

---

**Universidade Federal de São Paulo**

Instituto de Ciência e Tecnologia

---



**Projeto Final de Otimização Inteira  
Carregamento de cargas**

**Beatriz Viana Ferreira 133548  
Davi Juliano Ferreira Alves 133595**

São José dos Campos  
Março, 2021

## I. BREVE INTRODUÇÃO

---

O problema abordado neste trabalho é basicamente o problema da mochila que pode ser muito bem aplicado em questões do dia a dia, mais especificamente para empresas ou pessoas que tenham que fazer o transporte de produtos e mercadorias, mas o espaço é limitado, então uma boa maneira de resolver isso é escolher uma característica que será responsável por determinar quais produtos serão levados, se o desejo for selecionar a maior quantidade de elementos possíveis, então o algoritmo irá priorizar objetos que ocupem pouco espaço, o peso também pode ser uma característica, então o algoritmo irá priorizar objetos que possuem pouca ou muita massa, a depender do objetivo.



Desta forma, para que o problema seja solucionado sempre devemos levar em consideração os seguintes questionamentos:

- Quais são os critérios que levaremos em consideração para ocupar a mochila?
- Qual o peso e/ou o benefício (valor) de cada item?
- Qual é a capacidade limite de peso da mochila?
- A intenção é maximizar ou minimizar a FO?

**Palavras-chaves:** Problema da mochila, algoritmo genético, algoritmo guloso, solver

## II. FUNDAMENTAÇÃO TEÓRICA

---

### MODELAGEM DO PROBLEMA

De acordo com (CARVALHO, 2015), temos:

**Definição 1.** *O problema da mochila 0-1, ou mochila binária, consiste em, dada uma mochila de capacidade  $L$  e  $M$  itens cujos peso  $p_i$  e valor de utilidade  $v_i$  são dados, escolher quais itens serão alocados na mochila. Considere a variável de decisão  $x_j$  de maneira que, se o item  $j$  é alocado na mochila, então  $x_j = 1$ , caso contrário, então  $x_j = 0$ .*

Existem diversas variações do problema da mochila, pois é um problema de logística notável. Portanto, para resolver o problema da mochila com múltiplas mochilas, ou seja, receptores com tamanhos distintos entre si, temos o seguinte modelo, sofisticando o conceito de 0-1:

$$\begin{aligned} \text{MAX: } & \sum_{i=1}^n \sum_{j=1}^m (v_i x_{ij}) = Z. \\ \text{Sujeito à: } & \sum_{j=1}^m p_j x_{ij} \leq L_i, \forall i \in \{1, \dots, n\} \\ & \sum_{i=1}^n x_{ij} \leq 1, \forall j \in \{1, \dots, m\} \\ & x_{ij} \in \{0, 1\}, \\ & i = 1, \dots, n, j = 1, \dots, m. \end{aligned}$$

Sendo as variáveis,

- $x_{ij}$  diz se o item  $j$  está na mochila  $i$ .
- $L_i$  diz o volume da mochila  $i$ .
- $m$  é o número total de itens.
- $n$  é o número total de mochilas.

A extensão do conceito de diversos volumes pode ser aplicada à, por exemplo, uma frota de caminhões de tamanhos distintos que ocupam uma quantidade  $L_i$  de volumes de carga.

### III. APRESENTAÇÃO DO PROBLEMA

---

Suponha que fomos contratados por uma empresa X localizada no estado de São Paulo e desejamos maximizar seus lucros e para isso, pretende-se entregar com mais rapidez os produtos de maior valor (desconsiderando a localidade). No entanto, não é possível carregar o caminhão com todos os objetos disponíveis por causa de sua capacidade máxima. Logo, seremos responsáveis por selecionar os produtos que serão encaminhados de imediato.

Deste modo, visando maximizar o valor, concluímos que daremos preferência para produtos que tenham um valor elevado e que ocupem pouco espaço, a seguir tem-se uma tabela que mostra quais serão os produtos da base de dados:

Nome	Espaço (m³)	Valor (R\$)	Razão (R\$/m³)	Quantidade
Ar condicionado	0,7259	R\$ 3.379,00	4.654,91	10
Aspirador de pó	0,0054	R\$ 2.165,89	401.090,74	10
Cafeteira	0,0341	R\$ 3.484,19	102.175,66	10
Cama de casal	2,158	R\$ 1.551,03	718,73	10
Cama de solteiro	1,278	R\$ 930,90	728,40	10
Chuveiro	0,0107	R\$ 388,23	36.283,18	10
Conjunto sala de jantar	1,15	R\$ 1.488,02	1.293,93	10
Ferro de passar roupa	0,0259	R\$ 1.481,89	57.215,83	10
Fogão Brastemp	0,516	R\$ 4.859,10	9.416,86	10
Galaxy S20 Ultra	0,00011	R\$ 5.299,00	48.172.727,27	10
Geladeira Brastemp	1,155	R\$ 4.859,10	4.207,01	10
Guarda roupa	3,348	R\$ 1.367,90	408,57	10
Iphone 11	0,00091	R\$ 4.463,07	4.904.472,53	10
Lava louças	0,303	R\$ 3.999,00	13.198,02	10
Máquina de lavar roupa	0,518	R\$ 1.899,05	3.666,12	10
Microondas Electrolux	0,0865	R\$ 2.429,10	28.082,08	10
Notebook Acer	0,0016	R\$ 3.609,05	2.255.656,25	10
Notebook Dell	0,0019	R\$ 3.899,00	2.052.105,26	10
Notebook Lenovo	0,0018	R\$ 3.699,00	2.055.000,00	10
Painel para TV	1,294	R\$ 4.099,00	3.167,70	10
PS5	0,0105	R\$ 4.999,00	476.095,24	10
Purificador de água	0,028	R\$ 1.199,00	42.821,43	10
Sofá	2,898	R\$ 2.492,49	860,07	10
TV 65	0,37	R\$ 4.099,00	11.078,38	10
Ventilador	0,4	R\$ 284,90	712,25	10
<b>Total</b>	<b>16,32132</b>	<b>R\$ 72.424,91</b>		<b>250</b>

Figura 1: Produtos e quantidades disponíveis

Temos disponíveis 250 produtos, sendo 25 diferentes, onde cada objeto possui um valor e um espaço ocupado por ele. A partir das informações coletadas verificaremos a eficiência de três métodos de resolver o problema, sendo eles o **algoritmo genético**, o **algoritmo guloso** e o **solver** (uma ferramenta muito utilizada no excel).

## IV. ALGORITMO GENÉTICO

---

**Algoritmo genético (AG)** é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente pelo americano John Henry Holland. De acordo com (Carvalho, 2009), os Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, mas aleatória, que é voltada em direção ao reforço da busca de pontos de "alta aptidão", ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos).

Apesar de aleatórios, eles não são caminhadas aleatórias não direcionadas, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada iteração é chamada de *geração*. Durante cada iteração, os princípios de *seleção e reprodução* são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Através da seleção, se determina quais indivíduos conseguirão se reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pelo seu *índice de aptidão*. Em outras palavras, os indivíduos com maior adaptação relativa têm maiores chances de se reproduzirem.

Os Algoritmos Genéticos (AGs) diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos:

1. AGs trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros.
2. AGs trabalham com uma população e não com um único ponto.
3. AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
4. AGs utilizam regras de transição probabilísticas e não determinísticas.

Vislumbrando as etapas e especificações do código, temos:

- A **inicialização** é quando uma população de  $n$  indivíduos é gerada aleatoriamente. Cada um dos indivíduos da população representa uma possível solução para o problema, ou seja, um ponto no espaço de soluções.
- O **indivíduo** é o elemento que possui a resposta das escolhas, os valores que um indivíduo pode receber são binários, 0 o elemento não é selecionado, e 1 o elemento é selecionado, cada valor é considerado como gene.

- A **reprodução**, tradicionalmente, é dividida em três etapas: *acasalamento*, *recombinação* e *mutação*. O acasalamento é a escolha de dois indivíduos para se reproduzirem. A recombinação, ou crossing-over é um processo que imita o processo biológico homônimo na reprodução sexuada: os descendentes recebem em seu código genético parte do código genético do pai e parte do código genético da mãe. Esta recombinação garante que os melhores indivíduos sejam capazes de trocar entre si as informações que os levam a serem mais aptos a sobreviver, e assim gerar descendentes ainda mais aptos. A mutação cria diversidade, mudando aleatoriamente genes dentro de indivíduos, a mutação é aplicada de forma muito menos frequente do que a reprodução, assim como na natureza.

Existem diversas características dentro dos algoritmos genéticos, por isso, será feito um resumo das características principais do sofisticado algoritmo:

- O Algoritmo Genético é robusto até certo ponto, ou seja, ele possui um nível elevado de complexidade.
- A evolução é um processo natural e familiar para desvendar soluções.
- Ele busca uma solução ótima dentro de um espaço de soluções, filtrando o espaço.
- Quanto maior o número de gerações, maior a probabilidade de encontrar a solução ótima.

#### VANTAGENS DE UTILIZAR ALGORITMOS GENÉTICOS:

- Algoritmo genético é relativamente rápido;
- Direcionamento da soluções no espaço por meio das evoluções.

#### DESVANTAGENS DE UTILIZAR ALGORITMOS GENÉTICOS:

- Grande nível de complexidade em comparação com outros métodos de otimização;
- Aleatoriedade inicial quando se busca uma solução no espaço;
- Às vezes a solução ótima é inviável dentro das variáveis de contorno propostas.

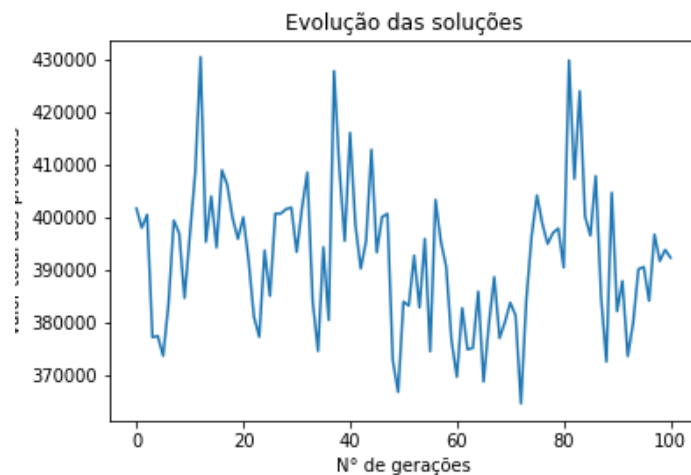
Após rodar o algoritmo, é plotado um gráfico que mostra a evolução das soluções de acordo com a sucessivas gerações e mostra-se a solução plausível. Como o algoritmo genético depende de uma probabilidade, então em cada compilação obteremos uma solução distinta. Em muitos casos, aumentar o tamanho da população e a quantidade de gerações nos possibilita uma maior chance de obter melhores soluções, mas não é regra, pode acontecer de piorar a solução, além de tornar o código mais custoso.

Limite de espaço $m^3$	Tamanho da população	Número de gerações	Tempo de execução (s)	Valor total	Espaço ocupado $m^3$
80	200	100	2,6262378	R\$ 449.345,68	75,566
80	200	300	7,8878338	R\$ 436.205,75	76,914
80	300	300	13,184637	R\$ 438.635,33	74,670

Note a tabela abaixo

A primeira linha da tabela mostra que mesmo tendo os valores dos parâmetros menores, o valor encontrado foi o melhor, além de ter executado em um tempo curto. O algoritmo trabalha com probabilidades, então o ideal é fazer uma lista com as melhores soluções, e a partir dessa lista extrair alguma ou então algumas respostas que satisfaçam o problema.

Na compilação do código, geramos um gráfico onde é possível notar as evoluções das soluções ao longo da reprodução. Logo, o código com 100 gerações e 50 indivíduos gerou o seguinte gráfico de *evolução de soluções*:



A melhor solução foi dada pela geração 12 (observe no gráfico que é onde atingiu o maior pico) igual a R\$430.441,26 para um espaço de  $79.82m^3$ , cuja restrição do espaço é  $80m^3$ .

O código do algoritmo genético está disponível para ser acessado em [https://colab.research.google.com/drive/110RFy6iMOzvcG3OoBt\\_N\\_5w1r4r6nPp9?usp=sharing](https://colab.research.google.com/drive/110RFy6iMOzvcG3OoBt_N_5w1r4r6nPp9?usp=sharing). Além disso, para a melhor compreensão do leitor, ele está comentado. Ressaltamos também que o código foi baseado em um curso da Udemy, mais especificamente, em <https://www.udemy.com/course/algoritmos-geneticos-em-python/>.

## V. ALGORITMO GULOSO

---

**Algoritmo guloso** ou míope é uma técnica de projeto de algoritmos que tenta resolver o problema fazendo a escolha localmente ótima em cada fase com a esperança de encontrar um ótimo global. Na solução de alguns problemas combinatórios a estratégia gulosa pode assegurar a obtenção de soluções ótimas, o que não é muito comum. No entanto, quando o problema a ser resolvido pertencer à classe NP-completo ou NP-difícil, a estratégia gulosa torna-se atrativa para a obtenção de solução aproximada em tempo polinomial.

A seguir, serão citadas algumas características do algoritmo guloso, características essas que são fundamentais para o funcionamento e pela lógica do mesmo:

- Jamais se arrepende de uma decisão, as escolhas realizadas são definitivas;
- Não leva em consideração as consequências de suas decisões;
- Podem fazer cálculos repetitivos;
- Nem sempre produz a melhor solução (depende da quantidade de informação fornecida);
- Quanto mais informações, maior a chance de produzir uma solução melhor.

### VANTAGENS DE UTILIZAR ALGORITMOS GULOSOS:

- Simples e de fácil implementação;
- Algoritmos de rápida execução;
- Podem fornecer a melhor solução (estado ideal).

### DESVANTAGENS DE UTILIZAR ALGORITMOS GULOSOS:

- Nem sempre conduz a soluções ótimas globais;
- Podem efetuar cálculos repetitivos;
- Escolhe o caminho que, à primeira vista, é mais econômico;
- Pode entrar em loop se não detectar a expansão de estados repetidos;
- Pode tentar desenvolver um caminho infinito.



O **código do algoritmo guloso** está disponível para ser acessado em <https://colab.research.google.com/drive/1UgqgeJVtddtqwls7QO3nEE-Maw47LknT?usp=sharing>. Além disso, para a melhor compreensão do leitor, ele está comentado.

## VI. SOLVER EXCEL

---

O Solver é um suplemento do Microsoft Excel que você pode usar para teste de hipóteses. O objetivo do processo de resolução do solver é encontrar os valores das variáveis de uma equação que resultem em um valor ótimo na célula alvo, também conhecida por "*objetivo*". Você pode escolher se o valor na célula alvo deve ser um máximo, um mínimo, ou se aproximar de um dado valor. Os valores iniciais das variáveis são inseridos no intervalo retangular de células da caixa alterando as células.

Você pode definir uma série de condições limitantes que definem restrições para algumas células. Por exemplo, você pode definir a restrição de que uma das variáveis ou células não pode ser maior que outra variável, ou não pode ser maior que um dado valor. Você também pode definir a restrição que uma ou mais variáveis devem ser inteiros (valores sem decimais), ou valores binários (onde somente 0 e 1 são permitidos).

Além disso, podemos escolher qualquer um dos três seguintes algoritmos ou métodos de solução na caixa de diálogo de Parâmetros do Solver:

- **Gradiente Reduzido Generalizado (GRG) - Não Linear** para problemas simples não lineares.
- **LP Simplex** para problemas lineares.
- **Evolucionário** para problemas complexos.

### AS VANTAGENS DO EXCEL SOLVER

- **Permite que você especifique várias células ajustáveis.** *Você pode usar até 200 células ajustáveis ao todo;*
- **Permite que você defina restrições nas células ajustáveis.** *Por exemplo, você pode dizer ao Solver para encontrar uma solução que não apenas maximize o lucro, mas também satisfaça certas condições, como alcançar uma margem bruta entre 20 e 30 por cento ou manter as despesas abaixo de R\$ 100.000. Essas condições são consideradas restrições à solução;*
- **Busca não apenas um resultado desejado (a “meta” em Goal Seek), mas também o ótimo.** *Por exemplo, procurar um resultado ideal pode significar que você pode encontrar uma solução que seja a máxima ou mínima possível.*
- **Para problemas complexos, o Solver pode gerar várias soluções.** *Você pode então salvar essas soluções diferentes em cenários diferentes.*

- **Praticidade.** *Gerenciar os dados usando o solver é extremamente rápido e prático.*

#### AS DESVANTAGENS DO EXCEL SOLVER

- **O tempo de processamento.** *Quando estivermos trabalhando com uma base grande de dados, o tempo de processamento do solver pode ser lento, uma vez que ele utiliza um método iterativo para realizar seus cálculos.*
- **Limitado.** *Se for um trabalho muito complexo ou com muitas variáveis, torna-se inviável.*

#### RESULTADOS

Para cada uma das soluções analisadas foram definidas três restrições:

- Tratamento da quantidade como inteira;
- Quantidade disponível de cada objeto;
- Espaço limite disponível.

Além disso, usamos o parâmetro LP Simplex, visando maximizar o valor total.

Segue a tabela abaixo com os resultados concatenados

Produto Espaço limite	Quantidade				
	5m <sup>3</sup>	20m <sup>3</sup>	40m <sup>3</sup>	60m <sup>3</sup>	80m <sup>3</sup>
Ar condicionado	0	8	10	10	10
Aspirador de pó	10	10	10	10	10
Cafeteira	10	10	10	10	10
Cama de casal	0	0	0	0	0
Cama de solteiro	0	0	0	0	0
Chuveiro	10	10	7	10	10
Conjunto sala de jantar	0	0	0	7	10
Ferro de passar roupa	10	10	10	10	10
Fogão Brastemp	0	10	10	10	10
Galaxy S20 Ultra	10	10	10	10	10
Geladeira Brastemp	0	0	10	10	10
Guarda roupa	0	0	0	0	0
Iphone 11	10	10	10	10	10
Lava louças	9	10	10	10	10
Máquina de lavar roupa	0	0	9	10	10
Microondas Electrolux	10	10	10	10	10
Notebook Acer	10	10	10	10	10
Notebook Dell	10	10	10	10	10
Notebook Lenovo	10	10	10	10	10
Painel para TV	0	0	2	10	10
PS5	10	10	10	10	10
Purificador de água	10	10	10	10	10
Sofá	0	0	0	0	6
TV 65	0	10	10	10	10
Ventilador	0	0	0	2	0
<b>Quantidade de produtos</b>	<b>129</b>	<b>158</b>	<b>178</b>	<b>199</b>	<b>206</b>
<b>Valor arrecadado</b>	<b>R\$ 407.948,64</b>	<b>R\$ 527.767,20</b>	<b>R\$ 607.240,96</b>	<b>R\$ 654.082,64</b>	<b>R\$ 672.931,84</b>
<b>Espaço ocupado</b>	<b>4,9963</b>	<b>19,7714</b>	<b>39,9911</b>	<b>59,7432</b>	<b>79,7812</b>

Figura 2: Soluções para 5m<sup>3</sup>, 20m<sup>3</sup>, 40m<sup>3</sup>, 60m<sup>3</sup> e 80m<sup>3</sup>

## VII. CONCLUSÃO

---

Para que possamos comparar a eficiência dos códigos, colocamos em uma tabela todos os resultados obtidos nos três métodos. Segue abaixo as tabelas:

Valores gerados com os produtos					
	5m³	20m³	40m³	60m³	80m³
Algoritmo Genético	No solution	No solution	R\$ 382.976,69	R\$ 425.076,09	R\$ 449.212,19
Algoritmo Guloso	R\$ 407.155,20	R\$ 527.767,20	R\$ 604.164,52	R\$ 625.680,17	R\$ 642.842,70
Solver Excel	R\$ 407.948,64	R\$ 527.767,20	R\$ 607.240,96	R\$ 654.082,64	R\$ 672.931,84

Note que o algoritmo guloso se saiu melhor em relação ao algoritmo genético, pois ele consegue ver quem são os produtos que mais compensam através do cálculo da razão, e vão inserindo os produtos até não caber mais nada, e como não existe aquela questão de aleatoriedade, o algoritmo pode ser executado diversas vezes com os mesmos parâmetros, que o resultado será sempre o mesmo. Além disso, perceba que o algoritmo genético não obteve a solução viável nos dois primeiros casos, em que o espaço era pouco e não foi suficiente para encontrar uma solução que satisfizesse as restrições impostas com 100 gerações e 50 indivíduos na população. Sendo assim, em todos os casos, o solver foi o que nos deu a melhor solução.

Espaços gerados com os produtos					
	5m³	20m³	40m³	60m³	80m³
Algoritmo Genético	No solution	No solution	37,529	56,297	70,220
Algoritmo Guloso	4,801	19,771	39,905	59,845	79,731
Solver Excel	<b>4,9963</b>	<b>19,7714</b>	<b>39,9911</b>	<b>59,7432</b>	<b>79,7812</b>

Assim como na tabela de valores as melhores soluções foram obtidas pelo Solver, no entanto o algoritmo guloso pareceu muito eficiente neste caso também, ambos ocuparam quase todo o espaço disponível. Enquanto, o algoritmo genético, mesmo sendo sofisticado e mais complicado de implementar, não obteve em nenhum caso uma solução próxima da esperada.

## REFERÊNCIAS

---

- UDEMY. Algoritmos Genéticos em Python. 2021. Disponível em: <https://www.udemy.com/course/algoritmos-geneticos-em-python/>. Acesso em: 17 jan. 2021.
- CARVALHO, Rubens. Problema da Mochila. Campinas, 2015. 14 p. Disponível em: <http://www.ime.unicamp.br/~eabreu/Projeto/RubensCarvalhoRA122181-MS777.pdf>. Acesso em: 17 jan. 2021.
- CARVALHO, André Ponce de Leon F. de. Algoritmos Genéticos. 2009. Disponível em: <https://sites.icmc.usp.br/andre/research/genetic/>. Acesso em: 17 jan. 2021.
- FEOFILOFF, Paulo. Algoritmos Gulosos. 2009. Disponível em: [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/guloso.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/guloso.html). Acesso em: 17 jan. 2021.
- LIBREOFFICE. Mozilla Public License V2.0. Solver. Disponível em: <https://help.libreoffice.org/latest/pt-BR/text/scalc/01/solver.html#:~:text=O%20objetivo%20do%20processo%20de%20resolu%C3%A7%C3%A3o%20do%20solver,m%C3%ADnimo%20C%20ou%20se%20aproximar%20de%20uma%20dado%20valor>. Acesso em: 17 jan. 2021.
- Definir e resolver um problema usando o Solver. Disponível em: <https://support.microsoft.com/pt-br/office/definir-e-resolver-um-problema-usando-o-solver-5d1a388f-079d-43ac>. Acesso em: 25 fev. 2021.
- McFedries, Paul. Excel Solver: What is It and When You Should Use It. Dummies. Disponível em: <https://www.dummies.com/software/microsoft-office/excel/excel-solver-what-is-it-and-when-you-should-use-it/>. Acesso em: 25 fev. 2021.