South American Development Society Journal
Vol.: 11| Nº.: 31| **Ano:** 2024 | **ISSN:** 2446-5763 | **DOI:** 10.24325/issn.2446-5763.v11i31p214-228
Data de Submissão: 09/04/2025 | **Data de Publicação:** 18/04/2025

PERFORMANCE DE UM ALGORITMO EXATO E DE UMA META-HEURÍSTICA NA RESOLUÇÃO DE UM PROBLEMA CLÁSSICO DE OTIMIZAÇÃO COMBINATÓRIA

Nathan Batistelli de Oliveira – LABICON
nathan.bo10@aluno.ifsc.edu.br

Eduardo Paz Putti – LABICON
eduardo.p2005@aluno.ifsc.edu.br

Carise Elisane Schmidt – LABICON
carise.schmidt@ifsc.edu.br

Resumo

Problemas de otimização combinatória podem modelar uma série de problemas reais. Contudo, a complexidade computacional que, em geral, está embutida nesses problemas é um desafio na busca por soluções. Este trabalho tem como objetivo avaliar a performance de duas abordagens de resolução em um problema clássico de otimização combinatória: o Problema do Caixeiro Viajante. Entre as abordagens aplicadas estão um modelo de programação linear baseado no fluxo de duas commodities e uma meta-heurística evolutiva, denominada A-BRKGA. Para atender o objetivo do estudo, esses algoritmos foram implementados computacionalmente e testes foram executados em um conjunto de instâncias da literatura. As abordagens foram comparadas em relação à qualidade da solução e o tempo de processamento. Os resultados mostraram que, nas instâncias avaliadas, ambos os algoritmos geraram soluções de qualidade e também similares entre si, com diferença média próxima a 1%. Nas instâncias testadas, o tempo médio para obtenção das soluções foi levemente inferior para a meta-heurística em relação ao algoritmo exato.

Palavras-chave: Problema do Caixeiro Viajante; Modelo de Fluxo de duas Commodities; Algoritmos Genéticos com Chaves Aleatórias.

Abstract

Combinatorial optimization problems can model a wide range of real-world problems. However, their typical computational complexity is a challenge when searching for solutions. This study aims to evaluate the performance of two solution approaches for a classic combinatorial optimization problem: the Traveling Salesman Problem. The approaches applied include a linear programming model based on a two-commodity flow formulation and an evolutionary metaheuristic called A-BRKGA. To achieve the proposed goal, these algorithms were implemented and computational tests were performed using a set of benchmark instances. The approaches were compared in terms of solution quality and processing time. The results show that, for the evaluated instances, both algorithms produced high-quality solutions that are also similar, with an average difference of approximately 1%. For the set of instances, the average processing time was slightly lower for the metaheuristic compared to the exact algorithm.

Keywords: Traveling Salesman Problem; Two-commodities Flow Model; Random-key Genetic Algorithm.

1 Introdução

A otimização combinatória é um campo de pesquisa que integra ciência da computação, pesquisa operacional e matemática aplicada. Ela compreende uma complexa classe de problemas de otimização, que possui variáveis de decisão discretas e um espaço finito de busca (KARIMI-MAMAGHAN et al. 2022).

Problemas de otimização combinatória podem modelar uma ampla variedade de aplicações e, por isso, podem ser encontrados em muitos campos da pesquisa. Muitos desses problemas pertencem à classe NP-Hard (MAZYAVKINA et al., 2021). Devido a essa complexidade computacional, resolver esses problemas de forma exata pode ser inviável dentro de um tempo computacional razoável. Para esses casos, algoritmos heurísticos são uma alternativa. Embora não tenham garantia teórica de otimalidade, eles podem gerar boas soluções em um tempo computacional aceitável.

Um clássico exemplo de problema de otimização combinatória, pertencente à classe NP-Hard, é o Problema do Caixeiro Viajante (PCV) (LAPORTE, 1992). Usando uma definição informal, este problema visa determinar uma rota de custo mínimo entre um conjunto de cidades, de forma que a rota inicie em uma dessas cidades, passe por todas as outras exatamente uma vez e retorne à cidade de origem. Mas, para além da questão logística, o problema tem muitas outras aplicações na vida real, como, por exemplo, a programação de máquinas, o planejamento de atividades, a fabricação de microchips e o sequenciamento de DNA (MATAI, SINGH, MITALL, 2010).

O PCV é um dos problemas mais estudados na literatura, sendo usado como referência para avaliar diversas técnicas de otimização (POP et al., 2024). Desde o seu surgimento, abordagens de solução exata, algoritmos heurísticos e abordagens combinadas já foram propostos e avaliados para a sua resolução.

Este estudo aborda o Problema do Caixeiro Viajante a partir de duas abordagens distintas de resolução, uma exata e uma meta-heurística. Nele, a performance de um modelo de programação linear baseado em fluxo de duas *commodities* é comparada à de um Algoritmo Genético Adaptativo com Chaves Aleatórias Viciadas. Para essa análise, é utilizado um conjunto de instâncias da literatura e executados testes computacionais.

A sequência deste documento está organizada conforme segue. Inicialmente, é apresentada uma revisão de literatura sobre o tema em estudo. Em seguida, a metodologia aplicada no seu desenvolvimento é detalhada. Os resultados obtidos são, então, mostrados e discutidos. Por fim, são apresentadas as conclusões.

2 Revisão de literatura

Nesta seção, será realizada uma breve revisão sobre a evolução, na literatura, das duas abordagens de resolução aplicadas neste estudo: os modelos de programação linear voltados para o PCV e a classe de meta-heurísticas denominada Algoritmos Genéticos (AG).

Em relação ao Problema do Caixeiro Viajante, a primeira menção conhecida foi encontrada em um manual de instruções direcionado a esses profissionais, datado de 1832 (IMPA, 2024). Porém, o problema só foi formalizado matematicamente muitos anos mais tarde. Desde então, o PCV foi ganhando espaço entre os pesquisadores, motivado pela ampla gama de aplicações e pela sua complexidade computacional.

Em termos de programação linear, o PVC foi definido pela primeira na década de 50, com base em um problema de designação (DANTZIG; FULKERSON; JOHNSON, 1954). A partir de então, diversos outros modelos foram propostos. Os principais modelos serão apresentados a seguir.

Miller, Tucker e Zemlin (1960) apresentaram uma formulação para o PVC, que também está vinculada ao problema de designação. Mais tarde, diferentes formulações baseadas em fluxo de commodities foram propostas.

Gavish e Graves (1978) propuseram uma formulação com fluxo de uma única commodity para solucionar o PCV. Usando essa mesma ideia, Wong (1980) e Claus (1984) apresentaram modelos baseados em fluxo de múltiplas commodities. Finke, Clauss e Gunn (1984) definiram um novo modelo de fluxo, que contempla duas commodities. E, da mesma forma, usando o fluxo de duas commodities, uma nova formulação também foi apresentada por Langevin et al. (1993).

A dinâmica do fluxo de mercadorias também foi aplicada na proposição de um modelo para a variante do PCV, denominada Problema do Caixeiro Viajante com Coletas e Entregas (BALDACCI; HADJICONSTANTINOU; MINGOZZI, 2003). Posteriormente, esses mesmos autores também propuseram um modelo desse tipo para o Problema de Roteamento de Veículos Capacitado (BALDACCI; HADJICONSTANTINOU; MINGOZZI, 2004).

Modelos de programação linear, como os aqui apresentados, têm a garantia de obtenção da solução ótima, mas estão sujeitos ao tempo computacional necessário para que ela seja alcançada (MAZYAVKINA et al., 2021). Isso pode, em muitas situações, tornar o problema intratável do ponto de vista prático.

Conforme Chaves, Gonçalves e Lorena (2018), meta-heurísticas são abordagens interessantes para obter soluções de boa qualidade em problemas de otimização combinatória, especialmente quando esses problemas são do tipo NP-Hard. Os Algoritmos Genéticos (AG) são um tipo de meta-heurística baseada na dinâmica de evolução das populações, com critérios de seleção natural e sobrevivência do mais apto (GOLDBERG; HOLLAND, 1988).

Os Algoritmos Genéticos com Chaves Aleatórias, propostos por Bean (1994), constituem uma classe de métodos para problemas de otimização combinatória. Seguindo a ideia dessa abordagem, Gonçalves e Resende (2011) apresentaram uma alteração na proposta, focada especialmente na forma de seleção dos indivíduos para o cruzamento e no formato como ele ocorre. Esse método é denominado Algoritmo Genético com Chaves Aleatórias Viciadas (do inglês, “*Biased Random-Key Genetic Algorithm*” – BRKGA).

Nesse tipo de meta-heurística, o decodificador da solução é a única parte da abordagem que depende do tipo de problema envolvido. As demais decisões são independentes. Contudo, a eficiência e efetividade de qualquer meta-heurística acaba sendo afetada pelos parâmetros envolvidos (CHAVES; GONÇALVES; LORENA, 2018).

Levando em conta que aplicar uma meta-heurística consiste em um processo dinâmico e adaptativo, onde o valor dos parâmetros ótimos pode variar conforme o processo avança, Chaves, Gonçalves e Lorena (2018) propuseram os Algoritmos Genéticos Adaptativos com Chaves Aleatórias Viciadas (do inglês, “*Adaptative Biased Random-Key Genetic Algorithm*” – A-BRKGA). Nessa abordagem, há um controle adaptativo da probabilidade de um indivíduo herdar os genes do genitor da elite, e regras determinísticas são usadas para definir parâmetros como o tamanho da população e as partições da elite e dos mutantes.

Além disso, os autores também incorporam ao método um componente de busca local em grupos de soluções promissoras. A motivação por trás do método é ajustar os parâmetros de forma ocorra uma exploração mais ampla da região das soluções na fase inicial da busca, e uma exploração intensificada das soluções existentes ao longo do processo evolucionário.

3 Metodologia

Esta seção detalha o modelo matemático e os parâmetros adotados para a meta-heurística, que foram aplicados neste estudo. Em seguida, são especificadas as instâncias da literatura usadas para avaliar a performance dos métodos. E, por fim, são descritos os testes computacionais que foram executados.

3.1 Modelo de programação linear

O Problema do Caixeiro Viajante pode ser formalizado usando a teoria dos grafos. O modelo de programação linear aplicado neste estudo é baseado no fluxo de duas commodities, gerado a partir da proposta de Baldacci, Hadjiconstantinou e Mingozzi (2003, 2004). O problema é definido conforme segue.

Dado um grafo completo $G = (V, E)$, onde V denota o conjunto de n vértices ponderados; E representa o conjunto de arestas, e $C = (c_{ij})$ expressa a matriz de custos, ou ponderações, associada ao conjunto E. Seja $\overline{G} = (\overline{V}, \overline{E})$ um grafo estendido de G, no qual é adicionado um novo vértice $n + 1$, denominado terminal, que representa uma cópia do vértice de origem 0. Sejam y_{ij} e y_{ji} duas variáveis de fluxo associadas com a aresta $\{i, j\}\in\overline{E}$; e seja x_{ij} uma variável binária que assume valor 1 caso a aresta $\{i, j\}\in\overline{E}$ esteja na solução, e 0 caso contrário. A formulação matemática, para o caso simétrico, é dada por:

$$Z = Min \sum_{\{i,j\} \in \overline{E}} c_{ij}x_{ij} \tag{1}$$

$$\sum_{j \in \overline{V}} (y_{ij} - y_{ji}) = 2, \; i \in \overline{V}\backslash\{0, n + 1\} \tag{2}$$

$$\sum_{j \in \overline{V}\backslash\{0,n+1\}} y_{0j} = n - 1, \tag{3}$$

$$\sum_{j \in \overline{V}\setminus\{0,n+1\}} y_{j0} = 0, \tag{4}$$

$$\sum_{j \in \overline{V}\setminus\{0,n+1\}} y_{n+1j} = n - 1, \tag{5}$$

$$\sum_{j \in \overline{V}\setminus\{0,n+1\}} y_{jn+1} = 0, \tag{6}$$

$$y_{ij} + y_{ji} = (n - 1)x_{ij}, \ \{i, j\} \in \overline{E} \tag{7}$$

$$\sum_{j \in \overline{V}\, i<j} x_{ij} + \sum_{j \in \overline{V}\, i>j} x_{ji} = 2, \ i \in \overline{V}\setminus\{0, n + 1\} \tag{8}$$

$$y_{ij} \geq 0; y_{ji} \geq 0, \ \{i, j\} \in \overline{E} \tag{9}$$

$$x_{ij} \in \{0, 1\}, \ \{i, j\} \in \overline{E}. \tag{10}$$

A função objetivo, expressa em (1), minimiza o custo total de execução da rota. Restrições dos conjuntos (2) – (6) definem um padrão de fluxo factível saindo da origem até o terminal. As restrições (7) indicam as arestas de uma solução factível, enquanto as restrições (8) forçam que, em qualquer solução factível, duas arestas incidam em cada um dos vértices intermediários. Restrições (9) e (10) definem o domínio das variáveis.

O modelo também é fortalecido com a inserção de algumas desigualdades válidas. Elas foram adaptadas ao problema a partir da proposta apresentada no estudo de Baldacci et al. (2003; 2004).

$$(n - 2)y_{ij} - y_{ji} \geq 0, \ \{i, j\} \in \overline{E} \tag{11}$$

$$(n - 2)y_{ji} - y_{ij} \geq 0, \ \{i, j\} \in \overline{E} \tag{12}$$

$$y_{ij} + y_{ji} \leq n - 1, \ \{i, j\} \in \overline{E}. \quad (13)$$

3.2 Parâmetros do A-BRKGA

O A-BRKGA utiliza cinco parâmetros presentes no BRKGA (p, p_e, p_m, ρ_e e max_{ger}), além de dois parâmetros adaptativos próprios (α e β). Os parâmetros ρ_e e β são autoadaptativos e os demais parâmetros são atualizados por regras determinísticas. Apenas o parâmetro γ é definido pelo usuário e ele indica o limite de tempo de processamento.

A população inicial é formada por p indivíduos. Cada gene de um indivíduo é gerado aleatoriamente, com probabilidade uniforme pertencente ao intervalo $[0; 1]$. A cada geração, os parâmetros são atualizados usando regras determinísticas, que consideram o progresso do processo evolutivo, conforme segue.

A população é dividida em dois grupos conforme aptidão. Um indivíduo da população é considerado apto se o seu fitness for superior a um valor limite f_e, calculado a partir do fitness do pior e do melhor indivíduo da população atual, e do parâmetro $\alpha \in [0; 1]$. O valor desse parâmetro é modificado periodicamente, conforme o progresso da busca. Para controlar a diversificação entre os indivíduos aptos e evitar ótimos locais, é aplicada uma perturbação. A força da perturbação é definida pelo parâmetro autoadaptativo $\beta \in [0,001; 0,01]$. A cada geração também são criados p_e indivíduos mutantes. O cruzamento uniforme parametrizado (SPEARS; DEJONG, 1991) é aplicado. Um dos genitores é selecionado aleatoriamente entre os indivíduos considerados aptos e o outro entre os demais indivíduos da população atual. O parâmetro ρ_e representa a probabilidade de que, no cruzamento, um indivíduo herde um gene do genitor classificado como apto e $1 - \rho_e$ que ele esteja entre os não aptos.

O critério de parada usado é o número máximo de gerações max_{ger}, calculado com base no parâmetro γ. Para os testes processados neste estudo, definiu-se $\gamma = 3600$ segundos.

3.3 Instâncias de teste

Dezenove instâncias de teste foram selecionadas da biblioteca TSPLIB (2024). Todas estão vinculadas ao Problema do Caixeiro Viajante simétrico. A Tabela 1 detalha o nome de cada instância e o número de pontos que ela contempla.

Tabela 1 – Informações sobre as instâncias de teste utilizadas

Instância	Nº de pontos	Instância	Nº de pontos
st70	70	kroB150	150
pr76	76	u159	159
rat99	99	rat195	195
kroA100	100	d198	198
rd100	100	kroA200	200
eil101	101	kroB200	200
bier127	127	ts225	225
ch130	130	tsp225	225
ch150	150	u574	574
kroA150	150		

Fonte: Autoria própria (2024)

3.4 Testes computacionais

Os algoritmos foram implementados em linguagem C++. Todos os testes foram conduzidos em uma máquina AMD Ryzen 7 2700, com processador de 8 núcleos e 4,1 GHz de frequência, 32 GB de memória RAM instalada, executando em sistema operacional Windows 11.

Para o modelo matemático de programação linear foi usado o solver Gurobi 11.0, na versão acadêmica e em modo default. Um limite de tempo de processamento de 3600 segundos foi aplicado a cada instância. Ao final do teste, foram obtidos: a melhor solução, o gap entre a melhor solução e o limitante inferior e o tempo de processamento usado.

Para o algoritmo A-BRKGA, foram executadas três repetições de cada teste para cada instância. Com os testes, foram extraídas as informações: melhor solução encontrada, menor tempo de execução entre as repetições realizadas, e o tempo total das três execuções.

Para avaliar a performance dos métodos, os resultados obtidos nos testes foram analisados e comparados para o conjunto de instâncias, tanto em relação à qualidade das soluções quanto ao tempo de processamento para sua obtenção.

4 Análise dos resultados e discussão

Na Tabela 2 são apresentados os resultados dos testes computacionais para as duas abordagens propostas. Após a identificação da instância, são reportados, respectivamente, a melhor solução, o gap entre a melhor solução e limitante inferior (%), e o tempo de processamento (s) do modelo matemático de programação linear. Na sequência, são apresentados, respectivamente, a melhor solução encontrada pelo algoritmo A-BRKGA nas três execuções, o menor tempo de processamento (s), o tempo total (s) e o acréscimo (%) na solução em relação ao modelo.

Tabela 2 – Resultados comparativos entre as abordagens exata e meta-heurística

Instância	Modelo matemático			A-BRKGA			
	Melhor solução	Gap (%)	Tempo (s)	Melhor solução	Melhor tempo (s)	Tempo total (s)	Acréscimo (%) na solução
st70	675	0,00	2	675	16	27	0,00
pr76	108159	0,00	13	108159	9	31	0,00
rat99	1211	0,00	1	1221	10	45	0,83
kroA100	21282	0,00	7	21282	28	52	0,00
rd100	7910	0,00	5	7916	29	61	0,08
eil101	629	0,00	1	640	27	60	1,75
bier127	118282	0,00	12	119366	35	76	0,92
ch130	6110	0,00	14	6160	76	98	0,82
ch150	6528	0,00	42	6640	70	145	1,72
kroA150	26524	0,00	43	26786	44	169	0,99
kroB150	26130	0,00	34	26445	57	135	1,21
u159	42080	0,00	20	42215	66	155	0,32
rat195	2323	0,00	70	2394	164	270	3,06
d198	15780	0,00	584	15886	100	239	0,67
kroA200	29368	0,00	101	29775	198	297	1,29
kroB200	29437	0,00	59	29846	41	388	1,39
ts225	126643	1,96	3600	126643	252	491	0,00
tsp225	3921	0,00	248	3989	230	350	1,73
u574	36905	0,00	2505	38731	1373	3799	4,95
Média	32100	0,10	388	32356	149	363	1,15

Fonte: Autoria própria (2024)

Os resultados mostraram que a abordagem exata conseguiu alcançar a solução ótima em 18 das 19 instâncias, dentro do limite de tempo estabelecido (3600 s). Apenas em uma das instâncias com 225 pontos o valor ótimo não foi alcançado (ts225). Nela, o gap foi de 1,96%.

Em relação ao A-BRKGA, os valores encontrados na melhor execução foram, em média, similares aos valores gerados pelo modelo matemático de programação linear. A diferença média foi de 1,15%, com rotas iguais ou mais curtas sendo geradas pela abordagem exata. Em 4 instâncias o algoritmo A-BRKGA gerou a solução ótima. A maior diferença foi de 4,95%, para a instância com o maior número de clientes. Quando avaliamos o tempo de processamento, mesmo realizando três execuções, a meta-heurística A-BRKGA teve um tempo médio menor em comparação à abordagem exata. Contudo, nas instâncias menores, a abordagem exata ainda foi mais rápida.

Com base no conjunto de instâncias analisadas, observou-se que as duas abordagens alcançaram as soluções ótimas ou chegaram muito próximas delas. Conforme o esperado, no algoritmo exato há uma tendência de aumento no tempo de processamento para obtenção do ótimo, conforme cresce o número de pontos. Em relação à meta-heurística, a variação entre o tempo de processamento das instâncias com menor número de pontos não diferiu na mesma proporção, corroborando com a tendência de maior eficiência em instâncias com maior número de pontos.

5 Conclusões

Este estudo teve como objetivo avaliar a performance de dois algoritmos na resolução do Problema do Carteiro Viajante. Uma dessas abordagens foi exata, dada por um modelo de programação linear e um conjunto de desigualdades válidas, e a outra foi meta-heurística, dada pelo algoritmo A-BRKGA. Ambos foram avaliados e comparados a partir de testes computacionais, executados em um conjunto de instâncias da literatura com até 574 pontos. Essas instâncias foram criadas para o Problema do Caixeiro Viajante simétrico.

Os resultados mostraram que, para o conjunto de instâncias utilizado, as duas abordagens forneceram soluções de qualidade, e que essa qualidade foi similar entre os métodos aplicados. Em relação ao tempo de processamento, a meta-heurística A-BRKGA foi, em média, mais rápida. Estudos mais amplos, em relação ao tamanho das instâncias testadas, são necessários para confirmar essa observação.

Referências

BALDACCI, R.; HADJICONSTANTINOU, E.; MINGOZZI, A. "An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation". Operations Research, v. 52, n. 5, p. 723–738, 2004.

BALDACCI, R.; HADJICONSTANTINOU, E.; MINGOZZI, A. "An exact algorithm for the Traveling Salesman Problem with Deliveries and Collections". Networks, v. 42, n. 1, p. 26-41, 2003.

BEAN, J. C. "Genetic algorithms and random keys for sequencing and optimization". ORSA J. on Computing, v. 6, p. 154–160, 1994.

CHAVES, A. A.; GONÇALVES, J. F.; LORENA, L. A. N. "Adaptive biased random-key genetic algorithm with local search for the capacitated centered clustering problem". Computers & Industrial Engineering, v. 124, p. 331-346, 2018.

CLAUS, A. "A new formulation for the traveling salesman problem". SIAM J. Alg. Discrete Methods, v. 5, p. 21–25, 1984.

DANTZIG, G.; FULKERSON, D.; JOHNSON, S. "Solutions of large-scale traveling salesman problem". Operations Research, v. 2, p. 393–410, 1954.

FINKE, G.; CLAUS, A; GUNN, E. "A two-commodity network flow approach to the traveling salesman problem". Congressus Numerantium, v. 41, p. 167–178, 1984.

GAVISH, B.; GRAVES, S. "The travelling salesman problem and related problems". Working Paper. Graduate School of Management, University of Rochester, 1978.

GOLDBERG, D.E.; HOLLAND, J. H. "Genetic Algorithms and Machine Learning".

Machine Learning, v. 3, p. 95–99, 1988.

GONÇALVES, J. F.; RESENDE, M. G. C. "Biased random-key genetic algorithm for combinatorial optimization". Journal Heuristics, v. 17, p. 487–525, out. 2011.

IMPA. Instituto de Matemática Pura e Aplicada. "O Problema do Caixeiro Viajante". Disponível em: https://impa.br/noticias/folha-o-problema-do-caixeiro-viajante/. Acesso em: 04 set. 2024.

KARIMI-MAMAGHAN, M. et al. "Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art". European Journal of Operational Research, v. 296, n.2, p. 393–422, 16 jan. 2022.

LANGEVIN, A.; DESROCHERS, M.; DESROSIERS, J.; GÉLINAS, S.; SOUMIS, F. "A two-commodity flow formulation for the traveling salesman and the makespan problems with time Windows". Networks, v. 23, p. 631–640, 1993.

LAPORTE, G. "The Traveling Salesman Problem: An overview of exact and approximate algorithms". European Journal of Operational Research, v. 59, n. 2, p. 231–247, 1992.

MATAI, R; SINGH, S. P.; MITALL, M. L. "Traveling salesman problem: an overview of applications, formulations, and solution approaches". In: DAVENDRA D. (Ed.). Traveling salesman problema, theory and applications. InTech Press, London, p. 1–26, 2010.

MAZYAVKINA, N.; SVIRIDOV, S.; IVANOV, S.; BURNAEV, E. "Reinforcement learning for combinatorial optimization: A survey". Computers & Operations Research, v. 134, 105400, out. 2021.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. "Integer Programming Formulations and Traveling Salesman Problems". Journal of the Association for Computing Machinery, v. 7, p. 326–329, 1960.

POP, P. C; COSMA, O.; SABO, C.; SITAR, C. P. "A comprehensive survey on the generalized traveling salesman problem". European Journal of Operational Research, v. 314, n. 3, p. 819–835, 1 mai. 2024.

SPEARS, W. M.; DEJONG, K. A. "On the virtues of parameterized uniform crossover”. Fourth International Conference on Genetic Algorithms. In: Proceedings of the Fourth International Conference on Genetic Algorithms, p. 230–236, 1991.

TSPLIB. “Traveling Salesman Problem Library”. Disponível em: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>. Acesso em: 11 mar. 2024.

WONG, R. T. “Integer programming formulations of the traveling salesman problem”. IEEE International Conference of Circuits and Computers. In: Proceedings of the IEEE International Conference of Circuits and Computers, p. 149–152, 1980.

O conteúdo expresso no trabalho é de inteira responsabilidade dos autores.