

Population Algorithm - Problema da Mochila Binária

Ana C. Knop¹, Carlos S. Mondo¹, Leandro M. da Silva¹

¹ENGETEC – Universidade da Região de Joinville (Univille)
Caixa Postal 246 – 89201-972 – Joinville – SC – Brazil

{ana.knop, carlosm, leandrosilva.1}@univille.br

Resumo. *Experimento com o algoritmo populacional evolucionário para resolver o problema da mochila binária.*

1. Problema da Mochila Binária

Este experimento com o algoritmo populacional evolucionário foi utilizado para resolver o problema da mochila binária, no qual dado uma mochila, qual a melhor solução a fim de maximizar o lucro.

Utilizando duas instâncias diferentes de um conjunto de benchmarks, devem ser realizados testes para cada instância no qual consistem em: executar 30 vezes, calcular a média e o desvio padrão. Com isso, calcular a melhor solução.

2. Algoritmo Populacional

São algoritmos geracionais que utilizam um conjunto de soluções candidatas para o determinado problema. No qual seu conjunto de soluções é chamado de população.

Uma geração representa um conjunto de soluções em um determinado período de tempo. Os algoritmos utilizam G gerações para manipular as soluções.

Uma solução é tipicamente um vetor, chamado de indivíduo que possui o tamanho M, sendo o conjunto de soluções uma matriz NxM (N é a quantidade de soluções e M é o tamanho de cada solução) em que cada linha é uma matriz é uma solução candidata.

Para cada solução é atribuído um valor de “fitness”, este valor de fitness ou aptidão define o qual bem a solução resolve o problema. Pode ser utilizado o valor calculado por uma função objetivo. Tipicamente é utilizado uma matriz 1xN (matriz coluna), onde N é a quantidade de soluções candidatas.

Assim podemos utilizar do elitismo para gerar uma solução melhor, no qual esta nova geração é incluída na próxima geração, o foco desta técnica é evitar perder a melhor solução encontrada durante o processo.

Incluindo o cruzamento, que consiste em selecionar dois indivíduos daquela melhor geração, selecionar suas metades e combinar para gerar dois novos filhos e assim utilizar eles no lugar de seus pais.

A mutação é outra técnica porem esta consiste em modificar um indivíduo.

E a reprodução, na qual a estratégia desse algoritmo é a reprodução de novos indivíduos pode ser realizada através de mutações, cruzamento entre outros operadores. Estas operações podem envolver 1 ou mais indivíduos. Novos indivíduos gerados são adicionados na nova geração, e todos os indivíduos da geração anterior são descartados.

Para finalizar o processo de busca quando não for mais capaz de gerar novas soluções por N gerações, quando atingir X avaliações de indivíduos entre outro.

3. Descrição dos Algoritmos

Para solucionar este problema, devemos ter como entrada de parâmetros lucro dos objetos, peso dos objetos, tamanho da mochila e a penalidade.

A penalidade serve para quando uma solução ultrapassa o tamanho da mochila. Para isso definimos que: Solução inválida - Fitness = 0, Penalizo a solução = Descontar o valor do fitness ou Corrijo a solução - Retirar algum item da mochila.

Com isso, podemos desenvolver uma função objetiva no qual calculamos o quão lucrativo é a mochila. Sendo 1 para itens na mochila e 0 para itens que não estão na mochila. Sendo assim multiplicamos o 1 ou 0 pelo valor do item.

E executamos o algoritmo de gerações. Inicialmente geramos a população com uma distribuição uniforme.

Fazemos uma avaliação da população e da solução usando a função objetivo.

Iniciamos um loop enquanto não atingimos nosso critério de parada. Enquanto este critério não é atingido, realizamos o algoritmo de elitismo e para cada item numa lista que representa o tamanho da população, realizamos o cruzamento utilizando os dois melhores filhos, separando eles em duas partes e juntando umas com as outras para gerar dois filhos que irão substituir o pai, realizamos a mutação naquele item, descobrimos o fitness da próxima população e incrementamos a quantidade de avaliações realizadas. Ao terminar o loop do tamanho da população, geramos uma nova população. Ao terminar todos os loops, somos capazes de identificar a melhor solução e o fitness.

4. Resultados

Executando 30 vezes o algoritmo para dois casos de testes com os seguintes dados:

| Parâmetro | Valor |
|---|-------|
| Penalidade | 25 |
| Tamanho da População | 4 |
| Quantidade total de Avaliações | 20 |
| Percentual p/ performar uma mutação total de Avaliações | 3 |

Os benchmarks utilizados podem ser encontrados neste link: https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/knapsack_01.html.

Comparando com os resultados atingidos na experiência anterior, a inclusão do cruzamento entre os indivíduos nos trouxe ganhos consideráveis em encontrar o melhor indivíduo da geração já que forçamos o cruzamento entre os melhores indivíduos.