Análise Comparativa de Desempenho de um Algoritmo Genético para o Problema da Mochila

Iuker de Souza Santos[1]

[1] Universidade Tuiuti do Paraná
Curitiba – PR

iuker.santos@utp.edu.br

Resumo. *Este trabalho detalha a implementação de um Algoritmo Genético (AG) para a resolução do Problema da Mochila 0-1, um clássico problema de otimização NP-difícil. O objetivo principal é conduzir uma análise quantitativa sobre o impacto de diferentes operadores e parâmetros no desempenho do algoritmo. Foram avaliadas variações nos métodos de crossover (ponto único, dois pontos, uniforme), taxas de mutação (1%, 5%, 10%), estratégias de inicialização da população (aleatória e heurística) e critérios de parada (gerações fixas e convergência). Os experimentos foram realizados em um conjunto de 10 instâncias com complexidades variadas. Os resultados, obtidos de forma reproduzível, indicam que a inicialização heurística e uma baixa taxa de mutação foram os fatores de maior impacto positivo na qualidade da solução, enquanto o critério de parada por convergência demonstrou uma drástica redução no tempo de execução, embora com um custo na otimalidade para instâncias maiores.*

1. Introdução

O Problema da Mochila 0-1 (Knapsack Problem) é um dos mais estudados em otimização combinatória e ciência da computação. O desafio consiste em selecionar um subconjunto de itens, cada um com um peso e um valor específicos, para maximizar o valor total sem exceder a capacidade de peso de uma mochila [Martello and Toth 1990]. Devido à sua natureza NP-difícil, encontrar uma solução ótima por meio de métodos exatos torna-se computacionalmente inviável para instâncias de grande porte.

Nesse contexto, meta-heurísticas como os Algoritmos Genéticos (AGs) surgem como uma alternativa robusta e eficiente. Os AGs são algoritmos de busca inspirados na teoria da evolução natural, que operam sobre uma população de soluções candidatas, aplicando operadores genéticos como seleção, crossover e mutação para evoluir em direção a soluções de alta qualidade [Goldberg 1989]. A eficácia de um AG, no entanto, é altamente dependente da configuração de seus parâmetros e operadores.

Este trabalho propõe a implementação de um AG para resolver o Problema da Mochila 0-1 e realiza uma análise experimental para quantificar o impacto de diferentes configurações de seus operadores. O estudo compara sistematicamente três tipos de crossover, três taxas de mutação, duas estratégias de inicialização e dois critérios de parada, visando identificar as configurações mais eficazes e discutir os trade-offs entre a qualidade da solução e o tempo de execução.

2. Metodologia

Para realizar a análise comparativa, foi implementado em Python um Algoritmo Genético modular, permitindo a fácil alteração dos operadores para a fase de experimentação.

2.1. Estrutura do Algoritmo Genético

Os componentes centrais do AG implementado são:

- **Representação do Cromossomo:** Uma solução é representada por um vetor binário de tamanho n, onde n é o número de itens. Um gene na posição i com valor 1 significa que o item i está na mochila.
- **Função de Aptidão (Fitness):** A aptidão de um cromossomo é a soma dos valores dos itens selecionados. Se a soma dos pesos exceder a capacidade da mochila, a solução é inválida e sua aptidão é 0.
- **Seleção de Pais:** Foi utilizado o método de seleção por torneio de 3 participantes.
- **Elitismo:** O melhor indivíduo de cada geração é copiado diretamente para a próxima, garantindo a preservação da melhor solução encontrada.
- **Reprodutibilidade:** A semente do gerador de números aleatórios foi fixada em 42 (‘random.seed(42)‘) para garantir que os resultados sejam consistentes e reproduzíveis.

2.2. Configurações dos Experimentos

Foram executados testes sobre 10 instâncias do problema. Para cada grupo de teste, um único parâmetro foi variado, mantendo os demais em uma configuração base (Crossover de Ponto Único, Mutação de 5%, Inicialização Aleatória, Parada por Gerações Fixas). As variações testadas foram:

- **Crossover:** Ponto Único, Dois Pontos, Uniforme.
- **Taxa de Mutação:** Baixa (1%), Média (5%), Alta (10%).
- **Inicialização da População:** Aleatória e Heurística.
- **Critério de Parada:** Gerações Fixas e Convergência.

As métricas de avaliação foram a qualidade da solução (valor do fitness) e o tempo de execução.

3. Resultados

Os resultados agregados dos experimentos, somando os valores e tempos de todas as 10 instâncias, são apresentados nas tabelas a seguir. Cada tabela foca na variação de um único parâmetro, permitindo uma análise clara do seu impacto isolado.

Tabela 1. Resultados Agregados por Tipo de Crossover.

Configuração	Soma dos Valores	Soma dos Tempos (s)
Ponto Único	19297	0.5805
Dois Pontos	19108	0.6997
Uniforme	19203	0.5971

A Tabela 1 detalha o desempenho dos diferentes operadores de crossover. O crossover de Ponto Único alcançou a maior soma de valores totais (19297), superando

o crossover Uniforme em 0.49% e o de Dois Pontos em 0.99%. Em termos de tempo de execução, o crossover de Ponto Único também se mostrou eficiente, sendo mais rápido que as outras duas variantes. Notavelmente, o crossover de Dois Pontos foi o que obteve o menor valor agregado e, simultaneamente, o maior tempo de execução, indicando uma ineficiência tanto na busca quanto na computação para este conjunto de problemas.

Tabela 2. Resultados Agregados por Taxa de Mutação.

Configuração	Soma dos Valores	Soma dos Tempos (s)
Baixa (1%)	19632	0.5645
Média (5%)	19293	0.5901
Alta (10%)	19157	0.6089

Conforme a Tabela 2, a taxa de mutação demonstrou uma correlação inversa com o desempenho. A taxa baixa (1%) apresentou o melhor resultado absoluto, com uma soma de valores de 19632 e o menor tempo de execução (0.5645s). Em comparação, a taxa média (5%) resultou em uma qualidade de solução 1.76% inferior, enquanto a taxa alta (10%) produziu um resultado 2.48% inferior ao da taxa baixa. O tempo de execução também aumentou ligeiramente com taxas de mutação mais altas, o que sugere que a maior aleatoriedade pode prolongar a busca sem trazer benefícios de qualidade.

Tabela 3. Resultados Agregados por Método de Inicialização.

Configuração	Soma dos Valores	Soma dos Tempos (s)
Aleatória	19241	0.5770
Heurística	19639	0.5590

Os dados da Tabela 3 indicam um ganho de performance expressivo ao utilizar uma inicialização heurística. A população inicializada com base em uma heurística de densidade de valor superou a inicialização aleatória em 2.07% na soma dos valores agregados. Além disso, a abordagem heurística resultou em um tempo de execução ligeiramente menor. Este resultado sugere que direcionar o início da busca para uma região promissora do espaço de soluções é uma estratégia altamente eficaz.

Tabela 4. Resultados Agregados por Critério de Parada.

Configuração	Soma dos Valores	Soma dos Tempos (s)
Gerações Fixas	19141	0.5601
Convergência	18731	0.1348

A Tabela 4 demonstra um claro trade-off entre qualidade e velocidade. O critério de parada por gerações fixas obteve uma soma de valores 2.19% superior à da parada por convergência. No entanto, o tempo de execução foi drasticamente diferente: a parada por convergência foi 75.9% mais rápida, executando em 0.1348 segundos contra 0.5601 segundos do critério de gerações fixas. Isso evidencia que, embora a parada por convergência seja extremamente veloz, ela corre o risco de interromper o processo prematuramente, antes de o algoritmo ter a chance de refinar a solução.

4. Discussão

A análise aprofundada dos resultados permite extrair insights sobre a dinâmica do Algoritmo Genético aplicado ao Problema da Mochila e a interação entre seus operadores.

O operador de **crossover de Ponto Único** mostrou-se o mais eficaz. Este resultado sugere que, para o Problema da Mochila, a preservação de grandes blocos de genes contíguos (conhecidos como esquemas) dos pais é uma estratégia vantajosa. Tais esquemas podem representar combinações sinérgicas de itens que se encaixam bem na capacidade da mochila. Operadores mais disruptivos, como o Uniforme, quebram esses blocos com maior probabilidade, potencialmente destruindo combinações promissoras e retardando a convergência para soluções de alta qualidade. O crossover de Dois Pontos, embora menos disruptivo que o Uniforme, parece oferecer um equilíbrio desfavorável entre preservação e exploração para este problema.

Em relação à **taxa de mutação**, os dados foram conclusivos em favor de uma taxa baixa (1%). Este fenômeno está ligado ao balanço entre exploração (busca por novas áreas) e intensificação (refinamento de boas soluções). Uma vez que o AG, auxiliado pelo elitismo, encontra uma região promissora do espaço de busca, perturbações pequenas e raras (mutação baixa) são mais eficientes para o refinamento fino da solução. Taxas mais altas (e.g., 5% ou 10%) introduzem um nível de ruído que pode ser contraproducente, destruindo indivíduos de alta qualidade com mais frequência do que os melhora. Isso prejudica a convergência, transformando a busca em um processo mais aleatório e menos direcionado.

O impacto mais significativo na qualidade da solução veio do **método de inicialização**. Ao iniciar a busca com uma população parcialmente ”inteligente”, baseada em uma heurística de densidade de valor (valor/peso), o algoritmo recebe um ”pontapé inicial”de alta qualidade. Em vez de gastar tempo computacional valioso explorando regiões aleatórias e de baixa aptidão do espaço de busca, o AG começa a partir de um patamar superior, focando seus esforços em refinar soluções que já são inerentemente promissoras. Isso não apenas justifica a melhoria no valor final, mas também a ligeira redução no tempo de execução, pois a convergência para uma solução de alta qualidade é acelerada.

Finalmente, a análise do **critério de parada** expõe o trade-off fundamental entre tempo de execução e otimalidade da solução. A parada por convergência é extremamente rápida porque encerra o processo assim que a diversidade da população cai abaixo de um limiar, ou seja, quando os indivíduos se tornam muito semelhantes. No entanto, isso representa um risco significativo de estagnação em ótimos locais. Uma população homogênea pode simplesmente significar que o AG encontrou um pico de aptidão local, mas não necessariamente o global. O critério de gerações fixas, embora mais custoso, força o algoritmo a continuar a busca, dando à mutação (mesmo que em taxa baixa) mais tempo e oportunidade para introduzir diversidade e potencialmente ”saltar”para fora de um ótimo local em direção a uma solução ainda melhor. A escolha ideal entre os dois depende diretamente da prioridade da aplicação: velocidade ou otimalidade.

5. Conclusão

A análise experimental demonstrou que a configuração dos operadores de um Algoritmo Genético impacta profundamente seu desempenho na resolução do Problema da Mochila

0-1. Os resultados, obtidos de forma reproduzível, revelam que a **inicialização heurística** e uma **taxa de mutação baixa (1%)** foram os fatores mais determinantes para a obtenção de soluções de alta qualidade.

A melhor configuração geral, visando a maximização do valor, seria a combinação de **inicialização heurística**, **crossover de ponto único**, **taxa de mutação de 1%** e parada por **gerações fixas**. Caso o tempo de execução seja um fator crítico, a substituição da parada por **convergência** oferece uma aceleração substancial, embora com um sacrifício na qualidade da solução final.

Como trabalhos futuros, sugere-se a investigação de métodos de seleção mais sofisticados, como a seleção por roleta ou classificação (rank selection), e a hibridização do AG com algoritmos de busca local para refinar as soluções encontradas. Adicionalmente, a implementação de mecanismos de ajuste dinâmico para a taxa de mutação poderia otimizar o equilíbrio entre exploração e intensificação da busca de forma adaptativa.

Referências

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.

Martello, S. and Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley.