

Algoritmo Genético Binário Para o Problema da Mochila Binária

Vítor Rezende Silva

Abril 2024

1 Implementação

Para que o algoritmo seja executado ele deve receber as seguintes variáveis como entrada:

- tc: taxa de cruzamento
- tm: taxa de mutação
- n: número de gerações
- num_individuos: número de indivíduos por geração
- filename: nome dos arquivos, sem extensão, contendo os valores de capacidade, valor e peso da mochila.

Os algoritmo irá ler três arquivos no formato *_c.txt (capacidade), *_p.txt (valor) e *_w.txt (peso), e somente o nome do arquivo (sem o identificador e extensão) deve ser passado como entrada, por exemplo, caso o nome dos arquivos sejam file.p.txt, somente a string "file" deve ser passada.

O algoritmo foi implementado utilizando a penalização mais branda para casos em que o peso encontrado ultrapassasse a capacidade da mochila:

$$fitness = \sum_{i \in X} v[i] * (1 - (\sum_{i \in X} p[i] - c)/c)$$

Além disso, os testes foram realizados utilizando tanto o método do torneio quanto o da roleta para a seleção de pais.

2 Testes

Ao realizar os testes, foi possível observar que os resultados sempre ficavam estagnados entre as gerações, e os únicos parâmetros que influenciaram num melhor resultado foram o número de gerações e de indivíduos.

A primeira etapa dos testes teve o objetivo de identificar o melhor valor, entre 100 e 500, para o número de indivíduos e gerações, com um salto de 100 entre cada iteração. O resultado foi registrado na seguinte tabela:

valores\teste	1	2	3	4	5
100	284,161	284,161	265,146	266,127	309,165*
200	309,165*	270,165	309,165*	284,161	269,149
300	266,127	276,159	276,159	284,161	269,149
400	309,165*	309,165*	309,165*	309,165*	309,165*
500	284,161	309,165*	269,149	284,161	309,165*

Cada célula recebe dois valores, o primeiro representa o valor encontrado e o segundo representa o peso da mochila, além disso as células marcadas com um '*' representam aquelas que obtiveram o maior valor dentro da capacidade da mochila, ou seja, 309 e 165. Observando os testes, os valores que obtiveram melhores resultados foi 400 para o número de indivíduos e gerações.

Por fim, o algoritmo foi executado 10 vezes com esses parâmetros para observar quantas vezes o valor ótimo foi encontrado. Nas 10 iterações realizados, o resultado ótimo foi encontrado 5 vezes, gerando o seguinte gráfico

