



Um Algoritmo Genético para o Problema da Árvore Geradora Mínima com Incertezas

Cassiano Blonski Sampaio¹

Bolsista de I.C. - Fundação Araucária,

Departamento de Ciência da Computação, UNICENTRO, Guarapuava, PR

Fábio Hernandes²

Departamento de Ciência da Computação, UNICENTRO, Guarapuava, PR

Resumo. O problema da árvore geradora mínima com incertezas possui aplicações em diversas áreas da Engenharia e da Informática, tais como: instalações de linhas telefônicas e elétricas, armazenamento de informações e transportes. Neste trabalho é proposto um algoritmo genético para o problema da árvore geradora mínima com incertezas nos pesos das arestas e na estrutura da rede. As incertezas são modeladas por meio da Teoria dos Conjuntos *Fuzzy*.

Palavras-chave. Teoria dos Grafos, Teoria dos Conjuntos Fuzzy, Número de Prüfer.

1 Introdução

O Problema da Árvore Geradora Mínima, PAGM, consiste em, a partir de um grafo não orientado, conexo e com pesos em cada aresta, gerar uma árvore que conecte todos os vértices do grafo e que tenha o menor peso total. Este problema possui aplicações nas mais diversas áreas da Engenharia e na Informática, dentre as quais se destacam: instalações de linhas telefônicas (ou elétricas), armazenamento de informações, transportes, dentre outras [3, 11]. Os principais algoritmos da literatura que resolvem este problema são os de Kruskal, Prim, Borůvka e Sollin [1].

Em algumas aplicações do PAGM os parâmetros da rede ou a estrutura em si não são precisos, pois em certos casos representam tempo, custos e conexões [13, 14], dados que dificilmente são exatos. Com isso, surgiu o PAGM com incertezas, proposto por Delgado et al. [4] e abordado por meio da Teoria dos Conjuntos *Fuzzy* [15–17], sendo este problema denominado de *Problema da Árvore Geradora Mínima Fuzzy*, *PAGM-Fuzzy*.

Os principais e mais recentes trabalhos da literatura que tratam do PAGM-*Fuzzy* devem-se a Delgado et al. [4], Chunde [3], Takahashi e Yamakami [13, 14], Janiak e Kasperski [9], Nayeem e Pal [10], Dhanalakshmi e Pavathi [5], Hernandes e Lourenço [6], Zhou

¹cassianoobsampaio@gmail.com

²hernandes@unicentro.br

et al. [19] e Hernandez e Lourenço [7]. O trabalho de Delgado et al. [4] foi um dos pioneiros da área, introduzindo o conceito da árvore geradora mínima *fuzzy*. Já em Chunde [3] foi proposto um algoritmo para o PAGM-*Fuzzy* com incertezas na estrutura da rede, sendo uma adaptação de Delgado et al. [4]. Takahashi e Yamakami [13, 14] trataram do problema com parâmetros *fuzzy*, resolvendo-o por meio de um algoritmo genético - AG [13], e também abordaram o problema em uma rede com estrutura incerta [14], baseando-se em Delgado et al. [4] e Chunde [3]. Janiak e Kasperski [9] utilizaram a teoria de possibilidade de Zadeh [17] nos pesos das arestas e resolveram o problema por meio de um algoritmo clássico da literatura. Nayeem e Pal [10] estudaram o problema com parâmetros *fuzzy*, resolvendo-o por meio de AGs. Dhanalakshmi e Pavathi [5] trataram do problema com os pesos das arestas modelados como números *fuzzy* trapezoidais, propondo um algoritmo que utiliza o centróide e o resolve como um problema *crisp*. Hernandez e Lourenço [6] resolveram o problema da árvore geradora mínima com estrutura incerta, propondo um algoritmo exato que apresenta um conjunto solução de árvores geradoras mínimas. Zhou et al. [19] introduziram o conceito de α -árvore geradora mínima, utilizando a definição de credibilidade e resolvendo o problema via algoritmos de Kruskal e Prim. Em um dos trabalhos mais recentes da área, Hernandez e Sampaio [7] propuseram um algoritmo adaptado de Prim para o problema da árvore geradora mínima com incertezas nos parâmetros e estrutura da rede, sendo uma abordagem inédita da literatura, porém não podendo ser aplicado em grandes redes.

Analisando os trabalhos descritos, verifica-se que estes, em sua maioria, resolvem o problema com incertezas nos parâmetros ou na estrutura, exceto Hernandez e Sampaio [7] que propuseram um algoritmo exato, porém, caro computacionalmente. Logo, visando continuar a abordagem com incertezas nos parâmetros e na estrutura da rede, simultaneamente, neste trabalho é proposto um AG a fim de contornar a dificuldade encontrada por Hernandez e Sampaio [7].

2 O Problema da Árvore Geradora Mínima Fuzzy

Definição 2.1. *Um grafo não orientado $G=(N,A)$, tal que N é o conjunto dos vértices e A o das arestas, é uma árvore se, e somente se, G for conexo e acíclico.*

Dado um grafo conexo $\tilde{G} = (N, A)$, tal que para cada aresta $(i, j) \in A$ associa-se um número triangular fuzzy $\tilde{C}_{ij}$ chamado *peso da aresta* (i, j) e seja $\tilde{G}' = (N, A')$ um subgrafo de $\tilde{G} = (N, A)$, denota-se *peso de $\tilde{G}' = (N, A')$* o valor:

$$\tilde{C}(\tilde{G}') = \sum_{(i,j) \in A'} \tilde{C}_{ij} \quad (1)$$

Definição 2.2. *O Problema da Árvore Geradora Mínima Fuzzy consiste em procurar uma árvore T^* de $\tilde{G} = (N, A)$, tal que:*

$$\tilde{C}(T^*) = \min_T C(T) \quad (2)$$

sendo o mínimo considerado sobre o conjunto de todas as possíveis árvores de $\tilde{G} = (N, A)$ e $C(T)$ o custo da árvore.

3 Relação de Ordem e Número de Prüfer

Nesta seção é apresentada a relação de ordem utilizada para transformar o problema incerto em clássico, bem como o conceito do número de Prüfer abordado para armazenar os cromossomos do AG proposto.

3.1 Relação de ordem

Em seu trabalho, Zhou et al. [19] trabalharam com o PAGM-*Fuzzy* e para eliminar as árvores consideradas "caras", foi utilizada uma relação de ordem para "defuzzificar" os pesos *fuzzy* dos arcos, convertendo-os em números *crisp*, denominada *distribuição de credibilidade inversa*, conforme segue:

Definição 3.1. A *distribuição de credibilidade inversa* de um número fuzzy triangular $\tilde{\alpha} = (a_i, b_i, c_i)$ é:

$$\Phi_i^{-1}(\alpha) = \begin{cases} a_i + 2(b_i - a_i)\alpha, & \text{se } 0 \leq \alpha \leq 0,5 \\ 2b_i - c_i + 2(c_i - b_i)\alpha, & \text{se } 0,5 < \alpha \leq 1 \end{cases} \quad (3)$$

3.2 Número de Prüfer

Segundo Zhou e Gen [18], para qualquer árvore em um grafo completo sempre há, pelo menos, dois vértices folhas, o que significa que apenas uma aresta está conectada a cada um deles. Com isso, Zhou e Gen [18] abordaram o PAGM por meio do número de Prüfer.

Na Figura 1, um exemplo é dado para ilustrar como uma árvore é armazenada por meio do número de Prüfer. Sendo o número de Prüfer [2 5 6 8 2 5] corresponde a uma árvore geradora em um grafo completo de 8 vértices, a construção do Número de Prüfer é descrita da seguinte forma: localize o vértice da folha com o menor rótulo. Neste caso, é o vértice 1. Uma vez que o vértice 2 (o único vértice) é incidente no vértice 1 na árvore, atribua 2 ao primeiro dígito na permutação, depois remova o vértice 1 e a aresta (1,2). Agora, o vértice 3 é o vértice de folha rotulado mais pequeno e o vértice 5 é incidente, atribui 5 ao segundo dígito na permutação e depois remove o vértice 3 e a aresta (3,5). Repita o processo na subárvore até a aresta (5,8) é deixada e o Número de Prüfer desta árvore com 6 dígitos é finalmente produzido.

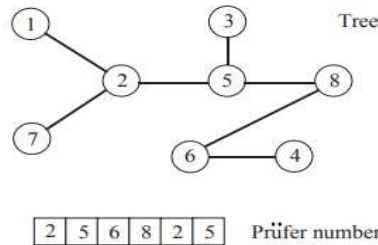


Figura 1: Exemplo de uma árvore armazenada por meio do Número de Prüfer.

4 Algoritmo Proposto

Trata-se da adaptação de um AG baseado nos trabalhos de Hernandez e Sampaio [7], Zhou et al. [19] e Almeida [2], cuja finalidade é abordar o PAGM em rede com incertezas nos parâmetros e na estrutura. Os pesos são tratados como números *fuzzy* triangulares, utilizando o conceito de credibilidade inversa [19] para eliminar as arestas de maior peso, enquanto na estrutura cada aresta possui sua pertinência de fazer parte da rede.

4.1 AG para o PAGM-Fuzzy

O AG proposto possui as seguintes características: (i) é capaz de gerar sua população aleatoriamente; (ii) avalia uma grande quantidade de árvores geradoras; (iii) usa relações de ordem para contornar o problema dos números *fuzzy*; e (iv) usa a representação do Número de Prüfer para verificar a solução.

Vale destacar que o AG proposto contorna a complexidade encontrada nos algoritmos exatos, pois com a codificação do cromossomo por meio do conceito do Número de Prüfer a população gerada aleatoriamente sempre será composta de elementos factíveis, o Número de Prüfer sempre garante isso. Além de gerar a solução factível, o mesmo número garante a facilidade no uso dos operadores de cruzamento e mutação.

O *fitness* é calculado a partir da decodificação do Número de Prüfer, podendo assim ranquear os indivíduos para encontrar a solução com o melhor custo. Para garantir que a melhor solução não se perca durante o avanço das gerações, foi aplicado o método elitismo.

As descrições do AG proposto são: **codificação** - cada cromossomo representa uma árvore geradora, sendo representado por um Número de Prüfer; **inicialização** - todos os cromossomos são gerados aleatoriamente como Números de Prüfer, dispensando validação; **cruzamento de um Ponto** - escolhe dois indivíduos da população de forma aleatória, $P1$ e $P2$, como os pais e cria um cromossomo filho que possui a primeira metade de $P1$ e a segunda de $P2$; **mutação** - após a escolha de um indivíduo, um determinado gene é alterado para um outro valor possível; **fitness** - o valor do fitness de cada cromossomo é calculado após a decodificação do Número de Prüfer, sendo a soma dos pesos das arestas que pertencem à árvore e "defuzzificado" por meio da relação de ordem de credibilidade.

4.2 Passos do algoritmo proposto

O algoritmo é composto de cinco passos, sendo que no primeiro é inicializada e avaliada a população de indivíduos; no segundo são selecionados os indivíduos para participarem da elite; no terceiro são selecionados os pais da nova geração e realizado o cruzamento, no quarto é selecionado uma percentagem de indivíduos e ocorre a mutação, no quinto são unidos os novos indivíduos com a elite para formar a nova população e verificado o critério de parada. A seguir é descrito o algoritmo proposto.

1. Criar a população inicial aleatória, com X indivíduos, sendo possível apenas valores de 0 até a quantidade de vértices para cada posição do cromossomo. Calcule o valor do fitness para cada indivíduo.
2. Selecione uma percentagem dos melhores indivíduos de P para o elitismo, e aloque os indivíduos na população elite (Pe).

3. Selecione uma percentagem dos indivíduos de P para cruzamento, proceda a operação e aloque os novos indivíduos na população intermediária (P_i) até $P_i = P - P_e$.
4. Selecione uma percentagem dos indivíduos de P_i para a mutação, proceda a operação. Faça $P \leftarrow P_i + P_e$. Calcule o valor do fitness de cada indivíduo de P .
5. Se ($i = \text{Geração}$) $\Rightarrow$ Fim. Senão retorne ao Passo 1

5 Resultados e Discussões

O algoritmo proposto foi implementado em Python e executado para o grafo proposto por Zhou et al. [19], cujos pesos e as pertinências das arestas estão na Tabela 1.

Tabela 1: Dados da rede da Zhou et al.

Aresta	Origem	Destino	Peso	Pertinência
1	1	2	(60, 65, 70)	0.3125
2	2	3	(58, 70, 75)	0.6823
3	3	4	(70, 74, 82)	0.7684
4	4	5	(62, 72, 80)	0.3357
5	1	5	(58, 66, 76)	0.2528
6	1	4	(64, 70, 74)	0.8132
7	1	3	(58, 62, 70)	0.2987
8	2	5	(65, 70, 80)	0.5255
9	2	4	(55, 60, 65)	0.6349
10	3	5	(62, 68, 72)	0.4966

Nas execuções foram utilizadas várias combinações de geração {5, 15, 20, 50}, população inicial {10, 30, 50, 100}, taxa de mutação {0.01, 0.1}, taxa de seleção {0.1, 0.01} e credibilidade {0.2, 0.5, 0.8}. Para cada combinação o algoritmo foi executado 10 vezes e então calculado a média aritmética, a moda e o melhor resultado obtido.

Considerando que a credibilidade 0.2 foi a que apresentou os melhores resultados, nesta seção são apresentados apenas seus resultados. Na primeira iteração foi executada a rede completa, sem cortes, gerando os resultados da Tabela 2. Na segunda iteração foi eliminada a aresta 5, com pertinência 0.2528, sendo que a melhor solução foi 243. Na terceira iteração foi eliminada a aresta 7, com pertinência 0.2987, além da aresta da iteração anterior, obtendo o melhor indivíduo com peso igual a 246.2. Na quarta iteração foi eliminada a aresta 1, com pertinência 0.3125, além das arestas eliminadas nas iterações anteriores, obtendo o melhor indivíduo com peso igual a 250.6. Na quinta e última iteração foi cortada a aresta 4, com pertinência 0.3357, e as das iterações anteriores, apresentando a melhor solução igual a 250.6.

Analisando os resultados obtidos, verifica-se que à medida em que as iterações ocorrem o peso das árvores aumentam, bem como as pertinências das mesmas. Pode-se notar, também, que o maior diferencial dentre as características do AG é o tamanho da população inicial, pois com uma população maior a chance de se criar a melhor solução nas primeiras gerações é elevada, e por meio do elitismo foi possível manter as melhores soluções para as seguintes gerações, garantindo assim que as mesmas não sejam perdidas. Já a taxa de mutação, em comparação com a taxa de seleção, apresentou uma influência reduzida, com uma maior taxa de seleção no elitismo uma maior quantidade de melhores indivíduos

Tabela 2: Resultados obtidos utilizando todas as arestas do grafo, Iteração 1.

Ger.	Pop. In.	Tx. Mutação	Tx. Seleção	Média	Moda	Melhor
5	10	0.01	0.1	243.56	243	240.6
15	10	0.01	0.1	243.86	240.6	239.8
20	10	0.01	0.1	243.54	243	239.8
20	10	0.01	0.01	244.04	239.8	239.8
5	30	0.01	0.1	240.92	240.6	239.8
20	30	0.01	0.1	241.48	239.8	239.8
20	30	0.2	0.1	240.44	239.8	239.8
50	30	0.01	0.1	241.08	239.8	239.8
5	50	0.01	0.1	240.29	239.8	239.8
5	50	0.01	0.01	240.92	239.8	239.8
20	50	0.01	0.1	239.88	239.8	239.8
20	50	0.01	0.01	239.8	239.8	239.8

pode sobreviver para as próximas gerações garantido melhor convergência do algoritmo. Em contrapartida, a mutação na representação com Número de Prüfer pode afetar cromossomos que poderiam ser candidatos a solução ótima, pois uma pequena alteração no valor da posição do cromossomo pode trocar arestas de forma que mude toda a árvore.

6 Conclusão, Limitações e Trabalhos Futuros

O PAGM é abordado em diversas aplicações da Engenharia e da Informática, sendo que em algumas destas as redes possuem incertezas nos pesos e até mesmo nas estruturas. Assim, o objetivo deste trabalho foi estudar o PAGM-*Fuzzy*, propondo um AG que abordasse incertezas nos pesos e na estrutura da rede simultaneamente.

O diferencial do AG proposto está na utilização do Número de Prüfer, o que faz com que este convirja rápido, pois todos os indivíduos da população são válidos, i.é, árvores.

Vale destacar que o algoritmo possui a limitação de trabalhar somente com grafos completos, o que não atrapalha aplicá-lo em um grafo não completo, sendo que neste caso é necessário completá-lo com arestas de pesos bastante grande, forçando assim que estas não sejam escolhidas, ou seja, não participem da solução final, podendo esta desvantagem ser tratada em trabalhos futuros.

Referências

- [1] R. K. Ahuja, T. L. Magnanti e J. B. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice Hall, (1993).
- [2] T. A. de Almeida, Computação evolutiva aplicada à resolução do problema da árvore geradora mínima com parametros fuzzy, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, (2006).

- [3] Y. Chunde, On the optimization problem of spanning tree in fuzzy network, *The Journal of China Universities of Posts and Telecommunications*, vol. 3, 22–25, (1996).
- [4] M. Delgado, J. L. Verdegay e M. A. Vila, On fuzzy tree definition, *European Journal of Operational Research*, vol. 22, 243–249, (1985).
- [5] S. Dhanalakshmi e N. Parvathi, Minimum spanning tree of a rough graph having weights as trapezoidal fuzzy numbers, *Global Journal of Pure and Applied Mathematics*, 0973–1768, (2015).
- [6] F. Hernandez e M. H. R. S. Lourenço, An Algorithm for the Minimum Spanning Tree Problem with Uncertain Structures, *IEEE Latin America Transactions*, vol. 13, 3885–3889, (2015).
- [7] F. Hernandez e C. B. Sampaio, O problema da árvore geradora mínima fuzzy: um algoritmo para o caso envolvendo incertezas nos pesos das arestas e na estrutura da rede, *Anais do IV Congresso Brasileiro de Sistemas Fuzzy*, vol. 1, 69–80, (2016).
- [8] W. Pedrycz e F. Gomide, *Fuzzy systems engineering: toward human-centric computing*, John Wiley & Sons, (2007).
- [9] A. Janiak e A. Kasperski, The minimum spanning tree problem with fuzzy costs, *Fuzzy Optimization and Decision Making*, vol. 7, 105–118, (2008).
- [10] S. M. A. Nayeem e M. Pal, Diameter constrained fuzzy minimum spanning tree problem, *International Journal of Computational Intelligence Systems*, vol. 6, 1040–1051, (2013).
- [11] G. R. Raidl e B. A. Julstrom, Edge sets: an effective evolutionary coding of spanning trees, *IEEE Transactions on Evolutionary Computation*, vol. 7, 225–239, (2003).
- [12] M. T. Takahashi, Contribuições ao estudo de grafos fuzzy: teoria e algoritmos, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, (2004).
- [13] M. T. Takahashi e A. Yamakami, Um estudo sobre árvore geradora mínima com parâmetros fuzzy, *Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional*, vol. 1, 2217–2227, (2003).
- [14] M. T. Takahashi e A. Yamakami, Um estudo sobre o problema da árvore geradora mínima com estrutura do grafo fuzzy, *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, vol. 1, 2278–2285, (2004).
- [15] L. Zadeh, Fuzzy sets, *Information and control*, vol. 8, 338–353, (1965).
- [16] L. Zadeh, Fuzzy algorithms, *Information and control*, vol. 12, 94–102, (1968).
- [17] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy sets and systems*, vol. 1, 3–28, (1978).
- [18] G. Zhou e M. Gen, Genetic algorithm approach on multi-criteria minimum spanning tree problem, *European Journal of Operational Research*, vol. 114, 141–152, (1999).
- [19] J. Zhou, L. Chen, K. Wang, e F. Yang, Fuzzy α -minimum spanning tree problem: Definition and solutions *International Journal of General Systems*, vol. 45, 311–335, (2016).