

Um modelo de programação matemática para o problema *Weighted Minimum Broadcast Time*

Alfredo Lima M. S.¹, Luiz Satoru Ochi¹, Bruno Nogueira², Rian G. S. Pinheiro²

¹Instituto de Computação – Universidade Federal Fluminense (UFF)
CEP 24210-346 – Niterói – RJ – Brasil

²Instituto de Computação - IC, Universidade Federal de Alagoas (Ufal)
Maceió, AL – Brasil.

alfredolima@id.uff.br, satoru@ic.uff.br, {bruno, rian}@ic.ufal.br

Abstract. *For a city to be classified as 'smart', should have sensors scattered across it. The dissemination of data in a sensor network is one of the challenges that must be overcome. In particular, a broadcasting problem is the WEIGHED MINIMUM BROADCAST TIME (WMBT). The WMBT is a data dissemination problem whose objective is to find a dissemination scheme that minimizes the time needed to perform the dissemination operation. An application of the WMBT for the process of updating the firmware of devices in a Bluetooth network will be presented. A mathematical model for the WMBT will be presented. Our model is compared with state-of-the-art algorithms for the problem. Experimental results show that it is able to attain competitive results.*

Resumo. *Para uma cidade ser classificada como "inteligente", ela precisa ter sensores espalhados por ela. A disseminação de dados em uma rede de sensores é um dos desafios que deve ser superado. Em particular, um problema de broadcasting é o WEIGHED MINIMUM BROADCAST TIME (WMBT). O WMBT é um problema de disseminação de dados cujo objetivo é encontrar um esquema de disseminação que minimize o número de passos necessários para executar a operação de disseminação. Será apresentada uma aplicação do WMBT para o processo de atualização de firmware de dispositivos em uma rede Bluetooth. Será apresentado um modelo matemático para o WMBT. Esta proposta comparou com adaptações de heurísticas do estado-da-arte. Os resultados experimentais mostram que o modelo pode ser aplicado para resolver o WMBT.*

1. Introdução

Uma rede de dispositivos inteligentes pode ser facilmente modelada como um grafo simples, não direcionado e sem pesos. Em que cada dispositivo é um nó do grafo, e as arestas indicam as conexões diretas entre os dispositivos. No entanto, alguns modelos mais realistas podem considerar o tempo de transmissão, ou seja, ter pesos nas arestas ou vértices do grafo. Um dos objetivos da modelagem de uma rede é observar, analisar e otimizar um processo. Nesse artigo será analisado o processo de disseminação de dados, especificamente, o *broadcasting*. O *broadcasting* é um processo no qual um ou mais nós da rede detêm uma informação que precisa ser enviada para os demais nós [Hedetniemi et al. 1988]. Um problema de *broadcasting* proeminente é o WEIGHTED MINIMUM BROADCAST TIME (WMBT) [Harutyunyan and Kamali 2017].

No WMBT, existe um conjunto de nós chamado de nós fontes, que possuem a informação a ser disseminada na rede. Cada nó deve enviar a informação aos seus vizinhos imediatos um por vez. Os nós que recebem a informação passam a ser emissores. Esse processo se repete até que todos na rede detenham a informação, respeitando as seguintes restrições: (i) o nó detém a informação se for um nó fonte ou se recebeu a informação anteriormente, (ii) qualquer nó pode no máximo enviar a informação a um vizinho por vez, (iii) qualquer cada nó necessita de um tempo interno para processar a informação, após este tempo pode transmitir aos demais, e (iv) o tempo de transmissão entre dois nós diferentes podem ser diferentes.

A motivação original para estudar o WMBT foi um problema de um parceiro industrial para cidades inteligentes. O parceiro construiu um relé fotoelétrico inteligente para ser utilizado no sistema de iluminação pública, o qual consegue identificar o consumo de energia das lâmpadas dos postes de iluminação, defeitos das lâmpadas, entre outras informações, e a partir desses dados é possível processá-los, e assim tomar alguma ação como a solicitação da troca de lâmpadas dos postes. Ou seja, são sensores que ficaram espalhados por toda a cidade, a qual é a base de qualquer cidade inteligente [Su et al. 2011, Perera et al. 2014]. No entanto, como qualquer dispositivo inteligente, o relé fotoelétrico inteligente pode necessitar de atualizações de *firmware* com a finalidade de aprimorar o equipamento. O parceiro requisitou para otimizar o processo de atualização dos dispositivos da rede. Em sua rede, os dispositivos usavam Bluetooth para comunicação em uma intranet. Alguns desses dispositivos (*gateways*) possuíam uma placa GPRS para comunicação externa. Dois tipos de processo de atualização foram considerados nas topologias ponto-a-ponto e *mesh*. Na topologia ponto-a-ponto, os nós funcionam semelhante ao WMBT, de modo que ocorre apenas uma transmissão por vez por dispositivo, enquanto na *mesh* essa restrição não existe e um dispositivo pode atualizar mais de um vizinho ao mesmo tempo. Os experimentos mostraram que no ponto-a-ponto o processo de atualização levou 2 minutos e no *mesh* levou 4,5 horas. A principal razão para essa diferença é que a técnica de *mesh* exigia mais largura de banda, o que, por sua vez, resultava em mais congestionamento na rede e perda de pacotes. Esta observação foi formalmente demonstrada por [Robledo et al. 2020].

O WMBT é uma generalização do MINIMUM BROADCAST TIME (MBT) com pesos nos vértices e/ou nas arestas. O modelo WMBT é mais realista do que o MBT, pois ele representa taxas de atraso de transmissão e/ou tempo de processamento interno de um nó. O MBT é um problema clássico \mathcal{NP} -difícil [Garey and Johnson 1979, problema ND49], a sua prova pode ser encontrada em [Jansen and Müller 1995]. Assim, o WMBT também é pelo menos \mathcal{NP} -difícil. Até onde os autores sabem, na literatura só existe definição do WMBT para o modelo com pesos nos vértices [Harutyunyan and Kamali 2008, Harutyunyan and Kamali 2017], e para o modelo com pesos nas arestas [Koh and Tcha 1991].

O WMBT pode ser formalmente definido da seguinte forma. Seja $G = (V, A, W_v, W_e)$ um grafo conexo ponderado não direcionado, em que V é um conjunto de vértices, A é um conjunto de arestas, $W_v(u)$ é uma função de pesos não-negativos para todo vértice $u \in V$ e $W_e(a)$ é uma função de pesos positivos para toda aresta $a \in A$. Além disso, seja $S = V_0 \subseteq V$ um subconjunto de vértices (nós fontes) que, inicialmente, contêm uma determinada informação a ser disseminada na rede. Sejam V_t um conjunto

de vértices que recebem a mensagem no tempo t ou anteriormente e E_t um conjunto de vértices que podem transmitir a partir do tempo t , com $1 \leq t \leq T_{max}$, em que T_{max} é um limite superior para o tempo de transmissão. A partir dessas definições, deve-se encontrar uma sequência $V_0, A_1, V_1, A_2, \dots, A_k, V_k$ que minimize k , tal que $V_k = E_k = V$. Inicialmente, todo vértice $s \in S$ pertence ao conjunto $E_{W_v(s)}$. Além disso, para cada $t \in \{1, \dots, k\}$, as seguintes restrições são válidas: (i) toda aresta $a \in A_t$ possui exatamente um nó em $E_{t-W_e(a)}$ (ii) toda aresta $a \in A_t$ não compartilha um extremo em comum com os conjuntos A_z , em que $t+1 \leq z \leq t+W_e(a)$, (iii) $V_t \subseteq E_z$, em que $t+W_v(v) \leq z \leq T_{max}$, e (iv) $V_t = V_{t-1} \cup \{v : (u,v) \in A_t, u \in E_{t-W_e((u,v))}\}$.

A Figura 1 em conjunto com as Tabelas 1-2 exemplificam um cenário simples do WMBT. Neste exemplo, $S = \{g_1\}$ representa um conjunto de gateways de rede, enquanto $D = \{d_1, d_2, d_3\}$ um conjunto de dispositivos comuns na rede. É possível encontrar duas soluções, uma viável (Figura 2) e a outra ótima (Figura 3). Em ambas figuras, um vértice com um círculo tracejado indica que o vértice ainda não pode transmitir a mensagem. Um vértice com um círculo completo indica que ele pode transmitir. Uma aresta tracejada indica uma possível transmissão que podem acontecer no instante t , e uma aresta completa indica que a transmissão já ocorreu. Um seta em um aresta tracejada indica que uma transmissão está ocorrendo, enquanto em uma aresta completa que a transmissão finalizou no instante t .

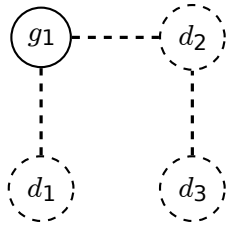


Figura 1. Cenário da rede.

Vértices	g_1	d_1	d_2	d_3
Pesos	1	0	2	1

Tabela 1. Pesos dos vértices.

Arestas	(g_1, d_1)	(g_1, d_2)	(d_2, d_3)
Pesos	7	2	1

Tabela 2. Pesos das arestas.

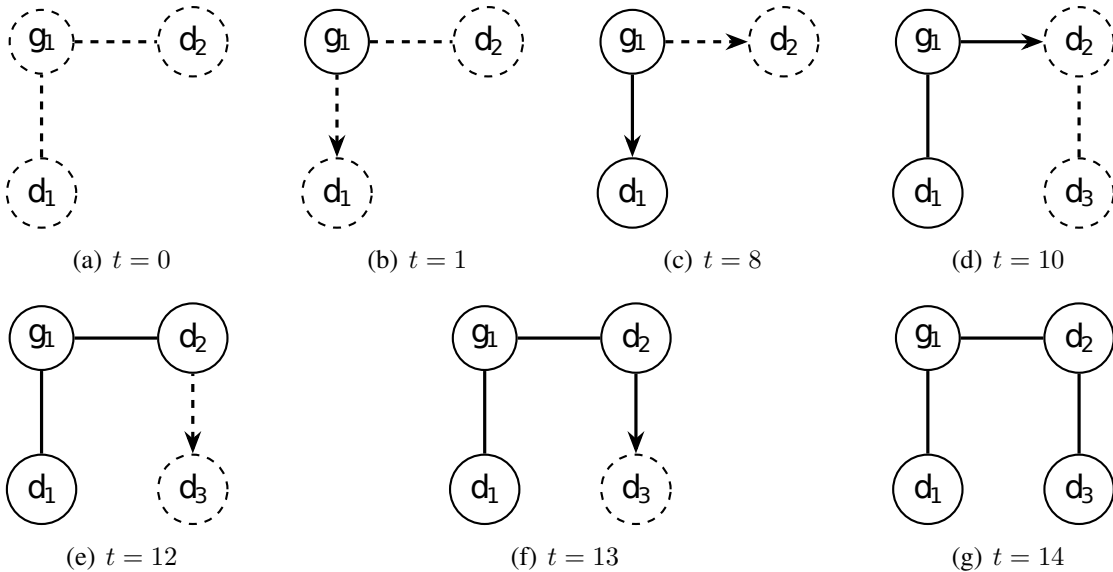


Figura 2. Solução viável.

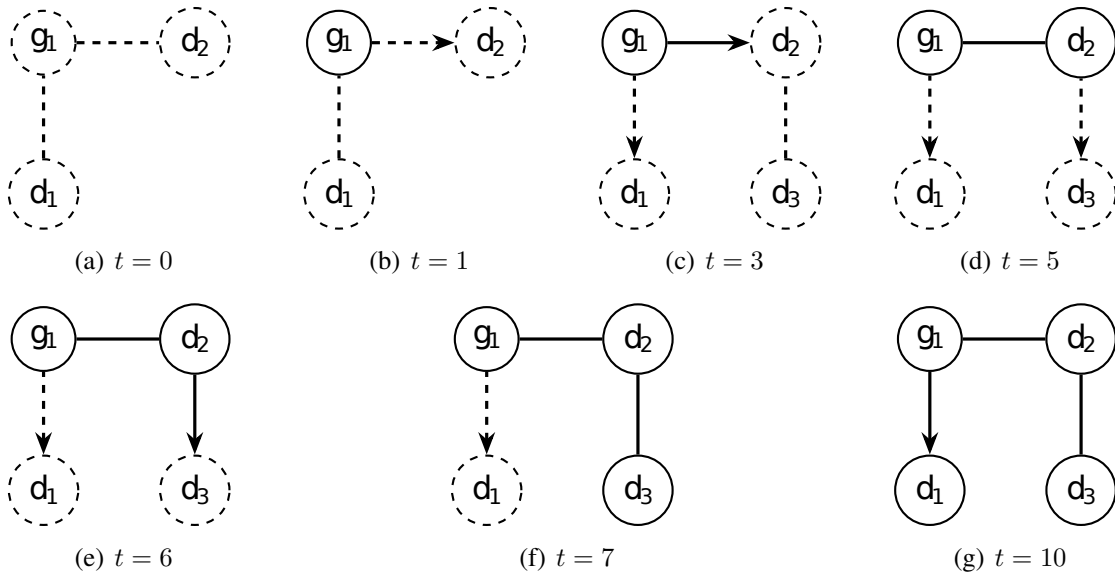


Figura 3. Solução ótima.

O MBT/WMBT possui diversas aplicações em sistemas distribuídos, mais especificamente em Internet das Coisas (IoT), Redes de Sensores Sem Fio (WSNs) e Indústria 4.0 [Shang et al. 2010, Hocoğlu and Genç 2019, Ivanova 2019]. Outras aplicações incluem comunicação entre redes telefônicas [Ivanova 2019], vigilância e reconhecimento [Dekker 2002], robótica [Bucantanschi et al. 2007, Keshavarz et al. 2011], redes de satélites [Chu and Chen 2018] e *direct memory access* (DMA) [Lazard 1992].

A contribuição deste trabalho consiste em apresentar um modelo matemático para o *Weighted Minimum Broadcast Time* (WMBT), o qual pode ser facilmente adaptado para as versões mais simples do WMBT, como somente com pesos nos vértices [Harutyunyan and Kamali 2008, Harutyunyan and Kamali 2017], e para somente pesos nas arestas [Koh and Tcha 1991]. A proposta deste artigo foi comparada com heurísticas do estado-arte. Os resultados experimentais indicam que a proposta é capaz de superar as heurísticas mais conhecidas para o WMBT, porém para instâncias grandes o modelo apresenta dificuldades.

O restante do artigo está organizado da seguinte forma. Na seção 2 é apresentado os trabalhos relacionado ao problema. Na seção 3 descrito o modelo matemático para a resolver o WMBT. Na seção 4 é discutido os resultados computacionais. Finalmente, a Seção 5 contém conclusão e trabalhos futuros.

2. Trabalhos relacionados

Os problemas MBT e WMBT atraíram interesse de vários pesquisadores devido ao seu potencial para modelar muitas aplicações do mundo real. Para o MBT, existem propostas de algoritmos exatos [Scheuermann and Wu 1984, de Sousa et al. 2018, Ivanova 2019], algoritmos aproximados [Elkin and Kortsarz 2003, Kortsarz and Peleg 1992], heurísticas, metaheurísticas e *matheuristic* [Scheuermann and Wu 1984, Hoelting et al. 1996, de Sousa et al. 2018, Hasson and Sipper 2004, Harutyunyan and Wang 2010, Harutyunyan and Jimborean 2014, Lima et al. 2022].

Para o WMBT com pesos apenas nas arestas, existe um algoritmo polinomial que resolve de forma ótima grafos do tipo árvore [Koh and Tcha 1991]. Para a versão com pesos nos vértices, existe um algoritmo guloso e um algoritmo evolucionário [Harutyunyan and Kamali 2008]. Resultados experimentais [Harutyunyan and Kamali 2008] mostram que o algoritmo guloso é melhor que o algoritmo evolucionário. Outros algoritmos [Averbuch et al. 2000] foram propostos para problemas similares ao WMBT, porém a função de custo é calculada de forma diferente dos trabalhos descritos anteriormente [Garey and Johnson 1979, Harutyunyan and Kamali 2008, Koh and Tcha 1991].

Até onde os autores conhecem, para grafos genéricos, não existem algoritmos exatos para o WMBT. Além disso, não existem heurísticas/metaheurísticas para o problema considerando pesos nos vértices e arestas ao mesmo tempo.

3. Modelo de programação matemática para o WMBT

Nesta seção será apresentado o modelo proposto de programação linear inteiro (PLI) para resolver o WMBT. Este modelo foi baseado no modelo do MBT [de Sousa et al. 2018]. Além das adições dos pesos dos vértices e arestas, essa adaptação também considera que o WMBT pode ser multi-fonte.

Seja $G = (V, V_0, A, W_v, W_e)$ um grafo conexo ponderado não direcionado, em que V é o conjunto de vértices, V_0 é o conjunto de vértices fontes, A é o conjunto de arestas, W_v é uma função de inteiros não-negativos de pesos dos vértices e W_e é uma função de inteiros positivos de pesos das arestas. Além disso, considere que $N(i)$ representa o conjunto de vizinhos do vértice $i \in V$, e $V_0 \in V$ é o conjunto de vértices fontes. Para auxiliar o modelo, K_i assume valor 1 caso o vértice $i \in V_0$ e 0, caso contrário. T_{max} um limite superior (*upper bound*) para que qualquer vértice do grafo consiga receber a informação. Por exemplo, $T_{max} = (\sum_{v \in V} W_v(v)) + (\sum_{a \in A} W_e(a))$ é o limite trivial. Por fim, seja T uma variável de decisão que representa o tempo mínimo de transmissão (WMBT), e x_{ij}^t como uma variável binária que tem valor 1 se o vértice i iniciar a transmissão da mensagem para o vértice j no tempo t e 0, caso contrário. Com base nessas definições, o modelo de PLI é definido da seguinte forma:

$$\min T \tag{1}$$

$$\text{s. a } K_i + \sum_{j \in N(i)} \sum_{t=1}^{T_{max}} x_{ji}^t = 1 \quad \forall i \in V \tag{2}$$

$$\sum_{j \in N(i)} x_{ij}^t = 0, \forall i \in V_0 \quad \forall t \in [0, W_v(i)) \tag{3}$$

$$\sum_{j \in N(i)} x_{ij}^t \leq 1 \quad \forall i \in V, \forall t \in [0, T_{max}] \tag{4}$$

$$x_{ik}^\tau \leq 1 - x_{ij}^t \quad \forall (i, j) \in A, \forall t \in [0, T_{max}], \tag{5}$$

$$\forall k \in N(i), \forall \tau \in [t + 1, t + W_e((i, j))]$$

$$x_{ij}^t \leq K_i + \sum_{k \in N(i) \setminus \{j\}} \sum_{\tau=0}^{t - W_e((k,i)) - W_v(i)} x_{ki}^\tau \quad \forall (i, j) \in A, \forall t \in [0, T_{max}] \quad (6)$$

$$\sum_{t=0}^{T_{max}} (t + W_e((i, j)) + W_v(j)) \cdot x_{ij}^t \leq T \quad \forall (i, j) \in A \quad (7)$$

$$T \in \mathbb{N} \quad (8)$$

$$x_{ij}^t \in \mathbb{B} \quad \forall (i, j) \in A, \forall t \in [0, T_{max}] \quad (9)$$

A função objetivo (1) minimiza o tempo de transmissão T . As restrições (2) restringem que todos os vértices devem receber em algum momento a informação, ou iniciar com a ela. As restrições (3) proibem que qualquer nó fonte envie uma informação antes de estar apto para transmitir. As restrições (4) limitam que em cada instante t qualquer vértice faz até uma transmissão. As restrições (5) proibem que um vértice inicie uma outra transmissão enquanto já iniciou uma, ou seja, espera $W_e((i, j))$ até iniciar outra transmissão. As restrições (6) restringem que um vértice só pode enviar caso seja um nó fonte ou que recebeu a informação anteriormente. As restrições (7) impõem que o valor de T deve ser maior ou igual ao tempo que inicia qualquer transmissão acrescido o período da transmissão e o período do processamento interno do vértice. Finalmente, as restrições (8) e (9) definem o domínio das variáveis de decisão.

O modelo anterior descrito é para a versão com pesos nos vértices e nas arestas. No entanto, para a versão com pesos somente nos vértices deve ser considerado algumas mudanças: (i) as restrições (5) devem ser removidas, (ii) os valores $W_e((k, i))$ e $W_e((i, j))$ nas restrições (6) e (7) devem ser alterados para o valor 1. Enquanto, para a versão com pesos somente nas arestas as seguintes mudanças devem ser consideradas: (i) as restrições (2) devem ser removidas, (ii) os valores $W_v(i)$ e $W_v(j)$ nas restrições (6) e (7) devem ser alterados para o valor 0.

4. Resultados computacionais

Nesta seção é apresentado os experimentos computacionais para avaliar a efetividade da proposta do modelo proposto para o WBMT. Nosso modelo de PLI é comparado a adaptações de uma heurística construtiva [Harutyunyan and Kamali 2008], a qual originalmente só considera pesos nos vértices. As duas adaptações foram estendidas para considerar pesos nos vértices e nas arestas. E uma delas foi acrescido uma busca local (refinamento) para o MBT [Slater et al. 1981], porém adaptada para o WMBT.

Todos os experimentos nesta seção foram conduzidos em um Intel Core i7-6700 com 3,40 GHz, 32 GB de RAM, rodando Ubuntu 18.04.5. A adaptação da heurística foi codificada em C++ com compilador g++ 7,5 e com as *flag* '-O3'. Além disso, foi utilizado o IBM Cplex 12.9 para resolver o modelo PLI. Foi adotado o limite de tempo de 3600 s (1 h) para o Cplex para resolver o modelo de PLI foi adicionado.

4.1. Instâncias testadas

Os algoritmos foram testados em um total de 44 instâncias, todas as instâncias são grafos aleatórios. Cada instância é composta por uma árvore aleatória $T = (V, E_T)$ e um grafo aleatório $G_r = (V, E_G)$, ou seja, a instância é um grafo $I = (V, E)$ com $E = E_T \cup E_G$,

assim sendo conexa e aleatória. O grafo aleatório é baseado no modelo $\mathbb{G}(n, p)$, também conhecido como modelo binomial [Gilbert 1959]. Cada grafo $G_r = (V, E_G)$ é gerado com n vértices e cada aresta potencial em E_r é criada com probabilidade p . Os pesos dos vértices e arestas foram criados usando uma distribuição uniforme.

4.2. Comparação entre o modelo PLI e heurística construtiva

Na Tabela 3 é apresentado a comparação entre o modelo PLI e as heurísticas construtivas [Harutyunyan and Kamali 2008]. A primeira coluna indica as características da instância, um grafo $G(n, g, p)$, em que n é a quantidade de vértices do grafo, g é a quantidade nós fontes e p a densidade em porcentagem do grafo. A segunda coluna (HARUTYUNYAN) indica o tempo de transmissão calculado pela adaptação da versão com pesos nas arestas e nos vértices da heurística construtiva [Harutyunyan and Kamali 2008]. A terceira coluna (HARUTYUNYAN+BL) indica o tempo de transmissão combinando a heurística anterior com uma busca local. A quarta e quinta colunas indicam, respectivamente, o tempo de transmissão e tempo de execução do modelo PLI utilizando o *solver* Cplex. O valor em negrito indica que o método obteve melhor desempenho ou igualou ao melhor método, e um astérisco ('*') indica que foi provado a otimalidade, enquanto a instância com travessão ('—') indica que o método não encontrou uma solução viável no tempo reservado.

Como esperado o modelo PLI conseguiu encontrar soluções ótimas em instâncias com poucos vértices e poucas arestas, mas à medida que a instância aumenta quantidade de vértices ou arestas - bem como os pesos -, o modelo tem mais dificuldade em resolver tais instâncias. Isto acontece devido ao aumento do número de variáveis necessárias para instância, tornando o modelo mais computacionalmente desafiador. Das 44 instâncias, o modelo PLI encontrou 31 melhores soluções, e provou a otimalidade de 26 soluções. A busca local na heurística construtiva da literatura melhorou 39 das 44 instâncias.

Tabela 3: Comparação dos algoritmos determinísticos.

Instância	HARUTYUNYAN	HARUTYUNYAN+BL	PLI	
	WMBT	WMBT	WMBT	t (s)
G(24,3,0,A)	35	32*	32*	0.04
G(24,3,0,B)	51	45*	45*	0.08
G(24,3,10,A)	20	19	17*	0.06
G(24,3,10,B)	22	22	19*	0.06
G(24,3,20,A)	17	16	14*	0.09
G(24,3,20,B)	18	17	16*	0.10
G(24,3,30,A)	18	18	15*	0.12
G(24,3,30,B)	19	18	16*	0.15
G(24,3,40,A)	15	14	13*	0.35
G(24,3,40,B)	16	14	12*	0.18
G(24,3,50,A)	16	15	13*	0.23
G(24,3,50,B)	15	14	13*	0.23
G(24,3,60,A)	16	14	13*	0.31
G(24,3,60,B)	15	12	11*	0.21
G(24,3,70,A)	12	10	9*	0.21
G(24,3,70,B)	13	12	10*	0.22
G(24,3,80,A)	15	13	11*	0.41
G(24,3,80,B)	16	15	13*	0.35
G(24,3,90,A)	13	12	11*	0.43
G(24,3,90,B)	13	11	9*	0.42
G(64,1,10,A)	28	27	23*	36.45
G(64,1,15,A)	25	24	22*	42.99

Continua na próxima página

Tabela 3 – continuação da página anterior

Instância	HARUTYUNYAN	HARUTYUNYAN+BL	PLI	
	WMBT	WMBT	WMBT	t (s)
G(64,1,20,A)	30	29	25	3600.00
G(64,1,25,A)	21	21	18*	51.16
G(64,1,10,A)	95	74*	74*	15.23
G(64,1,15,A)	128	128	109*	52.02
G(64,1,20,A)	99	88*	88*	57.86
G(64,1,25,A)	168	168	—	3600.00
G(128,1,10,A)	139	124	—	3600.00
G(128,1,15,A)	214	186	—	3600.00
G(128,1,20,A)	202	197	—	3600.00
G(128,1,25,A)	149	149	—	3600.00
G(128,1,10,A)	29	27	23	3600.00
G(128,1,15,A)	28	27	23	3600.00
G(128,1,20,A)	26	25	24	3600.00
G(128,1,25,A)	22	20	18	3600.00
G(256,1,10,A)	24	23	—	3600.00
G(256,1,15,A)	22	21	—	3600.00
G(256,1,20,A)	18	16	—	3600.00
G(256,1,25,A)	17	16	—	3600.00
G(256,1,10,A)	282	279	—	3600.00
G(256,1,15,A)	239	237	—	3600.00
G(256,1,20,A)	202	197	—	3600.00
G(256,1,25,A)	243	241	—	3600.00
# Melhor	2	17	31	
Média t (s)	—	—	1478.64	

5. Conclusão e trabalhos futuros

Neste trabalho, foi proposto um modelo de programação linear inteira (PLI) para o problema WEIGHTED MINIMUM BROADCAST TIME (WMBT). O modelo PLI apresentou um bom desempenho para instâncias com poucos vértices e poucas arestas superando as técnicas não exatas da literatura, foi capaz de resolver instâncias com até 64 vértices e aproximadamente 400 arestas. No entanto, como esperado, para instâncias maiores, o modelo teve dificuldades. Das 44 instâncias testadas, o modelo PLI encontrou 31 melhores soluções, e provou a otimalidade de 26 delas. A utilização de uma busca local na heurística construtiva da literatura melhorou 39 soluções nas 44 instâncias.

Para trabalhos futuros incluem a estudar a viabilidade de heurísticas e meta-heurísticas para WMBT. Assim, desenvolver outros algoritmos não ótimos, mas que possam ter soluções melhores do que os algoritmos da literatura. O *Biased Random Key Genetic Algorithm* [Toso and Resende 2015, Resende 2011] (BRKGA) é uma das meta-heurísticas a ser considerada, pois o BRKGA foi utilizado para solucionar o MBT [Lima et al. 2022].

Referências

- Averbuch, A., Roditty, Y., and Shoham, B. (2000). Computation of broadcasting multiple messages in a positive weighted tree. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 35:161–184.
- Bucantanschi, D., Hoffmann, B., Hutson, K. R., and Kretchmar, R. M. (2007). A neighborhood search technique for the freeze tag problem. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pages 97–113. Springer.

- Chu, X. and Chen, Y. (2018). Time division inter-satellite link topology generation problem: Modeling and solution. *International Journal of Satellite Communications and Networking*, 36(2):194–206.
- de Sousa, A., Gallo, G., Gutierrez, S., Robledo, F., Rodríguez-Bocca, P., and Romero, P. (2018). Heuristics for the minimum broadcast time. *Electronic Notes in Discrete Mathematics*, 69:165–172.
- Dekker, A. (2002). Applying social network analysis concepts to military c4isr architectures. *Connections*, 24(3):93–103.
- Elkin, M. and Kortsarz, G. (2003). Sublogarithmic approximation for telephone multicast: path out of jungle. In *SODA*, volume 3, pages 76–85.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA.
- Gilbert, E. N. (1959). Random graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144.
- Harutyunyan, H. A. and Jimborean, C. (2014). New heuristic for message broadcasting in networks. In *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pages 517–524. IEEE.
- Harutyunyan, H. A. and Kamali, S. (2008). Efficient broadcasting in networks with weighted nodes. In *2008 14th IEEE International Conference on Parallel and Distributed Systems*, pages 879–884.
- Harutyunyan, H. A. and Kamali, S. (2017). Efficient broadcast trees for weighted vertices. *Discrete Applied Mathematics*, 216:598–608. Levon Khachatryan’s Legacy in Extremal Combinatorics.
- Harutyunyan, H. A. and Wang, W. (2010). Broadcasting algorithm via shortest paths. In *2010 IEEE 16th International Conference on Parallel and Distributed Systems*, pages 299–305.
- Hasson, Y. and Sipper, M. (2004). A novel ant algorithm for solving the minimum broadcast time problem. In Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J. E., Tiño, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, pages 501–510, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Hedetniemi, S. M., Hedetniemi, S. T., and Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349.
- Hocaoğlu, M. F. and Genç, İ. (2019). Smart combat simulations in terms of industry 4.0. In *Simulation for Industry 4.0*, pages 247–273. Springer.
- Hoelting, C. J., Schoenefeld, D. A., and Wainwright, R. L. (1996). A genetic algorithm for the minimum broadcast time problem using a global precedence vector. In *ACM symposium on Applied Computing*, pages 258–262.
- Ivanova, M. (2019). *Optimization Problems in Communication Networks and Multi-Agent Path Finding*. PhD thesis, The University of Bergen.

- Jansen, K. and Müller, H. (1995). The minimum broadcast time problem for several processor networks. *Theoretical Computer Science*, 147(1):69 – 85.
- Keshavarz, H., Bagheri, A., Layeghi, K., and Seyed Iman Mahdavi (2011). A simulated annealing approach for the freeze-tag problem. In *International Conference on Recent Trends in Information Systems*, pages 94–98.
- Koh, J. and Tcha, D. (1991). Information dissemination in trees with nonuniform edge transmission times. *IEEE Transactions on Computers*, 40(10):1174–1177.
- Kortsarz, G. and Peleg, D. (1992). Approximation algorithms for minimum time broadcast. In Dolev, D., Galil, Z., and Rodeh, M., editors, *Theory of Computing and Systems*, pages 67–78, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Lazard, E. (1992). Broadcasting in dma-bound bounded degree graphs. *Discrete Applied Mathematics*, 37-38:387 – 400.
- Lima, A., Aquino, A. L. L., Nogueira, B., and Pinheiro, R. G. S. (2022). A matheuristic approach for the minimum broadcast time problem using a biased random-key genetic algorithm. *International Transactions in Operational Research*.
- Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Sensing as a service model for smart cities supported by internet of things. *Transactions on emerging telecommunications technologies*, 25(1):81–93.
- Resende, M. G. C. (2011). Biased random-key genetic algorithms with applications in telecommunications. *TOP*, 20(1):130–153.
- Robledo, F., Rodríguez-Bocca, P., and Romero, P. (2020). Optimal broadcast strategy in homogeneous point-to-point networks. In Nicosia, G., Ojha, V., La Malfa, E., Jansen, G., Sciacca, V., Pardalos, P., Giuffrida, G., and Umeton, R., editors, *Machine Learning, Optimization, and Data Science*, pages 448–457, Cham. Springer International Publishing.
- Scheuermann and Wu (1984). Heuristic algorithms for broadcasting in point-to-point computer networks. *IEEE Transactions on Computers*, C-33(9):804–811.
- Shang, W., Wan, P., and Hu, X. (2010). Approximation algorithms for minimum broadcast schedule problem in wireless sensor networks. *Frontiers of Mathematics in China*, 5(1):75–87.
- Slater, P. J., Cockayne, E. J., and Hedetniemi, S. T. (1981). Information dissemination in trees. *SIAM Journal on Computing*, 10(4):692–701.
- Su, K., Li, J., and Fu, H. (2011). Smart city and the applications. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 1028–1031.
- Toso, R. F. and Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93.