

Análise de Algoritmos Genéticos para o Problema da Mochila

Patryck Fragoso Dias^{*1}

¹Universidade Tuiuti do Paraná – Curso de Ciência da Computação

Este trabalho apresenta uma implementação de Algoritmos Genéticos (AGs) para resolver o Problema da Mochila, analisando o impacto de diferentes configurações dos operadores genéticos. Foram avaliadas variações nos operadores de crossover (um ponto, dois pontos e uniforme), taxas de mutação (moderada e alta), métodos de inicialização (aleatória e heurística), além de critérios de parada (gerações fixas e convergência). Os resultados mostram que as configurações são equivalentes para a instância testada, mas a inicialização heurística com parada por convergência teve melhor eficiência temporal.

Palavras-chave: Algoritmos Genéticos, Problema da Mochila, Otimização Combinatória.

Resumo

This work presents an implementation of Genetic Algorithms (GAs) to solve the Knapsack Problem, analyzing the impact of different genetic operator configurations. Variations in crossover methods (one-point, two-point, and uniform), mutation rates (moderate and high), population initialization (random and heuristic), and stopping criteria (fixed generations and convergence) were evaluated. Results show that configurations performed similarly on the test instance, but the heuristic initialization with convergence criterion was the most time-efficient.

Keywords: Genetic Algorithms, Knapsack Problem, Combinatorial Optimization.

1 Introdução

O Problema da Mochila é um clássico problema de otimização combinatória, onde o objetivo é selecionar itens com valor e peso de modo a maximizar o valor total, respeitando a capacidade da mochila (KELLERER; PFERSCHY; PISINGER, 2004). Algoritmos

^{*}Email: patryck.dias@utp.edu.br

Genéticos (AGs) são adequados para este problema devido à sua capacidade de busca eficiente em espaços complexos (GOLDBERG, 1989).

Este trabalho avalia o impacto de diferentes configurações dos operadores genéticos:

- Métodos de *crossover*: um ponto, dois pontos, uniforme;
- Taxas de mutação: 0.2 (moderada), 0.5 (alta);
- Inicialização: aleatória vs. heurística (baseada em valor/peso);
- Critério de parada: número fixo de gerações vs. convergência (20 gerações sem melhora).

2 Metodologia

2.1 Codificação e Aptidão

Cada indivíduo é representado por um vetor binário x_i , onde 1 indica que o item i foi selecionado. A função de aptidão retorna a soma dos valores dos itens, desde que o peso total não ultrapasse a capacidade C . Soluções inviáveis recebem valor 0.

2.2 Configurações Testadas

Foram avaliadas cinco configurações:

Tabela 1: Configurações dos Algoritmos Genéticos

Configuração	Crossover	Mutação	Inicialização	Parada
Base	Dois pontos	0.2	Aleatória	100 gerações
Uniforme	Uniforme	0.2	Aleatória	100 gerações
UmPonto	Um ponto	0.2	Aleatória	100 gerações
AltaMutação	Dois pontos	0.5	Aleatória	100 gerações
HeurísticaConverg	Dois pontos	0.2	Heurística	Convergência

3 Resultados

3.1 Resumo Numérico

Tabela 2: Comparação entre as Configurações

Configuração	Aptidão	Tempo (s)	Peso Total
Base	9.0	0.14	9
Uniforme	9.0	0.17	10
UmPonto	9.0	0.14	10
AltaMutaçao	9.0	0.14	10
HeurísticaConverg	9.0	0.02	9

3.2 Gráficos Comparativos

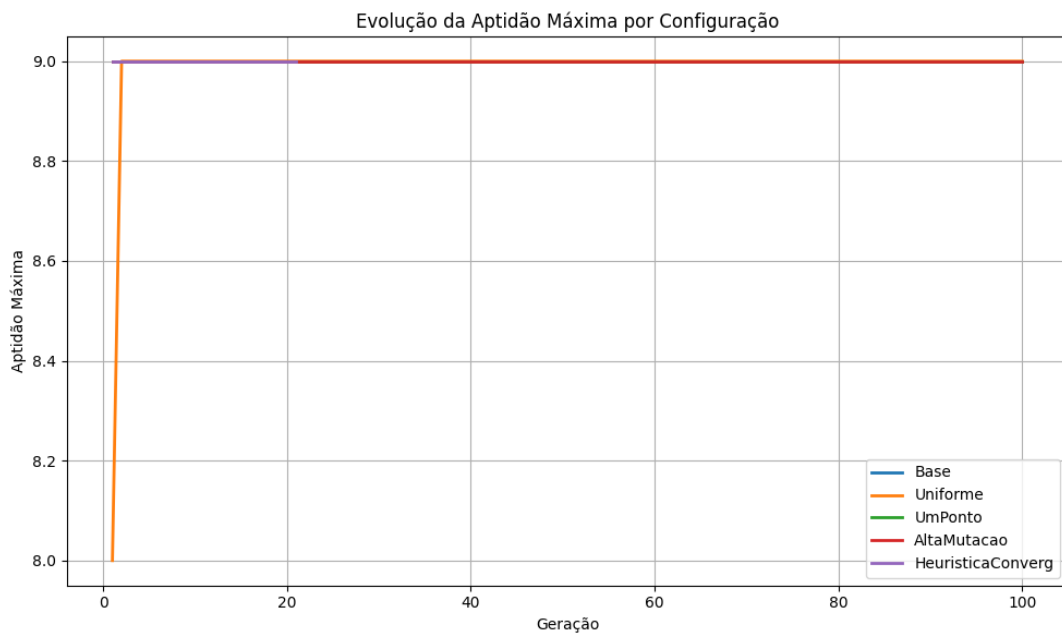


Figura 1: Evolução da Aptidão Máxima por Configuração

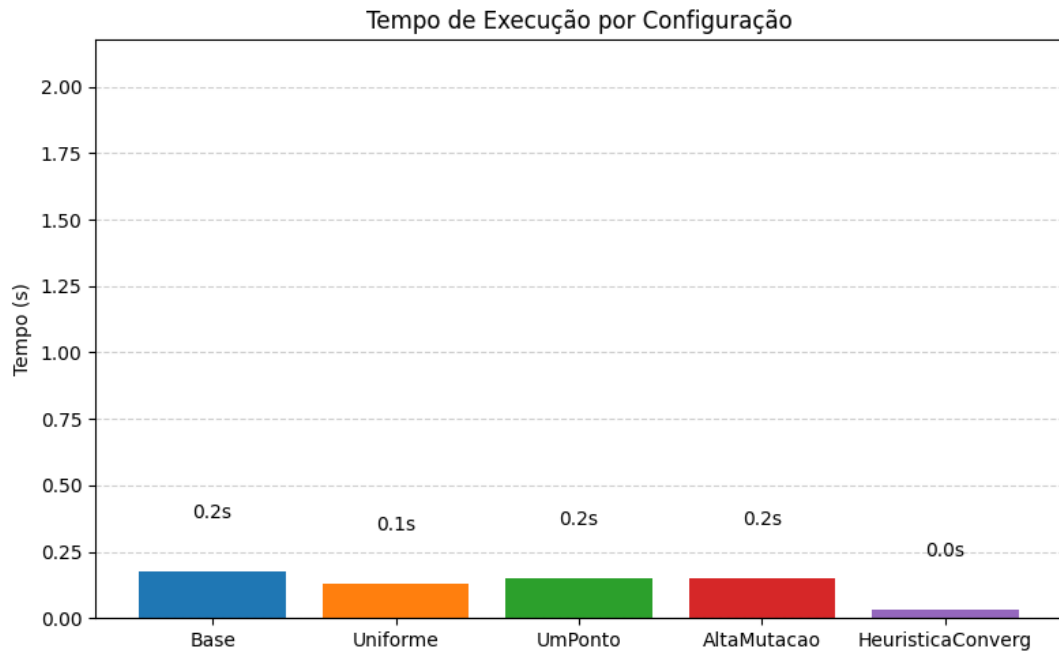


Figura 2: Tempo de Execução por Configuração

4 Discussão

As cinco configurações alcançaram aptidão máxima 9.0 rapidamente, indicando baixa complexidade da instância. Ainda assim, observou-se:

- Pequena vantagem de tempo para **HeurísticaConverg**, com tempo de execução 85% menor;
- Configurações com crossover *um ponto* e *uniforme* tiveram desempenho similar à base;
- A alta taxa de mutação (0.5) não trouxe melhorias claras;
- O critério de convergência foi eficaz ao encerrar antes das 100 gerações.

5 Conclusão

O uso de AGs no Problema da Mochila mostrou-se eficaz. As configurações testadas geraram soluções ótimas, mas com variações relevantes no tempo de execução. A inicialização heurística com parada por convergência destacou-se por sua eficiência.

Trabalhos futuros incluem:

- Testes com instâncias maiores e mais difíceis;
- Uso de operadores de mutação adaptativa;

- Avaliação em problemas multiobjetivo.

Referências

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. [S.l.]: Addison-Wesley, 1989. ISBN 978-0201157673.

KELLERER, H.; PFERSCHY, U.; PISINGER, D. *Knapsack Problems*. [S.l.]: Springer, 2004. ISBN 978-3-540-40286-7.