos *inoperantes* (em inglês *offline*). Sensores podem ficar inoperantes devido a insuficiência da carga em baterias ou falhas na alimentação elétrica, defeitos de fabricação, desgaste ou obstrução de componentes como lentes e antenas, condições ambientais adversas como chuvas ou poeira, ou interferência humana acidental e intencional. Um sensor que se torna inoperante em uma WSN pode comprometer a qualidade do serviço na rede, deixando um POI sem cobertura suficiente. Um sensor que se torna inoperante também pode tornar a WSN

https://proceedings.science/p/157559?lang=pt-br

desconexa ao tornar impossível que dados coletados em um ou mais POIs sejam transmitidos para alguma base.

O *problema da K-cobertura* recebe um conjunto de POIs, uma WSN e um valor para K como entrada. Na saída, retorna o menor subconjunto dos sensores da WSN que formam uma rede conexa em que cada POI é coberto por ao menos K sensores diferentes (Zhao e Gurusamy [2008]). O problema da K-cobertura pertence à classe NP-completo de complexidade (Ke et al. [2007]). É importante ressaltar que a qualidade de serviço em uma WSN não diz respeito à *estabilidade* do serviço, ou a capacidade da rede de se manter conexa em situações adversas.

WSNs que permanecem conexas mesmo se um ou mais sensores estiverem inoperantes simultaneamente são ditas *robustas* ou *resistentes a falhas*. Uma WSN que permanece conexa mesmo se qualquer um de seus sensores falhar é dita *2-conexa*, ou que detém a propriedade de *2-conectividade*. De forma generalizada, uma WSN que tem a propriedade de *M-conectividade* permanece conexa mesmo se qualquer subconjunto de até $M-1$ de seus sensores sofrerem falhas que os deixem inoperantes simultaneamente.

O *problema da M-conectividade* (também chamado na literatura de problema da K-conectividade) recebe um conjunto de POIs, uma WSN e um valor para M como entrada. Na saída, o problema retorna o menor sub-conjunto dos sensores da WSN que formam uma rede em que todo POI deve ser coberto por ao menos um sensor, e a rede deve permanecer conexa mesmo no caso de falhas em $M-1$ sensores ao mesmo tempo. O problema da M-conectividade pertence à classe NP-difícil (em inglês *NP-hard*) de complexidade (Gupta et al. [2016]). É importante ressaltar que a robustez em uma WSN não diz respeito à *qualidade* do serviço, tomando como suficiente apenas um sensor para cobrir cada POI.

WSNs projetadas para manter cobertura de K sensores em cada POI e serem resistentes a $M-1$ falhas em sensores devem manter as propriedades de K-cobertura e M-conectividade simultaneamente. Aplicações de WSNs podem ser beneficiadas pela identificação do menor conjunto de sensores capaz de manter as propriedades desejadas. A identificação de tal conjunto consiste no *problema da K-cobertura e M-conectividade* (em inglês *K-coverage M-connectivity* - KCMC). O problema KCMC recebe na entrada um conjunto de POIs, uma WSN, e valores para K e M. Na saída, retorna o menor subconjunto dos sensores da WSN que formam uma rede conexa com as propriedades de K-cobertura e M-conectividade. O problema KCMC foi recentemente estudado em vários trabalhos (Azharuddin et al. [2015]; Rebai et al. [2015]; Gupta et al. [2016]; Al-Karaki e Gawanmeh [2017]; Jehan e Punithavathani [2017]; Khanna e Chaturvedi [2018]; Gupta e Jha [2019]; Farsi et al. [2019]; Barkhoda e Sheikhi [2020]). Estudos sobre o problema KCMC estão relacionados com estudos nos dois sub-problemas que o compõem, o problema da K-Cobertura e o problema da M-Conectividade. Os três problemas pertencem à classe de complexidade NP-difícil (Ke et al. [2007], Zhao e Gurusamy [2008], Gupta et al. [2016]), de forma que o esforço computacional necessário para encontrar resultados ótimos cresce exponencialmente em proporção ao número de POIs e sensores na WSN.

Trabalhos que abordam o problema KCMC aceitam soluções que não são robustas, cujas WSNs obtidas podem ficar desconexas em caso de falhas simultâneas em subconjuntos de até $M-1$ sensores. Tomemos como exemplo a WSN ilustrada na figura 1, em que sensores estão representados por triângulos, POIs estão representados por asteriscos e o sensor-base por um "X" inscrito em um circulo. As áreas de cobertura de cada sensor coincidem com suas áreas de comunicação, e estão delimitadas por linhas pontilhadas. Para melhor visibilidade, os limites das áreas de cobertura e comunicação do sensor i_0 estão representadas por uma linha sólida. A WSN ilustrada na figura 1 garante que cada POI está coberto por ao menos 4 sensores diferentes (ou seja, a WSN ilustrada tem

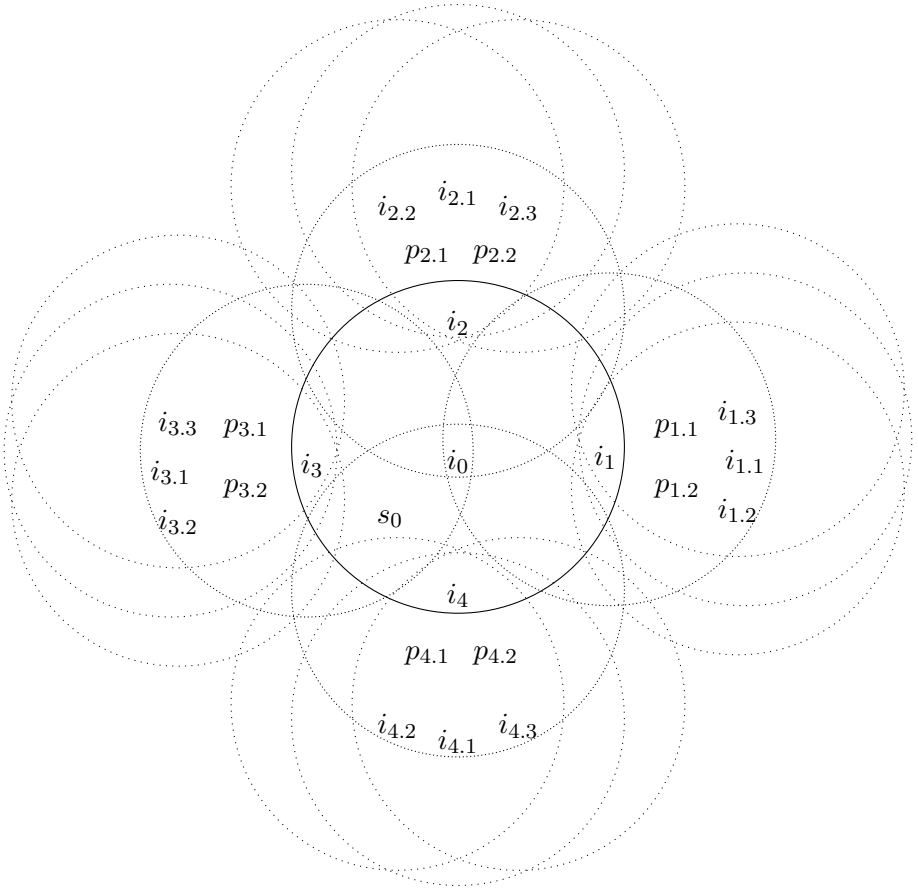


Figura 1: Uma instância de WSN na qual um único sensor inoperante torna a rede desconexa

a propriedade de K-cobertura para $K = 4$). Trabalhos relacionados afirmam que a WSN ilustrada na figura 1 tem a propriedade de M-Conectividade para $M = 3$, uma vez que cada sensor está conectado com ao menos três outros sensores. Contudo, acreditamos que a WSN ilustrada na figura 1 não é robusta, uma vez que uma falha em um único sensor tornaria a rede desconexa. No caso de falha do sensor i_0 a WSN ficaria desconexa pois não seria possível transmitir os dados coletados nos POIs $p_{1.1}$, $p_{1.2}$, $p_{2.1}$ e $p_{2.2}$ para a base s_0.

Neste trabalho, apresentamos uma formulação ILP para o problema KCMC que obtém soluções com as propriedades K-Cobertura e M-Conectividade, sempre com a menor quantidade possível de sensores. Nosso trabalho preenche a lacuna identificada na literatura, em que algoritmos apresentados em trabalhos relacionados não resultam em WSNs robustas.

2. Trabalhos relacionados

Rebai et al. [2015] estudaram uma versão do problema da K-Cobertura na qual POIs e sensores são posicionados em uma grade de formato regular. Propuseram uma formulação ILP, uma heurística de busca local, e um algoritmo genético para o problema, demonstrando com experimentos computacionais que a formulação ILP era capaz de obter boas soluções para grades de até $14 \cdot 14 = 196$ POIs e sensores. Rebai et al. [2015] também mostraram que seu algoritmo genético teve um desempenho superior ao da heurística de busca local em grades de até $50 \cdot 50 = 2500$ POIs e sensores.

Zhao e Gurusamy [2008] também estudaram o problema da K-Cobertura, considerando que cada sensor tem uma bateria de capacidade limitada, sendo capaz de monitorar todos os POIs em seus arredores, transmitir e receber dados de uma rede de sensores, e temporariamente interromper o monitoramento de POIs e transmissões para poupar a capacidade da bateria. Nesse contexto, o objetivo é preparar uma programação de atividades de cada sensor de forma a maximizar o tempo em que cada POI permaneça coberto por ao menos K sensores diferentes enquanto a rede de sensores permanece conecta.

Zhao e Gurusamy [2008] estudaram também outras duas variações do problema da K-Cobertura. Na primeira, um sensor pode escolher observar apenas um subconjunto dos POIs em seus arredores, conservando a carga de sua bateria. Os autores demonstraram que nessa variação o problema pode ser resolvido em tempo polinomial com um algoritmo de fluxo em redes. Na segunda variação proposta para o problema da K-Cobertura, um sensor sempre deve observar todos os POIs em seus arredores a todo tempo. Essa variação foi provada como pertencente à classe de complexidade NP-Completo. Os autores então propuseram limites inferiores e superiores para o problema, além de um algoritmo de aproximação e uma heurística *ad-hoc*.

Gupta et al. [2016] propuseram um algoritmo genético para o problema KCMC. Os autores adaptaram e extenderam a configuração proposta por Rebai et al. [2015] com um novo esquema de codificação de cromossomos e novos operadores evolutivos de baixa complexidade computacional. Em experimentos computacionais, realizados em instancias encontradas na literatura, os autores obtiveram melhores resultados com seu algoritmo genético que com o de Rebai et al. [2015].

Al-Karaki e Gawanmeh [2017] estudaram uma variação do KCMC na qual o objetivo era maximizar o tempo em que uma rede de sensores sem fio, de bateria limitada e organizados em formato de grade, conseguiria cobrir cada POI com K diferentes sensores enquanto cada sensor permanecesse conectado com M outros. Os autores exploraram cinco possíveis formas de calcular o tempo em que a rede permaneceria funcional. O problema KCMC foi modelado como uma rede fechada de filas, e resolvido com o uso de um algoritmo de convolução e três heurísticas gulosas.

Jehan e Punithavathani [2017] propuseram o uso de heurísticas baseadas no algoritmo de busca de gravitação oposta (do inglês, *Oppositional Gravitational Search algorithm*), enquanto Gupta e Jha [2019] propuseram um algoritmo de otimização baseado em biogeografia. As duas soluções propostas obtiveram melhores resultados que o algoritmo genético de Rebai et al. [2015] em experimentos usando instâncias encontradas na literatura. Contudo, não foi possível encontrar um trabalho que comparasse os resultados de Jehan e Punithavathani [2017] ou os de Gupta e Jha [2019] com os do algoritmo genético proposto por Gupta et al. [2016].

Barkhoda e Sheikhi [2020] estudaram o problema KCMC em ambientes bidimensionais e tridimensionais, nos dois casos usando o algoritmo IIC (do inglês, *Immigrant Imperialist Competitive algorithm*). Em experimentos bidimensionais, os resultados foram superiores aos do algoritmo genético de Gupta et al. [2016] e aos do algoritmo de otimização baseado em biogeografia de Gupta e Jha [2019]. Em experimentos tridimensionais, o algoritmo IIC proposto por Barkhoda e Sheikhi [2020] superou o desempenho de três outros encontrados na literatura.

Uma lista mais completa de algoritmos para os problemas de cobertura e conectividade em redes de sensores sem fios pode ser encontrada na revisão da literatura realizada por Farsi et al. [2019].

3. Programa Linear Inteiro - ILP

Obtemos soluções ótimas para o problema KCMC por meio de programas lineares inteiros (em inglês *Integer Linear Programs - (ILP)*). Um ILP consiste em uma função objetivo e um conjunto de restrições que sejam transformações lineares, sendo todas as variáveis inteiros escalares

ou vetores de inteiros. Resultados (também chamados de *soluções*) de um programa ILP consistem em valores para as variáveis utilizadas que respeitem todas as restrições. Para otimizar soluções do problema KCMC usando ILP, buscamos configurações das variáveis que levem a função objetivo a assumir valores mínimos. Algoritmos para encontrar soluções para programas ILP são *exatos*, obtendo sempre os melhores resultados possíveis na função objetivo. Contudo podem ter grande custo computacional, em alguns casos testando todas os valores possíveis para as variáveis.

Existem vários recursos matemáticos que minimizam o esforço computacional necessário para obter soluções de programas ILP. Por exemplo o algoritmo SIMPLEX ou o coeficiente de correlação de Pearson. Existem pacotes de *software* que implementam tais recursos de forma genérica, permitindo sua aplicação em novas formulações ILP. Estes pacotes são chamados de *solvers* (Luppold et al. [2018]). Dois *solvers* populares seriam o IBM CPLEX (Cplex [2009]) e o Gurobi (Gurobi Optimization, LLC [2021]). Implementamos nosso programa ILP usando o *solver* Gurobi e realizamos testes.

Nosso ILP é baseado nos *Problemas da Árvore de Steiner* (em inglês *Steiner Tree Problems - STP*). O STP pertence a classe de complexidade NP-Completo (Karp [1972]). Dado um grafo conexo $G = (V, E)$, onde N é o conjunto de vértices e E o conjunto de arestas, e cada aresta $(u, v) \in E$ tem um custo $c_{uv} \in \mathbb{N}_+$. Seja $T \subset V$ um conjunto qualquer de vértices denominado "conjunto de vértices terminais". Uma *Árvore de Steiner* no grafo G seria uma árvore composta apenas por vértices de G, contendo todos os vértices em T e possivelmente outros vértices do conjunto resultante $V \setminus T$. Esses vértices são chamados de *vértices de Steiner*. Nosso programas ILP buscam encontrar árvores de Steiner com a menor quantidade de vértices de Steiner, pois todas as arestas tem o mesmo custo.

Descrevemos uma instância de WSN como par de grafos (C, G), em que C é chamado de *grafo de cobertura* e G de *grafo de comunicação*. Nessa instância temos os conjuntos P de POIs e V de sensores que não são bases. Assumimos uma única base representada por s. O grafo C contém todos os elementos de P e o subconjunto $V_c \in V$ de sensores, tendo uma aresta ligando cada POI $p \in P$ com os sensores $i \in V_c$ que o cobrem. O grafo G contém todos os sensores $i \in V$, contendo uma aresta ligando cada par de sensores que se comunicam diretamente. Nos referimos ao conjunto de sensores como IS (em inglês *installation spots*) porque consideramos que cada sensor poderia ser "desativado"ou "desinstalado"em um esquema de agendamento. Nosso programa ILP identifica o menor conjunto de sensores que devem ser "instalados".

Nosso programa ILP identifica o menor conjunto de sensores que devem ser instalados buscando M arvores de Steiner disjuntas em vértices, em que I_S são os sensores candidatos a vértices de Steiner e P os vértices terminais. Definimos $N = P \cup I \cup \{s\}$, em que s é o sensor-base e I o conjunto IS. O conjunto $A = A^C \cup A^G \cup A^s$ contem arestas divididas em três sub-conjuntos disjuntos. O sub-conjunto A^C contém arestas $(p, i) \in A^C$ relacionando cada POI $p \in P$ com os IS $i \in I$ onde sensores, se instalados, cobririam o POI p. O subconjunto de IS onde sensores cobririam especificamente o POI p é denotado por $I(p)$. O sub-conjunto A^G contém arestas $(i, j) \in A^G$ ligando sensores comuns $i \in V$ aos IS $j \in I \setminus \{i\}$ com os quais comunicação seria possível. O sub-conjunto A^s contém arestas $(i, s) \in A^s$ ligando os IS ($i \in I$) com sensor-base s. Seja $L = \{1, \ldots, M\}$ o intervalo de 1 a M. Esses conjuntos são usados para definir as variáveis e restrições de nosso programa ILP.

Nosso ILP utiliza as variáveis $\mathbf{x} \in |I| \times |L|$, sendo que $x_i^\ell = 1$ se o IS $i \in I$ pertence a ℓ-ésima árvore de Steiner, e $x_i^\ell = 0$ em caso contrário. As variáveis binárias de fluxo $\mathbf{y} \in |A| \times |P| \times |L|$ devem ser configuradas de tal forma que $y_{ij}^{v\ell} = 1$ se a aresta $(i, j) \in A$ é utilizada para enviar uma unidade de fluxo do POI $v \in P$ para a base s, ou $y_{ij}^{v\ell} = 0$ em caso contrário.

Nosso programa ILP consiste na função objetivo (1) e nas restrições (2)–(7). A função-objetivo (1) minimiza a quantidade de sensores do conjunto IS instalados, ao minimizar o somatório das variáveis $\mathbf{x}$. A restrição (2) garante que cada IS só pode fazer parte de uma árvore de Steiner. A restrição (3) é uma clássica forma de conservação de fluxo que demanda fluxo M de cada POI até a base. A inequação (4) garante o valor correto das variáveis $\mathbf{x}$ relacionando-as com $\mathbf{y}$. O domínio das duas variáveis é dado pelas restrições (6) e (7), respectivamente. Finalmente, a restrição (5) garante a cobertura mínima em cada POI.

$$\min \sum_{i \in I} \sum_{\ell \in L} x_i^\ell \tag{1}$$

$s.a.$

$$\sum_{\ell \in L} x_i^\ell \leq 1, \quad \forall i \in I \tag{2}$$

$$\sum_{(i,j) \in A} y_{ij}^{p\ell} - \sum_{(j,i) \in A} y_{ji}^{p\ell} = \begin{cases} 1, \text{ se } i = p \\ \text{-1, se } i = s \\ 0, \text{ em caso contrário} \end{cases} \quad \forall p \in P, \ell \in L, i \in I \tag{3}$$

$$\sum_{(i,j) \in A} y_{ij}^{p\ell} \leq x_i^\ell, \quad \forall p \in P, \ell \in L, i \in I \tag{4}$$

$$\sum_{\ell \in L} \sum_{i \in I(p)} x_i^\ell \geq k, \quad \forall p \in P \tag{5}$$

$$x_i^\ell \in \{0,1\}, \quad \forall i \in I, \ell \in L \tag{6}$$

$$y_{ij}^{p\ell} \in \{0,1\}, \quad \forall (i,j) \in A, p \in P, \ell \in L \tag{7}$$

4. Experimentos e Resultados

Implementamos nosso ILP utilizando o pacote de *software* comercial Gurobi Optimization, LLC [2021]. Pacotes de *software* similares são chamados de *solvers*. Avaliamos o desempenho de nosso ILP por meio de experimentos computacionais. Os experimentos foram realizados em um ambiente com 16Gb de memória e processador Intel Xeon de 3.1GHz, utilizando o sistema operacional Amazon Linux 2.

Os experimentos computacionais avaliaram diferentes classes de instâncias criadas de forma *ad-hoc* para este trabalho, seguindo a metodologia de Gupta et al. [2016]. Classes de instâncias são diferenciadas quantidade de POIs ($|P|$), quantidade de ISs ($|I|$), e os valores de K e M para o problema KCMC que pode ser resolvido nas instâncias da classe com resultados ótimos. Em cada instância, as posições dos POIs e ISs foram definidas utilizando uma distribuição uniforme, sendo estes posicionados dentro de uma área quadrada de dimensões $300x300$ unidades. Além disso, os valores dos raios de cobertura e de comunicação dos sensores foram padronizados em todas as instâncias como 50 e 100 respectivamente. O sensor-base de cada instância é posicionado no centro da mesma, na posição $(150, 150)$. Criamos 9 classes de instâncias, todas com 100 POIs e 100, 300 ou 500 ISs. Utilizamos e valores inteiros para K em M de 1 a 2 sempre mantendo $K \geq M$. Criamos 10 instâncias diferentes para cada classe utilizando diferentes valores para o gerador de números pseudo-aleatórios, totalizando 90 diferentes instâncias.

Executamos nosso ILP para cada uma das instâncias que geramos e reportamos os resultados na Tabela 1. Nesta tabela, as colunas **I**, **K** e **M** representam, respectivamente, as quantidades

Tabela 1: Tabela de resultados de nosso ILP

$\|\|I\|\|$	K	M	#opt	#int	#n0	#nodes	LB	UB	%GAP	time(s)
100	1	1	10	10	10	0.2	5.6	5.6	0.0	11.69
100	2	1	10	10	10	0.1	11.6	11.6	0.0	7.78
100	2	2	10	10	10	0.5	11.4	11.4	0.0	102.08
300	1	1	10	10	10	0.1	5.0	5.0	0.0	1807.38
300	2	1	5	10	10	0.0	5.0	5.0	0.5	1435.26
300	2	2	2	9	10	0.2	3.0	3.0	0.7	1900.51
500	1	1	1	10	10	0.0	0.4	0.4	0.9	177.24
500	2	1	-	10	10	0.0	0.0	0.0	1.0	-
500	2	2	-	10	10	0.0	0.0	0.0	1.0	-

de sensores e os valores de K e M utilizados em cada instância, sendo que todas as instâncias tinham 100 POIs. A coluna **#opt** mostra em quantas das 10 instâncias foi possível obter resultados ótimos dentro do limite de tempo de uma hora (3600 segundos). As colunas **#int** e **#n0** mostram, respectivamente, em quantas das 10 instâncias foi possível obter resultados inteiros e resultados para os quais a função objetivo tem valores diferentes de 0, sempre dentro do limite de tempo. As colunas **LB** e **UB** apresentam, respectivamente, a média dos limites inferior e superior para valores da função objetivo calculados. A coluna $\%$**GAP** mostra a média da diferença entre os melhores valores obtidos para a função objetivo e um valor ótimo estimado de forma heurística, que não necessariamente poderia ser atingido. A coluna **time(s)** mostra a média de tempo necessária para obter valores ótimos. Observe que em muitos casos não foi possível obter valores ótimos no limite de tempo de uma hora (3600 segundos). Em casos onde não foi possível obter resultados ótimos, os valores da Tabela 1 estão preenchidos por "$-$".

Podemos perceber na Tabela 1 o impacto que o valor de M tem no tempo médio de processamento das instâncias. O único parâmetro com valores diferentes na segunda e terceira linha foi M, contudo o tempo de processamento médio subiu de $7,78$ segundos para $102,08$ segundos. Similarmente, M é único parâmetro diferente nas linhas 5 e 6, mas o tempo médio de processamento subiu de $1435,26$ segundos para $1900,51$ segundos, na sexta linha e só foi possível obter resultados ótimos em duas das dez instâncias testadas.

Podemos perceber na Tabela 1 o aumento do tempo médio de processamento com a quantidade de sensores na instância. Nas 5, 6 e 7, não foi possível obter resultados ótimos dentro do limite de tempo de 3600 segundos para todas as instâncias, e nas linhas 8 e 9 não foi possível obter resultados ótimos dentro do limite de tempo em nenhuma das instâncias. É importante ressaltar que nossos experimentos utilizaram sempre os mesmos valores para a quantidade de POIs, o tamanho da área em que POIs e sensores foram dispersos e o comprimento dos raios de cobertura e comunicação de cada sensor. Esses parâmetros tem impacto na quantidade de conexões entre sensores na WSN e na densidade do grafo de comunicação G.

O tempo de processamento de um ILP é afetado pelo tamanho de suas variáveis. Em nosso ILP, utilizamos as variáveis x e y que são vetores binários. x terá uma quantidade de posições igual ao tamanho do vetor L, que tem M posições, multiplicada pelo tamanho do vetor I, que tem uma posição para cada sensor que cobre um POI. y terá uma quantidade de posições igual ao tamanho do vetor L, multiplicada pelo tamanho do vetor P (que tem uma posição para cada POI), multiplicada pelo tamanho do vetor A, que tem uma posição para cada aresta no grafo G. Assim, o tempo de

processamento de nosso modelo será maior em instâncias com mais conexões entre os sensores, mas não necessariamente em instâncias com mais sensores caso esses tenham poucas conexões entre si. Por esse motivo explicamos como é possível que existam instâncias com 300 sensores que tenham tempo de processamento maior do que instâncias com 500 sensores com os mesmos valores de K e M, como pode ser observado respectivamente nas linhas 4 e 7 da Tabela 1.

5. Conclusões

Nosso programa ILP, diferentemente de soluções apresentadas em trabalhos relacionados, obtém apenas WSN robustas em seus resultados. Nas WSNs obtidas por nosso ILP, mesmo em caso de falhas em até $M - 1$ sensores simultaneamente os dados de cada POI ainda podem ser transmitidos para o sensor-base.

Nosso programa ILP pode tomar um longo tempo para processar instâncias com maiores quantidades de sensores, mas consegue obter eficientemente soluções para instâncias de até 300 sensores em condições como as de nossos testes.

6. Trabalhos Futuros

Trabalhos futuros podem trazer ILPs aprimorados que obtenham WSNs robustas utilizando menos variáveis, e portanto processem instâncias maiores no mesmo limite de tempo. Uma outra abordagem em trabalhos futuros poderia ser o uso de heurísticas no lugar de ILPs, que podem obter resultados mais rapidamente mesmo que não sejam resultados ótimos.

Referências

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., e Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422.

Al-Karaki, J. N. e Gawanmeh, A. (2017). The optimal deployment, coverage, and connectivity problems in wireless sensor networks: revisited. *IEEE Access*, 5:18051–18065.

Al-Mousawi, A. J. e AL-Hassani, H. K. (2018). A survey in wireless sensor network for explosives detection. *Computers & Electrical Engineering*, 72:682–701.

Azharuddin, M., Kuila, P., e Jana, P. K. (2015). Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks. *Computers & Electrical Engineering*, 41:177–190.

Barkhoda, W. e Sheikhi, H. (2020). Immigrant imperialist competitive algorithm to solve the multi-constraint node placement problem in target-based wireless sensor networks. *Ad Hoc Networks*, p. 102183.

Cplex, I. I. (2009). V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157.

Farsi, M., Elhosseini, M. A., Badawy, M., Ali, H. A., e Eldin, H. Z. (2019). Deployment techniques in wireless sensor networks, coverage and connectivity: A survey. *IEEE Access*, 7:28940–28954.

Ghosh, A. e Das, S. K. (2008). Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, 4(3):303–334.

Gupta, G. P. e Jha, S. (2019). Biogeography-based optimization scheme for solving the coverage and connected node placement problem for wireless sensor networks. *Wireless Networks*, 25(6): 3167–3177.

Gupta, S. K., Kuila, P., e Jana, P. K. (2016). Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks. *Computers & Electrical Engineering*, 56:544–556.

Gurobi Optimization, LLC (2021). Gurobi Optimizer Reference Manual. *Available at https://www.gurobi.com*. URL `https://www.gurobi.com`.

Jehan, C. e Punithavathani, D. S. (2017). Potential position node placement approach via oppositional gravitational search for fulfill coverage and connectivity in target based wireless sensor networks. *Wireless Networks*, 23(6):1875–1888.

Kandris, D., Nakas, C., Vomvas, D., e Koulouras, G. (2020). Applications of wireless sensor networks: an up-to-date survey. *Applied System Innovation*, 3(1):14.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, p. 85–103. Springer.

Ke, W.-C., Liu, B.-H., e Tsai, M.-J. (2007). Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete. *IEEE Transactions on Computers*, 56(5):710–715.

https://proceedings.science/p/157559?lang=pt-br

Khanna, G. e Chaturvedi, S. K. (2018). A comprehensive survey on multi-hop wireless networks: milestones, changing trends and concomitant challenges. *Wireless Personal Communications*, 101(2):677–722.

Luppold, A., Oehlert, D., e Falk, H. (2018). Evaluating the performance of solvers for integer-linear programming. Technical report.

Mahapatro, A. e Khilar, P. M. (2013). Fault diagnosis in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 15(4):2000–2026. ISSN 1553-877X.

Olatinwo, S. O. e Joubert, T.-H. (2020). Energy efficiency maximization in a wireless powered iot sensor network for water quality monitoring. *Computer Networks*, 176:107237.

Petrioli, C., Nati, M., Casari, P., Zorzi, M., e Basagni, S. (2014). Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(3):529–539.

Rebai, M., Snoussi, H., Hnaien, F., Khoukhi, L., et al. (2015). Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Computers & Operations Research*, 59:11–21.

Thakur, D., Kumar, Y., Kumar, A., e Singh, P. K. (2019). Applicability of wireless sensor networks in precision agriculture: A review. *Wireless Personal Communications*, 107(1):471–512.

Tripathi, A., Gupta, H. P., Dutta, T., Mishra, R., Shukla, K., e Jit, S. (2018). Coverage and connectivity in wsns: A survey, research issues and challenges. *IEEE Access*, 6:26971–26992.

Zhao, Q. e Gurusamy, M. (2008). Connected k-target coverage problem in wireless sensor networks with different observation scenarios. *Computer Networks*, 52(11):2205–2220.

Zhu, C., Zheng, C., Shu, L., e Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632.