UMA HEURÍSTICA PARA O PROBLEMA DE EMPACOTAMENTO DE BINS TRIDIMENSIONAIS

José Lassance de Castro Silva

Nei Yoshihiro Soma

Nelson Maculan

Departamento de Computação, Instituto Tecnológico de Aeronáutica
12228-900, São José dos Campos-SP, e-mail: {lassance, nysoma}@comp.ita.cta.br

Coppe/Sistemas, Universidade Federal do Rio de Janeiro
21945-970, Rio de Janeiro-RJ, e-mail: maculan@cos.ufrj.br

Abstract

The problem addressed in this paper is that of orthogonally packing a given set of box-shaped items into the minimum number of three-dimensional rectangular and equal bins. The problem is NP-hard in the strong sense and extremely difficult to be solved in the practice. We introduce a new heuristic for the problem based upon the way the items fit a given bin which is dictated by two lists: corner points and items yet to be packed. Extensive computational experiments are reported for instances with up to 90 items, and the results are compared with those obtained from the literature.

Key words: Three-Dimensional Bin Packing Problem, Lower Bound, Heuristic.

1 – Introdução

Neste trabalho, nosso principal objetivo é apresentar uma técnica que encontre boas soluções para o problema de empacotamento de bins tridimensionais, denotado na literatura como Three-Dimensional Bin Packing Problem(*3D-BPP*).

No 3D-BPP são dados um conjunto de n itens (caixas retangulares), cada um caracterizado pela largura w_j, altura h_j e comprimento d_j, para todo $j \in J=\{1, 2, ..., n\}$, e um número limitado de bins (conteineres ou caixas retangulares maiores), onde inicialmente há uma disponibilidade de pelo menos n bins idênticos e tridimensionais tendo como dimensões comuns: largura W, altura H e comprimento D.

O 3D-BPP consiste em se empacotar ortogonalmente todos os n itens no menor número possível de bins. Supomos que os itens não podem ser rotacionados, *i.e.* , que esses são empacotados com cada aresta paralela à aresta correspondente do bin, e que uma permutação da ordem na qual suas dimensões aparecem, implica na diversidade de itens, ou seja que $\{w_j, hj, d_j\} \neq \{h_j, dj, w_j\}$. Também supomos sem perda de generalidade, que as dimensões dos itens (dados de entrada) são números inteiros e positivos, satisfazendo às relações: $w_j \leq W$, $h_j \leq H$ e $d_j \leq D$, para todo $j \in J$.

O 3D-BPP aparece em uma grande quantidade de aplicações industriais, as quais variam do carregamento de cargas em aviões ao seqüenciamento (escalonamento) de tarefas em ambientes multi-processados. Sob o ponto de vista teórico, o problema é uma variação do bem conhecido e já clássico problema *Bin Packing* (BPP), *cf*. Garey e Johnson [1], o que implica, *a fortiori*, ser ele, NP-difícil no sentido forte, *cf*. Martello *et al.* [2].

Faz-se mister notar dois fatores quando da comparação em termos de resolução entre o BPP e o 3D-BPP, quais sejam:

- A resolução exata de ambos exige uma grande quantidade de recursos computacionais, sejam eles de espaço (memória) e/ou tempo (processamento máquina) e
- contrariamente ao *Bin Packing* usual (uni-dimensional), não se conhece heurística alguma, *s.m.j.*, para o 3D-BPP com garantia de desempenho, *i.e.* para o BPP, a bem conhecida heurística FFD implica em um erro assintótico de 11/9 e um tempo de processamento limitado por O(n.log n), onde n é a quantidade de itens (uni-dimensionais).

Verificamos que o 3D-BPP começa a ser pesquisado, mais intensivamente, a partir dos anos 90. Como especificado por Dowsland[3], há somente um número pequeno de trabalhos publicados com respeito aos problemas de empacotamentos tridimensionais. A primeira heurística apresentada para o 3D-BPP aparece em Scheithauer[4]. Chen, Lee e Shen[5] considerando uma generalização do problema, onde os bins podem ter diversos tamanhos, formularam o modelo matemático através de um problema de Programação Linear Inteira Mista.

Ainda em termos heurísticos, Pisinger, Faroe e Zachariasen[6] adaptaram ao 3D-BPP uma nova metodologia de aproximação para problemas da otimização combinatória, *i.e.* Busca Local Direcionada GLS (Guided Local Search). Das meta-heurísticas bastante utilizadas para resolver problemas de otimização tais como Simulated Annealing(SA), Tabu Search(TS), Genetic Algorithms(GA), Hill-Climbing(HC), etc, encontramos na literatura apenas uma aplicação de GA, desenvolvida por Corcoran e Wainwright[7], mas resolvendo o caso de preenchimento de itens em faixas (bi-dimensionais)

Em Martello, Pisinger e Vigo[2] tem-se o primeiro algoritmo exato para o problema (3D-BPP), assim como para o do Carregamento de Conteineres (Container Loading Problem), por meio da abordagem branch-and-bound.

O ataque ao 3D-BPP sugerido aqui baseia-se em uma abordagem heurística a qual apresenta bons resultados práticos quando comparada com os métodos sugeridos na literatura, todavia, não se tem a garantia desempenho de pior caso e/ou caso assintótico.

A organização do artigo segue à: Na Seção 2, apresentaremos as principais idéias da heurística, daqui para frente denominada por Heurística do Volume (HV). Na Seção 3, descrevemos os experimentos computacionais, com os resultados. Finalizamos nosso trabalho com a apresentação dos melhoramentos ora sendo desenvolvidos e ainda em fase de implementação, Seção 4 enquanto na Seção 5 tem-se o material bibliográfico consultado.

2 – A Heurística do Volume(HV)

HV é baseada em um princípio derivado do algoritmo exato proposto em Martello et al. [2], conhecido como sendo 3D-Corner, onde, todas as possíveis posições para colocar um item ainda não alocado, são consideradas. *Pontos de corner* (PC) foi o nome dado a estas posições, que é um ponto no espaço tridimensional com três coordenadas inteiras e não negativas (x , y e z). Supomos que a origem do sistema de coordenadas seja o canto inferior esquerdo do fundo do bin.

Seja (x_p, y_p, z_p) as coordenadas de um ponto de corner p: um item j não pode ser colocado em p caso $x_p+w_j > W$ ou $y_p+h_j > H$, ou ainda, se $z_p+d_j > D$. Se p é um ponto de corner então nenhum item de I, conjunto dos itens já empacotados, poderá ter alguma interseção à direita de p, ou acima de p, ou defronte a p. Em outras palavras, os itens de I definem duas regiões, uma região chamada de "envelope", composta pelos itens de I, e uma outra região "vazia" onde os itens de J\I ainda podem ser fisicamente alocados. Na Figura 1 tem-se um conjunto de itens já alocados e os respectivos pontos de corner.

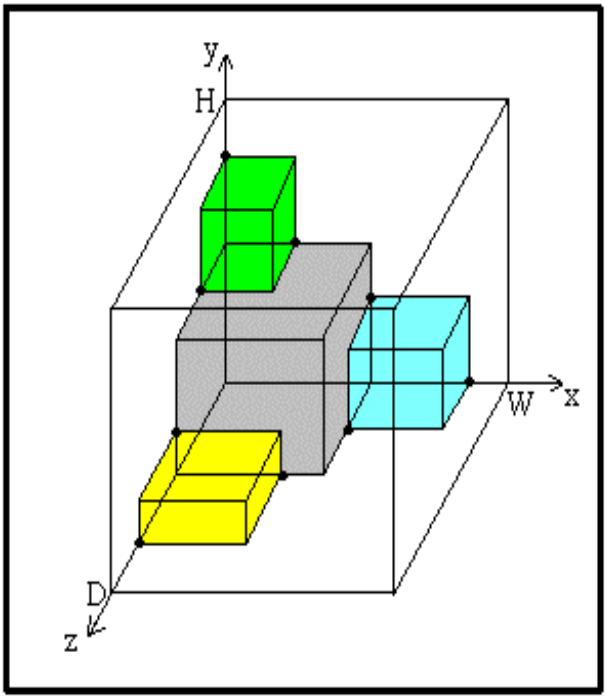

Figura 1 – Itens já empacotados com seus PC's(pontos preto).

A heurística do volume (HV) leva em conta a relação entre o espaço do envelope (região formada pelos itens já empacotados dentro de um bin), o espaço não ocupado dentro daquele envelope e os itens que ainda poderão ser empacotados no bin sendo analisado. A idéia é a de que o volume ocupado resultante seja o maior possível. Este procedimento é feito para cada um dos bins abertos e com os itens que ainda não foram empacotados. Um bin é considerado *aberto* caso à ele seja possível alocar itens ainda não empacotados e um bin é *fechado* quando nenhum item puder a ele ser alocado. O procedimento é repetido de maneira recursiva até que a totalidade dos itens tenha sido considerada.

A formulação recursiva é dada da forma: dado um conjunto I_k de itens já empacotados dentro do bin k, o próximo item a ser *potencialmente* alocado ao bin k, será aquele cujo o volume não ocupado dentro do envelope, formado pela soma do volume de I e deste item, seja o menor possível, dentre todos os itens que ainda estão para ser empacotados. Caso não seja este o caso, considera-se o primeiro bin aberto satisfazendo a esta restrição de volume, caso contrário mais um bin é aberto. Assim, sempre que um item é alocado a um

bin, segue-se o procedimento de atualização do conjunto de itens não empacotados, da determinação dos novos PC's e do cálculo do novo volume não ocupado do envelope. Há ainda um ulterior critério de fechamento de um bin, e este ocorre caso a solução encontrada atinja ao mesmo valor de um limitante inferior, no caso, escolheu-se o limitante de Martello, Vigo e Pinsinger (L_2), *cf*. Martello *et al*. [2].

A heurística proposta utiliza uma quantidade de recursos computacionais claramente polinomial, mais ainda, no pior caso tem-se $O(|J|^3) = O(n^3)$ de tempo de execução e $O(|J|^2) = O(n^2)$ de espaço (memória). Limitantes esses, que para os valores encontrados em aplicações industriais são bastante adequados.

3. Experimentos Computacionais

Um grande número de experimentos computacionais foram realizados, para uma melhor e mais fidedigna avaliação da heurística proposta. Comparamos HV com a heurística H1, desenvolvido por Martello et al.[2], HL e SA.

H1 é baseada no princípio da construção de camadas, conhecida como o método das prateleiras bi-ortogonais. HL e SA estão propostas em Silva *et al.* [9]. SA é uma meta-heurística baseada na meta-heurística simulated annealing. HL é uma heurística baseada na forma de como os itens se encaixam em PC's, em um bin dado, determinado por duas listas ordenadas: pontos de corner e itens, ainda não empacotados.

As classes de testes foram definidas como em Berkey e Wang[8] e Martello *et al.* [2]. As seguintes instâncias foram consideradas para problemas com uma quantidade de itens variando de 10 à 90.

- **Classe 1:** A maioria dos itens são muitos altos e profundos.
- **Classe 2:** A maioria dos itens são muitos largos e profundos.
- **Classe 3:** A maioria dos itens são muitos altos e largos.
- **Classe 4:** A maioria dos itens têm grandes dimensões.
- **Classe 5:** A maioria dos itens têm pequenas dimensões.
- **Classe 6:** Instâncias geradas aleatoriamente em um intervalo pequeno.
- **Classe 7:** Instâncias geradas aleatoriamente em um intervalo médio.
- **Classe 8:** Instâncias geradas aleatoriamente em um intervalo grande.
- **Classe 9:** Instâncias tendo uma solução conhecida com três bins.

A amostra considerada para cada uma das classes tem tamanho 10 (para cada um dos valores de n). Os algoritmos: H1, SA, HL and HV, foram executados em um Pentium III (600 MHz com 128 Mb de RAM) e o código foi implementado em Ansi C.

As Tabelas 1 e 2 apresentam o desvio médio em relação ao limite inferior L_2 e as médias dos tempo de CPU. O desvio é definido como $(Z - L_2)/L_2$, onde Z é o número mínimo de bins encontrado pelos algoritmos para empacotar todos os itens, em cada uma das instâncias.

Classe		10	15	20	25	30	35	40	45	50	60	70	80	90	TOTAL
1	H1	0,24	0,33	0,32	0,30	0,31	0,32	0,37	0,30	0,33	0,36	0,30	0,25	0,22	3,95
	SA	0,13	0,27	0,21	0,22	0,20	0,23	0,29	0,24	0,28	0,30	0,25	0,24	0,21	3,07
	HV	0,11	0,24	0,23	0,23	0,22	0,18	0,21	0,20	0,23	0,23	0,20	0,17	0,14	2,59
	HL	0,08	0,27	0,22	0,20	0,19	0,20	0,25	0,19	0,25	0,23	0,21	0,18	0,16	2,63
2	H1	0,30	0,49	0,22	0,42	0,41	0,26	0,30	0,30	0,37	0,24	0,27	0,24	0,22	4,04
	SA	0,15	0,29	0,19	0,30	0,31	0,20	0,25	0,25	0,26	0,20	0,24	0,23	0,23	3,10
	HV	0,15	0,31	0,19	0,28	0,25	0,18	0,19	0,19	0,23	0,16	0,18	0,14	0,15	2,60
	HL	0,15	0,29	0,19	0,28	0,25	0,19	0,22	0,21	0,24	0,16	0,18	0,17	0,17	2,70
3	H1	0,41	0,53	0,39	0,28	0,35	0,35	0,33	0,29	0,35	0,27	0,28	0,23	0,22	4,28
	SA	0,36	0,40	0,32	0,23	0,29	0,28	0,27	0,23	0,31	0,23	0,25	0,23	0,22	3,62
	HV	0,36	0,40	0,32	0,19	0,26	0,23	0,23	0,20	0,25	0,15	0,20	0,17	0,15	3,11
	HL	0,36	0,40	0,32	0,22	0,27	0,25	0,25	0,20	0,25	0,17	0,21	0,17	0,17	3,24
4	H1	0,03	0,03	0,03	0,05	0,05	0,04	0,04	0,05	0,04	0,03	0,06	0,03	0,05	0,53
	SA	0,03	0,01	0,01	0,00	0,04	0,01	0,02	0,03	0,03	0,01	0,02	0,00	0,02	0,23
	HV	0,03	0,01	0,01	0,02	0,05	0,03	0,02	0,03	0,03	0,02	0,03	0,03	0,03	0,34
	HL	0,03	0,01	0,01	0,02	0,04	0,03	0,03	0,03	0,04	0,02	0,04	0,03	0,03	0,36
5	H1	0,45	0,71	0,51	0,56	0,47	0,44	0,68	0,67	0,65	0,67	0,47	0,48	0,37	7,13
	SA	0,40	0,54	0,43	0,48	0,37	0,40	0,66	0,61	0,59	0,65	0,52	0,62	0,48	6,75
	HV	0,30	0,47	0,30	0,34	0,30	0,26	0,43	0,42	0,42	0,42	0,34	0,34	0,25	4,59
	HL	0,30	0,42	0,30	0,40	0,30	0,22	0,43	0,44	0,37	0,41	0,32	0,31	0,25	4,47
6	H1	0,47	0,47	0,32	0,32	0,36	0,39	0,30	0,31	0,37	0,30	0,31	0,25	0,22	4,39
	SA	0,33	0,29	0,20	0,23	0,34	0,35	0,29	0,33	0,41	0,37	0,40	0,37	0,35	4,26
	HV	0,28	0,32	0,17	0,20	0,20	0,22	0,15	0,18	0,21	0,17	0,17	0,16	0,13	2,56
	HL	0,28	0,29	0,20	0,21	0,20	0,22	0,17	0,20	0,24	0,20	0,21	0,19	0,14	2,75
7	H1	0,83	0,61	0,62	0,69	1,07	0,78	0,81	0,82	0,60	0,71	0,79	0,55	0,52	9,40
	SA	0,58	0,38	0,34	0,42	0,69	0,61	0,59	0,70	0,59	0,68	0,78	0,67	0,64	7,67
	HV	0,48	0,39	0,23	0,33	0,56	0,47	0,43	0,51	0,39	0,43	0,50	0,37	0,33	5,42
	HL	0,43	0,29	0,27	0,31	0,47	0,44	0,44	0,48	0,41	0,47	0,54	0,37	0,32	5,24
8	H1	0,68	0,55	0,33	0,58	0,58	0,44	0,70	0,60	0,57	0,38	0,40	0,42	0,47	6,70
	SA	0,40	0,35	0,28	0,34	0,43	0,37	0,56	0,57	0,58	0,42	0,50	0,53	0,62	5,95
	HV	0,35	0,25	0,21	0,29	0,30	0,26	0,44	0,34	0,39	0,19	0,28	0,29	0,35	3,94
	HL	0,40	0,25	0,21	0,27	0,32	0,30	0,42	0,42	0,38	0,23	0,32	0,30	0,36	4,18
9	H1	0,23	0,67	0,50	1,07	0,82	1,42	1,25	1,18	1,60	2,40	1,70	1,90	2,70	17,44
	SA	0,05	0,47	0,72	1,00	0,68	1,47	1,30	1,45	1,73	2,90	1,80	2,42	3,48	19,47
	HV	0,05	0,47	0,25	0,68	0,43	0,72	0,53	0,62	0,75	1,38	0,75	0,92	1,62	9,17
	HL	0,05	0,47	0,40	0,68	0,40	0,92	0,73	0,77	1,00	1,77	0,92	1,02	1,67	10,80

Tabela 1 – Desvio médio de L_2 com respeito ao valor Z da solução encontrada.

Verificamos pela Tabela 1, que HL e HV apresentaram os melhores resultados, tendo HV alcançado em média 65,5%, 64,3%, 72,6%, 64,1%, 64,3%, 58,3%, 57,6%, 58,8% e 52,5% dos resultados encontrados por H1 para as classes 1, 2, 3, 4, 5, 6, 7, 8 e 9, respectivamente. Isto representa um índice médio de desvio de 66,1% e 58,2%, dos resultados alcançados por HV em relação a H1, nas instâncias de Martello et al. e Berkey e Wang, respectivamente, que comprova que HV produz soluções bem melhores que H1.

Na média, HL teve um desempenho semelhante a HV. Na classe 4, SA teve, todavia, um desempenho melhor que os outros algoritmos. Analisamos a Tabela 2 e concluímos que HL, H1 e HV são bastante rápidas, *i.e.* apresentam soluções em tempo computacional razoável, porém o bom desempenho de HV está na qualidade das suas soluções

apresentadas para o problema. Na Tabela 2, vemos também que o tempo computacional consumido por SA é bem maior que o tempo gasto por H1, HV e HL, que já era esperado.

Classe		10	15	20	25	30	35	40	45	50	60	70	80	90	TOTAL
	H1	0,01	0,00	0,00	0,00	0,00	0,02	0,05	0,08	0,09	0,19	0,26	0,24	0,30	1,24
1	SA	0,08	0,53	1,85	2,88	5,73	9,92	14,73	21,79	37,53	49,98	72,76	105,86	147,78	471,42
	HV	0,00	0,00	0,01	0,02	0,04	0,07	0,12	0,18	0,25	0,45	0,80	1,24	1,85	5,03
	HL	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,03	0,03	0,04	0,20
	H1	0,00	0,00	0,00	0,01	0,01	0,04	0,05	0,10	0,13	0,23	0,23	0,27	0,30	1,37
2	SA	0,07	0,60	1,34	3,53	6,41	9,31	15,12	18,63	33,12	49,67	74,94	103,20	138,23	454,17
	HV	0,00	0,01	0,01	0,03	0,05	0,07	0,12	0,16	0,21	0,46	0,78	1,24	1,79	4,93
	HL	0,00	0,01	0,00	0,01	0,01	0,01	0,02	0,02	0,02	0,02	0,03	0,04	0,04	0,23
	H1	0,00	0,00	0,00	0,00	0,01	0,02	0,08	0,08	0,12	0,21	0,25	0,27	0,30	1,34
3	SA	0,14	0,76	1,58	4,02	6,70	11,26	14,23	21,26	30,96	48,71	73,20	100,69	148,85	462,36
	HV	0,00	0,01	0,02	0,02	0,05	0,07	0,12	0,18	0,24	0,44	0,74	1,14	1,68	4,71
	HL	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,02	0,03	0,04	0,04	0,22
	H1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,01	0,01	0,01	0,05
4	SA	0,01	0,04	0,09	0,82	2,30	3,21	3,46	4,88	10,19	14,15	39,27	75,19	121,71	275,32
	HV	0,00	0,00	0,00	0,01	0,02	0,02	0,05	0,06	0,11	0,14	0,48	0,96	1,50	3,35
	HL	0,00	0,00	0,00	0,01	0,00	0,01	0,01	0,01	0,01	0,01	0,02	0,03	0,04	0,15
	H1	0,00	0,00	0,01	0,00	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,12
5	SA	0,15	1,05	2,96	8,39	15,87	20,64	27,09	46,13	51,75	101,15	130,64	192,29	285,21	883,32
	HV	0,01	0,04	0,12	0,30	0,71	1,68	3,15	5,52	6,67	21,96	37,77	65,68	92,41	236,02
	HL	0,01	0,01	0,01	0,01	0,02	0,02	0,02	0,02	0,02	0,04	0,05	0,06	0,07	0,36
	H1	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,03	0,07
6	SA	0,13	0,80	1,53	5,02	8,74	14,92	17,69	20,50	28,04	42,77	65,93	92,91	128,19	427,17
	HV	0,00	0,01	0,02	0,04	0,07	0,13	0,15	0,26	0,44	0,63	0,80	1,14	1,64	5,33
	HL	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,02	0,03	0,03	0,18
	H1	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,08
7	SA	0,18	0,80	1,90	6,79	9,29	20,52	19,52	31,80	35,05	68,18	95,47	138,14	186,77	614,41
	HV	0,01	0,03	0,06	0,22	0,52	0,60	0,95	1,88	2,56	4,31	7,43	12,22	15,76	46,55
	HL	0,00	0,01	0,01	0,01	0,01	0,02	0,02	0,02	0,02	0,03	0,04	0,04	0,05	0,28
	H1	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,01	0,01	0,02	0,03	0,08
8	SA	0,18	0,58	1,88	4,76	8,89	17,89	22,90	30,99	36,73	56,00	84,97	130,48	235,99	632,24
	HV	0,00	0,01	0,03	0,10	0,20	0,43	0,59	0,77	1,67	2,41	4,31	6,66	9,50	26,68
	HL	0,01	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,03	0,02	0,03	0,04	0,05	0,27
	H1	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,01	0,01	0,01	0,05
9	SA	0,04	0,87	2,51	6,45	8,71	15,98	27,26	40,38	59,55	40,54	94,84	150,53	213,73	661,39
	HV	0,01	0,01	0,01	0,03	0,08	0,23	0,45	0,77	1,36	2,48	13,57	33,08	41,49	93,57
	HL	0,00	0,00	0,00	0,01	0,00	0,01	0,02	0,02	0,02	0,02	0,05	0,06	0,07	0,28

Tabela 2 – Tempo médio requerido de CPU para encontrar a solução das heurística.

4. Conclusões

Nossos experimentos computacionais baseados em problemas da literatura, indicam que HV pode ser aplicada ao 3D-BPP com sucesso, gerando boas soluções. A presente heurística faz parte de um trabalho maior, onde se está estudando o problema da estabilidade dos empacotamentos e, que *s.m.j.*, é inédito na literatura.

5. Bibliografia

[1] R. Garey e D.S. Johnson, Computers and Intractability: A guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[2] S. Martello, D. Pisinger, and D. Vigo. The three-dimensional bin packing problem. *Operations Research*, march-april, 2000.

[3] W. B. Dowsland. Three-dimensional packing solution approaches and heuristics development. *International Journal of Production Research*, 29(8): 1673-1685, 1991.

[4] G. Scheithauer. A three-dimensional bin packing algorithm. *J. Inform. Process. Cybernet.*, 27:263-271, 1991.

[5] C. S. Chen, S. M. Lee, and Q. S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80:68-76, 1995.

[6] D. Pisinger, O. Faroe and M. Zachariasen. Guided Local Search for the three-dimensional bin packing problem. *Techinical Report*. University of Copenhagen, Denmark, 13, 1999.

[7] A L. Corcoran and R.L. Wainwright. A genetic algorithm for packing in three dimensions. *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*, 1021-1030, 1992.

[8] J. O. Berkey and P. Y. Wang. Two dimensional finite bin packing algorithms. *Journal of the Operational Research Society*, 38:423-429, 1987.

[9] J.L.C. Silva, N.Y. Soma e N. Maculan. A heuristic and a simulated annealing approach for the three-dimensional bin packing problem. Aceito para apresentação na *IV Meta-heuristic International Conference*, Porto, Portugal, julho, 2001.