

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE GESTÃO E NEGÓCIO

ABNER SILVEIRA FACINA

Algoritmo Genético: um estudo comparativo dos métodos de seleção aplicado a um problema de otimização de carregamento em caminhão.

Uberlândia

2023

ABNER SILVEIRA FACINA

Algoritmo Genético: um estudo comparativo dos métodos de seleção aplicado a um problema de otimização de carregamento em caminhão.

Trabalho de Conclusão de Curso de Gestão da Informação da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Gestão da Informação.

Orientadora: Profa. Dra. Christiane Regina Soares Brasil.

Uberlândia

2023

ABNER SILVEIRA FACINA

Algoritmo Genético: um estudo comparativo dos métodos de seleção aplicado a um problema de otimização de carregamento em caminhão.

Trabalho de Conclusão de Curso de Gestão de Informação da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Gestão da Informação.

Uberlândia, 2023

Banca Examinadora:

Profa. Dra. Christiane Regina Soares Brasil
FACOM/UFU

Prof. Dr. Aleksandro Santos Soares
FACOM/UFU

Profa. Dra. Verônica Angélica Freitas de Paula
FAGEN/UFU

Dedico este trabalho aos meus pais e ao meu
irmão, pelo apoio, amparo e por acreditarem
em mim.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por ter me guardado e sustentado até aqui, porque sem Ele isso não seria possível.

Agradeço aos meus pais, principalmente minha mãe, Estela, por todo o apoio nessa jornada e meu irmão, Igor, por toda força e incentivo, e por sempre entenderem minha ausência em tempos de necessidade. Aos meus primos Pedro e Celso, por todo apoio.

Agradeço, especialmente, à professora Christiane pela paciência, e por toda atenção e apoio prestado no desenvolvimento deste trabalho.

Agradeço à Faculdade de Gestão e Negócio (FAGEN) e à Universidade Federal de Uberlândia (UFU) por me proporcionarem a oportunidade de me desenvolver intelectualmente em um ambiente incentivador, inovador e desafiador, por sempre acreditarem em seus alunos e sempre disponibilizar o melhor.

RESUMO

Os algoritmos genéticos simulam a seleção natural proposta por Darwin, sendo utilizados em várias áreas de conhecimento. Eles têm notoriedade em resolver problemas combinatórios, tais como o cacheiro viajante e o problema da mochila. Neste trabalho uma variante do problema da mochila é tratada com algoritmo genético, em que se busca a melhor maneira de transportar uma combinação de produtos que não exceda o limite de um caminhão e maximize o valor da venda (a soma do valor de todos os produtos neste carregamento) sem que seja repetido nenhum item. Além de encontrar a melhor solução para o problema proposto, também tem-se como objetivo secundário discutir a diferença no desempenho do algoritmo ao se utilizar o método da seleção da roleta e o método da seleção por torneio. O algoritmo foi codificado em Python, pois além de ser uma linguagem amplamente utilizada também por sua eficiência de trabalhar com listas. Ao final deste estudo, foi concluído que o algoritmo codificado com o método de seleção por torneio é mais eficiente que o método da roleta, ratificando a literatura desta área de pesquisa.

Palavras-chave: algoritmo genético; problema da mochila; métodos de seleção.

ABSTRACT

Genetic algorithms simulate the natural selection proposed by Darwin, being used in several areas of knowledge. They are famous for solving combinatorial problems, such as the traveling salesman problem and the knapsack problem. In this work, a variant of the knapsack problem is treated with a genetic algorithm, which seeks the best way to transport a combination of products that does not exceed the limit of a truck and maximizes the sales value (the sum of the price of all products in this load) without any item being repeated. In addition to finding the best solution to the proposed problem, the secondary objective is also to discuss the difference in the performance of the algorithm when using the roulette wheel selection method and the tournament selection method. The algorithm was coded in Python, as in addition to being a widely used language, it is also highly capable of working with lists. At the end of this study, it was concluded that the algorithm coded with the tournament selection method is more efficient than the roulette wheel method, confirming the literature in this area of research.

Keywords: genetic algorithm; knapsack problem; selection methods.

LISTA DE FIGURAS

Figura 1: Fluxograma do Algoritmo Genético.....	15
Figura 2: Método da roleta.....	16
Figura 3: Seleção por torneio	16
Figura 4: Recombinação 1-ponto.....	18
Figura 5: Recombinação n-pontos, n=2	18
Figura 6: Mutação simples.....	19
Figura 7: Pseudocódigo geral do algoritmo.....	23
Figura 8: Representação do indivíduo.....	23
Figura 9: Representação no pseudocódigo da função avaliação (fitness).....	24
Figura 10: Representação pseudocódigo da função crossover.....	24
Figura 11: Representação pseudocódigo da função mutação.....	25
Figura 12: Representação pseudocódigo da função seleção dos indivíduos por roleta.....	26
Figura 13: Pseudocódigo função seleção dos indivíduos por torneio.....	27
Figura 14: Representação pseudocódigo da função resolver: seleção de indivíduos, crossover, mutação e inserção dos filhos na nova geração.....	27
Figura 15: Representação pseudocódigo da função main.....	28
Figura 16: Exemplo de resultado apresentado pelo algoritmo.....	28
Figura 17: Sequência de testes 1.....	30
Figura 18: Resultados na sequência de testes 2.....	31
Figura 19: Resultados na sequência de testes 3.....	32
Figura 20: Resultados na sequência de testes 4.....	32
Figura 21: Resultados da sequência de testes 1, utilizando o método da roleta.....	35
Figura 22: Resultados da sequência de testes 2, utilizando o método por torneio.....	35
Figura 23: Resultados da sequência de testes 1, utilizando o método da roleta.....	36
Figura 24: Resultados da sequência de testes 2, utilizando o método por torneio.....	37
Figura 25: Resultados da sequência de testes 4, utilizando o método da roleta.....	39
Figura 26: Resultados da sequência de testes 3, utilizando o método por torneio.....	39

LISTA DE TABELAS

Tabela 1: A lista de 14 itens.....	22
Tabela 2: 14 itens adicionados.....	34
Tabela 3: 28 itens adicionados.....	35
Tabela 4: 56 itens adicionados.....	37

SUMÁRIO

1 INTRODUÇÃO	12
1.2 Objetivo	12
1.2 Justificativa	13
1.3 Organização do trabalho	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 Avaliação de aptidão (<i>fitness</i>).....	15
2.2 Seleção	16
2.3 Operadores de Reprodução.....	17
2.4 Cruzamento (<i>crossover</i>) ou recombinação	17
2.5 Mutação.....	18
2.6 Problema da Mochila.....	19
2.7 Trabalhos relacionados.....	20
2.8 Considerações finais do capítulo	21
3 METODOLOGIA	22
3.1 Linguagem.....	22
3.2 Implementação.....	22
3.2.1 Classe produto	23
3.2.2 Classe indivíduo	23
3.2.2.1 Função avaliação	23
3.2.2.2 Função <i>crossover</i>	24
3.2.2.3 Função mutação.....	24
3.2.3 Classe AlgoritmoGenetico	25
3.2.3.1 Função inicializa população	25
3.2.3.2 Função ordena população.....	25
3.2.3.3 Função melhor individuo.....	25
3.2.3.4 Função soma avaliação.....	26
3.2.3.5 Função seleciona pai (método da roleta).....	26
3.2.3.6 Função seleciona pai (torneio)	26
3.2.3.7 Função visualiza	27
3.2.3.8 Função resolver	27
3.2.4 Função main	28
3.3 Considerações finais do capítulo	28

4 RESULTADOS.....	30
4.1 Primeira etapa de experimentos	30
4.1.1 Sequência de Testes 1.....	30
4.1.2 Sequência de Testes 2.....	31
4.1.3 Sequência de testes 3.....	32
4.1.4 Sequência de testes 4.....	32
4.1.5 Discutindo resultados	33
4.2 Segunda etapa de experimentos	34
4.2.1 Sequência de testes 1.....	34
4.2.2 Sequência de testes 2.....	35
4.2.3 Sequência de testes 3.....	37
4.3.4 Discutindo resultados	39
4.4 Considerações finais do capítulo	40
5 CONSIDERAÇÕES FINAIS.....	41
REFERÊNCIAS	42

1 INTRODUÇÃO

A computação vem crescendo notavelmente, atuando em diferentes contextos, diante da grande demanda de problemas de alta complexidade a serem resolvidos. Diversos destes problemas são classificados como combinatórios e podem ser matemáticos ou do mundo real. Como exemplos famosos destes tipos de problemas, tem-se: o Problema do Caixeiro Viajante (DANTZIG, 1954) e o Problema da Mochila (WEINGART, 1966) e como exemplo de problemas do mundo real tem-se: programação de horários em escolas (SOUZA, 2002), predição de estruturas de proteínas (BRASIL, 2012), planejamento da manutenção preventiva na rede de fornecimento de energia elétrica (PEIXOTO, 2015), jogos (BARRETO, 2021), dentre tantos outros. Alguns problemas podem ser resolvidos usando métodos computacionais determinísticos, porém à medida que o tamanho do problema cresce se torna inviável por exigirem muita capacidade computacional, para isso existem diversas heurísticas computacionais que buscam a solução ótima ou a uma aproximação dela, em que se pode destacar o Algoritmo Genético.

Algoritmos Genéticos (AGs) são técnicas de otimização computacional, que tem como sua principal inspiração os conceitos da biologia. O principal fundamento é a Teoria da Evolução de Charles Darwin, tendo como destaque a seleção natural – em que os mais aptos sobrevivem – a reprodução genética e a mutação.

Esta junção interdisciplinar entre os campos computacionais e os campos biológicos intensificou-se nas décadas de 50 e 60, quando começaram a ser realizadas simulações computacionais de sistemas genéticos. Apesar desse avanço, foi no começo da década de 70, com John Holland que o tema Algoritmo Genético começou a ser seriamente pesquisado, dando origem ao seu livro "*Adaptation in Natural and Artificial Systems*" (HOLLAND, 1992), tido como a “bíblia” do Algoritmo Genético. Porém, foi com seu aluno, David E. Goldberg (GOLDENBERG, 1989), na década de 80, que esta tese foi perpetuada, uma vez que David teve sucesso em aplicar AGs industrialmente.

1.2 Objetivo

Este Trabalho de Conclusão de Curso visou aplicar um algoritmo para resolver o seguinte problema: carregar um caminhão que tem um espaço limitado, maximizando o valor total a ser carregado, sendo que se tem uma quantidade de itens à disposição, todos com suas dimensões e preços disponíveis. Neste sentido, nota-se uma semelhança ao famoso Problema da Mochila.

Portanto, o objetivo do Algoritmo Genético foi buscar a melhor maneira de transportar uma lista de produtos predefinida que não excedesse o limite de um caminhão e maximizasse o valor da venda (a soma dos preços de todos os produtos neste carregamento), desde que não fosse repetido nenhum item. Como objetivo secundário do trabalho, analisou-se a diferença no desempenho do algoritmo ao ser utilizado o método da seleção da roleta e o método da seleção por torneio.

1.2 Justificativa

Diversas aplicações de Algoritmos Genéticos para resolver problemas combinatórios semelhantes Problema da Mochila vêm sendo publicadas e discutidas há décadas, uma vez que este tema é bastante atual e pertinente por ser adaptado a diversas questões combinatórias encontradas no mundo real. Como existe uma dificuldade em resolver de forma eficiente esses problemas com algoritmos determinísticos à medida que o tamanho da instância cresce, os Algoritmos Genéticos têm se mostrado uma opção adequada (GOUVEIA, 2021).

A importância de estudar AG é a sua adaptabilidade em diversos outros problemas, que busquem otimizar o processo de transporte de carga, maximizando o valor a ser deslocado, seja ele terrestre, aéreo, ou fluvial, por exemplo. Em um país de dimensões continentais como o Brasil, estudar e buscar novas maneiras de melhorar este transporte se faz necessário, indubitavelmente. Neste sentido, este Trabalho de Conclusão de Curso visa estimular pesquisas neste tema, especialmente ao constatar a significativa melhora dos resultados do AG a partir de importantes ajustes de seus parâmetros.

1.3 Organização do trabalho

O trabalho foi estruturado em 6 capítulos, sendo divididos como:

- No Capítulo 2 são apresentados os principais conceitos sobre o Algoritmo Genético, técnica de otimização estudada, e sobre o Problema da Mochila, que é a fundamentação do problema do caminhão tratado neste trabalho.
- No Capítulo 3 é descrito o Algoritmo Genético desenvolvido neste trabalho.
- No Capítulo 4 são apresentados os resultados obtidos e as discussões geradas.
- No Capítulo 5 serão discutidas as considerações finais do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

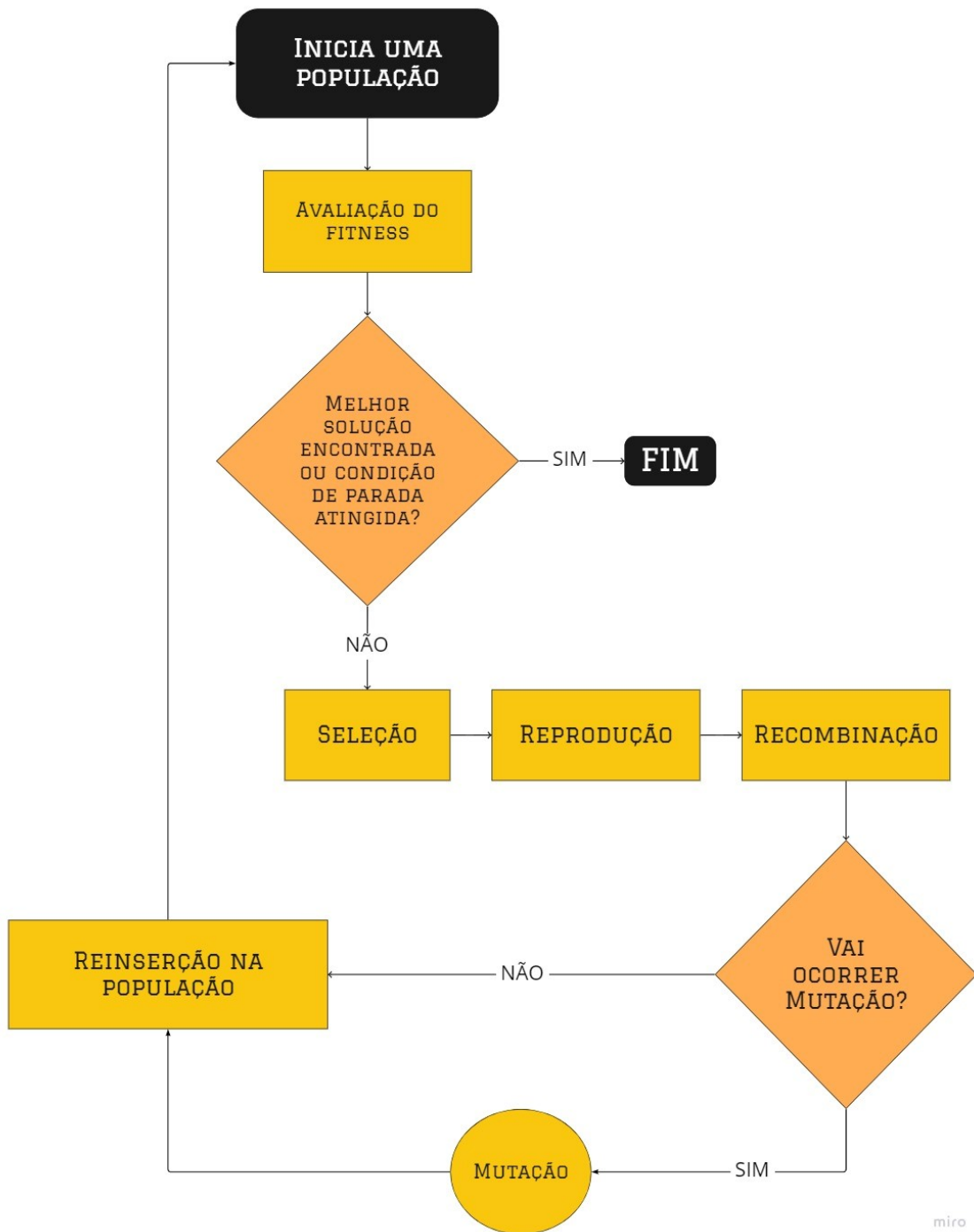
Neste capítulo alguns dos conceitos biológicos fundamentais serão apresentados, sendo eles: indivíduos, população, reprodução e mutação. Indivíduos são organismos vivos que pertencem a uma espécie e que se diferem dos demais devido às suas características particulares, enquanto que a população é o conjunto de indivíduos que habitam em um mesmo lugar e que concorrem entre si por recursos, como exemplificado por Mazza et al (1994). A reprodução é o processo em que um ou mais indivíduos produzem descendentes, passando para eles todos ou uma parte de seus genes. Pode-se classificar a reprodução como assexuada, onde um indivíduo sozinho gera descendentes por meio da mutação, e a sexuada: onde dois indivíduos geram um ou mais descendentes por meio da recombinação. A saber, a mutação é quando um ou mais genes de um indivíduo é aleatoriamente modificado.

Traduzindo alguns destes conceitos para a representação computacional tem-se, segundo Gabriel (2008): o indivíduo é representado por um vetor que guarda uma sequência de símbolos, sendo eles números ou caracteres, exclusivos dele e a população é um conjunto desses vetores. Para cada indivíduo é associado um valor de aptidão (*fitness*), que indica o quanto este indivíduo está apto para resolver o problema em questão.

De maneira bem concisa, os Algoritmos Genéticos (AGs) funcionam da seguinte maneira: é gerada uma população, de tamanho adequado ao problema, cada indivíduo é avaliado de acordo com seu *fitness* (o quão apto ele é). Com base nos *fitness* adquiridos, os indivíduos são selecionados: o modo de seleção varia para cada tipo de problema, porém a maioria deles busca selecionar os indivíduos que melhor solucionam o problema, onde os mais aptos são escolhidos (reforçando a teoria em que o mais “forte” sobrevive). Em seguida, os indivíduos selecionados se reproduzem, gerando novos indivíduos, que geram novas potenciais soluções, por meio da recombinação e/ou da mutação. Este processo é repetido até que a condição de parada seja satisfeita. A condição de parada varia de acordo com cada problema a ser solucionado e ela representa a solução “ótima” ou que foi atingido o número máximo de gerações.

Nas próximas seções serão descritas as principais funções do AG.

Figura 1: Fluxograma do Algoritmo Genético.



Fonte: autor.

2.1 Avaliação de aptidão (*fitness*)

Cada indivíduo de uma população tem seu *fitness* calculado (um valor numérico), isto é, o quão apto o indivíduo está de ser uma boa escolha para se obter a solução desejada ou a quão boa é a solução. Esta avaliação é essencial no Algoritmo Genético, sendo uma

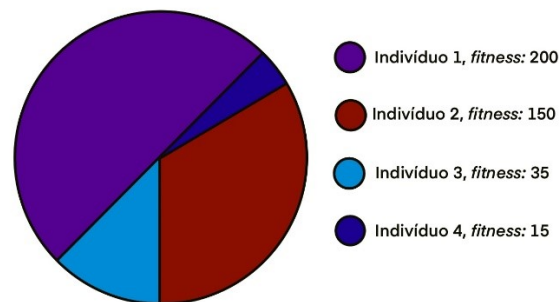
função elaborada de acordo com cada problema específico, a fim de qualificar as candidatas à soluções.

2.2 Seleção

Com os indivíduos devidamente avaliados, são selecionados os pais para gerar novos indivíduos por meio de operadores de reprodução. Dentre vários métodos de seleção, alguns dos mais usados são: método da roleta e a seleção por torneio (LUNARTI, 2020).

No método da roleta, os indivíduos são dispostos em uma roleta, onde os mais aptos ocupam uma porção maior da roleta, enquanto que os menos aptos são dispostos em porções menores. É ilustrado na Figura 2 um cenário onde o indivíduo 1 possui o melhor *fitness*, considerando que para este problema se deseja maximizar a aptidão.

Figura 2: Método da roleta.



Fonte: autor.

No método da seleção por Torneio de N, é selecionado aleatoriamente um grupo de N indivíduos, sendo esse grupo menor que a população inicial. Com esse grupo formado, o indivíduo com melhor aptidão é selecionado para ser um pai. Esse torneio será feito novamente a fim de selecionar do mesmo modo um outro pai. A partir destes pais é efetuada a formação de um novo indivíduo, aplicando os operadores de reprodução.

Figura 3: Seleção por torneio

Indivíduo 1:	0	1	1	0	0	1	1	1	0	1	1	1	0	0
	Fitness = 10													
Indivíduo 2:	1	1	1	1	0	0	0	1	0	1	0	0	1	0
	Fitness = 6													

Fonte: autor.

Como exemplificado na Figura 3, a seleção por torneio seleciona indivíduos aleatoriamente e, em seguida, compara seus respectivos *fitness*, caso o objetivo do *fitness*

seja maximizar o resultado, o indivíduo 1 será escolhido, caso seja para minimizar o resultado, o indivíduo 2 será escolhido.

2.3 Operadores de Reprodução

A reprodução, num contexto biológico, é vista de duas formas:

- 1- Assexuada: um único indivíduo gera descendentes por meio da mutação;
- 2- Sexuada: o envolvimento de dois indivíduos gera um ou mais descendentes, pelo processo de recombinação;

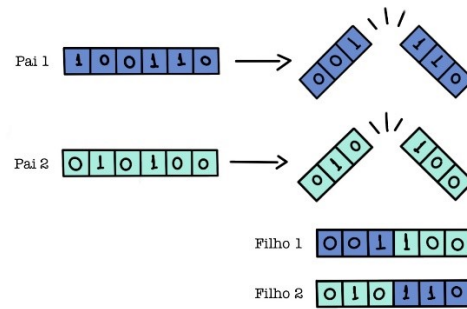
Tanto a mutação quanto a recombinação são operadores necessários para a diversificação da população, e para que a nova população mantenha as melhorias evolutivas obtidas anteriormente.

2.4 Cruzamento (*crossover*) ou recombinação

Este operador “mistura” as características de dois ou mais indivíduos, “pais”, para obter indivíduos novos, “filhos”. Essa mistura é a recombinação de alelos, que são as estruturas responsáveis por características específicas de um indivíduo. O cruzamento (*crossover*) acontece quando um ponto de corte é selecionado em um ponto aleatório do cromossomo dos pais, e eles são trocados de um pai para o outro, gerando dois filhos, descendentes. Este operador é vastamente utilizado (HASSANAT, 2019).

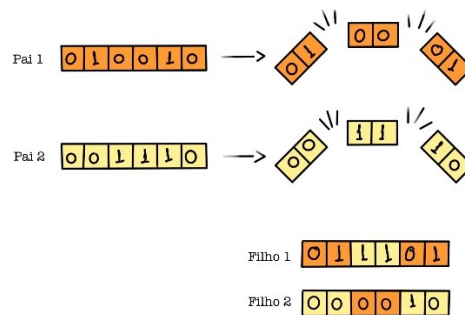
Existem diversos tipos de recombinação, tais como recombinação de 1-ponto, de n-pontos, discreta, recombinação aritmética, *blend crossover*, recombinação com múltiplos pais, dentre outros (GABRIEL, 2008). Neste trabalho foram utilizados dois tipos específicos: o de 1-ponto e o de n-pontos. Na recombinação de 1-ponto, o cromossomo é partido em um ponto, escolhido aleatoriamente, o dividindo em um pedaço à esquerda e outro à direita da partição. Cada novo indivíduo criado é o resultado da junção da partição esquerda do Pai 1 com a partição direita do Pai 2 e pela junção da partição esquerda do Pai 2 com a partição direita do Pai 1, como ilustrado na Figura 4. Na recombinação de n-pontos, os cromossomos são partidos em n posições que são recombinadas para criar novos descendentes (Figura 5).

Figura 4: Recombinação 1-ponto.



Fonte: autor.

Figura 5: Recombinação n-pontos, n=2

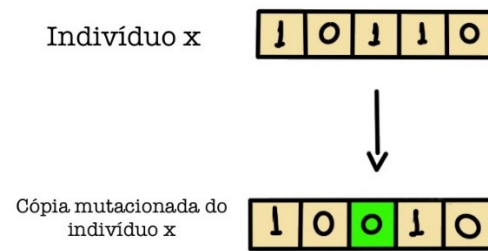


Fonte: autor.

2.5 Mutação

O operador de mutação é responsável pela modificação aleatória de um ou mais genes de um cromossomo. Um ou mais genes são escolhidos de forma aleatória em um cromossomo, tendo seu valor alterado também de modo randômico. Sendo assim, este operador introduz uma variabilidade genética na população que não existia, evitando uma eventual estagnação no processo evolutivo da mesma (convergência prematura). A probabilidade de que um gene sofra uma mutação é chamada de taxa de mutação, que é um valor geralmente pequeno, uma vez que pode mudar um indivíduo para melhor ou pior do que o original (Figura 6).

Figura 6: Mutação simples.



Fonte: autor

Existem também outros tipos de mutação, além da simples, como: mutação uniforme, mutação Gaussiana, mutação de ordem K, dentre outros (CORTES, 2019; NOGUEIRA, 1999; SOUZA, 2022).

2.6 Problema da Mochila

O Problema da Mochila – PM (*Knapsack Problem – KP*) é um dos problemas combinatórios clássicos de otimização (WEINGART, 1966), que busca elaborar uma configuração de objetos cujo valor ou lucro total seja maximizado, sendo restrita pela capacidade total da mochila (NAND & SHARMA, 2019). Há também o Problema da Mochila 0/1, que é um problema binário onde existe um conjunto de itens (com seus respectivos custos) e um conjunto de restrições, tendo como objetivo encontrar uma combinação de itens que respeite as restrições e maximize a função objetivo (MUNHÃO, 2019).

Existem muitos algoritmos propostos para a resolução do problema da mochila (ABDEL-BASSET et al., 2019), considerando a complexidade do problema e o consumo de recursos computacionais requeridos para resolvê-lo, que tendem a ter dificuldade em encontrar boas soluções à medida que o tamanho da instância cresce, ou seja, quando se tem um tamanho suficientemente grande. Como consequência, as meta-heurísticas são alternativas indicadas para encontrar soluções aceitáveis em tempo hábil. Dentre estes métodos, alguns são inspirados na natureza, tais como: Algoritmos Genéticos (HOLLAND, 1992), Otimização por Enxame de Partículas (EBERHART & KENNEDY, 1995) e Evolução Diferencial (STORN & PRICE, 1997).

Neste trabalho foi usado Algoritmo Genético para resolver uma variante do problema da mochila, em que existe uma lista de itens à disposição para serem colocados em um caminhão de tamanho delimitado visando maximizar o lucro. Este algoritmo foi implementado em duas versões: uma utilizando o método de seleção da roleta, e outra usando o método de seleção por torneio.

2.7 Trabalhos relacionados

Quando se trata do problema da mochila, existem vários tipos de soluções propostas na literatura, destacando-se os algoritmos genéticos e suas variações. A seguir, serão mostrados trabalhos com aplicações de AGs para a resolução do PM que foram publicados no intervalo entre 2019 e 2023, focando os mais citados deste período.

Ozsoydan e Baykasoglu (2019) propõem uma nova técnica de inteligência de enxame, simples e eficaz, baseada em algoritmo genético e otimização por enxame de partículas. A eficácia deste novo método foi aplicada em um novo problema da mochila, *the Set-union Knapsack Problem*, que tem uma vasta gama de aplicações reais como sistemas de segurança da informação. Além da facilidade da implementação e da não necessidade de funções de transferência e procedimentos de busca, também é proposta uma segunda mutação, sendo ela opcional, que diminui exponencialmente a diversidade introduzida na população. Dentre os resultados obtidos, é descoberto que esse novo processo de mutação pode ser adaptado para qualquer algoritmo meta-heurístico, contribuindo significativamente para o desempenho do algoritmo.

Ozsoydan (2019) apresenta um novo algoritmo de otimização baseado em enxame, composto por agentes artificiais de busca, tendo cada um deles uma inteligência cognitiva individual. Cada agente pode aprender com o espaço do problema, cada um deles busca uma característica de pesquisa diferente. A eficácia do novo método proposto é posta em prova com um conjunto de problemas de otimização binária, como o problema da mochila e o problema da localização de instalações incapacitadas. Os resultados obtidos destes testes comprovam que esse novo algoritmo proposto melhora significativamente os resultados, se comparados com os algoritmos não modificados.

Bayraktar, Ersoz e Kubat (2021) discutem a utilização do algoritmo de colônia de abelhas artificial para a resolução de problema da mochila. Os aspectos de desbravamento e exploração dos algoritmos de colônia são melhorados utilizando mecanismos de memória e operadores genéticos para desenvolver algoritmos híbridos diferentes. O algoritmo emergiu como uma abordagem promissora para resolver o problema de

carregamento de contêiner único com desempenho médio, e concluiu-se que os operadores genéticos são mais eficazes que o mecanismo de memória para desenvolver um algoritmo híbrido.

Yang, Kim e Yoon (2022) apresentam um algoritmo mimético para o problema da mochila multidimensional. Eles apresentaram um novo “reparo” heurístico baseado na função de tendência e uma busca genética por aproximação da função. O reparo pode transformar, com sucesso, soluções inviáveis em soluções viáveis. O algoritmo foi projetado para funcionar bem em instâncias de problemas com baixa densidade de soluções viáveis, transformando soluções inviáveis em viáveis. Após testes, concluiu-se que os algoritmos genéticos com o reparo apresentaram melhor desempenho ao comparado aos algoritmos sem o reparo.

Liu, Jing e Hou (2023) discutem uma solução para o problema de reconhecimento de incêndios florestais utilizando veículos aéreos não tripulados. Foi apresentado como solução um modelo que usa, em sua primeira fase, maximização de expectativas e, em sua segunda fase, o modelo do problema da mochila multidimensional para a atribuição de tarefas para o reconhecimento de incêndios florestais. Sensores via satélite informam sobre os incêndios, em seguida o algoritmo de maximização de expectativas é utilizado para obter a posição inicial de cada aeronave, em seguida o PM é usado para atribuir as tarefas de cara aeronave com base em sua posição inicial. Um algoritmo genético aprimorado baseado no algoritmo de fogos de artifício foi proposto para acelerar velocidade de iteração. Após testes realizados via simulação, concluiu-se que o modelo de atribuição de tarefas pode resolver de forma rápida e eficaz problemas de alocação de tarefas em grande escala. Além disso, o algoritmo híbrido (algoritmo genético e fogos de artifício) apresenta grandes vantagens sobre o GA tradicional, principalmente no tempo de resolução, na velocidade de convergência da iteração e na eficácia da solução.

2.8 Considerações finais do capítulo

Neste capítulo foi abordada a fundamentação teórica para o desenvolvimento deste trabalho. Foram explorados a definição de Algoritmo Genético e o seu funcionamento, e a apresentação do Problema da Mochila, que é a base conceitual do problema do caminhão tratado neste Trabalho de Conclusão de Curso.

No Capítulo 3 será apresentado o desenvolvimento do AG com a finalidade de maximizar o valor transportado por um caminhão.

3 METODOLOGIA

O cenário da problemática do algoritmo (problema original) é, primeiramente, carregar um caminhão que tem como limite de espaço 3 metros cúbicos e temos 14 produtos a disposição, como observado na Tabela 1, todos com suas dimensões e preço disponíveis. Tanto a limitação do caminhão quanto os itens iniciais foram usados como o cenário base a ser resolvido. A partir das melhorias encontradas neste formato original, foram realizados novos testes em diferentes cenários. Deste modo, buscou-se a melhor maneira de transportar uma combinação de produtos que não excedesse o limite do caminhão e maximizasse o valor da venda. O valor ideal (a soma de todos os valores dos produtos) era R\$27.042,35 e como volume desejado (soma dos espaços de todos os itens), 4,8 metros cúbicos.

Tabela 1: A lista de 14 itens.

Itens	Volume	Valor
Geladeira Dako	0.751	999.90
	0.000089	
Iphone 6	9	2911.12
TV 55'	0.400	4346.99
TV 50'	0.290	3999.90
TV 42'	0.200	2999.00
Notebook Dell	0.00350	2499.90
Ventilador Panasonic	0.496	199.90
Microondas Electro-lux	0.0424	308.66
Microondas LG	0.0544	429.90
Microondas Panasonic	0.0319	299.29
Geladeira Brastemp	0.635	849.00
Geladeira Consul	0.870	1199.89
Notebook Lenovo	0.498	1999.90
Notebook Asus	0.527	3999.00

Fonte: autor.

3.1 Linguagem

Este trabalho foi desenvolvido utilizando a linguagem Python, versão 3.11, para a codificação do algoritmo, por sua eficiência em trabalhar com listas e por sua simplicidade de uso. O ambiente de codificação usado foi o Pycharm.

3.2 Implementação

Nesta seção serão apresentadas as funções implementadas neste AG específico. A Figura 7 mostra uma visão geral do algoritmo implementado.

Figura 7: Pseudocódigo geral do algoritmo.

```

Importa bibliotecas

Classe produto
Classe individuo
    Função avaliação
    Função crossover
    Função mutação
Classe AlgoritmoGenetico
    Função inicializa população
    Função ordena população
    Função melhor indivíduo
    Função soma avaliação
    Função seleciona pai (método da roleta)
    Função visualiza geração
    Função resolver

Main

```

Fonte: autor.

3.2.1 Classe produto

Primeiramente, é criada uma classe produto, que recebe os atributos nome, espaço e valor, e os atributos para cada produto em uma variável (objeto). Em seguida, é criada e preenchida uma lista, onde cada posição dessa lista representa um objeto, que armazena todas as informações de um determinado produto.

3.2.2 Classe indivíduo

Cada indivíduo representa uma lista de objetos a serem levados no caminhão. Portanto, a classe indivíduo cria uma lista (indivíduo). A princípio, a “escolha” de qual produto deve ou não ser carregado é feita de forma randômica, onde cada produto teve 50% de chance de ser aleatoriamente escolhido ou não. O cromossomo gerado, como representado na Figura 8, é uma lista de 0's e 1's, onde os 1 representam que o produto foi escolhido e o 0 que o produto não foi escolhido.

Figura 8: Representação do indivíduo.

Cromossomo:	1	0	0	1	1	0	0	0	1	0	0	0	1	1
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fonte: autor.

3.2.2.1 Função avaliação

Após a criação da classe indivíduo, é criada a função avaliação que, como descrito na Figura 9, a função avaliação (*fitness*) deve medir a “qualidade” do cromossomo em resolver o problema. Logo, a função recebe o cromossomo, a soma dos valores (preço) de cada produto escolhido e o espaço que eles ocupam. Caso o espaço ocupado por essa solução seja maior que o limite do caminhão, ele recebe a nota = 1 (penalizando, desta

forma, quando a solução excede o limite). A função busca maximizar o lucro, ou seja, encontrar o maior valor a ser transportado.

Figura 9: Representação no pseudocódigo da função avaliação (*fitness*).

```

Função avaliacao():
    nota = 0
    soma_espacos = 0

    Para cada i no intervalo de 0 a tamanho do cromossomo:
        Se o cromossomo na posição i for igual a '1':
            Adiciona o valor na posição i aos valores da nota
            Adiciona o espaço na posição i à soma_espacos

    Se a soma_espacos for maior que o limite de espaços:
        Defina a nota como 1

    Defina a nota_avaliacao como nota
    Defina o espaco_usado como soma_espacos

```

Fonte: autor.

3.2.2.2 Função *crossover*

Em seguida, é criada a função *crossover*, com taxa de cruzamento de 100%, representada na Figura 10, que recebe os cromossomos dos pais e de forma randômica efetua o corte em uma posição do cromossomo (*crossover* de 1 ponto). Após os cromossomos serem divididos, o cromossomo filho1 deve receber a primeira metade do cromossomo do pai2 e a segunda metade do pai1, e o filho2 deve receber a primeira parte do pai1 e a segunda do pai2. Deste modo, a função gera novos indivíduos (filhos).

Figura 10: Representação pseudocódigo da função *crossover*.

```

Função crossover(outro_individuo):
    corte = arredonda(gerar número aleatório entre 0 e o tamanho do cromossomo)

    filho1 = primeira parte de genes do pai 2 + a segunda parte de genes do pai 1
    filho2 = primeira parte de genes do pai 1 + a segunda parte de genes do pai 2

    Crie um array de filhos com dois elementos:
        - Primeiro filho é um novo Indivíduo com os valores de espacos, valores, limite_espacos, e geracao + 1.
        - Segundo filho é um novo Indivíduo com os valores de espacos, valores, limite_espacos, e geracao + 1.

    Defina o cromossomo do primeiro filho como filho1
    Defina o cromossomo do segundo filho como filho2

    Retorne o array de filhos

```

Fonte: autor.

3.2.2.3 Função mutação

A função mutação muda aleatoriamente uma posição *i* do cromossomo. Esta função recebe um cromossomo e a taxa de mutação previamente definida. A função percorre posição por posição da lista e verifica se ocorrerá mutação ou não: se ocorrer, a função trocará 1 por 0 ou 0 por 1, caso não ocorra nada será mudado, como observado na Figura 11. A mutação promove a inserção de uma variabilidade genética na população,

que outrora não existia, evitando uma convergência prematura, ou seja, reduzindo a chance que a população fique estagnada em um ótimo local, sem ter como criar novas soluções.

Figura 11: Representação pseudocódigo da função mutação.

```

Função mutacao(taxa_mutacao):
  Para cada i no intervalo de 0 a tamanho do cromossomo:
    Se gerar um número aleatório entre 0 e 1 for menor que a taxa_mutacao:
      Se o cromossomo na posição i for igual a '1':
        Defina o cromossomo na posição i como '0'
      Senão:
        Defina o cromossomo na posição i como '1'

  Retorne o próprio objeto (self)

```

Fonte: autor.

3.2.3 Classe AlgoritmoGenetico

É criada a classe AlgoritmoGenetico, onde é armazenado o tamanho da população, os indivíduos dessa população, melhor solução e a geração.

3.2.3.1 Função inicializa população

A função de inicialização começa criando os primeiros indivíduos de forma aleatória, de acordo com o tamanho da população indicada e então, para fins de iniciar a busca pela melhor solução, é criado a variável `melhor_solucao` que recebe a primeira posição da lista população(`populacao[0]`). Essa atribuição é feita para que a variável `melhor_solucao` seja inicializada, uma vez que será aplicada a ordenação na população. Deste modo, a primeira posição conterà o melhor *fitness*.

3.2.3.2 Função ordena população

A função `ordena população` ordena a lista população de acordo com a `nota_avalicao` de maneira decrescente, da maior `nota_avalicao` até a menor.

3.2.3.3 Função melhor individuo

A função `melhor individuo` recebe um indivíduo e compara sua `nota_avalicao` com o valor da `melhor_avalicao`: se a `nota_avalicao` for maior que a `melhor_avalicao`, então a `melhor_avalicao` recebe o valor da `nota_avalicao`.

3.2.3.4 Função soma avaliação

A função soma avaliação percorre todos os indivíduos da população e soma as notas de avaliação, armazenando este somatório na variável soma.

3.2.3.5 Função seleciona pai (método da roleta)

A função seleciona_pai (na versão por roleta) recebe a soma_avaliacao e, em seguida, multiplica esse valor por um valor randômico, de 0 a 1, armazenando o resultado na variável valor_sorteado. Enquanto a soma da nota_avaliacao for menor que o valor_sorteado a função percorrerá a lista população somando as nota_avaliacao até chegar no valor mais próximo do valor sorteado e, então, retornar o índice desse indivíduo, como mostrado na Figura 12.

Figura 12: Representação pseudocódigo da função seleção dos indivíduos por roleta.

```

Função seleciona_pai(soma_avaliacao):
    pai = -1
    valor_sorteado = gerar número aleatório entre 0 e soma_avaliacao
    soma = 0
    i = 0

    Enquanto i for menor que o tamanho da população E soma for menor que valor_sorteado:
        soma += nota_avaliacao do indivíduo na posição i da população
        pai += 1
        i += 1

    Retorne o valor de pai

```

Fonte: autor.

3.2.3.6 Função seleciona pai (torneio)

A função seleciona_pai (na versão por torneio), ilustrada na Figura 13 recebe o tamanho_torneio, que é o tamanho do torneio. Deste modo, é escolhida uma amostra da população de forma aleatória, logo o participante com a nota de avaliação mais alta dentro do torneio é determinado pela função “max”, e esta nota é o critério de comparação de cada indivíduo. Uma vez encontrado o melhor indivíduo, a função retorna o índice deste indivíduo na população inicial.

Figura 13: Pseudocódigo função seleção dos indivíduos por torneio.

```

Função seleciona_pai(tamanho_torneio):
  # Seleciona aleatoriamente um subconjunto da população para o torneio
  participantes = AmostraAleatória(populacao, tamanho_torneio)

  # Encontra o participante com a nota de avaliação mais alta no torneio
  melhor_participante = Máximo(participantes, função de comparação: lambda x: x.nota_avaliacao)

  # Retorna o índice do melhor participante na população original
  Retorne ÍndiceDe(populacao, melhor_participante)

```

Fonte: autor

3.2.3.7 Função visualiza

A função visualização imprime o melhor resultado encontrado na geração mostrando o melhor resultado, o valor, o espaço ocupado, o cromossomo e os itens selecionados.

3.2.3.8 Função resolver

É nesta função que acontece grande parte do funcionamento do Algoritmo Genético. A função resolver recebe a taxa de mutação, o número de gerações, os espaços, valores e o limite de espaço. Primeiramente, a população é inicializada, em seguida ocorre a primeira avaliação dos indivíduos e a ordenação deles do maior ao menor. Então, começa a execução das gerações, onde da primeira geração até a geração limite, deve ser criada a respectiva população. Para cada geração, deve-se realizar a seleção dos pais e *crossover*, onde a taxa de seleção é 100%. Em seguida, são realizadas: o *crossover* por recombinação de 1 ponto e a mutação simples, com uma taxa de ocorrência predefinida. Logo após, são adicionados os filhos na nova geração, como ilustrado na Figura 14.

Figura 14: Representação pseudocódigo da função resolver: seleção de indivíduos, crossover, mutação e inserção dos filhos na nova geração.

```

Para cada indivíduo gerado no intervalo de 0 a tamanho da população:
  pai1 = seleciona_pai(soma_avaliacao)
  pai2 = seleciona_pai(soma_avaliacao)

  filhos = crossover(populacao[pai1], populacao[pai2])

  Adicione filhos[0] após aplicar mutação com taxa_mutacao à nova população
  Adicione filhos[1] após aplicar mutação com taxa_mutacao à nova população

  Atribua nova população à população

```

Fonte: autor.

A nova geração é avaliada e imprime a melhor solução de todas encontradas.

3.2.4 Função main

Na função main (Figura 15) é feita a entrada de dados – itens, com nome, espaço (em metros cúbicos) e valor – e é criada a lista de itens. Após esta ação, é chamada a função AlgoritmoGenetico, que recebe a taxa de mutação, o número de gerações, os espaços, os valores, o limite, retornando o melhor resultado. O último passo imprime a lista de produtos que configura a melhor solução.

Figura 15: Representação pseudocódigo da função main.

```

Cria listas vazias de espaços, valores e nome

Para cada produto na lista de produtos:
  Adiciona o espaço do produto à lista espaços
  Adiciona o valor do produto à lista valores
  Adiciona a nome do produto à lista nome

Define o limite da mochila
Define o tamanho da população
Define a taxa de mutação
Define a taxa de elitismo
Define o número de gerações

Cria uma instância do Algoritmo Genético chamada ag com o tamanho da população

Executa o algoritmo genético para resolver o problema, passando (taxa de mutação, número de gerações, listas de
espaços e valores, limite, taxa elitismo)

Armazena o resultado em 'resultado'

Para cada posição i no intervalo de 0 a tamanho da lista de produtos:
  Se o resultado na posição i for igual a '1':
    Imprimi "Nome: [nome do produto] R$[valor do produto]"

```

Fonte: autor.

É mostrado na Figura 16 um exemplo de possível resultado obtido:

Figura 16: Exemplo de resultado apresentado pelo algoritmo.

```

Melhor Solucao -> G:8 Valor:20446.57 Espaco:2.3982899 Cromossomo:
['1', '1', '0', '1', '1', '1', '0', '1', '1', '1', '0', '0', '1',
'1']
Nome:Geladeira Dako R$999.9
Nome:Iphone 6 R$2911.12
Nome:TV 50' R$3999.9
Nome:TV 42' R$2999.0
Nome:Notebook Dell R$2499.9
Nome:Microondas Electrolux R$308.66
Nome:Microondas LG R$429.9
Nome:Microondas Panasonic R$299.29
Nome:Notebook Lenovo R$1999.9
Nome:Notebook Asus R$3999.0

```

Fonte: autor.

3.3 Considerações finais do capítulo

Neste capítulo foram apresentadas as implementações dos métodos utilizados neste trabalho. O Capítulo 4 mostrará os experimentos realizados com o AG desenvolvido

neste trabalho, bem como os resultados obtidos a partir de uma sequência de testes e seus respectivos ajustes de parâmetros.

4 RESULTADOS

Os resultados gerados durante o desenvolvimento deste Trabalho de Conclusão de Curso serão apresentados neste capítulo. Estes resultados foram obtidos a partir da execução de um conjunto de 10 testes para cada configuração do AG a ser analisada, calculando a média dos melhores *fitness* de cada execução e o desvio padrão de cada sequência de teste. Também é mostrada a média do tempo de execução para cada teste. O sistema computacional utilizado para realizar os testes possui 16 GB de RAM, um processador de 2.60GHz, e foi executado no sistema operacional Windows 11.

4.1 Primeira etapa de experimentos

Nesta primeira rodada de testes foram experimentados os parâmetros do algoritmo, com os devidos ajustes, a fim de encontrar a configuração em que o resultado – melhor valor – mais se aproxima do valor ideal.

Para tal, manteve-se em todas as execuções os seguintes parâmetros: taxa de cruzamento 100% e recombinação de 1-ponto, mutação simples e método da seleção por roleta. Foram realizados testes alterando: a taxa de mutação, elitismo, tamanho da população e o número de gerações.

4.1.1 Sequência de Testes 1

Todos os testes descritos nesta subseção tiveram como padrão: taxa de elitismo 10%, população fixa de 20 indivíduos e 100 gerações. A taxa de mutação foi modificada a fim de analisar o melhor valor para a mesma.

Figura 17: Sequência de testes 1.

	Teste 0	Teste 1	Teste 2	Teste 3
Mutação	0,01	0,03	0,05	0,1

	Teste 0			Teste 1		
	Média	D. Padrão	M. Resul.	Média	D. Padrão	M. Resul.
Melhor Valor:	23292,51	1271,841	24694,26	23888,02	731,1977	24793,56
Tempo de Execução:	0,0193	0,002946	0,019	0,0205	0,004353	0,017

Teste 2			Teste 3		
Média	D. Padrão	M. Resul.	Média	D. Padrão	M. Resul.
22270,43	1335,303	24694,26	22527,57	1331,889	24343,37
0,0247	0,017449	0,024	0,0247	0,016479	0,017

Fonte: autor.

Comparando os resultados obtidos, mostrados pela Figura 17, com tempo de execução, Média, desvio padrão, e o valor ideal, chegou-se à conclusão de que os melhores resultados foram encontrados utilizando a variável proposta no Teste 1 – taxa de mutação 3% – onde a média foi de R\$23.888,02, desvio padrão de 731,1977 com o melhor resultado encontrado de R\$24.793,56. Além de ter a melhor média e o melhor valor mais próximo do esperado (R\$27.042,35), foi a alternativa com o menor desvio padrão, ou seja, os dados estavam mais próximos ao serem comparados a média.

4.1.2 Sequência de Testes 2

Todos os testes descritos nesta subseção tiveram como padrão: taxa de mutação 3%, obtida nos testes anteriores, população fixa de 20 indivíduos e 100 gerações. A taxa de elitismo foi modificada a fim de analisar o melhor valor para a mesma.

Figura 18: Resultados na sequência de testes 2.

	Teste 0	Teste 1	Teste 2	Teste 3
Elitismo	10%	5%	15%	20%

	Teste 0			Teste 1		
	Média	D.Padrão	M. Resul.	Média	D.Padrão	M. Resul.
Melhor Valor:	23337,81	968,7544	24642,66	23746,93	1063,01	24793,56
Tempo de Execução:	0,0153	0,015699	0,018	0,058	0,022236	0,053

Teste 2			Teste 3		
Média	D.Padrão	M. Resul.	Média	D.Padrão	M. Resul.
24086,88	565,8029	24694,26	22122,01	1818,619	24484,9
0,03	0,03716	0,027	0,0229	0,025783	0,096

Fonte: autor.

Comparando os resultados obtidos, mostrados na Figura 18, tempo de execução, média, desvio padrão, e o valor ideal, chegou-se à conclusão de que os melhores resultados foram encontrados utilizando a variável proposta no Teste 2, taxa de elitismo 15%, onde a média foi de R\$24.086,88, desvio padrão de 565,8029 com o melhor resultado encontrado de R\$24694,26. Além de ter a melhor média e o valor mais próximo do esperado (R\$27.042,35), foi a alternativa com o menor desvio padrão. Note que este teste foi considerado melhor que o anterior, pois a média foi maior e o desvio padrão menor, ainda que o melhor resultado encontrado não tenha sido melhor que o anterior.

4.1.3 Sequência de testes 3

Todos os testes descritos nesta subseção tiveram como padrão: mutação 3%, taxa de elitismo 15%, obtidos dos testes anteriores, e 100 gerações. O tamanho da população será modificado a fim de analisar o melhor valor para a mesmo.

Figura 19: Resultados na sequência de testes 3.

	Teste 0	Teste 1	Teste 2	Teste 3	Teste 4
População	20	40	60	80	100

	Teste 0			Teste 1		
	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
Melhor Valor:	24086,88	565,8029	24694,26	23964,14	675,1362	24642,66
Tempo de Execução:	0,03	0,03716	0,027	0,0157	0,001418	0,018

	Teste 2			Teste 3			Teste 4		
	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M. Resul.
	24563	362,5369	24993,55	24540,99	299,4882	24793,56	24766,01	273,1298	24993,55
	0,0202	0,001989	0,019	0,0301	0,00743	0,032	0,0202	0,001989	0,017

Fonte: autor.

Comparando os resultados obtidos, mostrados na Figura 19, tempo de execução, média, desvio padrão, e o valor ideal, chegou-se à conclusão de que os melhores resultados foram encontrados utilizando a variável proposta no teste 4, tamanho da população igual a 100 indivíduos, onde a média foi de R\$24.766,01, desvio padrão de 273,1298 com o melhor resultado encontrado de R\$24.993,55. Além de ter a melhor média e o melhor valor mais próximo do esperado (R\$27.042,35), foi a alternativa com o menor desvio padrão. Note que este teste foi melhor que os anteriores considerando tanto a média, quanto o desvio padrão e o melhor valor encontrado.

4.1.4 Sequência de testes 4

Todos os testes descritos nesta subseção tiveram como padrão: taxa de mutação 3%, taxa de elitismo 15% e população fixa de 100, definidos empiricamente nos testes anteriores. A quantidade de gerações foi modificada a fim de analisar o melhor valor para a mesma.

Figura 20: Resultados na sequência de testes 4

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
Geracões	50	100	150	200	250

Teste 6	Teste 7	Teste 8	Teste 9	Teste 10
300	350	400	450	500

Teste 11	Teste 12	Teste 13	Teste 14	Teste 15
550	600	650	700	750

Teste 16	Teste 17	Teste 18	Teste 19	Teste 20
800	850	900	950	1000

	Teste 1			Teste 2			Teste 3			Teste 4		
	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
Melhor Valor:	24697,02	89,65342	24793,56	24741,79	221,6453	24993,55	24723,91	221,017	24993,55	24609,01	268,0479	24993,55
Tempo de Execução:	0,015	0,002	0,014	0,0222	0,001317	0,021	0,0247	0,00116	0,024	0,0296	0,004222	0,03

Teste 5			Teste 6			Teste 7			Teste 8		
Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
24665,19	328,0953	24993,55	24837,67	240,0143	24993,55	24665,76	318,2632	24993,55	24767,74	153,1481	24993,55
0,033	0,002906	0,029	0,035	0,003944	0,025	0,04	0,002404	0,038	0,0459	0,003985	0,043

Teste 9			Teste 10			Teste 11			Teste 12		
Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
24710,92	211,3213	24993,55	24664,89	207,1352	24993,55	24755,37	202,3166	24993,55	24654,18	251,4547	24993,55
0,0465	0,002224	0,045	0,0539	0,008386	0,05	0,0575	0,003504	0,054	0,0585	0,003779	0,06

Teste 13			Teste 14			Teste 15			Teste 16		
Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
24699,73	205,8855	24993,55	24547,91	397,3188	24993,55	24671,83	201,0231	24993,55	24672,95	199,7178	24993,55
0,0653	0,004218	0,069	0,0663	0,002497	0,063	0,0716	0,0043	0,08	0,1492	0,221782	0,086

Teste 17			Teste 18			Teste 19			Teste 20		
Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.	Média	D.Padrão	M.Resul.
24642,45	196,929	24993,55	24727,88	224,8617	24993,55	24728,18	140,2274	24993,55	24762,72	175,4752	24993,55
0,0837	0,004057	0,089	0,083	0,004738	0,082	0,09711	0,00787	0,087	0,099	0,012147	0,092

Fonte: autor.

A necessidade de tantas tentativas ocorreu devido à busca por uma análise mais refinada dos resultados, uma vez que o melhor resultado foi obtido desde o Teste 2, igual ao teste anterior. De fato, estava se buscando a melhoria da média e do desvio padrão.

Dos resultados encontrados, mostrado pela Figura 20, os mais relevantes foram os Testes 6, 8 e 16, onde o valor das médias mais se aproximam do valor esperado (R\$27.042,35) e foram melhores que o valor encontrado na sequência de testes anterior, o critério de desempate foi o desvio padrão, sendo assim o melhor teste foi o 8, onde a média encontrada foi de R\$24.767,7, desvio padrão de 153,148 e o melhor valor foi de R\$24.993,55.

4.1.5 Discutindo resultados

Analisando os testes realizados e as soluções obtidas, de acordo com os melhores resultados, as suas médias e desvio padrão a melhor configuração encontrada foi taxa de mutação 3%, taxa de elitismo 15%, tamanho da população de 100 indivíduos e o número de gerações de 400. Com estes experimentos, observou-se a importância da análise de

resultado, e, principalmente, do ajuste adequado destes parâmetros para que se alcançasse melhores resultados.

4.2 Segunda etapa de experimentos

Esta etapa buscou melhorar ou encontrar o valor ideal com novos itens adicionados, por meio de experimentos com os métodos de seleção. Brasil (2012) discorre sobre a tendência do método da seleção por torneio obter melhores resultados da resolução de problemas. Santana (2018) também discute o bom desempenho da seleção por torneio e ainda sugere uma nova variação deste método. Deste modo, os testes a seguir tem os mesmos parâmetros que o anterior, porém o método de seleção será modificado do método da roleta para o método de seleção por torneio, buscando comprovar a literatura, variando a capacidade do caminhão e o tamanho da lista de produtos.

4.2.1 Sequência de testes 1

Foram adicionados 14 itens, observados na Tabela 2, aos 14 anteriores, totalizando 28 itens. O novo valor ideal R\$31.992,04, que é a soma do valor dos 14 objetos antigos com o valor de 14 objetos novos, em que o volume esperado é aproximadamente 5,5 metros cúbicos.

Tabela 2: 14 itens adicionados.

Itens	Volume	Valor
Echo Show 8	0,0026989	949,05
A História da Terra-média - Box 2	0,002893	169,99
Oster, PrimaLatte II - Cafeteira Espresso	0,030949	1499,9
UMIDIFICADOR WAP AIR FLOW	0,013445	199,99
AQUECEDOR WAP AIR HEAT	0,006393	138,9
TORRE WAP AIR SILENCE, 220	0,0927	479,9
Ventilador rajada arno	0,39984	284,29
Risqué Top Coat	0,000071	10,16
Truss Hidratação Intensa	0,000008	379
Cd - DUA LIPA - FUTURE NOSTALGIA	0,000201	52,9
Mop Plano 3 EM 1	0,057443	100,73
Kit Casa Conectada Positivo	0,005	499,99
Mouse com fio USB Logitech	0,001315	44,9
Difusor De Aromas	0,004677	139,9

Fonte: autor.

Figura 21: Resultados da sequência de testes 1, utilizando o método da roleta.

	28 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	27898,41	580,9002	28728,68	30129,28	503,7634	30898,43	30855,48	603,3711	31792,05
Tempo de Execução:	0,0612	0,014367	0,06	0,0522	0,003011	0,05	0,0554	0,004971	0,062

Fonte: autor.

Com os 28 itens, o novo valor esperado passa a ser R\$31.992,04, e como observado na Figura 21 o teste com a média mais próxima do valor esperado foi o terceiro, sendo a configuração com o caminhão de 9 metros cúbicos a melhor para 28 itens em cada carregamento.

Figura 22: Resultados da sequência de testes 2, utilizando o método por torneio.

	28 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	28620,29	475,5457	29297,9	31506,47	337,8714	31797,08	31307,28	500,2814	31741,25
Tempo de Execução:	0,0779	0,006082	0,073	0,0816	0,010167	0,074	0,0817	0,006567	0,077

Fonte: autor.

Observe que no método da roleta, em que o caminhão de 6 metros cúbicos, apesar de apresentar um desvio padrão menor, não foi a melhor configuração. Foi a configuração com 9 que apresentou média melhor, a qual chegou mais próximo do valor ideal mais vezes (8 vezes das 10) ao comparar com a com 6 metros cúbicos (6 vezes de 10). Com a seleção por torneio, é possível verificar que os testes alcançaram resultados melhores (tanto com desvio padrão menor quanto média maior e maiores valores) comparado ao método da roleta. Note que a configuração com 6 metros cúbicos obteve as melhores soluções, conforme Figura 22.

4.2.2 Sequência de testes 2

Foram adicionados 28 itens, observados na Tabela 3, aos 28 anteriores, totalizando 56 itens. O novo valor ideal é de R\$ 141.043,5, que é a soma do valor dos 28 objetos antigos com o valor de 28 objetos novos, onde o volume aproximado é de 10,4 metros cúbicos.

Tabela 3: 28 itens adicionados.

Itens	Volume	Valor
Echo Studio	0,000748	1709,05
Echo Show 10	0,001097	1899,05
Kindle 11ª Geração	0,000137	474,05

Multiplicador 5 Tomadas	0,000768	111,66
Boneco Dungeons & Dragons	0,002537	199,99
LEGO® Ideas Farol Motorizado	0,025422	2601,49
LEGO® Indiana Jones™ O Templo do Ídolo Dourado	0,01914	1399,99
Mochila Casual	0,002821	61,99
Midea - Frigobar, 45L	0,104501	788,5
Fritadeira Airfryer 4L	0,039336	599,9
Arno Cafeteira Espresso Nescafé	0,004448	929,99
Forno Elétrico Britânia	0,066041	599,99
Micro-ondas Electrolux	0,047059	616,55
Lavadora de Roupas Consul	0,35148	2534,7
Lavadora de Roupas Brastemp	0,561	3979
Geladeira Brastemp Frost Free	0,825185	3710
Cooktop 4 Bocas Mondial	0,0414	489,99
Refrigerador Brastemp Frost Free - French Door	1,237213	8503,32
Fogão 4 bocas Dako	0,309513	1829
Lava e Seca Samsung	0,447225	6599
Freezer Horizontal Consul	0,772367	3979
Depurador Slim SUGGAR	0,02448	379
Apple TV	0,000336	1999
MacBook Air	0,001089	7999
iPhone 14 Pro	0,000083	9499
MacBook Pro	0,001483	22999
iPad Pro	0,000386	22475,75
JOGO FACAS TRAMONTINA	0,00418	84,5

Fonte: autor.

Figura 23: Resultados da sequência de testes 1, utilizando o método da roleta.

	56 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	92240,612	10452,27621	105196,75	116215,358	3423,490269	121234,55	126455,91	3939,3377	131980,5
Tempo de Execução:	0,0848	0,00820298	0,084	0,082	0,007958224	0,08	0,0852	0,0050288	0,085

Fonte: autor.

Com os 56 itens, o novo valor ideal passa a ser R\$141.043,50, e como observado na Figura 23, a configuração de 6 metros cúbicos apresentou um desvio padrão menor, enquanto que a configuração de 9 metros cúbicos apresentou uma média maior e chegou perto do valor ideal (6 vezes de 10) mais vezes que a de 6 metros cúbicos (4 vezes de 10), sendo assim considerada a melhor configuração.

Figura 24: Resultados da sequência de testes 2, utilizando o método por torneio.

	56 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	100985,11	5145,173	107865,6	123339	1769,725	125858,3	131514,9	2150,017	134736,3
Tempo de Execução:	0,124	0,014621	0,156	0,119	0,005735	0,115	0,1226	0,009617	0,124

Fonte: autor.

Ao compararmos os resultados encontrados com os resultados do método da roleta (Figura 24), observou-se uma semelhança nas conclusões. No entanto, a configuração com 6 metros cúbicos apresentou um desvio padrão menor, enquanto que a configuração com 9 metros cúbicos apresentou uma média maior, valores maiores e aproximou-se do valor ideal mais vezes. Portanto, a configuração com o caminhão de 9 metros cúbicos do método por torneio apresentou melhores soluções.

4.2.3 Sequência de testes 3

Foram adicionados 56 itens, observados na Tabela 4, aos 56 anteriores, totalizando 112 itens. O novo valor ideal R\$158.390,95, que é a soma do valor dos 56 objetos antigos com o valor de 56 objetos novos, ilustrados na Tabela 4, onde o volume aproximado é de 12,7 metros cúbicos.

Tabela 4: 56 itens adicionados.

Itens	Volume	Valor
God Of War Ragnarok Ps4 Sony Físico Standard Edition	0,00077	160
Ventilador Mondial	0,042017	130
Liquidificador Turbo Power mundial	0,01722	99
Lavadora De Alta Pressão WAP	0,04394	448
Furadeira parafusadeira WAP	0,002819	172
Fonte MSI MAG	0,001806	588,22
Monitor Gamer LG	0,037278	459,99
Utensílios De Cozinha De Madeira	0,00385	384
Estante Aparador	0,608	108,85
3 Prateleiras	0,012	99,99
Panela De Pressao 4,5 Polida	0,012	78,72
Varal Dobravel	0,00364	50,9
Modolo Automotivo Soundigital	0,000776	399,9
Churrasqueira Elétrica Giratória De Inox	0,152898	2077,67
Balcão Caixa Recepção Wallet	0,264	499,99
Fogão Cooktop Chamalar	0,046512	614,92
Multiprocessador Philco	0,03159	568,9
Cortador Elétrico Mondial Spiralize	0,01036	308,9

passadeira de roupas	0,04446	215
Mala de Viagem Rose Em ABS - P	0,06591	490
Fritadeira Industrial Elétrica Cuba Profissional	0,025461	270,9
Box O Fantasma da Ópera	0,001137	79,9
Bauleto 28 Litros Mod, Smart Box Pro Tork	0,048	105,99
Taça Prime Scotland Degustação Whisky	0,00098	120,9
Copo Trainer 220 ml, MAM, Rosa	0,000936	64,9
Glencairn Copo de uísque, conjunto de 15 cm pacote comercial	0,007704	103,9
Lâmpada Inteligente 15W Smart	0,00101	57,49
Câmera de Esgoto Com Tela LED	0,0168	1503
A História da Terra-média - Box 2	0,003005	289,9
MAM, Escova de Dentes Training Brush, Azul	0,000073	27,46
Mini desembaraçador de escova úmida exclusiva da Amazon – Aqua	0,000713	30,73
Umificador Difusor de Ar BANpinSH	0,001604	113,9
Faqueiro Amazonas com 48 Peças em Aço Inox - Tramontina	0,002728	469,9
Controle Remoto Universal Compatível Para todas Smart Tv's	0,000313	19,9
Balcão Cozinha Armário Gabinete para Pia	0,6179	434,9
Nebzmart, , Portatil	0,001954	398,59
Pruie Mini carregador de bateria portátil	0,000019	55,99
Shampoo Desamarelador Truss Blond 300ml	0,000568	89,9
Mop Flat Chenile com cabo Telescópico	0,00712	34,99
Magic Mixies - Caldeirão Mágico, Rosa, Candide	0,020384	732,97
Barbeador série Flex	0,001861	1089,79
Barbeador elétrico seco Philips	0,001888	450
Candide MAGIC MIXIES - MIXLINGS TWIN PACK, Modelo: 2453, Cor: Multicolor	0,001803	169,89
Sérum Facial Vitamina C 10, Tracta	0,000132	69,99
Jogo de Facas Tramontina Colorcut	0,004292	169,9
Multiplicador 5 Tomadas Bivolt	0,000768	110,57
BLACK+DECKER Ferro de Passar a Vapor	0,004461	120,05
Headset Gamer Hyperx Cloud II	0,006218	590
NATURA ESSENCIAL UNICO DEO PARFUM FEMININO 90ML	0,000815	156,81
PHEBO - Sabonete Líquido 320ml	0,000581	19,9
Jogo Rummikub Júnior	0,005944	69,99
Lixeira para Banheiro 3L	0,007225	49,9
Caixa Organizadora Empilhável Bambu	0,003291	89,9
Arandela Carvalho	0,014896	129,9
Gaveteiro Madeira Branco	0,1364	999,9
Vela Aromatizada Jasmim Silvestre Branco	0,000475	99,9

Fonte: autor.

Figura 25: Resultados da sequência de testes 4, utilizando o método da roleta.

	112 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	87871,572	10345,77939	98451,52	122748,805	8440,930765	134823,24	129778,74	4387,352	138399,89
Tempo de Execução:	0,1625	0,018149074	0,153	0,1395	0,010522463	0,16	0,1496	0,0162358	0,136

Fonte: autor.

Com os 112 itens, o novo valor ideal passa a ser R\$158.391,00, e como observado na Figura 25 o teste com a média mais próxima do valor ideal e com o menor desvio padrão foi o 3, sendo a configuração com o caminhão de 9 metros cúbicos e o melhor cenário. Entre os 10 testes realizados foi observado que em média foi possível carregar 63 itens.

Figura 26: Resultados da sequência de testes 3, utilizando o método por torneio.

	112 Itens								
	3m ³			6m ³			9m ³		
	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.	Média	D. Padrão	M.Resul.
Valor:	91928,23	8542,169	101381,7	124506,4	4666,163	129308,8	137263,5	1917,144	140106,4
Tempo de Execução:	0,1988	0,007843	0,197	0,2022	0,00992	0,201	0,2012	0,007786	0,194

Fonte: autor.

A partir dos resultados gerados com o torneio (Figura 26), observou-se a mesma conclusão em que a configuração com o caminhão de 9 metros cúbicos foi o melhor cenário, tendo o menor valor padrão, a maior média e seus resultados mais próximos do valor ideal, ao serem comparados com as configurações com caminhões de 3 e 6 metros cúbicos. Além disso, o método da roleta obteve desvio padrão de 4.387,352 e valor médio de R\$129.778,74 e os resultados por torneio foram de desvio padrão de 1.917,144 e valor médio de R\$137.263,50, evidenciando o melhor desempenho da seleção por torneio neste experimento.

4.3.4 Discutindo resultados

Na sequência de testes 1 foi encontrado o melhor resultado quando a configuração com o caminhão de 6 metros cúbicos apresentou o menor desvio padrão, a maior média e onde foram encontrados os valores mais próximos do valor ideal. Neste experimento o método do torneio

Na sequência de testes 2 foi verificado que a configuração de 6 metros cúbicos apresentou um desvio padrão menor, mas a configuração de 9 metros cúbicos apresentou uma média maior e aproximou-se do valor ideal mais vezes que a de 6 metros cúbicos.

Vale ressaltar que os resultados encontrados por torneio foram melhores que utilizando a roleta neste experimento.

Na sequência de testes 3 alcançou-se o melhor resultado com a configuração com o caminhão de 9 metros cúbicos, onde se encontra o menor desvio padrão, a maior média e os valores mais próximos do ideal, confirmando mais uma vez que o método do torneio obteve melhor resultado que a roleta.

Com estes resultados foi possível comprovar a literatura que indica que o método de seleção por torneio tem um desempenho melhor ao ser comparado com o da roleta.

4.4 Considerações finais do capítulo

Após a análise de todos os resultados na busca do melhor cenário para a resolução do problema apresentado, pode-se concluir que a melhor configuração dos parâmetros para otimizar o algoritmo genético para este problema foram: taxa de mutação 3%, elitismo 15%, tamanho da população de 100 indivíduos e o número de gerações de 400. Testes mais avançados, aumentando o tamanho da instância, mostraram que usando o método da roleta alcançou-se o valor ideal com 14 itens e um caminhão de 6 metros cúbicos, e nos outros cenários propostos com 28, 56 e 112 itens, foi possível alcançar resultados numericamente mais próximos do esperado no caminhão de 9 metros cúbicos, apesar do valor ideal não ter sido obtido. Ao comparar os resultados alcançados pelos testes usando o método da roleta e os usando o método por torneio, pode-se verificar que o método por torneio mostrou um melhor desempenho em todos os casos, com 28, 56 e 112 itens, confirmando que o método por torneio garante maior eficiência e robustez no AG do que o método da roleta, uma vez que ele prioriza os melhores de uma amostra da população para cada torneio aplicado.

O Capítulo 5 trará as considerações finais deste trabalho, assim como propostas de trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

Este Trabalho de Conclusão de Curso teve como objetivo principal implementar um Algoritmo Genético aplicado a uma variante do Problema da Mochila que é descrita como um caminhão com uma limitação volumétrica predefinida e uma dada lista de itens, com seus respectivos preços e volumes, a ser carregada, a fim de maximizar o valor transportado. Um objetivo secundário foi comparar o desempenho deste AG utilizando o método de seleção da roleta e o método da seleção por torneio.

Neste sentido, o algoritmo implementado com o método da roleta alcançou o valor ideal com os primeiros 14 itens, e com um caminhão de 6 metros cúbicos, nos demais cenários propostos, com 28, 56 e 112 itens, apesar de não alcançar o resultado esperado, gerou soluções próximas ao desejado no caminhão de 9 metros cúbicos. Com o algoritmo com o método de seleção por torneio, os cenários propostos com 28, 56 e 112 itens apresentaram melhores resultados ao serem comparados ao método da roleta, confirmando a literatura que indica que o método por torneio obtém melhores soluções que o método da roleta, principalmente quando o tamanho da instância aumenta.

Como proposta para trabalhos futuros vislumbra-se refinar os resultados obtidos, por meio de: (i) melhoria da função de avaliação de aptidão, utilizando ponto flutuante para penalizar a solução que excede o limite (e não de modo discreto, como foi implementado neste trabalho); (ii) desenvolvimento de diferentes formas de inicialização da população, e não somente a aleatória, como foi feito neste AG; (iii) realização de novos experimentos com um conjunto maior de instâncias de tamanhos diferentes. A partir destas adaptações do AG, espera-se obter soluções melhores, especialmente quando se tratar de instâncias de dimensões maiores.

REFERÊNCIAS

BARCELLOS, João Carlos Holland de. **Algoritmos genéticos adaptativos: um estudo comparativo**. 2000. Dissertação de Doutorado Universidade de São Paulo, São Paulo – SP. Disponível em: <https://doi.org/10.11606/D.3.2000.tde-05092001-141334>. Acesso em: 04 de dezembro de 2023.

BARRETO, Débora Barbosa. **Piratas VS Marinha: um jogo de perseguição utilizando teoria dos grafos e pygame**. 2021. Trabalho de conclusão de curso PUC Goiás, Goiânia – GO. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3538>. Acesso em: 04 de dezembro de 2023.

BAYRAKTAR, Tuğrul; ERSÖZ, Filiz; KUBAT, Cemalettin. Effects of memory and genetic operators on Artificial Bee Colony algorithm for Single Container Loading problem. **Applied soft computing**, v. 108, n. 107462, p. 107462, 2021. Disponível em: <https://doi.org/10.1016/j.asoc.2021.107462>. Acesso em: 04 de dezembro de 2023.

BORGUESAN, Bruno. **GARTS: um algoritmo genético baseado no método de seleção por torneio restrito adaptativo para o problema de predição de estruturas 3D de proteínas**. 2016. Dissertação de Mestrado Universidade Federal do Rio Grande do Sul. Porto Alegre -RS. Disponível em: <http://hdl.handle.net/10183/197308>. Acesso em: 04 de dezembro de 2023.

BRAGA, Edgar Augusto Silva. **Modelagem e otimização do problema do caixeiro viajante com restrições de tempo, distância e confiabilidade via algoritmos genéticos**. 2007. Trabalho de conclusão de curso. Universidade Federal de Pernambuco, Recife – PE. Disponível em: <https://repositorio.ufpe.br/handle/123456789/5672>. Acesso em: 04 de dezembro de 2023.

BRASIL, Christiane Regina Soares. **Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. 2012. Tese de Doutorado Universidade de São Paulo. São Carlos -SP. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-20072012-163056/>. Acesso em: 04 dez. 2023.

CORTES, Omar Andres Carmona; DA SILVA, Josenildo Costa. Unconstrained numerical optimization using real-coded genetic algorithms: a study case using benchmark functions in R from Scratch. **Revista Brasileira de Computação Aplicada**, v. 11, n. 3, p. 1-11, 2019. Disponível em: <https://doi.org/10.5335/rbca.v11i3.9047>. Acesso em: 04 de dezembro de 2023.

DANTZIG, George B.; FULKERSON, Delbert R.; JOHNSON, Selmer M.. Solution of a large-scale traveling-salesman problem. **Journal of the operations research society of America**, v. 2, n. 4, p. 393 410, 1954. Disponível em: https://www.hexaly.com/wp-content/uploads/2022/04/TSP_MIP.pdf?redirect. Acesso em: 04 de dezembro de 2023.

DARWIN, Charles. Sobre a Origem Das Especies: On the Origin of Species (Portuguese Edition). Tradução: Andrew Ryan. North Charleston, SC, USA: **CreateSpace Independent Publishing Platform**, 2015.

DE CARVALHO, Vinicius Renan; POZO, Aurora Trinidad Ramirez. Um estudo sobre otimização por partículas aplicado ao problema de roteamento de veículos com demandas estocásticas. Proceedings of the **Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)**, São Carlos, Brazil, p. 19-23, 2014. Disponível em: <10.13140/2.1.1673.4727>. Acesso em: 04 de dezembro de 2023

GABRIEL, Paulo Henrique Ribeiro; DELBEM, Alexandre Cláudio Botazzo, 2008, Fundamentos de Algoritmos Evolutivos. **Notas Didáticas do ICMC-USP**, ISSN 0103-2585, No. 75, 35 pg. Disponível em: https://repositorio.usp.br/directbitstream/7472618b-87b3-4077-a1ca-eb5f40a0542c/nd_75.pdf. Acesso em: 04 dez. 2023.

GOLDBERG, David E. Genetic algorithms in search, optimization, and machine learning. **Choice** (Chicago, Ill.), v. 27, n. 02, p. 27- 0936-27-0936, 1989. Disponível em: http://www2.fiit.stuba.sk/~kvasnicka/Free%20books/Goldberg_Genetic_Algorithms_in_Search.pdf. Acesso em: 04 de dezembro de 2023.

GOUVEIA, Luis Borges et al. Algoritmos genéticos: aplicando a teoria a um estudo de caso. **Brazilian Journal of Development**, v. 7, n. 3, p. 21053-21077, 2021. Disponível em: <https://doi.org/10.34117/bjdv7n3-016>. Acesso em: 04 de dezembro de 2023.

GRIGOLI, Flávio Adriano Giaretti. **Grafos e o Problema da Distribuição: Uma introdução com Análise Combinatória**. 2015. Trabalho de conclusão de curso em Licenciatura em Matemática – Instituto Federal de Educação, Birigui – SP. Disponível em: <<https://vdocuments.com.br/licenciatura-em-matematica-flavio-adriano-giaretti-a-teoria-dos-grafos-sera.html?page=1>>. Acesso em: 04 dezembro de 2023.

HASSANAT, Ahmad et al. **Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach**. **Information**, v. 10, n. 12, p. 390, 2019. Disponível em: <https://doi.org/10.3390/info10120390>. Acesso em: 04 de dezembro de 2023.

HOLLAND, John H. Genetic algorithms. **Scientific american**, v. 267, n. 1, p. 66-73, 1992. Disponível em: <http://dx.doi.org/10.1038/scientificamerican0792-66>. Acesso em: 04 de dezembro de 2023.

LACERDA, E. G. M; Carvalho, A. C. P. L. F. Introdução aos algoritmos genéticos. In: **Galvão, C. O.; Valença, M. J. S. Sistemas inteligentes**. Porto Alegre: Editora da Universidade da UFRGS – ABRH, 1999. Disponível em: <<https://repositorio.usp.br/item/001079374>>. Acesso em: 04 de dezembro de 2023.

LIU, Xinlin; JING, Tian; HOU, Linyi. An FW–GA Hybrid Algorithm Combined with Clustering for UAV Forest Fire Reconnaissance Task Assignment. **Mathematics**, v. 11, n. 10, p. 2400, 2023. Disponível em: <https://doi.org/10.3390/math11102400>. Acesso em: 04 de dezembro de 2023.

LUNARTI, Lucas L.; VALADÃO, Victor L.; JUNIOR, Maury M.G. . Alocação Ótima de um Sistema de Medição Fasorial Sincronizada utilizando Algoritmo Evolucionário. **Simpósio Brasileiro de Sistemas Elétricos-SBSE**, v. 1, n. 1, 2020. Disponível em: <https://doi.org/10.48011/sbse.v1i1.2453>. Acesso em: 04 de dezembro de 2023.

MAZZA, Maria Cristina Medeiros et al. Etnobiologia e conservação do bovino pantaneiro. **Corumbá: Embrapa-CPAP**; Brasília, DF: Embrapa-SPI, 1994., 1994. Disponível em: <<http://livimagens.sct.embrapa.br/amostras/00060760.pdf>>. Acesso em: 04 de dezembro de 2023.

MUNHAO, Bárbara S. et al. Implementações GPGPU do Algoritmo de Otimização por Enxame de Partículas para o Problema da Mochila Multidimensional. In: **Anais Estendidos do XX Simpósio em Sistemas Computacionais de Alto Desempenho**. SBC, p. 96-107, 2019. Disponível em: https://doi.org/10.5753/wscad_estendido.2019.8703. Acesso em: 04 de dezembro de 2023.

NAND, Ravneil; SHARMA, Priynka. Iteration split with firefly algorithm and genetic algorithm to solve multidimensional knapsack problems. **2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)**. Anais...IEEE, 2019. Disponível em: <http://dx.doi.org/10.1109/CSDE48274.2019.9162422>. Acesso em: 04 de dezembro de 2023.

NOGUEIRA, Marcelo Luis Lobato; SAAVEDRA, Osvaldo Ronald. Estratégias evolutivas aplicadas à resolução de otimização multimodal. In: **Simpósio Brasileiro de Automação Inteligente**. 1999. Disponível em: https://www.researchgate.net/profile/Osvaldo-Saavedra-3/publication/260629462_Estrategias_Evolutivas_Aplicadas_a_Resolucao_de_Otimizacao_Multimodal/links/580f5a3608ae009606bb6b0a/Estrategias-Evolutivas-Aplicadas-a-Resolucao-de-Otimizacao-Multimodal.pdf. Acesso em: 06 de dezembro de 2023.

OZSOYDAN, Fehmi Burcin. Artificial search agents with cognitive intelligence for binary optimization problems. **Computers & industrial engineering**, v. 136, p. 18–30, 2019. Disponível em: <https://doi.org/10.1016/j.cie.2019.07.007>. Acesso em: 04 de dezembro de 2013.

OZSOYDAN, Fehmi Burcin; BAYKASOGLU, Adil. A swarm intelligence-based algorithm for the set-union knapsack problem. **Future generations computer systems: FGCS**, v. 93, p. 560–569, 2019. Disponível em:

<https://doi.org/10.1016/j.future.2018.08.002>. Acesso em: 04 de dezembro de 2023.

PACHECO, Marco Aurélio Cavalcanti. Algoritmos genéticos: princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada**, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro – RJ, 1999. Disponível em:

<https://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf>.

Acesso em: 04 de dezembro de 2023.

PEIXOTO, Alan Bessa Gomes. **O Problema de inspeções na rede de distribuição de energia elétrica: uma abordagem evolutiva**. 2015. 114 f. Dissertação de Mestrado em Logística e Pesquisa Operacional – Pró-Reitoria de Pesquisa e Pós-Graduação, Universidade Federal do Ceará, Fortaleza. Disponível em:

<<http://repositorio.ufc.br/handle/riufc/14674>>. Acesso em: 05 de dezembro de 2023.

ANTANA, Ivan; TOREZZAN, Cristiano. Um novo método para seleção de indivíduos em algoritmos genéticos inspirado em torneios de chaves. In: **Anais do Simpósio Brasileiro de Pesquisa Operacional**. 2018. Disponível em:

<<https://proceedings.science/sbpo/sbpo-2018/trabalhos/um-novo-metodo-para-selecao-de-individuos-em-algoritmos-geneticos-inspirado-em-t?lang=pt-br>>. Acesso em: 04 de

dezembro de 2023.

SOUZA, Marcone Jamilson Freitas, et al. Um Algoritmo Evolutivo Híbrido para o Problema de Programação de Horários em Escolas. **Anais do XXII Encontro Nacional de Engenharia de Produção**. Curitiba – PR: 2002. Disponível em:

<https://abepro.org.br/biblioteca/enegep2002_tr68_0896.pdf>. Acesso em: 04 de

dezembro de 2023.

TAYLOR, Charles E. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Complex adaptive systems. John H. Holland. **The quarterly review of biology**, v. 69, n. 1, p. 88–89, 1994. Disponível em: <<https://www.journals.uchicago.edu/doi/10.1086/418447>>.

Acesso em: 05 de dezembro de 2023.

WEINGART, H. Martin; NESS, David N. . Methods for solution of multi-dimensional 0/1 knapsack problem. **Operations research**, v. 14, p. 67. 1966. Disponível em: <https://doi.org/10.1287/opre.15.1.83>. Acesso em: 05 de dezembro de 2023.

YANG, Jaeyoung; KIM, Yong-Hyuk; YOON, Yoon. A memetic algorithm with a novel repair heuristic for the multiple-choice multidimensional knapsack problem. **Mathematics**, v. 10, n. 4, p. 602, 2022. Disponível em: <https://doi.org/10.3390/math10040602>. Acesso em: 05 de dezembro de 2023.