

Ciência da Computação - Graduação

Projeto 2: Algoritmo Genético para a solução do problema da Mochila Ilimitada

Jeferson Alves (175715)

Victor Andrietta Razoli Silva (187835)

1. Resumo

Uma empresa de transportes dispõe de um avião cargueiro para transportar alimentos de um país A para um país B. A empresa recebe N tipos de alimentos, cada um com uma quantidade q_i e um valor v_i (em USD\$) associados e deve compor uma carga de volume L (em m^3) de forma a maximizar o valor do carregamento. O volume máximo que o avião cargueiro é capaz de transportar é V (em m^3). Portanto o valor L da carga não pode ultrapassar V (mas pode ser menor, desperdiçando espaço).

Palavras-chave: evolutivo, variação, genético, ótima, taxa.

2. Método utilizado para encontrar os melhores parâmetros

Para avaliar a qualidade do resultado obtido, assim como para obter os melhores parâmetros possíveis para o Algoritmo Genético, comparamos a proximidade da solução do Algoritmo Genético com o resultado ótimo calculado por Programação Dinâmica.

Foram testadas 300 instâncias diferentes com parâmetros aleatórios; cada grupo de 10 instâncias testadas com a mesma porcentagem de mutação que variava de 0 a 3%. Todos os algoritmos utilizados para efetuar a análise do Algoritmo Genético no espaço amostral se encontram no arquivo **analysis.py**.

Os produtos utilizados para a análise do Algoritmo Genético foram gerados aleatoriamente e armazenados em um arquivo na aplicação local.

3. O Algoritmo Genético

3.1 Função de Fitness

A função de fitness adotada estabelece a pontuação do cromossomo pelo valor total dos produtos codificados nos alelos uma vez que o objetivo do problema da mochila é maximizar o valor da carga. Se o volume total dos itens codificados no cromossomo ultrapassa o volume V máximo o valor de avaliação passa a ser 1 (a pior avaliação possível).

3.2 Modelagem do Cromossomo

3.2.1 Cromossomo

Para o problema da mochila inteira ilimitado, como enunciado, o cromossomo foi modelado como uma lista de números inteiros não negativos. Cada loco gênico representa um produto e a respectiva quantidade a ser levada da seguinte maneira: se o loco gênico contiver valor 0 o item não será levado; se o loco gênico contiver valor diferente de 0 o item será levado e a quantidade a ser levada é dada pelo valor numérico inteiro gravado no loco.

3.2.2 Loco Gênico

Cada loco representa informações a respeito de um produto dentro da carga L que estamos interessados em construir. O i -ésimo loco representa o i -ésimo produto, como representado no **Diagrama 1**.

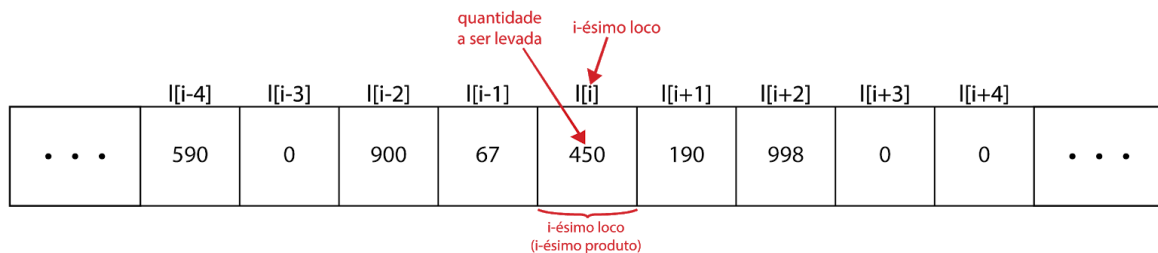


Diagrama 1 (autoral): informações de um loco dentro de um cromossomo.

3.3 Geração de prole

3.3.1 Escolha dos ancestrais

Método 1: Roleta Viciada

Escolhemos os pais na geração G que irão gerar filhos para a geração G+1 utilizando o método da roleta viciada. O método funciona da seguinte maneira: cada indivíduo recebe uma parcela da roleta proporcional à avaliação que recebeu da Função de Fitness, de forma que indivíduos bem avaliados (mais aptos/adaptados) possuem maior chance de passar seus genes adiante; isso não exclui os indivíduos com avaliação inferior, apenas fornece a eles chance menor de passar seus genes adiante.

O método da roleta viciada assegura que mesmo indivíduos pouco adaptados possuam a possibilidade de passar seus genes adiante, evitando assim o Elitismo, e possibilitando melhoras globais ao sistema. O **Gráfico 1** ilustra o processo de escolha de ancestrais para uma população de 8 indivíduos. O ancestral A_i possui a i -ésima melhor avaliação dada pela Função de Fitness.

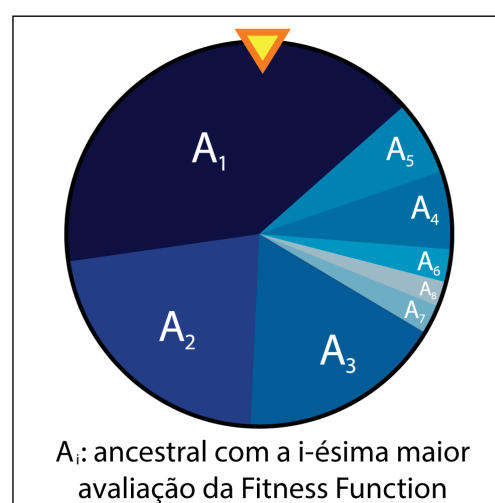


Gráfico 1 (autoral): método da roleta viciada para uma população de 8 indivíduos.

Método 2: Elitismo

O segundo método de seleção consiste na seleção elitista, escolhe-se o cromossomo pai entre os **X** melhores cromossomos (melhores avaliações dadas pela Função de Fitness), sendo **X** um parâmetro informado ao método de crossover.

Análise de métodos de Seleção de Ancestrais

Qual método de seleção é o melhor? Para responder essa pergunta foi feita uma análise prática, avaliando os dois métodos pela proximidade das soluções que geram com relação à solução ótima e escolhendo **X** aleatoriamente entre 10% e 30% do tamanho da população aleatória gerada.

Visualizando qual dos dois métodos possui maior soma de taxa de proximidade com a solução ótima para cada solução obtida, chegamos à conclusão que a roleta viciada performa melhor que a seleção elitista, o que pode ser explicado pela introdução de maior variação à população. No entanto, como pode ser analisado pelo **Gráfico 2**, a diferença de qualidade entre os dois métodos não passou de 20% no pior caso.

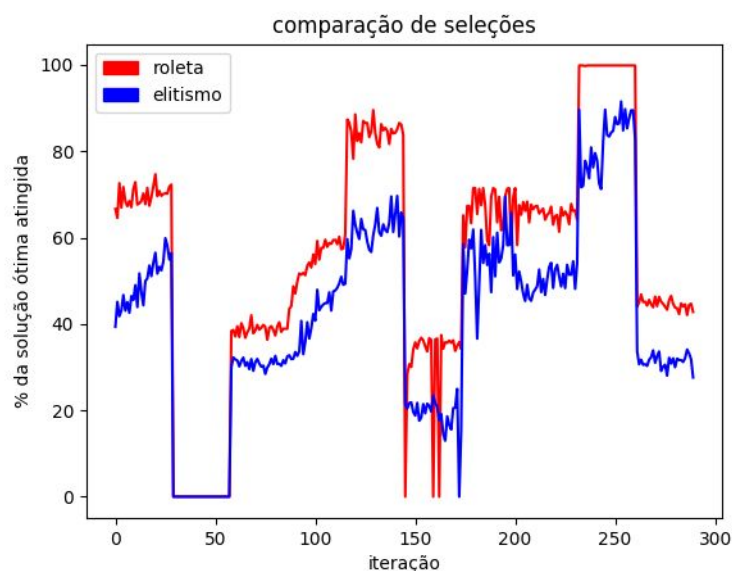


Gráfico 2: comparação entre os métodos para seleção de ancestrais.

3.3.2 Crossover

Método 1: Crossover de 1 Ponto

O crossover de um ponto gera os cromossomos de dois descendentes a partir dos cromossomos de dois ancestrais. O **Diagrama 2** ilustra o processo utilizado.

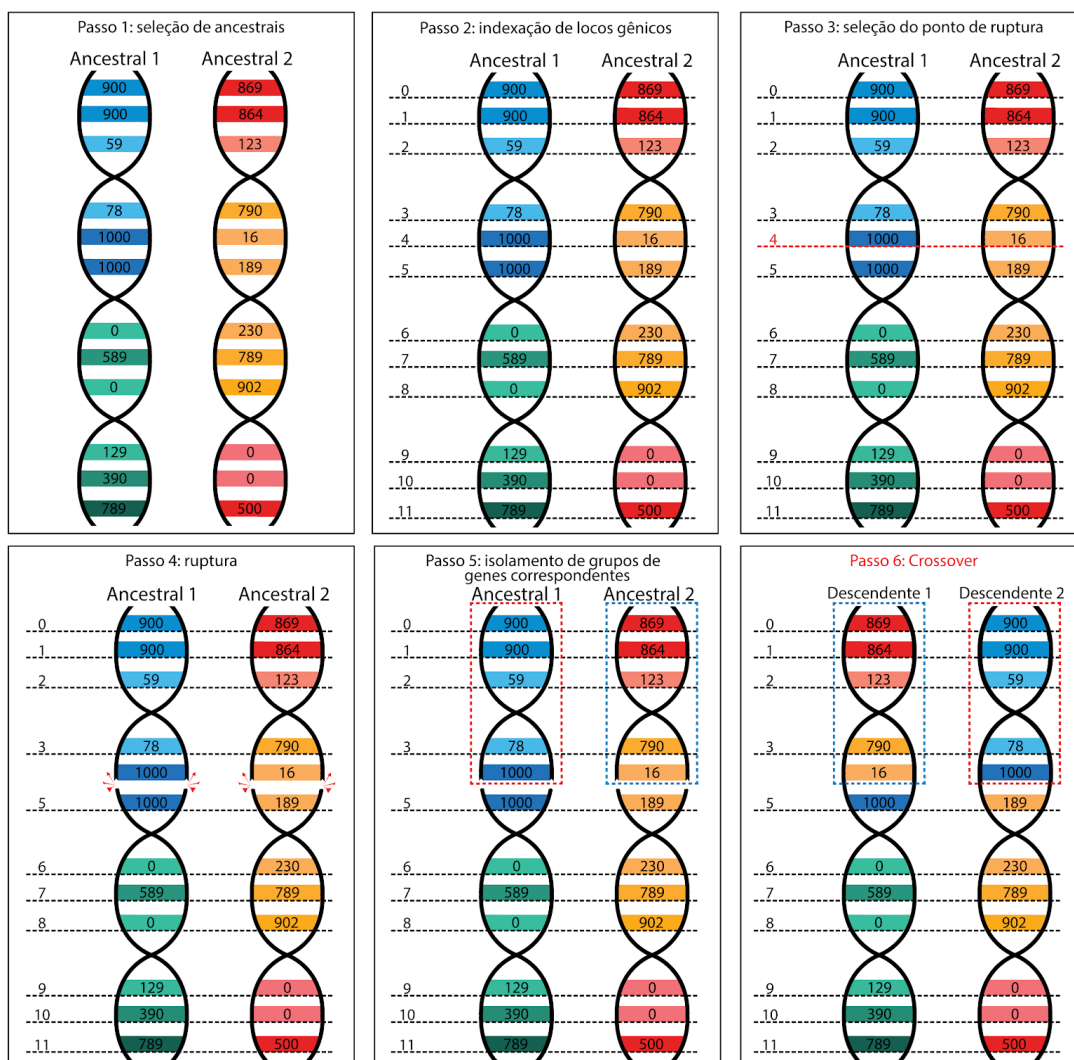


Diagrama 2 (autoral): passo a passo da simulação do Crossover de 1 Ponto.

Método 2: Crossover de 3 Pais

Em vez de utilizar o material genético de 2 pais em um ponto de cisão aleatória são utilizados 3 pais, sendo que cada filho receberá aleatoriamente uma porcentagem do material genético dos 3 pais escolhidos pelo método da roleta (definido como melhor pela análise).

Análise de métodos de Crossover

Analisando os dois métodos de crossover a partir da proximidade das soluções que geram com a solução ótima identificamos o crossover de 3 pais como sendo o melhor, como mostra o **Gráfico 3**. Tal resultado é bastante razoável pois a adição de mais um cromossomo selecionado pelo método da roleta viciada aumenta a variação dentro da população e, por conseguinte, a possibilidade de se encontrar um melhor encaixe de alelos a longo prazo.

Da análise do **Gráfico 3** percebemos que as soluções obtidas pelos dois métodos distam em no máximo 10% do valor da solução ótima, a variação entre os dois métodos não altera substancialmente a solução.

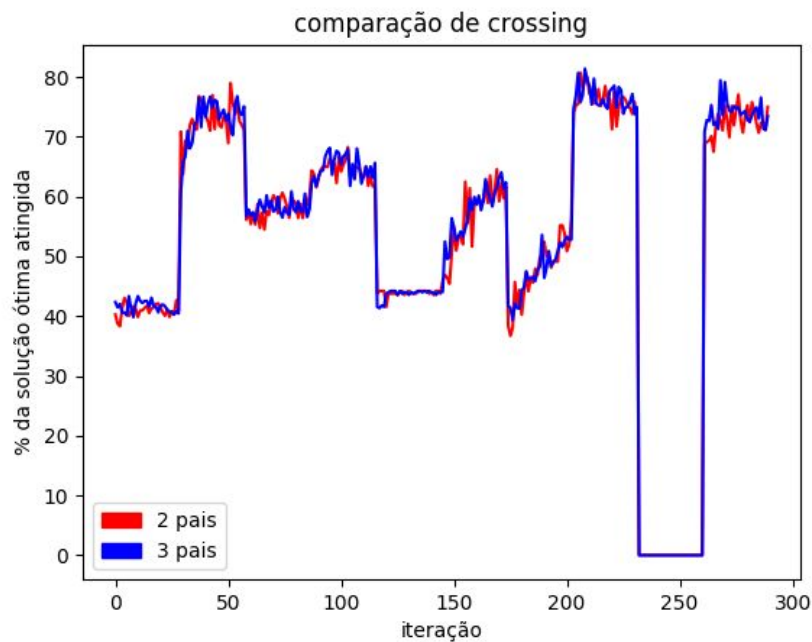


Gráfico 3: comparação entre os métodos de crossover.

3.3.3 Fator de Crossover

Em vez de efetuar o crossover usando todos os genes de um cromossomo pai podemos querer usar apenas uma parte do material genético dele e o restante de outro(s). A questão é: qual é a porcentagem dos genes do primeiro pai que queremos manter e qual é porcentagem que queremos mudar? Para isso fizemos um estudo variando o fator de crossover de 10% a 100% (índice de compartilhamento de genes entre o primeiro pai e os outros para compor o novo cromossomo).

Realizando a média do fator de crossover testado ponderado pela proximidade de sua solução com a solução ótima chegamos em 50,05% como fator de crossover ótimo, ou seja, mantém-se 50% do material genético do primeiro pai e pega-se o restante de outro(s) pai(s). Esse método garante uma variação controlada para a geração futura. A comparação feita pode ser acompanhada no **Gráfico 4**. Não é trivial identificar, mas o gráfico evidencia que os pontos associados ao fator de 50% (0.5) estão mais próximos à solução ótima em relação aos associados aos demais fatores.

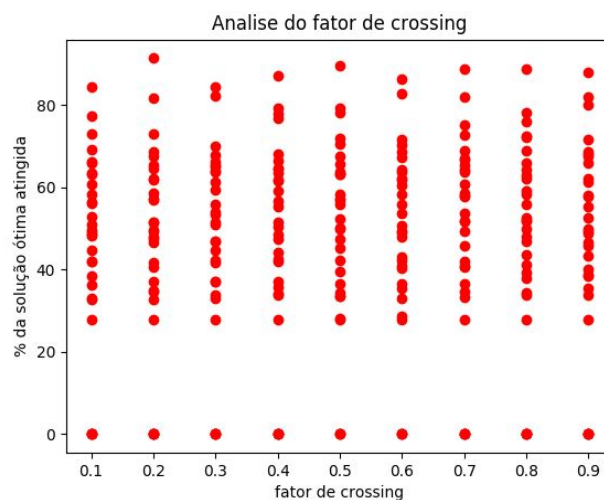


Gráfico 4: análise do fator de crossover.

3.3.4 Métodos de Mutação

Método 1: variação aleatória

Um cromossomo tem a probabilidade da taxa de mutação de ter um de seus locos gênicos alterado para uma quantidade aleatória de produtos, que tem a quantidade disponível como limite. A tentativa de mutação ocorre logo após a geração do filho a partir de cada conjunto de ancestrais no crossover.

Método 2: variação em termos de parcela

Um cromossomo tem a probabilidade da taxa de mutação de ter o seu loco gênico alterado para uma parcela da quantidade disponível do produto, calculada pela proporção do seu valor de item dividido pela soma do valor de todos os tipos de item do problema. Ou seja, prioriza-se em maior quantidade produtos que possuem maior valor.

Análise de métodos de Mutação

Antes de determinar qual a taxa de mutação ótima primeiramente foram testados os dois métodos de mutação utilizando fatores de mutação com instâncias aleatórias do problema para evitar overfit e encontrar o melhor entre os dois. Da análise do **Gráfico 5** chegamos à conclusão que a mutação aleatória representa o método de mutação mais eficiente (isso pode ser explicado pelo fato de que produtos com alto valor podem não ter um bom encaixe em uma solução intermediária).

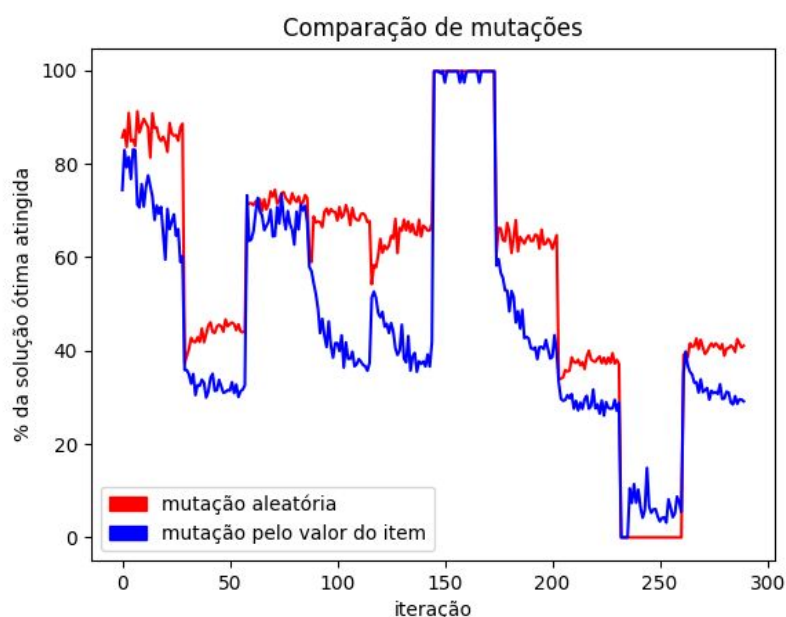


Gráfico 5: comparação dos dois métodos de mutação.

3.3.5 Taxa de Mutação

Para encontrar a melhor taxa de mutação para o Algoritmo Genético utilizamos a mesma estratégia utilizada até o momento para a seleção de uma taxa: geração de instâncias com a taxa variando em um intervalo controlado (1% a 3%) para comparar a proximidade da solução obtida

com a utilização dessa taxa com relação à solução ótima. O **Gráfico 6** ilustra o processo mencionado.

Efetuada a média ponderada das taxas pela proximidade da solução por elas gerada com solução ótima, foi encontrado o valor de 1.5% como valor ótimo para a taxa de mutação.

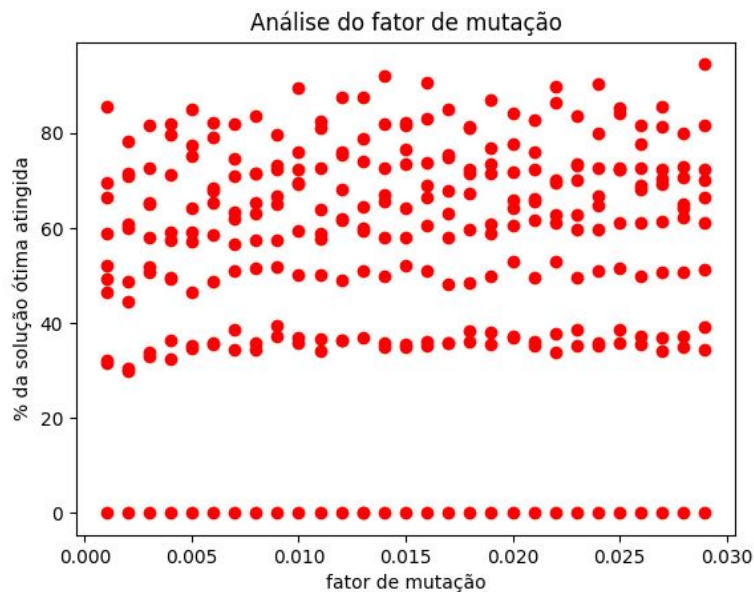


Gráfico 6: análise da taxa de mutação via geração aleatória em intervalo controlado.

Da análise do **Gráfico 6** percebemos que com aumento do fator de mutação a soma das distâncias entre cada ponto aumenta, portanto é menor a taxa de convergência para a solução ótima, conforme a fator de mutação aumenta a solução se aproxima da aleatoriedade e da busca da solução pelo método da força bruta, por isso restringimos a análise do fator de mutação ao máximo de 3%. Entretanto devemos deixar claro que o fator de mutação deve existir, para englobar casos em que apenas a seleção pela Função de Fitness e pela Roleta Viciada escolhem arranjos onde outros produtos poderiam ter um encaixe e valor de fitness melhores.

4. Complexidade

Como sabemos pela literatura o Problema da Mochila possui complexidade de tempo e uso de memória da ordem de $O(W.n)$, sendo W o peso dos itens e n o número de itens. Como o problema pesquisado apresenta repetição de itens, não podemos aplicar o problema da mochila diretamente, então foi feita uma redução utilizando uma função $\sigma(x)$ que mapeia cada um dos itens disponíveis para q_i repetições, sendo q_i a quantidade existente de cada item.

A redução aplicada $\sigma(x)$ tem complexidade $O(\sum(q_i))$, i de 1 a N_{produtos} , sendo: q_i a quantidade disponível do i -ésimo produto; N_{produtos} a quantidade de produtos diferentes. Dessa forma, contando a redução, esse problema representado pelo problema da mochila binária tem complexidade de tempo e de espaço dada por: $O(\sum(q_i).Volume)$ i de 1 a N_{produtos} .

A complexidade do Algoritmo Genético enunciada foi obtida analisando o código produzido. A complexidade dos métodos que foram usados para compô-lo assim como seu raciocínio pode ser encontrado no código do método **solve** da classe **Genetic Algorithm** no arquivo **main.py** em forma de comentário.

4.1 Complexidade de espaço e tempo do algoritmo da mochila ilimitada obtido:

$$O(\sum(q_i).Volume), i \text{ de } 1 \text{ a } N_{\text{produtos}}$$

4.2 Complexidade de espaço do algoritmo genético implementado

$$O(N_{\text{população}} \cdot N_{\text{produtos}})$$

4.3 Complexidade de tempo do algoritmo genético implementado

$$O(N_{\text{geração}} \cdot (N_{\text{população}} + N_{\text{produtos}}) + N_{\text{população}}^2 \cdot N_{\text{produtos}} \cdot \log(N_{\text{produtos}}))$$

Há alguma vantagem de utilizar o Algoritmo Genético em relação à Programação Dinâmica? Primeiro deve-se ter em mente que a quantidade da soma de itens de produtos e do volume de itens próximo a problemas reais são gigantescos, o que torna a aplicação do algoritmo de Programação Dinâmica inviável. Portanto devemos encontrar situações onde o Algoritmo Genético performa melhor em tempo quando temos uma tolerância quanto a otimicidade (por exemplo se quisermos chegar a 80% do valor ótimo), sendo que nesses casos:

$$O(\sum(q_i) \cdot \text{Volume}) > O(N_{\text{geração}} \cdot (N_{\text{população}} + N_{\text{produtos}}) + N_{\text{população}}^2 \cdot N_{\text{produtos}} \cdot \log(N_{\text{produtos}})), \text{ i de 1 a } N_{\text{produtos}}$$

Podemos comparar a performance assintótica dos dois algoritmos através do seguinte **Gráfico 7**, que considera como critério de parada uma taxa de 70% de convergência com a solução ótima da Programação Dinâmica ou após 1000 gerações terem sido analisadas. Os métodos de execução de cada instância foram realizados considerando os parâmetros ótimos encontrados na análise prévia.

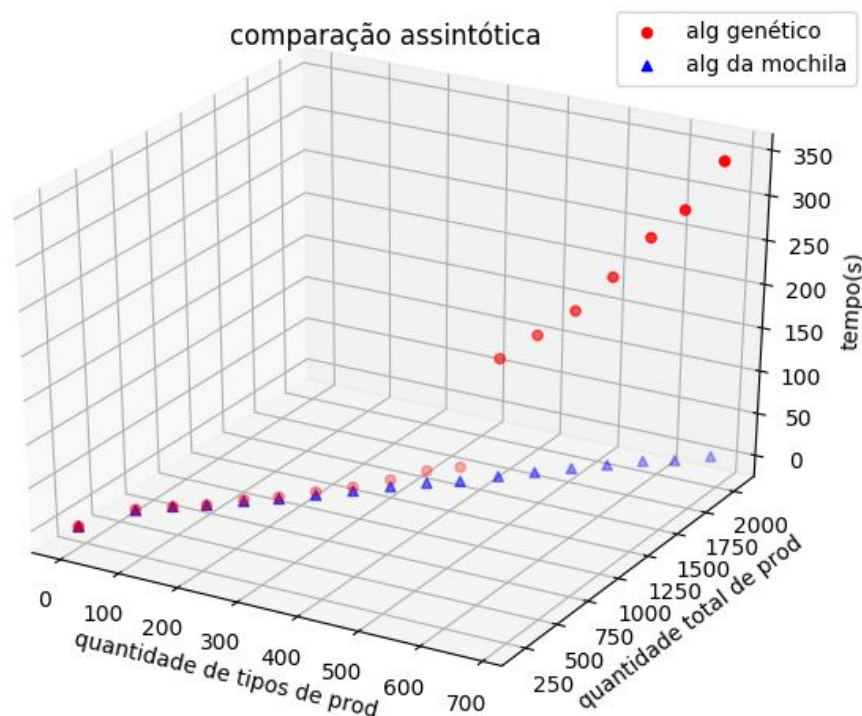


Gráfico 7: análise assintótica das soluções.

Com uma quantidade de tipos de produtos crescente a complexidade do Algoritmo Genético tem comportamento exponencial por conta do fator $O(N_{\text{geração}} \cdot (N_{\text{população}} + N_{\text{produtos}}))$. Comparando os dois algoritmos utilizando os valores e métodos ótimos para o Algoritmo Genético obtidos nas análises anteriores para menores instâncias, também chegamos aos dados mostrados na **Tabela 1**:

População	Gerações	Tipos de Produtos	Unidades	Porcentagem da solução ótima requerida	Porcentagem da solução ótima	Tempo decorrido via Algoritmo Genético (s)	Parou na Geração	Tempo decorrido via Programação Dinâmica (s)
3	100	3	300	72.0	74.63	0.001	52	0.143
13	150	13	372	72.0	73.29	0.004	27	0.197
23	200	23	397	80.0	70.33	0.084	200	0.215
33	250	33	426	73.0	73.01	0.044	55	0.245
43	300	43	457	78.0	77.22	0.411	300	0.278
53	350	53	475	80.0	76.63	0.669	350	0.295
63	400	63	500	74.0	74.09	0.469	182	0.323
73	450	73	528	83.0	77.83	1.608	450	0.358
83	500	83	553	75.0	75.33	0.143	31	0.403
750	750	750	3116	60.0	60.62	28.823	76	8.437
800	800	800	3280	60.0	60.42	26.374	61	9.444
850	850	850	3397	60.0	60.35	28.645	59	10.396
900	900	900	3524	60.0	60.60	48.871	89	10.987
950	950	950	3667	60.0	60.04	64.834	105	12.196
1000	1000	1000	3783	60.0	60.16	69.078	101	13.240

Tabela 1: compara diversas instâncias do problema utilizando Algoritmo Genético e Programação Dinâmica.

5. Análise da Tabela 1

É perceptível que com uma quantidade pequena de produtos, mas com uma grande quantidade de indivíduos, chegamos a um caso que para atingir uma convergência de 70% a 80% da solução ótima o Algoritmo Genético performa melhor que o algoritmo de Programação Dinâmica para mochila ilimitada. Esse fato se dá pela complexidade do algoritmo da mochila adaptado ter complexidade de tempo diretamente vinculada à quantidade de produtos e seus volumes, enquanto que a complexidade do Algoritmo Genético desenvolvido não possui dependência direta com a quantidade de exemplares de produtos mas sim com a quantidade de tipos de produtos.

Com uma distribuição mais esparsa de quantidades de produtos com menor volume, o algoritmo de Programação Dinâmica tem melhor performance temporal de caso médio que o Algoritmo Genético; com uma maior população de tipos de produtos o tempo para efetuar o crossover cresce multiplicado pelo número de gerações.

6. Conclusão

Algoritmos genéticos, no geral, fornecem uma solução aproximada, portanto não devem ser usado no caso em que a única solução exigida para o problema é a solução ótima. Dessa forma o Algoritmo Genético para a solução do problema da mochila, em casos que se tem um **threshold** de tolerância não muito estrito, pode performar em tempo melhor que a solução da mochila ilimitada.

A utilização de taxas de Crossover e de Mutação proporcionam melhora sensível nos resultados obtidos. A consequente adição de maior variação à população em estudo com a utilização destas taxas pode ser entendida como principal responsável pela melhora nos resultados. Essa é uma das principais características dos algoritmos evolutivos, classe de algoritmos à qual o Algoritmo Genético pertence.

A análise estatística para a obtenção dos melhores parâmetros para o Algoritmo Genético se mostrou efetiva possibilitando uma maior taxa de convergência para a solução ótima, entretanto para um melhor entendimento de como os parâmetros se relacionam entre si na solução do problema seria necessário um estudo exaustivo dos parâmetros, estudo esse que poderia ser feito testando todos os possíveis agrupamentos de parâmetros e métodos e analisando como eles impactam no desempenho do algoritmo.

Uma vez que o Algoritmo Genético é um algoritmo evolutivo sua abertura para otimização combinatória é de grande valor para problemas específicos, onde a complexidade temporal/espacial da solução ótima não segue ritmo polinomial.