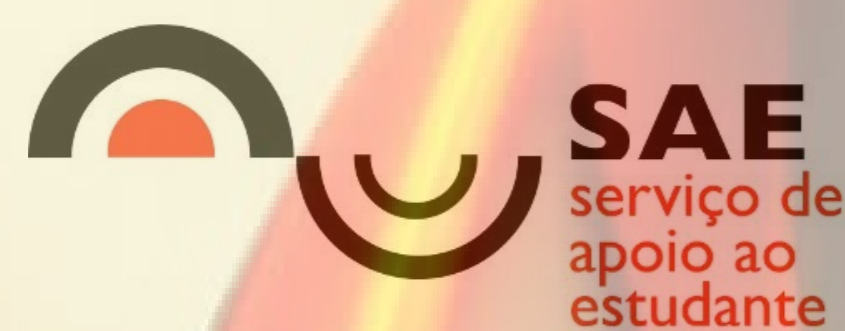




PROBLEMA DE EMPACOTAMENTO

Débora Saraiva Mendes - d135464@dac.unicamp.br
Francisco A. M. Gomes. Neto - chico@ime.unicamp.br

Palavras-chave: Empacotamento – Matlab – Algoritmos Genéticos



Introdução

O problema de empacotamento consiste em armazenar um conjunto de itens em recipientes com capacidade limitada, usando o menor número possível de recipientes.

No problema unidimensional, os itens a serem alocados possuem uma única variável, sendo que o valor dessa variável não pode ultrapassar a do recipiente. Um exemplo disso é o **problema da mochila**, no qual é preciso colocar um conjunto de itens de pesos variados em mochilas com determinado limite de peso.

Já no problema bidimensional, os recipientes e os itens possuem formato retangular, e deseja-se encaixá-los de forma a economizar o máximo de espaço possível dentro do recipiente. Um problema prático bidimensional, consiste em colocar itens de altura (ai) e largura (li) variadas e conhecidas, em contêineres também com altura (A) e largura (L) conhecidas, supondo que nenhum item tenha dimensão superior à do recipiente e que o item não possa ser rotacionado. Em ambas as dimensões do problema, a melhor solução é aquela em que se usa o menor número de recipientes. Geralmente, usa-se um método heurístico para resolver esse tipo de problema.

Metodologia

Estudamos e desenvolvemos algumas estratégias para a solução do problema unidimensional através da utilização de algoritmos de aproximação. Escolhemos e destacamos os algoritmos **gulosos** *Next Fit*, *First Fit*, e *Best Fit*, além de suas versões *Next Fit Decreasing*, *First Fit Decreasing* e *Best Fit Decreasing*, as quais a única diferença é que, antes de aplicar os comandos os itens são ordenados em forma decrescente de valor.

O algoritmo *Next Fit* aloca os itens um a um no recipiente. Quando o próximo item excede o limite do recipiente atual, o item é colocado em um recipiente novo e continua-se o processo. Já o algoritmo *First Fit* aloca os itens nos recipientes de forma que o próximo item a ser alocado pertença ao recipiente cujo valor dimensional seja o **menor** entre os já utilizados. O algoritmo *Best Fit* assemelha-se ao *First Fit*. A única diferença é que os itens são colocados nos recipientes de **maior** valor entre eles.

Para a solução do problema bidimensional, estamos estudando a aplicação de uma variante do algoritmo genético de Gonçalves e Resende. Os algoritmos genéticos são métodos baseados na teoria da evolução de Darwin, que combinam diversas soluções (cromossomos) para a criação de outras novas. Foram desenvolvidos pequenos programas para a localização do item dentro do contêiner.

Resultados

Os algoritmos gulosos mencionados foram implementados no ambiente Matlab, e realizou-se diversos testes a fim de encontrar a melhor solução para o problema da mochila. Dentre todos os algoritmos, *First Fit Decreasing* e *Best Fit Decreasing* foram os que mostraram os melhores resultados (Ver Quadro 1), pois conseguiram alocar os itens utilizando a menor quantidade de recipientes.

Quadro 1 - Resultado de teste do algoritmo Best Fit para o problema da mochila com 20 itens.

BEST FIT																				
Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
mdest=	1	1	1	2	3	4	5	2	6	7	8	4	3	6	9	1	5	10	9	11
mochila	1	2	3	4	5	6	7	8	9	10	11									
pmoc=	4.7	5	5	5	4.8	5	3.5	2.5	5	4	3									
Mochilas utilizadas:	11																			

BEST FIT DECREASING																				
item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
mdest=	1	2	3	4	5	6	7	8	9	10	9	7	8	5	6	4	1	2	3	3
mochila	1	2	3	4	5	6	7	8	9	10										
pmoc=	5	5	5	5	5	5	5	5	5	2.5										
Mochilas utilizadas:	10																			

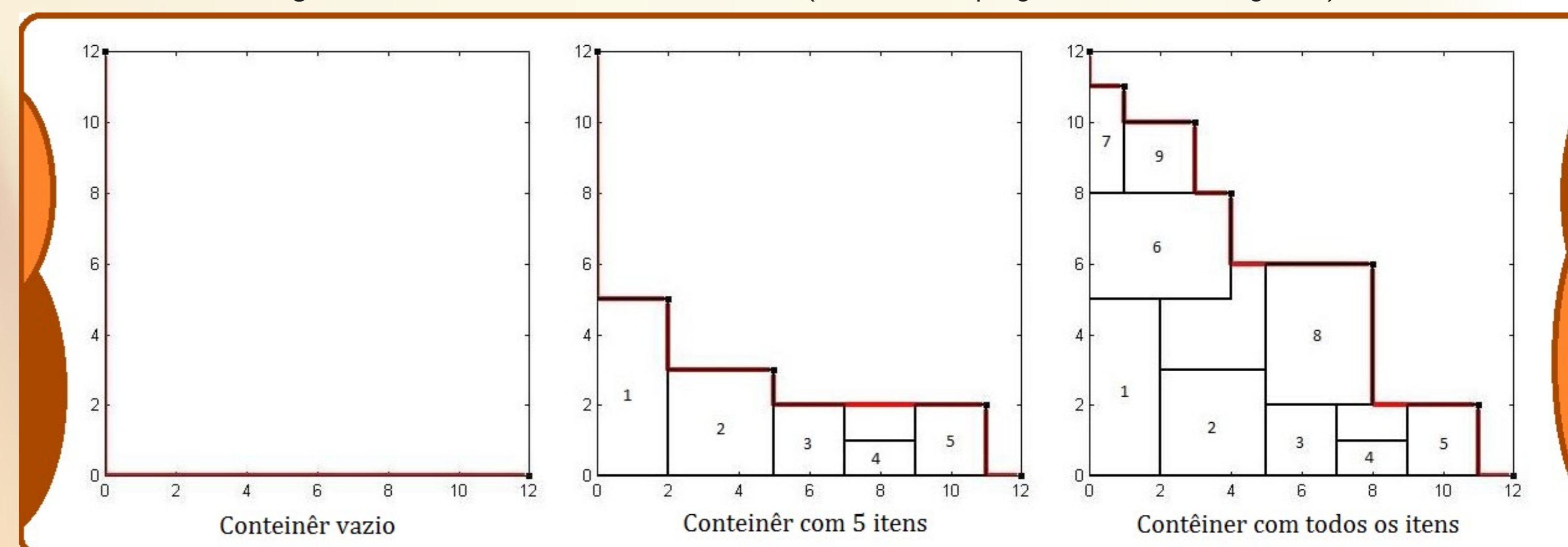
Legenda: **mdest** é a mochila destino de cada item e **pmoc** é o peso de cada mochila ao final.

Para a solução do problema prático bidimensional, numa primeira etapa, foi desenvolvido um programa que inclui os itens no contêiner e cria um gráfico com a imagem dos itens dentro do contêiner. Tal programa é composto por duas funções criadas no *Matlab*: **incluir_ponto** e **desenha_linha**. Primeiramente, o programa delimita o espaço já utilizado pelos itens no contêiner, através de uma linha vermelha (vetor contendo os pontos superiores direitos dos itens). Logo em seguida, o programa **incluir_ponto** inclui os pontos dos itens, um a um, dentro do contêiner, atualizando a linha vermelha. Ao final, o programa **desenha_linha** retorna o desenho final (Ver Figura 1).

Conclusão

Os algoritmos gulosos *First Fit Decreasing* e *Best Fit Decreasing*, destacam-se no nosso trabalho como os melhores métodos gulosos para a solução do problema de empacotamento unidimensional. Quando ao problema bidimensional, esperamos concluir a implementação do algoritmo genético até o fim desse semestre, usando o programa já desenvolvido.

Figura 1 - Desenho dos itens no contêiner (resultado do programa Teste, na Figura 2).



Referências

- [1] BERNARDI, R. - Aplicando a técnica de times assíncronos na otimização de problemas de empacotamento unidimensional. São Paulo. p. 17-27, 2001.
- [2] GONÇALVES, J. F. & RESENDE, M.G.C. – A biased random-key genetic algorithm for a 2D and 3D bin packing problem. AT&T Labs Research Technical Report, 2012.