



BCC466 - Técnicas Metaheurísticas de Otimização Combinatória

Professor: Marcone Jamilson Freitas Souza

Aluno: André Ribeiro de Brito 11.2.4985

Problema de Alocação de caixas em contêiner usando Algoritmo Genético

Definição

- O carregamento de *containers* é um problema clássico em pesquisas de otimização, cujo objetivo, em geral, é aumentar o volume utilizado em relação ao volume total disponível. Em outras palavras, deseja-se encontrar uma configuração de arrumação de cargas de tal maneira que o volume ocioso do *container* seja o mais próximo de zero.
- Quando existem mais cargas a serem carregadas do que volume disponível nos *containers*, acrescenta-se ao problema a necessidade de reduzir o número de *containers* para dispor todas as cargas.
- A melhor utilização do volume disponibilizado pelo *container* e a menor quantidade de *containers* utilizados reduz despesas com o transporte de mercadorias, o que deve refletir no preço dos produtos e nos resultados financeiros das companhias.

Problemas semelhantes

- O problema combinando *knapsack* e *bin packing* é denominado *Multiple Container Packing Problem* (MCP) ou Problema de Empacotamento em Múltiplos *Containers* (RAIDL, 1999), onde se tem o objetivo múltiplo de maximizar o espaço utilizado e, ao mesmo tempo, minimizar o número de *containers*. O *knapsack* pode ser encarado como um problema MCP com apenas um *container*, onde o objetivo passa a ser unicamente de maximizar o espaço utilizado.

Modelagem

- **As instâncias foram geradas da seguinte maneira:**
 - Conjunto de caixas N.
 - Altura x da caixa N.
 - Comprimento y da caixa N.
 - Tamanho x Container.
 - Tamanho y Container
 - $\text{tempoExecucao} = 60.0 * 2.0$

Modelagem

- A restrição é conseguir colocar um numero maximo de caixas em um container, com isso , a aptidaoParcial tem que ser proximo de 1.
- Onde:

aptidaoParcial =

volumeAcumulado da caixa(altura * comprimento)

volumeTotal do container(altura * comprimento)

Esse resultado tem que ser proximo de 1.

Função Objetiva

- Função Objetiva é Maximizar a quantidade de caixas em um container

$$\text{Max} \sum_{i=1}^n x_i y_i$$

N= quantidade de caixas.

Algoritmo Genético

- Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, mas aleatória, que é voltada em direção ao reforço da busca de pontos de "alta aptidão", ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos).

- O que um algoritmo genético faz é criar uma população de possíveis respostas para o problema, sendo constituído pelas seguintes etapas:
- avaliação: avalia-se a aptidão das soluções (indivíduos da população) — é feita uma análise para que se estabeleça quão bem elas respondem ao problema proposto;
- seleção: indivíduos são selecionados para a reprodução. A probabilidade de uma dada solução i ser selecionada é proporcional à sua aptidão;
- cruzamento: características das soluções escolhidas são recombinadas, gerando novos indivíduos;
- mutação: características dos indivíduos resultantes do processo de reprodução são alteradas, acrescentando assim variedade à população;
- atualização: os indivíduos criados nesta geração são inseridos na população;
- finalização: verifica se as condições de encerramento da evolução foram atingidas, retornando para a etapa de avaliação em caso negativo e encerrando a execução em caso positivo.

Algoritmo Genético

Procedimento AG{

$t = 0$;

$inicia_população(P, t)avaliação(P, t)$;

$repita\ até\ (t = d)\{$

$t = t + 1;seleção_dos_pais(P, t)$;

$recombinação(P, t)$;

$mutação(P, t)$;

$avaliação(P, t)$;

$sobrevivem(P, t)\}$

}

• onde:

t - tempo atual;

d - tempo determinado para finalizar o algoritmo;

P - população

Aplicação do AG em relação a outros problema proposto

- Sistemas de classificação
- Teoria dos Jogos
- Escalonamento e grade horária

Conclusão

- Algoritmos Genéticos estão entre as mais utilizadas dentre das técnicas metaheurísticas , pela sua flexibilidade, facilidade de programação, possibilidade de busca de solução em um espaço grande de soluções e pelos estudos e aplicações já realizadas que demonstram sua eficácia (HE, CHA, 2002).

Referências

- <http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>
- <http://www.inf.ufpr.br/aurora/tutoriais/Ceapostila.pdf>