# Computação Evolutiva - Algoritmos Genéticos
# Problema da Mochila

**Adlla Katarine Aragão Cruz Passos**[1]

[1]Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, s/n – Novo Horizonte - 44036-900 – Feira de Santana – BA – Brasil

adllakatarine@hotmail.com

***Resumo.*** *O Problema da Mochila é um problema de otimização combinatória, aplicado em Algoritmos Genéticos, que busca preencher uma mochila com objetos de diferentes pesos e valores, objetivando preencher a mochila com o maior valor possível, não ultrapassando o peso máximo. Este relatório apresenta a resolução de problema proposto na disciplina EXA867 – Computação Evolutiva 2019.2 E aplicando o Problema da Mochila na compra de elementos para montar uma sala de jogos. É apresentado todas as técnicas aplicadas ao desenvolvimento do código para que se obtivesse um resultado satisfatório.*

## 1. Introdução

Um algoritmo genético (AG) é uma técnica de busca para encontrar soluções aproximadas em problemas de otimização e busca, desenvolvida por John Henry Holland em 1975. De acordo com a teoria evolucionária de Charles Darwin, os seres sofrem mutações e os mais aptos (melhores) vão sobreviver. E os indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações.

A partir disso, um algoritmo genético procura imitar esses princípios enquanto buscam uma melhor solução para um determinado problema, através da evolução de populações de soluções codificadas através de cromossomos artificiais [Pacheco ]. Ele é uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou crossing over).

A tabela da Figura 1 abaixo representa a analogia entre o Sistema Natural e os Algoritmos Genéticos:

**Figura 1. Tabela do Sistema Natural x Algoritmos Genéticos**

| Natureza | Algoritmos Genéticos |
|---|---|
| Cromossoma | Palavra binária, vetor, etc |
| Gene | Característica do problema |
| Alelo | Valor da característica |
| Loco | Posição na palavra, vetor |
| Genótipo | Estrutura |
| Fenótipo | Estrutura submetida ao problema |
| Indivíduo | Solução |
| Geração | Ciclo |

Fonte: Pacheco

Dentre os problemas que o Algoritmo Genético busca resolver está o Problema de Empacotamento. Este problema consiste em colocar objetos menores, chamados de itens, dentro de objetos maiores chamado de recipientes. Ao empacotar os itens, deve-se considerar ao máximo o aproveitamento do espaço dentro do recipiente ou, se escolher apenas alguns itens, considerando que os itens devem ser empacotados de forma otimizada [Moura and de Queiroz ].

Existem muitas variações deste problema, dentre eles o Problema da Mochila que é um problema de otimização combinatória. O nome dá-se devido ao modelo de uma situação em que é necessário preencher uma mochila com objetos de diferentes pesos e valores. O objetivo é que se preencha a mochila com o maior valor possível, não ultrapassando o peso máximo [Wikipédia 2014].

Sendo assim, foi proposto o desenvolvimento de um projeto relacionado a Algoritmos Evolutivos, cujo tema escolhido foi o Problema de Empacotamento, mais especificamente o Problema da Mochila. O problema será aplicado na compra de elementos para montar uma **sala de jogos**. O tema foi baseado em uma monografia que aplica o problema da mochila em decisões de compra de aparelhos destinados a atividades físicas, sendo uma das soluções para a situação apresentada a otimização dos espaços, onde procurasse alocar o maior número de equipamentos sem infringir as restrições utilizando o problema da mochila inteira [Bretas 2016].

O objetivo será, dentre um grupo de jogos selecionados, encontrar as melhores escolhas de compra que ocupe, no total, pelo menos o tamanho mínimo disponibilizado no ambiente e que não ultrapasse o tamanho máximo que poderá ser ocupado. Também não poderá ser ultrapassado o valor máximo especificado de gasto nas compras.

Considerando as especificações do projeto e os estudos feito relacionado as subcategorias do Problema da Mochila, foi decidido que o Problema da Mochila Binária (booleana ou 0/1) é que melhor se encaixa para esta resolução. Consiste, resumidamente, em um *vetor principal binário*, onde cada espaço representa um item e sempre que seu valor for 1 considera-se que o item foi escolhido, ou seja, "*colocado na mochila*".

## 2. Metodologia

Para o desenvolvimento do código, foi escolhida a linguagem de programação *Python* (na versão 3.7), a mais popular no meio da Ciência de Dados [Python ]. Também foi necessário o uso da biblioteca *Random* para preenchimento (aleatório) dos cromossomos de cada individuo da primeira geração e mutação [Random ].

Foi preciso criar uma base de dados com os possíveis jogos, simulando seus espaços, preços e valores, como pode ser observado na Figura 2 abaixo. A base contém 11 jogos de mesa e fliperama, cada um com seu **nome**, **espaço** (área) em metros quadrados, **preço** em real (R$) e **valor** com um número inteiro variando de 1 à 5. É importante ressaltar que os valores são irreais, apenas para representação.

| Index | Nome | Espaço ($m^2$) | Preço (R$) | Valor |
|---|---|---|---|---|
| 0 | Mesa de Sinuca | 0.751 | 999.9 | 5 |
| 1 | Máquina de Pinball | 0.199 | 2911.12 | 2 |
| 2 | Pebolim | 0.4 | 1346.99 | 3 |
| 3 | Pingue-Pongue | 0.39 | 2999.9 | 3 |
| 4 | Mesa de Jogo Poker/Truco | 0.3 | 1999 | 5 |
| 5 | Mesa de Hockey | 0.35 | 2499.9 | 3 |
| 6 | Mesa de Xadrez | 0.596 | 499.9 | 4 |
| 7 | Mesa de Futebol de Botão | 0.0499 | 508.66 | 1 |
| 8 | Máquina de Fliperama | 0.0599 | 429.9 | 4 |
| 9 | Mesa Arcade | 0.319 | 499.29 | 2 |
| 10 | Boliche | 0.835 | 849 | 4 |

**Figura 2. Base de dados**

Inicialmente, foi criada uma classe Jogo para criação dos objetos contendo nome, espaço, preço e valor de cada um. Foi necessário implementar a estrutura do Algoritmo Genético. A Figura 1 já apresentou as características principais de um AG. Ele é composto por uma população que é o conjunto dos indivíduos. Cada indivíduo é representado pelos seus cromossomos. E em geral, pode-se representar uma população como sendo vetor de indivíduos.Mas para esse problema, quem seriam os indivíduos? Primeiro, é importante saber do que o Problema da Mochila Binária precisa.

Como já explicado anteriormente, o Problema da Mochila (PM) busca preencher uma mochila com objetos de diferentes pesos e valores, objetivando preencher a mochila com o maior valor possível. Assim, o PM é composto de um conjunto de itens, peso e valor associado a cada item e a capacidade da mochila. Para este problema da *Sala de Jogos*, segue a analogia:

- **Conjunto de itens**: é representado pelos jogos presentes na base de dados.
- **Peso associado ao item**: como a ”mochila”neste problema é representado por uma ”sala”, ao invés do peso, cada item (ou jogo) contém o espaço que ocupa no ambiente em metros quadrados.
- **Valor associado ao item**: um número inteiro variando de 1 à 5.
- **Capacidade da mochila**: representado por um vetor com o espaço mínimo e máximo que pode ser ocupado na sala.

Agora pode-se entender melhor que cada individuo será uma solução de compra. O cromossomo será uma sequência de bits onde cada posição é um jogo, se seu valor for 1 o jogo deverá ser comprado e se for 0 não deverá. A partir disso foram criadas três classes: **Individuo**, **Populacao** e **AlgoritmoGenetico**.

A classe Individuo será responsável por armazenar as informações sobre a possível solução do problema como: cromossomo; fitness; espaço total usado; preço total gasto;

geração; e 3 vetores, cada um contendo os espaços, valores, preços de todos os jogos. O fitness é uma nota responsável pela avaliação do indivíduo, quanto mais alto o valor, melhor e mais adaptado ele está. Também é aqui onde se realiza a mutação do cromossomo.

Na classe Populacao, é mantido todos os indivíduos da geração atual, ela é a responsável por eles. Também contém uma lista com todos os jogos e suas informações, o limite de espaço que poderá ser usado na sala e o gasto máximo permitido. Por fim a classe AlgoritmoGenetico que se responsabiliza pela inicialização do algoritmo, da primeira população e próximas gerações e realiza o crossover entre os indivíduos.

Abaixo estão os métodos usados no processo da função de avaliação, seleção, crossover e mutação.

### 2.1. Função de Avaliação

A função de avaliação é a função que mede o quão adaptado está o indivíduo ao ambiente. Essa função é implementada de acordo a cada problema. Para este foi necessário a avaliação das variáveis espaço total usado e valor para a definição final do valor do fitness.

Para iniciar a avaliação, foi usada a Função 1 de soma abaixo onde a = espaço, b = valor, c = bit do cromossomo e k = posição do elemento nos vetores de todos os jogos.

$$\sum_{k=1}^{N} a_k * b_k * c_k \tag{1}$$

A soma total resulta no valor no fitness até o momento. Contudo ainda é necessário verificar se as restrições de espaço e valor gasto foram atendidas. Para isso é feita apenas a soma total de cada um e comparara com as restrições definidas. Se os valores estiverem dentro do limite, o fitness continua com seu valor atual, caso um ou os dois limites tenham sido quebrados o fitness recebe o valor 1.

### 2.2. Seleção

O processo de seleção está baseado no princípio da sobrevivência dos melhores indivíduos, ou seja, os indivíduos com melhor aptidão têm maior chance de perpetuar seu material genético, ou ao menos parte deste material. Dentre os vários métodos existentes, foram aplicados neste problema o Elitista e a Seleção por Torneio.

O método Elitista consiste em clonar uma certa quantidade dos indivíduos melhores avaliados de uma geração para a seguinte, evitando a perda de informações importantes presentes em indivíduos de alta avaliação e que podem ser perdidas durante os processos de seleção e cruzamento [Bento and Kagan 2008].

A Seleção por Torneio promove um torneio entre dois indivíduos aleatoriamente tomados na população. Assim, o indivíduo que vencer este torneio (indivíduo com melhor valor de aptidão entre o grupo), é selecionado para fazer parte da geração da nova população ou para participar do crossover.

### 2.3. Crossover

O crossover é o processo pelo qual indivíduos mais adaptados, ou seja, com melhores valores para a função de avaliação, têm a chance de perpetuar parte do seu material genético.

Foi usado o método de crossover de um ponto. Ele define um ponto de corte aleatório em cada cromossomo e os segmentos de cromossomo criados a partir dos pontos de corte são trocados, criando dois novos indivíduos.

### 2.4. Mutação

A mutação insere modificações aleatórias no cromossomo, com o objetivo de permitir maior variabilidade genética na população, impedindo que a busca fique estagnada em um mínimo local. A mutação usada sorteia uma posição aleatória do cromossomo e troca o valor do gene.

## 3. Resultados e Discussões

Finalizada a implementação do Algoritmo Genético, agora é o momento de verificar os resultados obtidos. Mas antes, foi preciso definir os valores de algumas variáveis. Essas variáveis são passadas por parâmetros na instância da classe *AlgoritmoGenetico* e determinam quantas gerações serão criadas, a quantidade de população por geração, quantidade de indivíduos o método elitista irá copiar, entre outras variáveis como pode ser observada na Figura 3 abaixo:

```
TAXA_CROSSOVER = 0.8
TAXA_MUTACAO = 0.1
GERACOES = 100
NUM_POPULACAO = 100
LIMITE_ESPACO = [2.4, 3]
GASTO_MAXIMO = 10000.00
QTD_ELITE = 20
```

**Figura 3. Variáveis globais**

Foram feitos alguns testes com os valores destas variáveis afim de observar com quais o algoritmo se adaptaria melhor e essa foi a configuração que chegou nos melhores resultados. A Figura 4 abaixo mostra duas melhores soluções encontradas, a primeira sendo a melhor encontrada dentre todas as gerações e a segunda a melhor solução da ultima geração. É importante observar que as duas soluções trazem o mesmo resultado e que a geração 34 já tinha encontrado a melhor. Também destaca-se que a melhor solução da geração atual na verdade foi da geração 81 e que chegou até a última pelo método elitista.

Apesar do ótimo resultado destas soluções em que o espaço usado não ultrapassou o limite determinado e quase o preencheu completamente, o valor gasto foi bem menor do que o máximo disponibilizado e mais da metade dos itens foram comprados, houve uma média de 20% da população que excedeu os limites impostos e consequentemente é uma solução ruim.

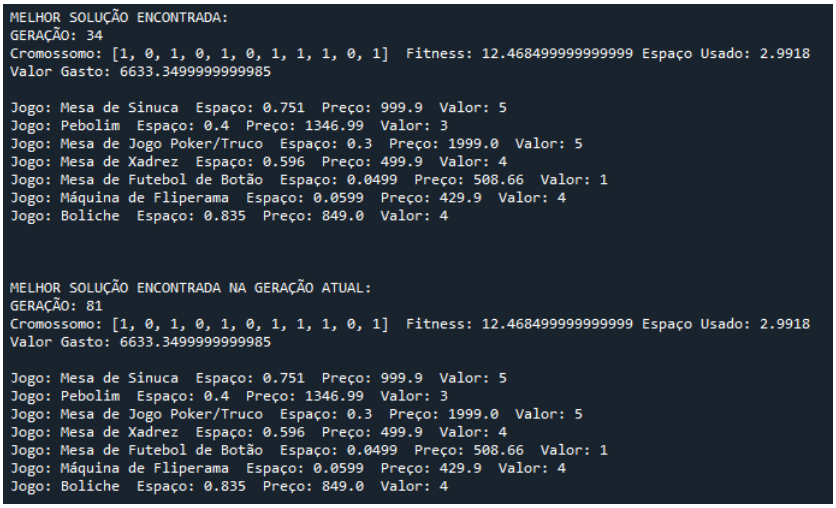


**Figura 4. Resultado final**

## 4. Conclusão

O problema proposto foi importante para o reforço do aprendizado de Algoritmos Genéticos e seus métodos e para o entendimento do Problema da Mochila, principalmente por ser sido implementado. Foi interessante ver como um Algoritmo Genético se comporta na prática em um problema "*real*".

Os requisitos propostos foram cumpridos, o AG obteve um resultado final esperado e satisfatório. Contudo, mais testes poderiam ter sido feitos utilizando outros métodos de crossover. Melhorias também poderiam ser adicionadas, até mesmo uma pequena mudança do problema, afim de ser possível a aplicação de outra variação mais complexa do Problema da Mochila.

## Referências

Bento, E. P. and Kagan, N. (2008). Algoritmos genéticos e variantes na solução de problemas de configuração de redes de distribuição. Disponível em: https://www.scielo.br/pdf/ca/v19n3/06.pdf.

Bretas, T. L. (2016). Problema da mochila inteira aplicado em decisões de compra em aparelhos de atividade desportivas: Modelo e aplicação. Disponível em: http://professor.ufop.br/sites/default/files/andre/files/thiago_lomas_bretas.pdf.

Moura, M. A. S. and de Queiroz, T. A. Investigando o problema da mochila irrestrita em sua versão bidimencional. Disponível em: https://files.cercomp.ufg.br/weby/up/506/o/resumo_mirella.pdf.

Pacheco, M. A. C. Algoritmos genéticos: Princípios e aplicação. Disponível em: http://www.inf.ufsc.br/ ∼ mauro.roisenberg/ine5377/CursosICA/CEintro_apost.pdf.

Python. Disponível em: https://docs.python.org/3/. Acesso em Outubro de 2020.

Random. Disponível em: https://docs.python.org/3/library/random.html. Acesso em Outubro de 2020.

Wikipédia, E. L. (2014). Problema da mochila. Disponível em: https://pt.wikipedia.org/wiki/Problema_da_mochila.