

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Projeto 01 - Computação Bioinspirada

Aluno: Thiago Vasconcelos Braga

Matrícula: 11921BSI225

UBERLÂNDIA, SETEMBRO 2023.

Esse trabalho trata-se do desenvolvimento de um algoritmo para solução do problema da mochila binária e uma comparação da solução desenvolvida com programação dinâmica. Abaixo traremos alguns passos utilizados pela solução bem como um comparativo de desempenho:

Etapas de Inicialização da População:

O primeiro passo é gerar uma população inicial de indivíduos, sendo que, cada indivíduo representa uma solução inicial, neste trabalho utilizaremos uma heurística gulosa para criar indivíduos iniciais que são de alta qualidade, ou seja, são candidatos a serem soluções ótimas para o problema. A heurística gulosa é utilizada para selecionar os itens que serão incluídos na mochila para cada indivíduo da população que será gerada, os ordenando com base em sua relação valor/peso em ordem decrescente, priorizando os itens mais valiosos em relação ao peso. Em seguida, os itens são selecionados até que a capacidade máxima da mochila seja alcançada, garantindo que a solução seja válida.

Algoritmo Genético:

Após a criação da população inicial, o algoritmo genético prossegue com as etapas de seleção, crossover e mutação, onde os indivíduos mais aptos são mais propensos a serem selecionados como pais e combinados para criar novos indivíduos.

Seleção:

A seleção dos pais é realizada com base na aptidão dos indivíduos. Indivíduos com maior aptidão têm uma maior probabilidade de serem selecionados como pais.

Crossover:

Os pais selecionados são combinados através de um operador de crossover, no caso foi utilizado o ponto de corte (one-point crossover), o que cria novos indivíduos (filhos) que herdam características de seus pais.

Mutação:

É aplicada uma taxa de mutação aos filhos gerados, isso envolve a alteração aleatória de genes em um indivíduo com uma pequena probabilidade com objetivo de implantar diversidade na população.

Substituição da População:

Os indivíduos da população anterior são substituídos pelos filhos gerados e possivelmente mutados o que garante que a população evolua ao longo das gerações.

Convergência:

O algoritmo genético continua a evoluir a população por um número especificado de gerações, ao longo do tempo, espera-se que a população se aproxime de soluções melhores para o Problema da Mochila Binária.

Resultado Final esperado:

Após as iterações, o algoritmo genético retorna o melhor indivíduo encontrado, que representa uma solução aproximada para o problema da mochila.

Execução:

A fim de realizar o comparativo, 4 instâncias do problema foram executadas 6 vezes em cada algoritmo, avaliamos a capacidade de gerar boas soluções obtendo a média do valor máximo obtido em cada uma das instâncias e o desvio padrão dessas médias:

Programação Dinâmica				
	Instância 01	Instância 02	Instância 03	Instância 4
Média do valor máximo obtido	31621	67829	143449	28840
Desvio Padrão do valor máximo médio	0	0	0	0
Algoritmo Genético				
	Instância 01	Instância 02	Instância 03	Instância 4
Média do valor máximo obtido	30783,16667	60737	116936,5	24649
Desvio Padrão do valor máximo médio	646,9238535	2566,194524	8678,17223	621,6778373

Tabela 01 - Média e desvio padrão em cada instância

Por meio da tabela acima, elaboramos um gráfico comparativo entre as médias de valor máximo obtidas por cada algoritmo em cada instância, que pode ser visualizado abaixo:

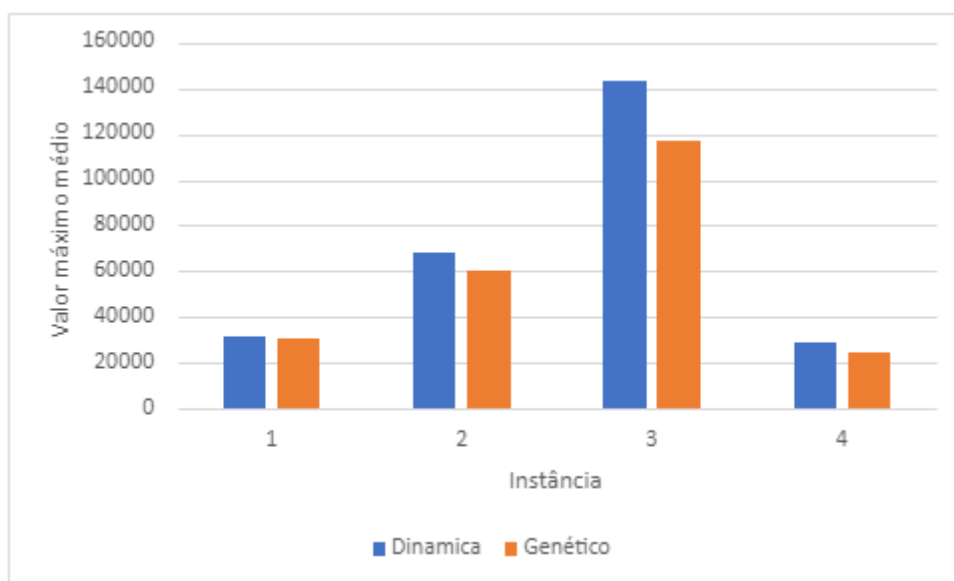


Gráfico 01 - Comparação de valor máximo médio obtido em cada instância

É possível analisar que a programação dinâmica tem o desempenho levemente superior ao genético desenvolvido no quesito de gerar o máximo valor possível.

Outro comparativo relevante é o tempo de execução de cada uma das soluções, pois podem representar de certa maneira a complexidade de cada um dos algoritmos. Abaixo podemos visualizar uma tabela com cada um o tempo de execução para às 4 instâncias de cada implementação:

	Genético	Dinâmico
Execução 01	0.6910014152526855	0.45499205589294434
Execução 02	0.6369991302490234	0.47295546531677246
Execução 03	0.9339959621429443	0.44903111457824707
Execução 04	0.6829590797424316	0.5039961338043213
Execução 05	0.6770007610321045	0.46300292015075684
Execução 06	0.701068639755249	0.47598814964294434

Tabela 02 – Tempo de execução em segundos de todas as instâncias para cada vez que o algoritmo foi executado.

Conclusão:

Como o problema da mochila binária trata-se de um caso muito complexo de se obter a solução ótima, tudo o que temos são implementações que garantem uma solução próxima à ótima, na medida do possível. Analisando os resultados obtidos, podemos concluir que, no quesito, gerar soluções próximas à ótima sem gasto desproporcional de recursos computacionais, o algoritmo genético se assemelha muito nos resultados obtidos pela

programação dinâmica, que é considerada um dos métodos mais eficientes em gerar soluções para esse tipo de problema.