

Uma análise de múltiplos critérios do Problema de Roteamento de Veículos usando o AEMMT.

Augusto Fernandes Macri*

augusto.macri@hotmail.com

Universidade Federal de Uberlândia
Uberlândia, Minas Gerais, Brasil

Christiane Regina Soares Brasil

Universidade Federal de Uberlândia

Uberlândia, Brasil

christiane.ufu@gmail.com

Abstract

The following work presents a resolution of the Vehicle Routing Problem, focusing on fuel distribution, using Evolutionary Algorithms, which is a bio-inspired Artificial Intelligence technique. The study examines the behavior of a monoobjective version compared to the 'MultiObjective Evolutionary Algorithm with Many Tables' (AEMMT, in portuguese), which operates simultaneously with different subpopulations, where each one is dedicated to minimizing a specific objective function. Therefore, the objective is to determine which of the two produces the best results when considering several factors such as distance, time and fuel. As a result, in this work the version algorithm with many objectives proved to be superior to the single-objective version in relation to the value of the improvement brought, especially analyzing the evolutionary behavior of each criterion.

Keywords

Artificial Intelligence, Evolutionary Algorithms, Multi-Objective Optimization, Vehicle Routing Problem.

1 Introdução

É inquestionável a importância da aplicação da Inteligência Artificial (IA) nas mais diversas áreas do conhecimento, a fim de auxiliar nas buscas por soluções de problemas de alta complexidade. Neste sentido, este trabalho focou na implementação de técnicas de IA para a resolução do Problema de Roteamento de Veículos (PRV), um dos mais famosos problemas da área de otimização computacional e que representa uma das questões mais desafiadoras da atualidade. O objetivo principal deste problema é determinar uma rota, visitando uma sequência de locais, com o uso de veículos, ao menor custo possível, atendendo a todas as imposições do problema ([3]). A partir disto, este custo é calculado utilizando critérios de distância, tempo, violações de janelas de tempo e gasto de combustível, por exemplo. Portanto, é um tema muito pertinente do ponto de vista logístico de distribuição de produtos, de modo geral, a nível regional e/ou nacional.

Para este trabalho, foi escolhida uma técnica bioinspirada de IA denominada Algoritmo Evolutivo (AE). A escolha desta técnica foi feita, principalmente, por sua literatura mostrar resultados relevantes com aplicações em problemas combinatórios complexos [9], [7], [18], [19], [1], [2], [16] e [13], trabalhando bem tanto com uma abordagem monoobjetivo quanto com muitos objetivos (dois ou mais critérios).

Para o desenvolvimento deste trabalho, foram implementadas duas abordagens do AE: (a) uma versão monoobjetivo do algoritmo, com a finalidade de minimização de apenas uma função objetivo,

calculada pela ponderação de três critérios diferentes (distância, tempo e combustível); (b) uma versão multiobjetivo, o AEMMT - Algoritmo Evolutivo MultiObjetivo com Muitas Tabelas, que compreende na minimização de quatro funções objetivos distintas de modo simultâneo, referentes à distância, ao tempo, ao combustível e à ponderação dos três critérios. Esta versão multiobjetivo foi inspirada no trabalho de [1]. Há diversas outras abordagens bem sucedidas de algoritmos multiobjetivos, tais como NSGA ([11]), NSGA-II ([5]), SPEA-II ([20]), MOEA-D ([18]) e NSGA-III ([6]). No entanto, neste trabalho desejou-se implementar o AEMMT por ser uma abordagem de múltiplas tabelas, e por isto, ter uma codificação mais próxima do monoobjetivo, além de mostrar-se eficiente em diversos problemas ([2], [14] e [8]). Deste modo, pretendia-se analisar o desempenho do AEMMT confrontando-o ao AE monoobjetivo, ambos aplicados ao PRV com dados hipotéticos, a fim de ratificar a superioridade das soluções obtidas pelo AEMMT, o que, de fato, foi observado com os resultados obtidos, especialmente examinando o comportamento evolutivo de cada critério. Futuramente, almeja-se realizar uma análise comparativa do AEMMT aplicado ao PRV com outros algoritmos evolutivos multiobjetivo e com muitos objetivos presentes na literatura.

A seguir, será explicada a metodologia de desenvolvimento do trabalho, com as principais técnicas abordadas e os detalhes de implementação e experimentação.

2 Metodologia

2.1 Algoritmos Evolutivos

Os Algoritmos Evolutivos são técnicas da grande área de Inteligência Artificial, sendo uma vertente que utiliza como fundamento principal a Teoria da Evolução de Charles Darwin [4]. Portanto, os AEs são algoritmos que mimetizam o comportamento evolutivo de uma população, que inspirados neste processo buscam otimizar a resolução dos mais variados problemas. A população em um AE representa um conjunto de possíveis soluções de um determinado problema. Desta maneira, os AES utilizam conceitos biológicos, como mutação e recombinação gênica, com o objetivo de melhorar uma possível solução para o problema que esteja sendo trabalhado.

Na próxima seção, será descrito brevemente o funcionamento de um Algoritmo Evolutivo.

2.2 Funcionamento de um Algoritmo Evolutivo

O Algoritmo Evolutivo pode ser descrito basicamente como um ciclo iterativo, que se inicia com a geração de uma população inicial composta por soluções aleatórias, usualmente. Com isto, a cada iteração do algoritmo, ocorre uma avaliação dos indivíduos desta população por meio de uma função *fitness*, que determina numericamente a qualidade de cada solução de uma população para um

*Both authors contributed equally to this research.

determinado problema. Com base no *fitness*, pode ser realizada a seleção dos indivíduos para cruzamento, priorizando aqueles com melhores soluções, dependendo do tipo de seleção [1]. Os indivíduos selecionados participam do processo de cruzamento, onde suas características são recombinadas para dar origem a uma nova solução. Estas novas soluções podem, ou não, passar por um operador de mutação. Este operador realiza pequenas mudanças na solução do indivíduo com objetivo de gerar novas características nos indivíduos, a fim de expandir o espaço de busca das soluções [1]. O AE também pode utilizar o recurso do elitismo, que é manter uma taxa dos melhores indivíduos para cada nova geração. Por fim, este ciclo é repetido até atingir a condição de parada estabelecida, que pode ser o número de gerações a ser alcançada ou a solução ótima ser encontrada [1].

Desta forma, com o objetivo de simular um processo evolutivo em uma população, são utilizados parâmetros para representar importantes aspectos de um AE, tais como: tamanho da população, número de rodadas da evolução (gerações), elitismo e taxa de mutação. Neste trabalho específico, também houveram variáveis para tratar do Problema de Roteamento de Veículos, como número de clientes a serem atendidos, número de veículos utilizados, capacidade de armazenamento de cada veículo e diferentes tipos de combustíveis.

A seguir, será descrito o AE aplicado ao PRV, e suas versões mono e multiobjetivo.

2.3 Algoritmo Evolutivo aplicado ao PRV

Neste trabalho, a criação da população inicial foi realizada de forma aleatória. Para seleção, foi utilizada a seleção por roleta, em que as chances de um indivíduo ser escolhido aumentam quanto melhor seu valor de aptidão (*fitness*). Após a seleção de indivíduos, realizou-se o cruzamento com apenas um ponto de corte, com taxa de 100%, em que um ponto de corte é delimitado em ambos os pais com o filho herdando a primeira metade do primeiro pai e a segunda metade do segundo pai. Em seguida, é decidido, baseado na taxa de mutação determinada como 1% neste trabalho, se ocorrerá ou não a mutação do filho criado durante o cruzamento. Por fim, o filho é introduzido na próxima população, repetindo este processo até completar a população da próxima geração do algoritmo. É importante destacar a aplicação do elitismo no algoritmo, onde este garantiu que uma porcentagem dos melhores indivíduos de uma geração sejam mantidos na próxima, visando preservar os melhores resultados em gerações seguintes. Neste trabalho, a taxa de elitismo foi de 5%. O AE foi executado até atingir o número de gerações (3000). Estes parâmetros foram definidos empiricamente, e foram usados tanto para versão mono quanto para o AEMMT. Para tratar do PRV, utilizou-se: o número de clientes (64), o número de veículos (2), a capacidade de armazenamento de cada veículo (32) e diferentes tipos de combustíveis (gasolina, diesel, etanol).

A versão monoobjetivo trabalha com apenas uma população, minimizando apenas uma função objetivo. Assim sendo, esta única função objetivo calcula a aptidão (*fitness*) do indivíduo com base em uma ponderação da distância, tempo, infrações de janela de tempo e gasto de combustível. Desta forma, caracteriza-se um bom indivíduo quanto menor o *fitness* retornado por esta função.

Por outro lado, o AEMMT, diferente da mono, foi implementado para trabalhar com múltiplas subpopulações simultaneamente por meio de tabelas, onde cada uma trata de uma função objetivo específica. Com isto, tem-se quatro subpopulações, uma para cada critério considerado no problema (distância, tempo e gasto de combustível), e uma última subpopulação para a minimização da ponderação dos três critérios. Deste modo, é esperado que o algoritmo com múltiplas tabelas expanda o alcance no espaço de busca para o problema em questão (segundo os pesquisadores [7], [1] e [2]), o que foi constatado com os resultados deste trabalho.

A seguir, serão apresentadas as equações utilizadas para o cálculo do *fitness* nas duas versões.

2.4 Equações utilizadas no AE monoobjetivo

A função objetivo para a versão monoobjetivo, adaptada de [12], foi a seguinte:

$$ct = \sum_{k=1}^K \sum_{i=0}^{nc} \sum_{j=0}^{nc} ((d_{ij} \cdot w_{distance}) + (t_{ij} \cdot w_{time}) + (f_{ij} \cdot w_{fuel}) \cdot x_{ijk}) \quad (1)$$

$$fit = (K \times w_v) + \left(\sum_{i=1}^K v_i \times w_{viol} \right) + ct \quad (2)$$

Onde: K é o número de veículos, nc é o número de clientes, d_{ij} é a distância entre os pontos i e j , t_{ij} é o tempo de viagem entre i e j , f_{ij} é o consumo de combustível para o trecho entre i e j , x_{ijk} indica se o percurso do vértice i ao vértice j foi feito pelo veículo k , w_v é o peso do número de veículos, v_i é o número de violações do veículo i , w_{viol} é o peso das violações, ct é o custo total, $w_{distance}$ é o peso da distância (1.0), w_{time} é o peso do tempo (0.5), e w_{fuel} é o peso do combustível (0.75). A Eq. (1) assegura que K veículos sejam utilizados, todos tendo início e término no mesmo ponto, e atendendo todos os cliente da rota. Logo, o custo total (ct) é dado pela somatória da distância, do tempo e do combustível gasto para cada veículo percorrer o trajeto entre os clientes de sua rota. A Eq. (2) representa o cálculo final do *fitness*, que multiplica o número de veículos pelo seu respectivo peso, somado ao somatório de todas as infrações de janela de tempo causadas por cada veículo multiplicada pelo peso das violações, e, enfim, somada ao custo total dado pela Eq. (1).

2.5 Equações utilizadas no AEMMT

Para o AEMMT, a Eq. (1) foi decomposta para cada subpopulação minimizar separadamente um aspecto do problema. Deste modo, uma subpopulação tratou da distância percorrida (Eq. (3)), outra dedicou-se ao tempo gasto e às violações de janela de tempo (Eq. (4)), a terceira ateve-se ao gasto de combustível na rota (Eq. (5)), e a última referiu-se à ponderação, sendo esta a mesma Eq. (1) da versão mono. Portanto, quatro subpopulações foram gerenciadas a fim de minimizar simultaneamente as quatro funções objetivos no AEMMT.

$$fitnessDistance = \left(\sum_{k=1}^K \sum_{i=0}^{nc} \sum_{j=0}^{nc} d_{ij} \cdot x_{ijk} \right) \cdot w_{distance} \quad (3)$$

$$\text{fitnessTime} = \left(\sum_{i=1}^K v_i \times w_{viol} \right) + \left(\sum_{k=1}^K \sum_{i=0}^{nc} \sum_{j=0}^{nc} t_{ij} \cdot x_{ijk} \right) \cdot w_{time} \quad (4)$$

$$\text{fitnessFuel} = \left(\sum_{k=1}^K \sum_{i=0}^{nc} \sum_{j=0}^{nc} f_{ij} \cdot x_{ijk} \right) \cdot w_{fuel} \quad (5)$$

Na próxima seção, são descritos alguns dos principais algoritmos multiobjetivos da literatura, que inspiraram o desenvolvimento do AEMMT [1][2].

2.6 Algoritmos Evolutivos Multiobjetivo

Os Algoritmos Evolutivos MultiObjetivo (AEMO's) têm sido amplamente utilizados há vários anos na resolução de problemas complexos, onde é necessário trabalhar com múltiplos critérios simultaneamente. A primeira geração iniciada em 1985 por [15], a partir de modificações realizadas no algoritmo proposto por [10]. Nesta geração, também havendo a introdução do conceito de Pareto por [9]. No entanto, os AEMOs tradicionais começaram a ser amplamente desenvolvidos a partir de 1990, com destaques como: NSGA [11] com ranqueamento de indivíduos não dominados; SPEA [19]; SPEA-II [20] com função de aptidão baseada no número de indivíduos que dominam ou são dominados; NSGA-II [5], que utiliza ordenação em ranking por dominância e originou o NSGA-III [6], capaz de tratar de mais de três objetivos de modo eficiente; e o MOEA-D [18], que decompõe o problema em subproblemas tratando-os simultaneamente, lidando com problemas com muitos objetivos. Neste sentido, surgiu o Algoritmo Evolutivo MultiObjetivo Baseado em Tabelas (AEMT) [7] em 2002, em que apresentou a separação da população em subpopulações organizadas em tabelas, cada uma possuindo uma função de aptidão diferente, tendo como principais vantagens a transferência de características entre as subpopulações e a possibilidade de trabalhar com diversas subpopulações simultaneamente. Como continuidade deste AE, O Algoritmo Evolutivo MultiObjetivo com Muitas Tabelas (AEMMT) foi aplicado pela primeira vez no trabalho de [1], possuindo similaridades quando comparado ao AEMT, principalmente pelas subpopulações divididas em tabelas. No entanto, ele se destaca por possibilitar trabalhar com um número muito maior de tabelas simultaneamente. Isto explica o maior alcance na busca por melhores soluções quando comparado ao mono e ao AEMT, como apresentado em [1].

A seguir, são apresentados detalhes do desenvolvimento deste trabalho, bem como: ambiente de desenvolvimento, dados utilizados e experimentos.

2.7 Ambiente de Desenvolvimento

Neste trabalho, foi utilizado predominantemente o sistema operacional Linux Ubuntu 22.04.5, a linguagem de programação C com bibliotecas padrão, usando o Visual Studio Code como editor de código.

A máquina utilizada para realizar os experimentos possui as seguintes configurações: processador AMD Ryzen 7 4800H, com 16gb de memória ram DDR4 e armazenamento SDD de 256gb.

Tabela 1: Evolução do Fitness ponderado no Monoobjetivo

Execução	Melhor da primeira geração	Melhor da última geração	Diferença
1	1217	1217	0
2	1337	1335	2
3	1349	1315	34
4	1198	1198	0
5	1333	1305	28
6	1248	1248	0
7	1230	1211	11
8	1310	1266	44
9	1219	1219	0
10	1273	1245	28

2.8 Dados utilizados nos experimentos

Para a realização dos experimentos, os indivíduos foram construídos de modo aleatório, onde os clientes foram gerados e posicionados aleatoriamente em um plano cartesiano. Deste modo, a cada teste realizado, os clientes foram inicializados novamente com posições x e y aleatórias.

Para a definição da melhor configuração possível dos parâmetros dos algoritmos, foram executados diversos testes até que se obtivesse as melhores soluções. Deste modo, empiricamente determinou-se o tamanho da população em 300, número de veículos em 2, número de clientes em 64, capacidade de cada veículo em 32, 1% de taxa de mutação, 5% de taxa de elitismo, 3000 gerações com seleção por roleta e cruzamento de um ponto (com taxa de 100%). Para lidar com o PRV, utilizou-se: o número de clientes (64), o número de veículos (2), a capacidade de armazenamento de cada veículo (32) e diferentes tipos de combustíveis (gasolina, diesel, etanol).

Foram realizadas 10 execuções com cada algoritmo, utilizando os parâmetros apresentados anteriormente para as duas versões. Ao final, foram averiguados os melhores valores do fitness da rota inicial, o melhor fitness encontrado em todas as gerações, o fitness médio da população final, o desvio padrão da população final e o tempo médio de execução.

3 Resultados e Discussão

3.1 Resultados

Os resultados obtidos a partir dos experimentos com o Algoritmo Evolutivo monoobjetivo e o AEMMT estão expostos a seguir. Os testes foram realizados utilizando a máquina descrita na Seção 2.7 e as configurações expostas na Seção 2.8.

Primeiramente, ressalta-se a superioridade da versão monoobjetivo na questão de tempo de execução, em que atingiu cerca de 17,5 segundos (em média) em cada execução, mais eficiente que a versão multiObjetivo, que necessitou de 350 segundos (em média) para realizar a execução do algoritmo.

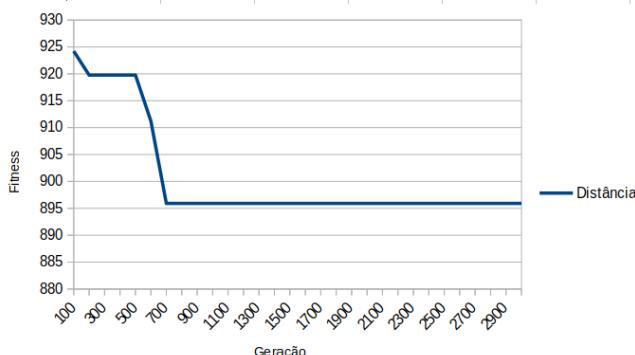
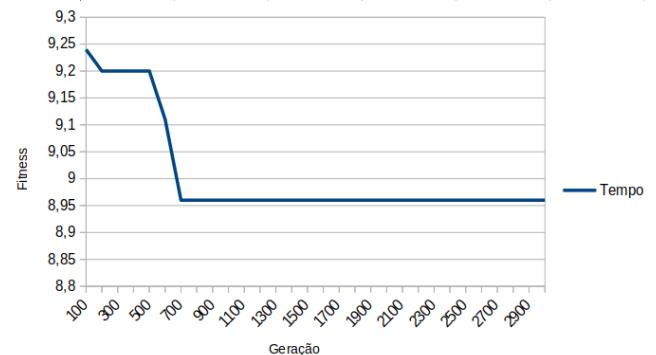
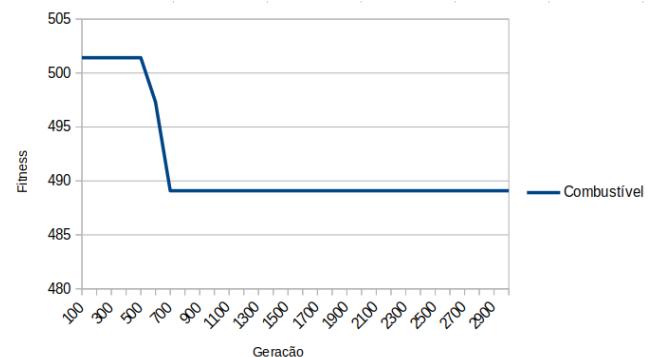
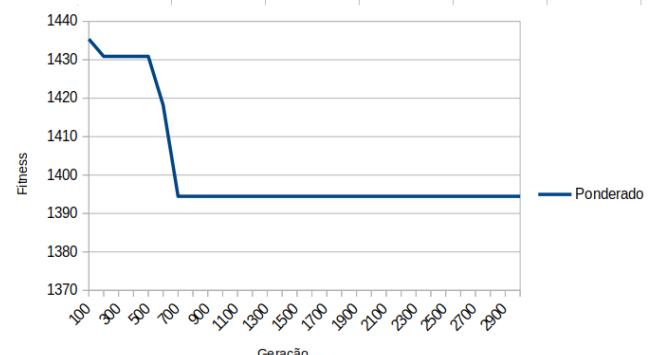
Em relação aos resultados gerados, nas Tabelas 1 e 2 são apresentados o melhor valor do fitness da primeira geração, o melhor fitness encontrado na última geração e a diferença entre eles. Analisando o fitness ponderado nas Tabelas 1 e 2, o AEMMT demonstrou ser superior sobre a versão monoobjetivo, uma vez que é possível observar

Tabela 2: Evolução do Fitness ponderado no AEMMT

Execução	Melhor da primeira geração	Melhor da última geração	Diferença
1	1307,6064	1297,2113	10,3951
2	1297,8458	1209,7522	88,0936
3	1242,7572	1183,914	58,8432
4	1371,8091	1286,874	84,9351
5	1124,5465	1089,9476	34,5989
6	1196,5895	1194,7303	1,8592
7	1303,0045	1221,8177	81,1868
8	1159,621	1137,1162	22,5048
9	1312,4554	1305,015	7,4404
10	1396,9568	1348,9772	47,9796

que em todas as execuções da versão multiobjetivo houve melhoria do primeiro melhor *fitness* encontrado até o melhor na última geração, isto ocorrendo em todos os testes. Por outro lado, a versão mono obteve melhoria em 6 de 10 execuções, conforme mostrado na Tabela 1. Perceba que, além do AEMMT produzir melhorias de *fitness* em todos os testes, a melhoria adquirida neste é maior que aquela quando obtida na versão mono.

Ao analisar a evolução de cada *fitness* separadamente na versão mono a partir das Figuras 1 a 4, nota-se que os valores de aptidão convergem até a geração 750, aproximadamente, permanecendo estagnados após esta geração, sugerindo que houve uma convergência prematura em um ótimo local. Em contrapartida, nas Figuras 5 a 8 pode-se observar claramente que o processo evolutivo no AEMMT gera novos melhores valores de *fitness* até a geração 2800, aproximadamente, com exceção do combustível, em que houve melhoria de aptidão em torno da geração 1700. Isto aponta que o AEMMT tem um poder de evolução maior que a versão mono quando se trata de múltiplos critérios, como é este caso.

**Figura 1: Evolução do fitness da distância percorrida ao longo das gerações para versão monoobjetivo****Figura 2: Evolução do fitness do tempo gasto ao longo das gerações para a versão monoobjetivo****Figura 3: Evolução do fitness do combustível gasto para percorrer as rotas ao longo das gerações para a versão monoobjetivo****Figura 4: Evolução do fitness ponderado ao longo das gerações para a versão monoobjetivo**

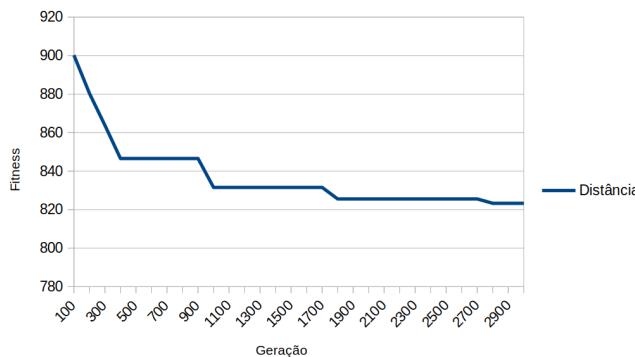


Figura 5: Evolução do *fitness* da distância percorrida ao longo das gerações para o AEMMT

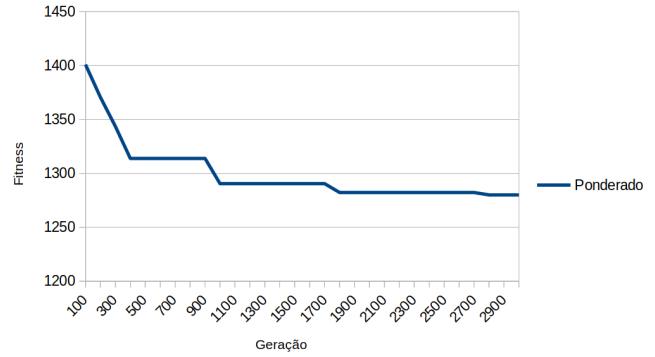


Figura 8: Evolução do *fitness* ponderado ao longo das gerações para o AEMMT

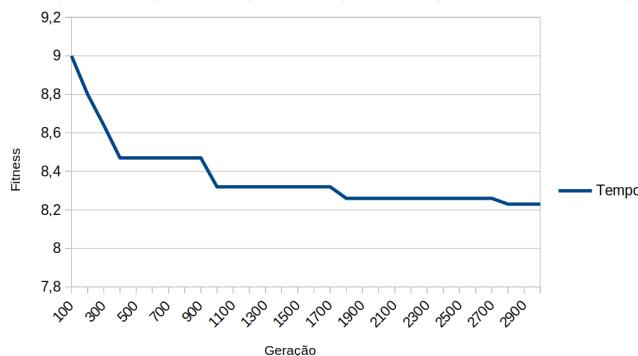


Figura 6: Evolução do *fitness* de tempo gasto ao longo das gerações para o AEMMT

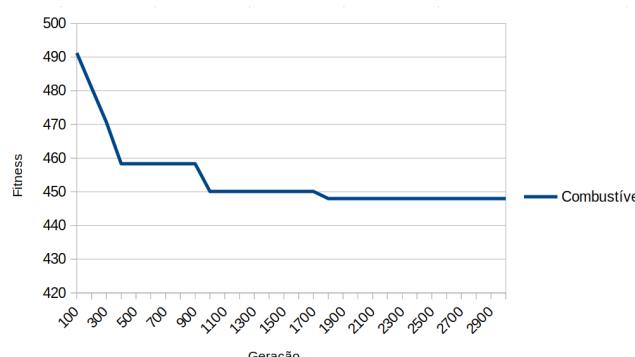


Figura 7: Evolução do *fitness* do combustível gasto para percorrer as rotas ao longo das gerações para o AEMMT

4 Conclusão

Após as análises realizadas sobre os resultados provenientes deste estudo, foi possível observar a superioridade do AEMMT na obtenção de melhores soluções para o problema de Roteamento de Veículos se comparado à versão monoobjetivo do AE. De fato, a versão com muitos objetivos obteve vantagens em relação à implementação da mono, tais como: (i) o AEMMT conseguiu melhorar o resultado do *fitness* em todos os 10 experimentos, comparando o melhor da primeira geração e o melhor da última geração, enquanto o monoobjetivo conseguiu em 6 de 10 experimentos; (ii) a melhoria do valor obtido do *fitness* é maior no AEMMT do que no monoobjetivo, quando este consegue obtê-la; (iii) ao se verificar a evolução de cada *fitness* separadamente na versão mono, observou-se que os valores de aptidão convergem até a geração 750, aproximadamente, permanecendo estagnados após esta geração, sugerindo que houve uma convergência prematura em um ótimo local, o que não é o esperado. Por outro lado, mostrou-se por meio dos experimentos que o processo evolutivo no AEMMT foi capaz de gerar novos melhores valores de *fitness* até a geração 2800, aproximadamente, com exceção do combustível, em que houve melhoria de aptidão em torno da geração 1700. Isto indica que o AEMMT tem um potencial evolutivo maior que a versão mono quando se trabalha com múltiplos critérios no problema de Roteamento de Veículos.

Estes resultados convergem com os trabalhos presentes na literatura [15], [10], [9], [11], [19], [20], [5], [18], [7], [1], [2], que apontam frequentemente que as versões multiobjetivos se apresentam superiores às mono em diversas aplicações.

Por fim, alguns aprimoramentos podem ser feitos neste trabalho, vislumbrando-se a obtenção de melhores resultados em momentos futuros. Deste modo, pretende-se realizar os experimentos com instâncias de Solomon[17] nas duas versões, dando continuidade a esta pesquisa, uma vez que, atualmente, os cenários são gerados aleatoriamente. Também almeja-se refinar o código do algoritmo evolutivo com muitos objetivos, visando tornar sua execução mais rápida e, deste modo, possibilitar a aplicação do mesmo em cenários reais. Além disto, futuramente deve ser abordada uma análise comparativa do AEMMT com importantes algoritmos multiobjetivos existentes na literatura, tais como NSGA-II [5], NSGA-III [6] e/ou MOEA-D [18].

References

- [1] Christiane Regina Soares Brasil. 2012. *Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas*. Ph.D. Dissertation. Escola de Engenharia de São Carlos, Universidade de São Paulo. Tese de Doutorado.
- [2] Christiane Regina Soares Brasil, Alexandre Cláudio Botazzo Delbem, and Fernando Luís Barroso da Silva. 2013. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. *Journal of Computational Chemistry*, 34, 20, 1719–1734. doi: 10.1002/jcc.23315.
- [3] George B. Dantzig and John H. Ramser. 1959. The truck dispatching problem. *Management Science*, 6, 1, 80–91.
- [4] Charles Darwin. 1859. *A origem das espécies*. John Murray.
- [5] Kalyanmoy Deb. 2001. *Multiobjective Optimization*. Springer. ISBN: 978-0-471-87339-6.
- [6] Kalyanmoy Deb and Himanshu Jain. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18, 4, 577–601. doi: 10.1109/TEVC.2013.2281535.
- [7] Alexandre Cláudio Botazzo Delbem. 2002. *Restabelecimento de energia em sistemas de distribuição por algoritmo evolucionário associado a cadeias de grafos*. Tese de Doutorado. Universidade de São Paulo, São Paulo, Brasil.
- [8] Thiago Fialho de Queiroz Lafetá. 2016. *Algoritmos evolutivos many objectives aplicados ao problema de roteamento Multicast com qualidade de serviço*. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Uberlândia, Uberlândia, Brasil, 148. doi: 10.14393/ufu.di.2016.58. Programa de Pós-graduação em Ciência da Computação.
- [9] David E. Goldberg. 1989. Genetic algorithms in search, optimization, and machine learning. *Addison Wesley*, 1989, 102, 36.
- [10] John H. Holland. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press. ISBN: 978-0262581110.
- [11] Amrit Pratap Kalyanmoy Deb Samir Agrawal and Tanaka Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: nsga-ii. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings* 6. Springer, 849–858.
- [12] Stanley Jefferson De Araujo Lima. 2015. *Otimização do problema de roteamento de veículos capacitado usando algoritmos genéticos com heurísticas e representações cromossómicas alternativas*. Master's thesis. Universidade Nove de Julho. Dissertação de Mestrado.
- [13] M. Oliveira Nascimento and C. Regina Soares Brasil. 2024. Uma investigação sobre os operadores genéticos no algoritmo evolutivo aplicado ao problema do percurso do cavalo. *Revista Brasileira de Computação Aplicada*, 16, 3, (dezembro 2024), 48–62. doi: 10.5335/rbca.v16i3.15725.
- [14] Johnata Ferreira Santos. 2023. *Algoritmos evolutivos multiobjetivo baseados em tabelas para escalonamento de tarefas em ambientes multiprocessados*. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Uberlândia, Uberlândia, Brasil, 113. doi: 10.14393/ufu.di.2023.163. Programa de Pós-graduação em Ciência da Computação.
- [15] J. Schaffer. 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates, Pittsburgh, PA, USA, 93–100.
- [16] Rafael Melo Silva and Christiane R. S. Brasil. 2020. A adaptabilidade de npcs em jogos com estilo presa-predador usando algoritmo genético. In *Anais do Computer on the Beach*. doi: 10.14210/cotb.v11n1.p056-058.
- [17] Marius M. Solomon. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35, 2.
- [18] Qingfu Zhang and Hui Li. 2007. Moea/d: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11, 6, 712–731.
- [19] Eckart Zitzler and Lothar Thiele. 1998. An evolutionary algorithm for multiobjective optimization: the strength pareto approach. *TIK report*, 43.
- [20] Eckart Zitzler and Lothar Thiele. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *ETH Zurich, Computer Engineering and Networks Laboratory*.