



Universidade Federal
de São João del-Rei

Departamento de Ciência da Computação

Vítor Augusto Niess Soares Fonseca

Relatório Algoritmo Genético Para Mochila Binária

São João del-Rei, Abril de 2024

Sumário

1	Introdução	3
2	Metodologia	3
2.1	Geração da População Inicial	3
2.2	Avaliação dos Indivíduos	3
2.3	Seleção dos Pais	3
2.4	Cruzamento e Mutação	4
2.5	Geração da Próxima Geração	4
2.6	Parâmetros e Execução	4
3	Resultados e Discussões	4
4	Considerações Finais	6

1 Introdução

Os algoritmos genéticos (AGs) são uma classe de algoritmos de otimização baseados em princípios biológicos de evolução natural. Inspirados no processo de seleção natural, os AGs são capazes de resolver uma ampla variedade de problemas de otimização em diferentes domínios. Uma das áreas em que os algoritmos genéticos têm sido amplamente aplicados é a resolução do problema da mochila binária.

O problema da mochila binária é um problema clássico de otimização combinatória que envolve selecionar um subconjunto de itens de uma lista, cada um associado a um peso e a um valor, de modo a maximizar o valor total, enquanto se mantém o peso total dentro de um limite predefinido. Esta restrição binária impõe que cada item seja selecionado ou não, sem frações. Apesar de sua simplicidade conceitual, o problema da mochila binária é NP-difícil, o que significa que não existe uma solução algorítmica eficiente para encontrar a solução ótima em tempo polinomial para instâncias de tamanho arbitrário.

Neste trabalho, propomos a utilização de algoritmos genéticos para abordar o problema da mochila binária. Especificamente, investigaremos o impacto de diferentes métodos de seleção de pais e funções de penalização na eficácia do algoritmo genético em encontrar soluções de alta qualidade para o problema da mochila binária. Duas abordagens distintas serão comparadas: o método de seleção de pais pelo torneio e pelo método da roleta, além da comparação entre uma função de penalização branda e outra mais rígida.

2 Metodologia

Para abordar o problema da mochila binária utilizando algoritmos genéticos, implementamos um conjunto de funções em Python que compõem o núcleo do nosso algoritmo. As principais etapas do algoritmo incluem a geração da população inicial, a avaliação dos indivíduos, a seleção dos pais, o cruzamento, a mutação e a geração da próxima geração.

2.1 Geração da População Inicial

A função `genRandBin(bits)` gera números binários aleatórios de tamanho `bits`, representando indivíduos na população inicial. A função `genIniPop(popSize, bits)` utiliza a função anterior para gerar uma população inicial de tamanho `popSize`.

2.2 Avaliação dos Indivíduos

Duas funções de avaliação foram implementadas: `objFuncMin` e `objFuncMax`. A primeira calcula o fitness de cada indivíduo utilizando uma função objetivo com penalização mais branda, enquanto a segunda utiliza uma função objetivo com penalização mais rígida. Ambas as funções retornam uma lista de tuplas, onde cada tupla contém o valor de fitness e a posição do indivíduo na população.

2.3 Seleção dos Pais

Implementamos dois métodos de seleção de pais: o método do torneio e o método da roleta. A função `tourn(fits)` realiza um torneio entre dois candidatos selecionados aleatoriamente da população, enquanto a função `roul(pop, fits, scalFactor)` utiliza o método da roleta,

onde a probabilidade de seleção de um indivíduo é proporcional ao seu valor de fitness após um escalonamento exponencial.

2.4 Cruzamento e Mutação

A função `crossOver(fat, mot, bits, mutRate)` realiza o cruzamento de dois pais individuais utilizando um ponto de corte aleatório. Em seguida, a função `mutate(son, mutRate)` realiza a mutação de um indivíduo, onde cada gene tem uma probabilidade de ser alterado de acordo com a taxa de mutação `mutRate`.

2.5 Geração da Próxima Geração

Por fim, a função `iniCross(pop, fits, bits, popSize, mutRate, selMet, scalFactor)` combina todas as etapas anteriores para gerar a próxima geração de indivíduos. Esta função recebe como entrada a população atual, os valores de fitness, o tamanho dos indivíduos, o tamanho da população, a taxa de mutação, o método de seleção de pais e o fator de escalonamento para o método da roleta.

2.6 Parâmetros e Execução

Definimos os parâmetros do algoritmo, incluindo as listas de pesos e valores dos objetos, a capacidade máxima da mochila, o tipo de função objetivo, o método de seleção de pais, o fator de escalonamento e as taxas de mutação. Em seguida, executamos o algoritmo por um número específico de gerações, registrando o melhor fitness de cada geração.

3 Resultados e Discussões

Para avaliar o desempenho dos algoritmos genéticos na resolução do problema da mochila binária, foram realizados testes utilizando diferentes configurações. Cada teste consistiu em executar o algoritmo em quatro cenários distintos, variando o método de seleção de pais (torneio ou roleta) e a função de penalização (branda ou rígida).

Foram realizados cinco testes para cada um dos quatro casos, totalizando 20 experimentos. Em cada teste, variamos dois parâmetros: o tamanho da população (10, 50 e 100 indivíduos) e a taxa de mutação (0.01, 0.05 e 0.10).

Os resultados foram registrados na Tabela 1. Cada entrada na tabela representa o resultado de um teste específico, indicando o método utilizado, o tamanho da população, a taxa de mutação e a geração em que o valor ótimo foi alcançado.

Para garantir a consistência dos resultados, estabelecemos um critério de desclassificação: se um teste não encontrasse o valor ótimo (309) em mais de três execuções, ele seria desclassificado.

A análise dos resultados mostrou que diferentes combinações de parâmetros influenciaram significativamente o desempenho do algoritmo. Por exemplo, observamos que o método do torneio com penalização rígida tendeu a encontrar o valor ótimo em menos gerações em comparação com outras configurações. Além disso, pudemos observar como o tamanho da população e a taxa de mutação afetaram a convergência do algoritmo e o tempo necessário para alcançar o valor ótimo.

Esses resultados fornecem insights valiosos para a escolha adequada dos parâmetros do algoritmo genético na resolução do problema da mochila binária, permitindo uma abordagem mais eficiente e eficaz para problemas semelhantes.

Tabela 1: Resultados dos testes para cada caso

Caso	Método	População	Taxa de Mutação	Geração do Ótimo
1	Torneio - Penalização Branda	10	0.01	Desclassificado
1	Torneio - Penalização Branda	10	0.05	Desclassificado
1	Torneio - Penalização Branda	10	0.1	33
1	Torneio - Penalização Branda	50	0.01	Desclassificado
1	Torneio - Penalização Branda	50	0.05	12
1	Torneio - Penalização Branda	50	0.1	5
1	Torneio - Penalização Branda	100	0.01	Desclassificado
1	Torneio - Penalização Branda	100	0.05	5
1	Torneio - Penalização Branda	100	0.1	2
2	Torneio - Penalização Rígida	10	0.01	Desclassificado
2	Torneio - Penalização Rígida	10	0.05	Desclassificado
2	Torneio - Penalização Rígida	10	0.1	50
2	Torneio - Penalização Rígida	50	0.01	Desclassificado
2	Torneio - Penalização Rígida	50	0.05	30
2	Torneio - Penalização Rígida	50	0.1	10
2	Torneio - Penalização Rígida	100	0.01	Desclassificado
2	Torneio - Penalização Rígida	100	0.05	5
2	Torneio - Penalização Rígida	100	0.1	3
3	Roleta - Penalização Branda	10	0.01	Desclassificado
3	Roleta - Penalização Branda	10	0.05	Desclassificado
3	Roleta - Penalização Branda	10	0.1	Desclassificado
3	Roleta - Penalização Branda	50	0.01	Desclassificado
3	Roleta - Penalização Branda	50	0.05	40
3	Roleta - Penalização Branda	50	0.1	25
3	Roleta - Penalização Branda	100	0.01	Desclassificado
3	Roleta - Penalização Branda	100	0.05	Desclassificado
3	Roleta - Penalização Branda	100	0.1	25
4	Roleta - Penalização Rígida	10	0.01	Desclassificado
4	Roleta - Penalização Rígida	10	0.05	Desclassificado
4	Roleta - Penalização Rígida	10	0.1	50
4	Roleta - Penalização Rígida	50	0.01	Desclassificado
4	Roleta - Penalização Rígida	50	0.05	Desclassificado
4	Roleta - Penalização Rígida	50	0.1	30
4	Roleta - Penalização Rígida	100	0.01	Desclassificado
4	Roleta - Penalização Rígida	100	0.05	44
4	Roleta - Penalização Rígida	100	0.1	21

4 Considerações Finais

Este trabalho proporcionou uma experiência valiosa no desenvolvimento e análise de algoritmos genéticos para resolver o problema da mochila binária. A implementação dos algoritmos de seleção de pais por torneio e roleta, juntamente com diferentes estratégias de penalização, permitiu uma compreensão mais profunda do funcionamento e da influência desses componentes no desempenho do algoritmo genético.

Além disso, a realização dos testes com diferentes configurações de população e taxa de mutação proporcionou insights sobre a sensibilidade desses parâmetros e sua influência na convergência e eficácia do algoritmo. Observou-se que a escolha adequada desses parâmetros pode ter um impacto significativo no desempenho do algoritmo e na capacidade de encontrar soluções ótimas para o problema da mochila binária.

Em suma, este trabalho contribuiu para uma melhor compreensão dos algoritmos genéticos e sua aplicação para resolver problemas de otimização, além de fornecer uma oportunidade para aplicar conceitos teóricos em um contexto prático. O aprendizado adquirido durante o desenvolvimento deste trabalho será fundamental para futuras explorações em algoritmos evolutivos e para abordagens de resolução de problemas de otimização mais complexos.