

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**METAHEURÍSTICA TABU APLICADA AO PROBLEMA DE
CARREGAMENTO DE CONTÊINER COM CAIXAS IDÊNTICAS**

GUILHERME IZIDORO POLI

Orientador(a): Prof(a). Dr(a). Vitória Pureza

Agência Financiadora: CAPES

São Carlos

2009

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**METAHEURÍSTICA TABU APLICADA AO PROBLEMA DE
CARREGAMENTO DE CONTÊINER COM CAIXAS IDÊNTICAS**

Guilherme Izidoro Poli

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.
Orientadora: Prof(a). Dr(a). Vitória Pureza

Agência Financiadora: CAPES

São Carlos

2009

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

P766mt

Poli, Guilherme Izidoro.

Metaheurística tabu aplicada ao problema de carregamento de contêiner com caixas idênticas / Guilherme Izidoro Poli. -- São Carlos : UFSCar, 2009.
94 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2009.

1. Pesquisa operacional. 2. Sistema de unitização de cargas. 3. Modelagem matemática. 4. Heurística. I. Título.

CDD: 658.4034 (20^a)



FOLHA DE APROVAÇÃO

Aluno(a): Guilherme Izidoro Poli

DISSERTAÇÃO DE MESTRADO DEFENDIDA E APROVADA EM 25/09/2009 PELA
COMISSÃO JULGADORA:

Prof.ª Dr.ª Vitória Maria Miranda Pureza
Orientador(a) PPGE/UFSCar

Prof. Dr. Reinaldo Morabito Neto
PPGE/UFSCar

Prof. Dr. Horacio Hideki Yanasse
LAC/INPE

Prof. Dr. Roberto Antonio Martins
COORDENADOR DO PPGE

AGRADECIMENTOS

À Prof.(a) Vitória Pureza pela orientação e pela amizade.

Aos meus familiares pelo incentivo nos momentos difíceis.

Aos colegas do grupo de Modelagem do DEP-UFSCAR pelas dicas e pela convivência.

Ao Prof. Fábio Molina pelas dicas de implementação.

Aos Profs. Reinaldo Morabito e Horácio Yanasse pelo tempo dedicado e pelas sugestões na dissertação.

À CAPES pelo apoio financeiro e a todos que de alguma forma colaboraram com este trabalho.

ABSTRACT

Intermodal transport, that is, the integration between different transport modes was facilitated by the use of containers. The stowage devices in the breakdown of the cargo is an important logistics activity, since the total volume actually used can affect significantly the cost of transporting the products.

This approach addresses the Container Loading Problem, which more generally consists in arranging items (for example, products packaged in boxes) of various sizes within larger objects (for example, containers) with the maximum use of the available volume. In particular, it is considered the special case where the boxes to be arranged are rectangular and identical (single sized) and there is only a single container. In order to tackle these problems, the 2006's Tabu Search heuristic by Pureza and Morabito, originally proposed for the Manufacturer's Pallet Loading Problem, was extended. From an initial solution generated by a block heuristic, moves are applied to selected blocks in order to change their box orientation and/or to expand them in one of the six directions, which result in the reduction, elimination and creation of other blocks. Criteria for stability of the load are also addressed in this work. Computational experiments using a set of instances in the literature demonstrate the performance of the proposed approach.

Key words: Intermodal Transport; Container Loading; Single Size of Box; Tabu Search.

RESUMO

O transporte intermodal, ou seja, a integração entre diversos modos de transporte foi facilitada com o uso de contêineres. O acondicionamento da carga em dispositivos de unitização de cargas é uma importante atividade logística, uma vez que o volume total efetivamente utilizado pode afetar de maneira significativa o custo de transporte dos produtos.

Este trabalho aborda o problema do carregamento de contêineres, cuja forma mais geral consiste em arranjar itens (por exemplo, produtos embalados em caixas) de vários tamanhos dentro de objetos maiores (por exemplo, contêineres) com máximo aproveitamento do volume disponível. Em particular, é considerado o caso especial em que as caixas a serem arranjadas são retangulares e idênticas e dispõe-se de apenas um único contêiner. Com vistas à resolução destes problemas, estendemos a heurística de busca tabu de Pureza e Morabito (2006), originalmente projetada para o problema de carregamento de paletes do produtor. Partindo-se de uma solução inicial gerada por uma heurística de blocos, são realizados movimentos de troca de orientação e/ou expansão de blocos selecionados, e que resultam na diminuição, eliminação e criação de outros blocos. Critérios de estabilidade da carga são também abordados neste trabalho. Experimentos computacionais utilizando um conjunto de instâncias da literatura demonstram o desempenho da abordagem proposta.

Palavras-chave: Transporte Intermodal; Carregamento de Contêiner; Caixas idênticas; Busca Tabu.

LISTA DE FIGURAS

FIGURA 2.1 – Padrão de corte de itens de comprimento l_i ($i=1,\dots$, número de tipos de itens) em uma bobina de comprimento L	6
FIGURA 2.2 – Padrão de corte de peças retangulares com dimensões (l,w) em uma chapa de dimensões (L,W)	7
FIGURA 2.3 – Padrão de carregamento de caixas retangulares de dimensões (l_i, w_i, h_i) ($i=1,\dots$, número de tipos de caixas), em um contêiner de dimensões (L,W,H)	8
FIGURA 2.4 – Tipos básicos.....	16
FIGURA 3.2 - Pilha de caixas.....	28
FIGURA 3.3 – Exemplo de padrão de corte guilhotinado.....	29
FIGURA 3.4 – Cubóide com 8 caixas.	29
FIGURA 3.5 – Prisma.	37
FIGURA 3.6 – Estabilidade vertical segundo Gehring e Bortfeldt (1997): (a) Pilha aceitável (estável), (b), e (c) Pilhas não aceitáveis (instáveis).	41
FIGURA 3.7 – Carregamento estável verticalmente segundo critério 1 de Liu e Hsiao (1997).	42
FIGURA 3.8 – Estabilidade vertical segundo critério 2 de Liu e Hsiao (1997): (a) carregamento estável; (b) carregamento instável.	42
FIGURA 4.1 – Exemplos de padrões de carregamento para (a) PCPD, e (b) PCPP (Fonte: MORABITO e PUREZA, 2008).	44
FIGURA 4.2 – Um padrão de carregamento com 4 blocos para a instância $(L,W,l,w)=(61,44,9,7)$	45
FIGURA 4.3 - (a) padrão guilhotinado, (b) corte (em negrito) e padrão não-guilhotinado de 1ª ordem, (c) padrão não-guilhotinado de ordem superior. (Fonte: PUREZA e MORABITO, 2006).....	46
FIGURA 4.4 – Passos do algoritmo PM para o PCPP.....	47
FIGURA 4.5 – Direções de expansão.....	48
FIGURA 4.6 – Expansão de um bloco para a direita: (a) 2 camadas da face expandidas; (b) face inteira expandida; (c) bloco ativo é dividido em até 3 outros blocos.....	49
FIGURA 4.7 – Quatro possíveis configurações das áreas remanescentes do bloco passivo j com a expansão para a direita do bloco ativo i . (FONTE: PUREZA e MORABITO, 2006).....	50

FIGURA 4.8 – (a) Padrão de carregamento com 51 caixas (iteração it); (b) aplicação de deslizamento no padrão em (a); (c) padrão de carregamento com 52 caixas após dois movimentos (iteração $it + 2$). (Adaptada de PUREZA e MORABITO, 2006).	52
FIGURA 4.9 – Adição da área da tira de espessura e entre os blocos j_1 e j_3 à área do bloco j_1 . (FONTE: PUREZA e MORABITO, 2006).	52
FIGURA 4.10 – Opções de carregamento do chão do contêiner para a instância $(L,W,H,l,w,h)=(50,50,50,9,7,5)$: (a) caixas sobre a face (l,w) , (b) caixas sobre a face (w,h) , (c) caixas sobre a face (l,h) .	53
FIGURA 4.11 – As 6 direções de expansão em um carregamento tridimensional.	54
FIGURA 4.12 – As 4 configurações adotadas dos volumes remanescentes de blocos passivos.	56
FIGURA 4.13 – Vista superior de um carregamento, antes e durante a aplicação dos procedimentos de divisão e deslizamento de blocos.	57
FIGURA 4.14 – União dos blocos i e j .	58
FIGURA 4.15 – Passos da adaptação proposta (algoritmo TSI) para o PCCCI.	60
FIGURA 5.1 – Interface gráfica da implementação.	61
FIGURA 5.2 – Vistas isométricas de carregamentos obtidos pelo algoritmo TSI : (a) Instância 2, (b) Instância 5, (c) Instância 8 e (d) Instância 14.	65
FIGURA 5.3 – Número de caixas do padrão inicial (s_0) vs. tamanho da lista ($tlista$).	67
FIGURA 5.4 – Passos do procedimento PE.	70
FIGURA 5.5 – Estabilidade critério M1.	72
FIGURA 5.6 – Estabilidade critério M2.	73
FIGURA A.1 – Orientação dos blocos	83
FIGURA A.2 – Posição do bloco no contêiner.	84
FIGURA A.3 – Padrão de carregamento.	85
FIGURA A.4 – Aplicação do movimento à direita: $numx=1$.	86
FIGURA A.5 – Aplicação do movimento à direita: $numx=2$.	87
FIGURA A.6 – Aplicação do movimento de deslizamento à direita: <i>iteração 0</i> .	92
FIGURA A.7 – Aplicação do movimento de deslizamento à direita: <i>iteração 1</i> .	94
FIGURA A.8 – Aplicação do movimento de deslizamento à direita: <i>iteração 2</i> .	94

LISTA DE TABELAS

Tabela 2.1 – Tipos intermediários/minimização de entradas.....	17
Tabela 2.2 – Tipos intermediários/maximização de saídas.....	17
Tabela 5.1 – Resultados computacionais: Número de caixas no carregamento.	63
Tabela 5.2 – Frequência, média e variância de obtenção da solução provida por <i>TS1</i> com maior número de caixas.....	64
Tabela 5.3 – Resultados computacionais: Porcentagem de utilização do volume do contêiner.....	68
Tabela 5.4 – Resultados computacionais: Estabilidade do carregamento.	71
Tabela A.1 – Valores de tamanho de lista tabu (<i>tlist</i>) utilizados.	82

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Justificativa	2
1.2	Objetivos	3
1.3	Estrutura do trabalho	3
2	PROBLEMAS DE CORTE E EMPACOTAMENTO (PCE)	5
2.1	Caracterização de DYCKHOFF e FINKE (1992)	6
2.2	Tipologia de DYCKHOFF (1990)	12
2.3	Tipologia de WASCHER et al. (2007)	14
3	O PROBLEMA DE CARREGAMENTO DE CONTÊINERES	19
3.1	Modelagem Matemática	23
3.2	Métodos Exatos	26
3.3	Métodos Heurísticos	27
3.3.1	Métodos Heurísticos para Carregamentos Heterogêneos	30
3.3.2	Métodos Heurísticos para Carregamentos Homogêneos	36
3.4	Estabilidade do Carregamento	40
3.4.1	Carregamento heterogêneo	40
3.4.2	Carregamento homogêneo	41
4	UM ALGORITMO DE BUSCA TABU PARA O PROBLEMA DE CARREGAMENTO DE CONTÊINER COM CAIXAS IDÊNTICAS	43
4.1	O Problema do Carregamento de Paletes	43
4.2	O Algoritmo de Pureza e Morabito	46
4.2.1	Avaliação de movimentos candidatos	50
4.2.2	Elementos de restrição tabu	51
4.2.3	Recursos adicionais	51
4.3	A Extensão Proposta	52
4.3.1	Passo 1 – Geração do Padrão Inicial	53
4.3.2	Passo 3.1 - Movimentos de expansão	54
4.3.3	Passo 3.1b - Movimentos adicionais	55
4.3.4	Reconfiguração de volumes remanescentes	55
4.3.5	Recursos Adicionais	56
5	EXPERIMENTOS COMPUTACIONAIS	61
5.1	Qualidade da Solução e Tempo Computacional	62
5.2	Robustez e o tamanho da lista tabu	66
5.4	Sensibilidade à solução inicial	68
5.5	Estabilidade	69
6	CONCLUSÕES E PERSPECTIVAS DE PESQUISA FUTURA	74
7	REFERÊNCIAS BIBLIOGRÁFICAS	75
	APÊNDICE A	82
8.1	Tamanhos de lista tabu	82
8.2	Estruturas de dados	82
8.2.1	Armazenagem	83
8.2.2	Cálculo e realização de movimentos	85

8.2.3 Deslizamento de blocos	89
------------------------------------	----

1 INTRODUÇÃO

Nas últimas décadas, verifica-se um crescente interesse na aplicação de técnicas de pesquisa operacional na resolução de problemas da cadeia logística (BISCHOFF e DOWSLAND, 1982; LEE *et al.*, 1993; ARNTZEN *et al.*, 1995; LEACHMAN *et al.* 1996; CAMM *et al.* 1997 e LIN *et al.*, 2000). Este fenômeno é em parte devido ao aumento nos custos de estocagem e distribuição, o que levou à necessidade premente do uso racional dos recursos disponíveis. Mais recentemente, muitas empresas se inseriram no mercado global, e questões de custo, rapidez e preço, tornaram-se aspectos primordiais para sua sobrevivência neste ambiente altamente competitivo.

No que diz respeito a atividades logísticas, o transporte é a atividade mais importante porque absorve, em média, de um a dois terços dos custos logísticos (BALLOU, 1993). Neste contexto, a *unitização das cargas* tornou-se uma estratégia eficiente para transportar as cargas ao longo da cadeia de suprimentos. A unitização de cargas consiste no carregamento de caixas sobre paletes ou contêineres de forma que a configuração final se conforme exatamente com dimensões especificadas de comprimento, largura e altura. Estas dimensões são geralmente selecionadas para maximizar o espaço utilizado nos veículos selecionados para o transporte dos produtos.

O transporte intermodal, ou seja, a integração entre os diversos modos de transporte foi facilitada com o uso de contêineres. Contêineres são grandes caixas de metal de dimensões padronizadas. De acordo com Morabito (1992), os volumes unitários a serem carregados dentro dos contêineres podem estar soltos ou paletizados para facilitar a estofagem (carregamento do contêiner) e a desova ou descarregamento do contêiner, utilizando empilhadeiras.

Em linhas gerais, o problema do carregamento de contêineres consiste em carregar caixas de produtos, na grande maioria dos casos, retangulares. Admite-se em geral que as caixas estão disponíveis em grandes quantidades e que devem ser arranjadas ortogonalmente, isto é, com os seus lados paralelos aos lados do contêiner. Sob a perspectiva da pesquisa operacional, este problema é classificado como um *problema de empacotamento* cujo principal objetivo é a otimização de espaço utilizado pela carga (MORABITO, 1992).

As principais vantagens do uso do contêiner são: diminuição dos custos de carga, descarga e manipulação nos terminais, e diminuição de perdas por danificação e furto. Uma função importante dos contêineres é naturalmente a de facilitar a transferência dos

materiais de um modo de transporte a outro com um mínimo de manipulação ou, pelo menos, com manipulação mecanizada (MAGEE, 1977). No empacotamento da carga, um grau mínimo de estabilidade e de distribuição de peso no contêiner (DAVIES e BISCHOFF, 1999) devem ser considerados. Muitos trabalhos tratam da estabilidade estática da carga, considerando a estabilidade apenas vertical ou horizontal (GEHRING e BORTFELDT, 1997; MORABITO e ARENALES, 1997; DAVIES e BISCHOFF, 1999; SILVA e SOMA, 2003), no entanto, a literatura é escassa no que diz respeito à estabilidade quando o contêiner é movimentado pelos diferentes modais de transporte.

As principais desvantagens do contêiner são seu custo e peso, a necessidade de consolidação de carga, a programação e manipulação do seu retorno vazio, ainda que alguns contêineres possam ser desmontados no seu retorno (MORABITO, 1992; CHOONG *et al.*, 2002). De acordo com Araújo (2006), a organização de cargas em contêineres, seja ela paletizada ou distribuída em caixas, é uma das tarefas mais complexas em empresas que tem no transporte de carga um fator logístico importante e de alto custo.

Neste trabalho tratamos do caso especial de carregamento de contêiner em que as caixas a serem arranjadas são retangulares e idênticas, visando otimizar a utilização do volume do contêiner, ou seja, maximizar o número de caixas empacotadas em um único contêiner. No entanto, mesmo com a tendência do mercado de substituir grandes pedidos por muitos pedidos pequenos com grande variedade de itens (CECÍLIO, 2003), há a necessidade comercial de vários produtores, que em muitos casos possuem cargas de tamanho considerável de um único tipo de produto.

1.1 Justificativa

A escolha do problema em si é justificada pela sua relevância prática. O problema do carregamento de contêineres, como já discutido, aparece freqüentemente na armazenagem e transporte de produtos. Dado o grande volume envolvido nestas operações (BALLOU 1993, DIAS 1995), um pequeno aumento no número de caixas carregadas, pode representar diversas vantagens técnicas e uma economia significativa (MORABITO, 1992), tais como redução de custos de movimentação, armazenagem, transporte de itens produzidos, tempo de carregamento e descarregamento dos contêineres. A melhoria da eficiência de operações logísticas está freqüentemente associada à utilização de estratégias adequadas de carregamento. Com o auxílio de um programa computacionalmente eficiente, a empresa pode, por exemplo, simular padrões de empacotamento para um conjunto de produtos e contêineres

e, a partir dos resultados obtidos, escolher o contêiner mais adequado. Outra possibilidade diz respeito ao desenvolvimento de novas embalagens de produtos, a qual depende tanto de fatores logísticos como mercadológicos (OLIVEIRA, 2004). Nesse caso, a metodologia pode auxiliar a análise do arranjo físico de embalagens dentro de uma unidade de carga intermediária (DOWSLAND, 1995; MORABITO e ARENALES, 2000).

O problema de carregamento de contêineres pertence à classe de problemas NP-difíceis, o que torna um desafio a obtenção de soluções ótimas dentro de um limite de tempo razoável. De fato, a complexidade do problema geralmente prescreve a utilização de métodos heurísticos para a resolução do problema, o que por sua vez, comumente motiva a busca por algoritmos eficazes.

1.2 Objetivos

Os objetivos deste trabalho são:

- (i) Revisar métodos de solução exatos e aproximados existentes da literatura para o problema de carregamento de contêineres;
- (ii) Estender a heurística de busca tabu de Pureza e Morabito (2006), originalmente proposta para o caso bidimensional (problema de carregamento de paletes do produtor) para o caso tridimensional (problema de carregamento de contêineres com caixas idênticas);
- (iii) Analisar e comparar os resultados do método proposto com os resultados dos métodos existentes na literatura.

1.3 Estrutura do trabalho

O restante desta dissertação está organizado como se segue. No Capítulo 2 são apresentadas algumas categorizações e tipologias propostas na literatura para problemas de corte e empacotamento. O problema de carregamento de contêineres e uma revisão bibliográfica de métodos heurísticos são discutidos no Capítulo 3. No Capítulo 4, são descritas a heurística de Pureza e Morabito (2006) para o problema do palete do produtor, e sua extensão para o caso tridimensional. O Capítulo 5 apresenta os resultados computacionais

obtidos pelo algoritmo proposto para um conjunto de instâncias da literatura, seguido dos próximos passos da pesquisa no Capítulo 6.

2 PROBLEMAS DE CORTE E EMPACOTAMENTO (PCE)

Problemas de Corte e Empacotamento (PCE) consideram tanto objetos reais como abstratos e estão presentes em diversas áreas, tais como ciências de gestão, ciências de informação e computação, engenharia, matemática e pesquisa operacional (MORABITO, 2004). Devido a sua relevância tanto prática como científica, os PCE têm atraído considerável atenção de praticantes e pesquisadores.

Em contextos industriais, problemas de corte consistem em cortar unidades grandes (*objetos*) em um conjunto de unidades pequenas (*itens*) demandadas, de maneira a otimizar determinado objetivo, por exemplo, minimizar a quantidade de objetos necessários. Problemas de empacotamento, por outro lado, consistem em empacotar unidades pequenas em unidades grandes, visando, por exemplo, a maximização da utilização do volume. Cortar unidades maiores em unidades menores ou empacotar unidades menores em unidades maiores são vistos como problemas duais, uma vez que um item *cortado* em uma determinada posição do objeto pode ser considerado *alocado* àquela posição (ARENALES *et al.* 2004).

A relevância dos PCE é estimulada, em parte, pela necessidade de indústrias em otimizar seus processos. Na medida que transformações econômicas vêm tornando os mercados cada vez mais competitivos, vêm-se exigindo mudanças profundas nos setores produtivos no sentido de maior automatização de seus processos, melhorias qualitativas de seus produtos e redução de preços praticados. Para que isso seja praticado, torna-se imprescindível a racionalização de recursos e insumos. Ao buscar soluções que minimizem a quantidade de objetos utilizados (problemas de corte) ou maximizem a utilização de espaços (problemas de empacotamento), os PCE se alinham a esta meta. Um dos reflexos da importância destes problemas é o ESICUP (**E**URO *S*pecial *I*nterest *G*roup on *C*utting and *P*acking Problems) (<http://ctpc22.inescn.pt/esicup/>), organização internacional com mais de 500 membros, cujo objetivo é o de reunir e difundir os trabalhos na área.

Devido à grande variedade de PCE, alguns autores desenvolveram critérios para caracterizá-los a fim de simplificar o desenvolvimento ou a escolha de métodos de solução para cada caso. Na seção 2.1 é apresentada a caracterização de Dyckhoff e Finke (1992), consistindo de seis critérios, discutidos a seguir.

2.1 Caracterização de DYCKHOFF e FINKE (1992)

1. Dimensão do problema

A dimensionalidade de um problema é dada pelas dimensões do espaço efetivamente relevantes para sua resolução. Segundo este critério, os PCE podem ser classificados em: (i) unidimensional, (ii) 1.5-dimensional, (iii) bidimensional, (iv) 2.5-dimensional, (v) tridimensional, e (vi) multidimensional.

O Problema da Mochila e o Problema de Corte industrial de bobinas são respectivamente, exemplos típicos de problemas unidimensionais de empacotamento e corte. No Problema da Mochila tem-se um conjunto de itens, cada qual com valor v_i e peso p_i , e deseja-se obter a combinação mais valiosa para carregar uma mochila de capacidade limitada C . No Problema de Corte de bobinas, bobinas de comprimento L devem ser cortadas a fim de satisfazer a demanda de n itens menores de comprimento l_i com mínima perda de material (Figura 2.1). Note que em ambos os problemas, apenas uma dimensão é relevante para a resolução do problema: para o primeiro, o peso, e para o segundo, o comprimento. Exemplos de métodos de solução e variações do problema são encontrados em Martello e Toth (1990), Yanasse *et al.* (2000), Arenales e Marques (2002), Hoto *et al.* (2003), e Poldi e Arenales (2006).

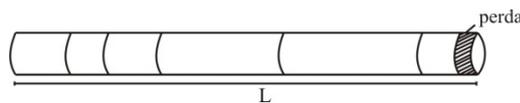


FIGURA 2.1 – Padrão de corte de itens de comprimento l_i ($i=1, \dots$, número de tipos de itens) em uma bobina de comprimento L .

Um exemplo de problema de corte *bidimensional* é o corte de placas retangulares de dimensões (L, W) e espessura constante em peças menores de dimensões idênticas (l, w) , também de espessura constante. A Figura 2.2 mostra um padrão de corte para este tipo de problema, em que a área escura representa a perda de material, a qual deve ser minimizada. A Figura 2.2 serve também como ilustração de um exemplo de aplicação de empacotamento bidimensional. Neste contexto, caixas retangulares para as quais a orientação vertical é fixada, são arranjadas sobre um palete retangular, e a área escura corresponde à área não utilizada do palete. Note que este problema (denominado *Problema de Carregamento de*

Paletes do Produtor e discutido no próximo capítulo) envolve de fato três dimensões. Entretanto, como as caixas têm a mesma altura, a resolução requer apenas a determinação de uma camada de caixas, a qual é repetida até que a altura máxima de carregamento seja atingida. Por esta razão, o problema é bidimensional (MORABITO, 1992). Exemplos de métodos de solução e variações do problema são encontrados em Bischoff e Dowsland, (1982), Hodgson (1982), Nelissen, (1994, 1995), Lodi *et al.* (2002), Yanasse e Morabito (2006), Alvarez-Valdes *et al* (2005), e Morabito e Pureza (2008).

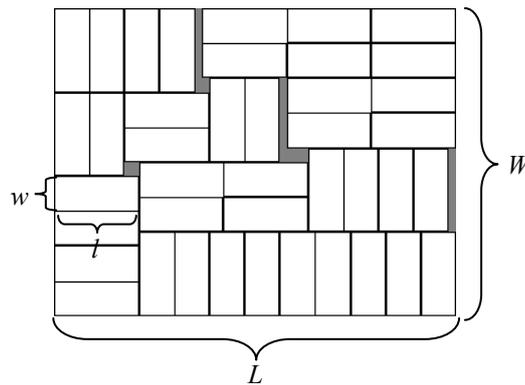


FIGURA 2.2 – Padrão de corte de peças retangulares com dimensões (l, w) em uma chapa de dimensões (L, W) .

Um exemplo de problema tridimensional de empacotamento consiste no arranjo de caixas retangulares de diferentes dimensões (l_i, w_i, h_i) (em que i é o tipo da caixa) em uma caixa maior (contêiner, caminhão ou vagão) de dimensões (L, W, H) . Um exemplo clássico é o carregamento de contêineres, conforme ilustrado na Figura 2.3. Problemas de corte tridimensionais surgem em indústrias de colchões, em que peças de espuma devem ser cortadas em peças menores com diferentes espessuras.

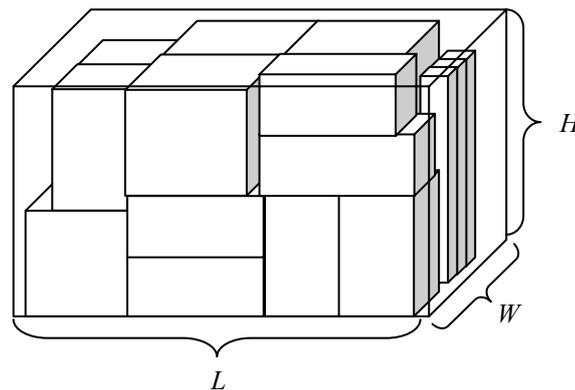


FIGURA 2.3 – Padrão de carregamento de caixas retangulares de dimensões (l_i, w_i, h_i) ($i=1, \dots, \text{número de tipos de caixas}$), em um contêiner de dimensões (L, W, H) .

Problemas 1.5-dimensional têm duas dimensões relevantes, mas uma delas é variável. Considere o exemplo de uma lâmina de vidro com largura fixa: qual o menor comprimento possível da lâmina, necessário para produzir uma determinada quantidade de peças de vidro? No problema 2.5-dimensional, três dimensões são relevantes, mas uma delas é variável. Um exemplo é o carregamento de caixas sob um palete com altura variável e comprimento e largura fixos. Exemplos de métodos de solução são encontrados em Haessler (1978) e Hinxman (1980).

Problemas Multidimensionais são aqueles que possuem mais de 3 dimensões relevantes para sua resolução. Um exemplo deste tipo de problema é o problema de alocação de tarefas, apresentado em Morabito (1992).

2. Forma de alocação das unidades

As unidades podem ser alocadas de duas formas: (i) pela seleção de unidades grandes e, (ii) pela seleção de unidades pequenas. No primeiro caso, um conjunto de unidades grandes deve ser escolhido para incluir todas as unidades pequenas. Um exemplo desta situação é o Problema do Empacotamento de *Bins* (clássico). No segundo caso, um conjunto de unidades pequenas deve ser escolhido para ocupar todas as unidades grandes. Um exemplo é o Problema do Carregamento de Paletes.

3. Ambiente de planejamento

Problemas como programação da produção, sequenciamento de cortes de chapas, planejamento de estoque, entre outros, fazem parte do ambiente de problemas de

corte. Para se alcançar uma solução ótima, o planejamento das decisões neste tipo de ambiente deve considerar todas as inter-relações entre estes problemas. Dois tipos de procedimentos podem ser adotados: planejamento sucessivo e planejamento simultâneo.

No planejamento sucessivo ou seqüencial, os problemas são tratados um após o outro. Um caso típico é o problema de corte de estoque e sequenciamento dos padrões de corte. Possíveis objetivos do primeiro problema (corte de estoque) incluem minimizar a perda de material, maximizar o lucro ou minimizar os custos. Possíveis objetivos do segundo problema incluem minimizar o tempo máximo em que uma pilha (com peças idênticas) em uma estação de descarregamento fica aberta ou minimizar a quantidade de pilhas abertas (YANASSE e LAMOSA, 2007). A solução do primeiro problema define o número de chapas e das peças produzidas, estabelecendo condições de contorno para o segundo problema (MORABITO, 1992). Porém, em geral, uma boa solução para o problema de geração de padrões não corresponde a uma boa solução para o problema de sequenciamento de padrões e vice-versa, evidenciando o conflito de objetivos entre esses dois problemas (PILEGGI *et al.* 2007). Exemplos de abordagens que realizam planejamento sucessivo são encontrados em Dyson e Gregory (1974), Madsen (1979, 1988), e Yanasse (1996, 1997).

No planejamento simultâneo ou integrado os problemas são tratados como um único problema. Duas estratégias básicas podem ser exploradas para resolver dois problemas de forma integrada. A primeira estratégia consiste em otimizar a função objetivo de um dos problemas, transformando o objetivo do segundo problema em uma restrição. A outra estratégia consiste em agregar os problemas e otimizar uma função que combine os objetivos (PILEGGI *et al.* 2007). Exemplos de abordagens que realizam planejamento simultâneo são encontrados em Armbruster (2002), Pinto, (2004), Pileggi *et al.* (2005), Belov e Scheithauer (2007), Pileggi *et al.* (2007), e Yanasse e Lamosa (2007).

4. Objetivos

Os objetivos das decisões a serem tomadas devem ser descritos de acordo com o ambiente de planejamento. No caso de um ambiente com planejamento sucessivo, o objetivo de cada problema é endereçado de forma independente. No caso do planejamento simultâneo em que os objetivos são combinados em uma única função, estes precisam ser expressos em unidades comuns como, por exemplo, em unidades monetárias.

5. Restrições

Restrições geométricas estão naturalmente presentes na maioria dos problemas de corte e empacotamento; elas prescrevem que unidades pequenas devem caber inteiramente na(s) unidade(s) grandes, e não podem estar sobrepostas entre si. À parte destas, outras restrições são categorizadas pelo objeto de sua aplicação, especificamente, se endereçadas (i) às unidades grandes, (ii) às unidades pequenas, ou (iii) ao processo de produção e planejamento.

Restrições de unidades grandes, por exemplo, dizem respeito a suas formas, tamanhos e quantidades disponíveis. Na maioria dos casos abordados na literatura, as unidades grandes têm formas retangulares (paletes, contêineres, caixas de papelão). Contêineres em aeronaves, por sua vez, podem ter formas não retangulares a fim de se adequar ao *layout* da área de carga da aeronave.

No tocante às unidades pequenas, as restrições se aplicam a suas formas e tamanhos disponíveis, quantidade demandada e orientação de posicionamento nas unidades grandes. Assim como no caso das unidades grandes, grande parte da literatura considera unidades de formas retangulares (caixas de embalagem de produtos). Aplicações mais recentes tratam unidades não retangulares, como no empacotamento de cilindros (BIRGIN *et al.*, 2005), encontrados na armazenagem e distribuição de diferentes tipos de gás, e unidades de formas irregulares (ANAND *et al.*, 1999; BENNELL e DOWSLAND, 1999), presentes em problemas de corte em indústrias de manufatura de roupas. A restrição de orientação de posicionamento de unidades pequenas é importante nos casos em que produtos como remédios e cosméticos precisam ser transportados com uma determinada face da embalagem voltada para cima.

Restrições dos processos, por sua vez, são relativas aos equipamentos de corte e empacotamento, propriedades dos materiais e condições organizacionais. Processos de corte são definidos principalmente pela tecnologia dos equipamentos, os quais limitam os possíveis padrões a serem produzidos (um exemplo típico ocorre no problema de corte de bobinas de papel, em que o equipamento possui n facas de corte, produzindo, desta forma, no máximo $n+1$ bobinas). A maioria dos trabalhos considera equipamentos que produzem cortes ortogonais, ou seja, produzidos paralelamente a um dos lados das unidades grandes. Cortes ortogonais podem ser classificados em cortes guilhotinados e cortes não guilhotinados.

Cortes não guilhotinados são produzidos por equipamentos que permitem interromper o corte antes de chegar à borda oposta da unidade (MORABITO, 1992). Cortes guilhotinados, por sua vez, são produzidos a partir de uma borda da unidade grande até a

borda oposta, de maneira a dividir um retângulo em dois retângulos menores e adjacentes. Estes cortes podem ser classificados como n -estagiados ou multi-estagiados, em que um estágio marca a mudança na *direção* atual dos cortes, de maneira que cortes produzidos em um estágio k sejam ortogonais aos cortes produzidos no estágio $k-1$. Em cortes multi-estagiados não se impõem limitações ao número máximo de estágios. Cortes n -estagiados, por outro lado, têm o número de estágios limitado a n devido a restrições do equipamento. Em Belluzzo e Morabito (2005) é apresentado um estudo de caso em uma fábrica de móveis em que além da limitação de cortes em até 3 estágios, existem restrições para as dimensões das chapas e dos cortes longitudinais e transversais, além de outras considerações como cortes de cabeça e giro do pacote de chapas, número de estações de descarregamento automáticas e manuais, e grandezas (profundidade) das incisões das serras e remates.

No tocante a propriedades dos materiais, aspectos como resistência a cortes próximos, peso e densidade do material devem ser considerados. A estabilidade da carga é também um importante aspecto a ser considerado, uma vez que restringe o número de carregamentos considerados viáveis, em particular, em problemas de empacotamento tridimensionais. A fragilidade dos itens em certos casos pode também ocupar um papel determinante na solução apresentada.

Certas condições organizacionais podem interferir diretamente nos padrões de corte e empacotamento a serem obtidos. Considere como exemplo, processos de corte de chapas sujeitos à norma de atender exclusivamente um cliente por vez. Esta política limita os possíveis padrões de corte a serem formados se comparados aos obtidos caso fossem incluídas peças de outros clientes. Em relação ao empacotamento da carga, cabe ressaltar que apesar da perda de eficiência, padrões com agrupamentos de cargas para cada destino facilitam a desova do material.

6. Métodos de solução

Não existem métodos de solução gerais e eficientes para os PCE. Isto é devido à complexidade computacional dos algoritmos, causada por sua natureza combinatória, e a diversidade de situações em que os problemas podem surgir. A associação entre métodos de solução e problemas específicos é importante, no entanto esta associação não é encontrada efetivamente na literatura. Pode-se, no entanto, destacar que as técnicas mais empregadas são otimização discreta, programação linear com geração de colunas, heurísticas, metaheurísticas, e relaxação lagrangiana.

Nas duas seções seguintes, são apresentadas duas tipologias (DYCKHOFF, 1990; e WASCHER *et al.*, 2007) fundamentadas na estrutura lógica dos problemas.

2.2 Tipologia de DYCKHOFF (1990)

O propósito da tipologia de Dyckhoff foi o de unificar os diferentes usos de PCE na literatura e concentrar pesquisas futuras sobre tipos de problemas melhor definidos. A seguir são apresentadas as 4 características principais da tipologia, subdivididas nos seguintes tipos (entre parênteses):

1. Dimensão do problema
 - (1) unidimensional
 - (2) bidimensional
 - (3) tridimensional
 - (N) N- dimensional ($N > 3$)

2. Forma de alocação das unidades
 - (V) seleção de unidades grandes
 - (B) seleção de unidades pequenas

3. Sortimento de unidades grandes
 - (O) uma unidade
 - (I) unidades de tamanhos iguais
 - (D) unidades de tamanhos diferentes

4. Sortimento de unidades pequenas
 - (F) poucas unidades de tamanhos diferentes
 - (M) muitas unidades de muitos tamanhos diferentes
 - (R) muitas unidades de poucos tamanhos diferentes
 - (C) unidades de tamanhos iguais

Combinando todos os tipos destas 4 características principais, obtém-se $4 \times 2 \times 3 \times 4 = 96$ tipos diferentes para os problemas de corte e empacotamento. A falta de um componente na quádrupla indica que o PCE pode ocorrer em várias situações definidas. O problema da mochila (clássico) é representado por 1/B/O/ enquanto o problema do carregamento de contêiner com caixas idênticas, foco deste trabalho, é representado pela notação 3/B/O/C.

Embora tenha destacado a estrutura comum entre os PCE, a tipologia de Dyckhoff (1990) não obteve unanimidade no meio científico. Wascher *et al.* (2007) destaca as seguintes desvantagens:

- A tipologia de Dyckhoff não necessariamente designa cada problema a uma única codificação. O problema do carregamento de veículos foi tipificado por Dyckhoff como sendo 1/V/I/F e 1/V/I/M, onde “F” caracteriza a situação em que poucos itens de diferentes formas estão presentes, e “M” quando existem muitos itens de muitas diferentes formas;
- A tipologia de Dyckhoff apresenta também inconsistências. Um exemplo diz respeito ao problema de empacotamento por faixas (*strip-packing problem*), um caso particular do problema de empacotamento bi-dimensional em que um conjunto de itens de diferentes tamanhos é empacotado em retângulos individuais de largura fixa e comprimento mínimo. Enquanto este problema foi codificado (classificado) por pesquisadores e profissionais da área como 2/V/O/M, na tipologia de Dyckhoff ele recebe a notação 2/V/D/M. Dyckhoff (1990) justifica sua notação dizendo que este problema “...é equivalente ao problema da seleção de variedades, em que o estoque é dado por uma infinidade de objetos de mesma largura e com todas as possibilidades de comprimento e somente um objeto será escolhido do estoque, isto é, o de menor comprimento”. Além disso, Dyckhoff (1990) denomina o problema como de empacotamento de *bins* bidimensional, denominação esta que para diversos pesquisadores (Lodi *et al.*, 2002; Miyazawa e Wakabayashi, 2003; Faroe *et al.*, 2003; apud Wäscher *et al.*, 2007) deveria ser reservada para a extensão natural do problema clássico de empacotamento de *bins* unidimensional;
- A aplicação da tipologia de Dyckhoff não resulta necessariamente em categorias de problemas homogêneos. No caso do corte unidimensional

onde um grande número de itens com poucas formas diferentes é produzido de uma quantidade ilimitada de objetos, pode-se incorrer em duas situações caso os objetos possuam diferentes tamanhos. Na primeira situação, os objetos são separados em pequenos grupos de elementos idênticos; na segunda, todos os objetos são diferentes. Ambas as situações são classificadas por Dyckhoff como 1/V/D/R (Gradisar *et al.*, 2002).

2.3 Tipologia de WASCHER et al. (2007)

Wascher *et al.* (2007) propõem que a definição dos tipos de PCE utilize os seguintes cinco critérios:

1. Dimensionalidade

É distinguida em 1, 2 ou 3 dimensões, similarmente a Dyckhoff (1990). Problemas com mais de 3 dimensões são considerados variantes dos anteriores.

2. Tipo de designação

Corresponde ao critério “Forma de alocação das unidades” de Dyckhoff, e considera duas situações básicas:

- Problemas do tipo maximização de saídas (*output maximisation types*): um dado conjunto de itens deve ser designado a um dado conjunto de objetos e o conjunto de objetos não é suficiente para acomodar todos os itens. Neste caso, todos os objetos serão utilizados e será selecionado o subconjunto de itens de máximo valor total.
- Problemas do tipo minimização de entradas (*input minimisation types*): um dado conjunto de itens cuja demanda deve ser satisfeita completamente deve ser designado a um dado conjunto de objetos em número suficiente para acomodar todos os itens. Desta forma, todos os itens serão designados e será selecionado o subconjunto de objetos de mínimo valor total.

3. Sortimento (variedade) de itens

Corresponde ao critério “Sortimento de unidades pequenas” de Dyckhoff, sendo os problemas classificados como tendo: (i) itens idênticos, (ii) variedade fracamente heterogênea, e (iii) variedade fortemente heterogênea. A classificação *i* corresponde à

situação em que os itens possuem a mesma forma e tamanho, e é idêntica à categoria C de Dyckhoff. A classificação *ii* corresponde a casos em que existem muitos itens de poucos tamanhos/formas diferentes, e é idêntica à categoria R de Dyckhoff. Finalmente, na classificação *iii* existem os itens têm muitos tamanhos/formas diferentes, correspondendo às categorias F e M de Dyckhoff.

4. Sortimento (variedade) de objetos

Corresponde ao critério “Sortimento de unidades grandes” de Dyckhoff, e considera dois tipos principais: *(i)* um objeto e *(ii)* vários objetos. No caso de um único objeto (tipo O de Dyckhoff), admite-se que o mesmo possui todas as dimensões fixas ou uma ou mais dimensões variáveis. No caso de vários objetos, admite-se que os mesmos são idênticos (tipo I de Dyckhoff), tem variedade fracamente heterogênea, ou variedade fortemente heterogênea.

Para a definição da tipificação básica de problemas, admite-se que para casos de duas e três dimensões, todos os objetos têm forma retangular e consistem de material homogêneo. Objetos não-retangulares e/ou não homogêneos resultam em variantes dos problemas anteriores.

5. Forma dos itens

Em relação a problemas de 2 e 3 dimensões, os itens são distinguidos entre regulares (retângulos, círculos, caixas, cilindros, bolas, etc.) e irregulares.

De acordo com o que é geralmente assumido na literatura, itens retangulares são posicionados ortogonalmente nos objetos, ou seja, os itens são dispostos paralelamente às faces do(s) objeto(s). Problemas que permitem *layouts* não-ortogonais e/ou mistos de itens regulares e irregulares são vistos como variantes do problema de itens retangulares posicionados ortogonalmente.

Com base nas características acima, Wascher *et al.* (2007) propõem para os PCE, as tipificações básica, intermediária e refinada, descritas a seguir.

A) Tipos básicos

Os tipos básicos contemplam as categorias “Problemas do tipo maximização de saídas” e “Problemas do tipo minimização de entradas”, levando em consideração a

combinação dos critérios “Tipo de designação” e “Sortimento (variedade) de itens”. A Figura 2.4 mostra as combinações relevantes e problemas associados.

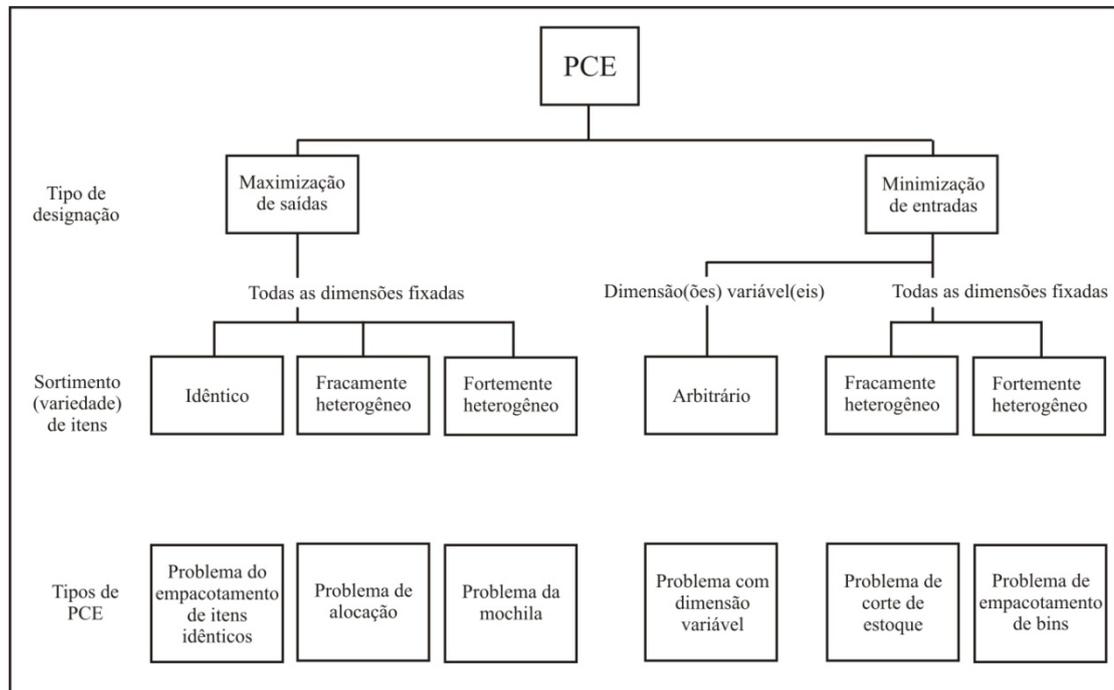


FIGURA 2.4 – Tipos básicos.

B) Tipos intermediários

Para definir os tipos intermediários é adicionado o critério “Sortimento (variedade) de objetos” aos utilizados para os tipos básicos. Desta forma, obtêm-se uma maior diferenciação entre os problemas. As Tabelas 2.1 e 2.2 sumarizam o sistema de problemas do tipo minimização de entradas e maximização de saídas, respectivamente. As siglas utilizadas nas tabelas são abreviaturas do termo em inglês de cada tipo do problema. Maiores informações podem ser obtidas em Wascher *et al.* (2007).

Tabela 2.1 – Tipos intermediários/minimização de entradas.

Sortimento (variedade) de itens		Características dos objetos	
		Fracamente heterogêneo	Fortemente heterogêneo
Todas as dimensões fixadas	Idênticos	Problema de corte de estoque de objetos de um único tamanho SSSCSP	Problema de empacotamento de <i>bins</i> de um único tamanho SBSBPP
	Fracamente heterogêneo	Problema de corte de estoque de objetos de tamanhos diferentes MSSCSP	Problema de empacotamento de <i>bins</i> de tamanhos diferentes MBSBPP
	Fortemente heterogêneo	Problema de corte de estoque residual RCSP	Problema de empacotamento de <i>bins</i> residual RBPP
Apenas um objeto com uma ou mais dimensões variáveis		Problema de dimensão “aberta” ODP	

Tabela 2.2 – Tipos intermediários/maximização de saídas.

Sortimento (variedade) de itens		Características dos objetos		
		Idênticos	Fracamente heterogêneo	Fortemente heterogêneo
Todas as dimensões fixadas	Um objeto	Problema de empacotamento de itens idênticos IIPP	Problema de alocação em um único objeto SLOPP	Problema da mochila SKP
	Idênticos	X	Problema de alocação em múltiplos objetos idênticos MILOPP	Problema de múltiplas mochilas idênticas MIKP
	Heterogêneos		Problema de alocação em múltiplos objetos heterogêneos MHLOPP	Problema de múltiplas mochilas heterogêneas MHKP

C) Tipos refinados

A obtenção de tipos refinados é feita adicionando-se o critério “Dimensionalidade” aos utilizados para o tipo intermediário e, para problemas com duas e três dimensões, o critério “Forma dos itens”. As subcategorias resultantes são caracterizadas por adjetivos adicionados aos nomes dos problemas do tipo intermediário de acordo com o seguinte sistema:

$$\{\text{dimensão}\} + \{\text{forma}\} + \{\text{IPT}\}$$

Em que:

Dimensão: 1,2 ou 3;

Forma: retangular, circular,..., irregular;

IPT: nome do problema do tipo intermediário.

Wascher *et al.* (2007) afirma que esta tipologia fornece uma estruturação mais abrangente e precisa dos PCE do que aquela proposta em Dyckhoff (1990).

3 O PROBLEMA DE CARREGAMENTO DE CONTÊINERES

O Problema de Carregamento de Contêineres (PCC) consiste em arranjar ortogonalmente e sem sobreposição, caixas retangulares disponíveis em n tipos de dimensões (l_i, w_i, h_i) ($i=1..n$) dentro de contêineres retangulares disponíveis em m tipos de dimensões (L_j, W_j, H_j) ($j=1..m$). Deve-se notar que apesar das situações práticas não estarem limitadas a itens e objetos com estruturas retangulares, a dificuldade no tratamento de problemas cujos elementos possuam estruturas não retangulares ou irregulares tem restringido a maioria das contribuições na literatura ao empacotamento de cubos em cubos maiores.

Segundo Bischoff e Marriott (1990), o PCC aparece na prática em dois casos importantes:

- (i) Quando uma combinação de diferentes contêineres deve ser escolhida para transportar uma dada carga ($m \geq 1$ e $n \geq 1$). Conforme discutido no Capítulo 2, este problema pode ser classificado em Dyckhoff (1990) como 3/V/ / em que os últimos 2 valores omitidos correspondem respectivamente ao sortimento de unidades grandes (objetos) e ao sortimento de unidades pequenas (itens). Wascher *et al.* (2007) incluem estes problemas dentre as classes SSSCSP, SBSBPP, MSSCSP e MBSBPP (veja Tabela 2.1).
- (ii) Quando o maior volume de uma dada carga deve ser escolhido para ser transportado em um único contêiner. Em relação ao *mix* de caixas, o carregamento é classificado como *homogêneo* quando as caixas são idênticas, e *heterogêneo*, caso contrário. Neste último caso, o carregamento pode ser ainda classificado como fortemente heterogêneo caso haja muitos tipos de caixas, e como fracamente heterogêneo se o número de tipos de caixas for pequeno (BORTFELDT, 1994). De acordo com a classificação de Dyckhoff (1990), quando as caixas são idênticas, o problema é classificado como 3/B/O/C; se a carga é fortemente heterogênea, o problema é classificado como 3/B/O/M. Em Wascher *et al.* (2007), esses 2 problemas podem ser classificados respectivamente como IIPP e SKP (veja Tabela 2.2).

Além de restrições geométricas (a carga deve caber dentro do contêiner e as caixas não podem estar sobrepostas), podem-se considerar várias outras, desta vez dependendo do tipo de produto a ser transportado, ou mesmo impostas pela empresa responsável pela armazenagem e distribuição, como limitações de custos. Pisinger (2002) apresenta as seguintes variantes do PCC de acordo com a função objetivo e restrições presentes:

- **Empacotamento por faixas (*Strip packing*)**. O contêiner tem fixa a largura e altura, e o problema consiste em empacotar todas as caixas de forma que a profundidade utilizada do contêiner seja mínima. Este problema é aplicado em situações *multi-drop*, onde o carregamento deve ser dividido em áreas distintas do contêiner, cada qual correspondendo a um determinado destino. Várias metodologias para este problema são comparadas em Bischoff e Marriott (1990) e o problema geral é discutido em Bischoff e Ratcliff (1995a).
- **Carregamento da Mochila (*Knapsack loading*)**: Cada caixa está associada a um valor de lucro, e o problema consiste em escolher um subconjunto de caixas que maximizem o lucro. Se a lucratividade de cada caixa está associada ao volume, então o objetivo corresponde a maximizar o espaço utilizado do contêiner. Aplicações deste problema contemplam carregamentos heterogêneos (GEORGE e ROBINSON, 1980; GEHRING *et al.*, 1990; SCHEITHAUER, 1992; ABDU e YANG, 1994; GEHRING e BORTFELDT, 1997; MORABITO e ARENALES, 1994, 1997; ELEY, 2002; PISINGER 2002; CECÍLIO 2003; ARAÚJO e ARMENTANO, 2007) e carregamentos homogêneos (HAN *et al.*, 1989; GEORGE, 1992; LIU E HSIAO, 1997; LINS *et al.*, 1999, 2002).
- ***Bin-packing***. Um número limitado de caixas deve ser empacotado utilizando o menor número de contêineres idênticos. Este problema é tratado em Scheithauner (1991), Faroe *et al.* (1999) e Martello *et al.* (2000).
- **Carregamento de múltiplos contêineres (*Multi-container loading*)**. Similar ao *Bin-Packing*, exceto que os contêineres podem ter diferentes dimensões e o objetivo é o de escolher um subconjunto de contêineres para o qual os custos de transporte sejam minimizados. Abordagens para este problema podem ser encontradas em Chen *et al.* (1995).

Aspectos de fragilidade, manuseio e, em especial, de estabilidade da carga, desempenham um papel relevante no PCC e vêm mais recentemente atraído a atenção de pesquisadores. Em particular, Gehring *et al.* (1990) discutem as seguintes restrições que prescrevem o balanceamento de peso e localização dos itens dentro do contêiner:

- Orientação das caixas: limitadas para certos tipos de produto, por exemplo, cosméticos e tintas.
- Limitações de peso bruto (DAVIES e BISCHOFF, 1999): geralmente presentes quando a carga tem densidade considerável, a qual pode, até certo ponto, ser compensada pela escolha das dimensões dos contêineres (MORABITO e ARENALES, 1997). A orientação é que contêineres maiores sejam utilizados para cargas pouco densas e contêineres menores para cargas mais densas. George e Robinson (1980) argumentam que, em muitos casos, considerações geométricas limitam a quantidade de carga a ser carregada nos contêineres, antes que restrições de peso sejam efetivamente limitantes. Esta restrição é considerada em Liu e Hsiao (1997).
- Empilhamento máximo: necessário quando existe limitação do peso que as caixas podem suportar.
- Estabilidade da carga: mesmo tomando precauções durante o transporte, a movimentação das caixas dentro do contêiner, devido a um carregamento instável, pode levar a danos e até perda de parte da carga.

Bischoff e Raticliff (1995b) classificam as restrições entre fortes e fracas e adicionam os seguintes tipos:

- Manejo: o tamanho ou o peso de um item e equipamentos utilizados podem de certa forma ditar o posicionamento dentro do contêiner.
- Agrupamento de itens: Podem facilitar a verificação da carga, por exemplo.
- *Multi drop situations*: se a carga será entregue em vários destinos, é desejável não somente carregar itens de mesmo destino próximos, mas também a ordem de carregar o contêiner deve ser levada em

consideração para evitar o descarregamento e carregamento desnecessário da carga.

- Separação de itens no contêiner: presentes no carregamento de gêneros alimentícios e artigos de perfume ou diferentes produtos químicos.
- Carregamento completo de grupos de itens: um subconjunto da carga, por exemplo, peças de uma máquina, constitui uma única entidade. É necessário assegurar que se uma peça é carregada então todo o subconjunto será carregado.
- Prioridade de carregamento: alguns itens podem ser mais importantes do que outros. Isto pode estar relacionado, por exemplo, a prazos de entrega.
- Complexidade do arranjo do carregamento: padrões de carregamento mais complexos geralmente resultam em um maior esforço dos recursos para manuseio da carga, o que pode representar um fator de restrição para o carregamento.
- Distribuição de peso no contêiner: é desejável que o centro de gravidade esteja próximo do centro geométrico do contêiner. Esta restrição é abordada em Davies e Bischoff (1999).

Restrições de peso máximo do contêiner, orientação das caixas e empilhamento são restrições fortes e devem ser necessariamente satisfeitas. Restrições como agrupamento de itens, por outro lado, não são claramente definidas e podem ser consideradas restrições fracas. As restrições fortes devem ser adicionadas ao modelo como restrições uma vez que delimitam o espaço de soluções factíveis, enquanto as restrições fracas podem ser consideradas na função objetivo.

Devido a fatores geométricos (dimensões do contêiner e das caixas) mesmo que o volume total das caixas a serem empacotadas seja menor do que o volume do contêiner pode não ser possível empacotar todas as caixas no contêiner. Em geral, haverá uma maior utilização do volume do contêiner em carregamentos que possuam caixas de dimensões relativamente pequenas do que em carregamentos que possuam caixas de dimensões relativamente grandes. A imposição de restrições como as citadas anteriormente contribuem também para que a utilização do volume do contêiner seja menor.

Segundo a teoria de complexidade (GAREY e JOHNSON, 1979) problemas de carregamento de contêineres são classificados como *NP-difíceis*. Isso impossibilita, na maioria das situações práticas, a obtenção de soluções exatas, seja por limitações de memória

computacional, seja por limitações de tempo para tal obtenção (PISINGER, 2002). Por este motivo, grande parte das abordagens de resolução para o problema consiste de métodos heurísticos (ARAÚJO, 2007). Ainda assim, o grande número de variações do problema limita a aplicação de métodos de resolução existentes a uma pequena parcela de situações existentes (BISCHOFF e RATCLIFF, 1995b).

3.1 Modelagem Matemática

Nesta seção apresenta-se o modelo de programação linear inteira mista 0-1 proposto em Chen *et al.* (1995). O modelo considera m contêineres (possivelmente de diferentes dimensões), N caixas (possivelmente de diferentes dimensões), e inclui restrições de orientação das caixas. Como normalmente adotado, as caixas devem ser posicionadas ortogonalmente nos contêineres. A maior dimensão de cada caixa e contêiner é definida como comprimento, a menor dimensão como altura, e a dimensão intermediária como largura. O objetivo é minimizar o volume não utilizado dos contêineres, o que é equivalente a maximizar o volume total utilizado. Este problema pode ser classificado em Dyckhoff (1990) como 3/V/ / , enquanto em Wascher *et al.* (2007), este tipo de problema pode ser classificado dentre SSSCSP, SBSBPP, MSSCSP e MBSBPP (veja Tabela 2.1).

Parâmetros

- N : número total de caixas a serem empacotadas
 m : número total de contêineres disponíveis.
 (p_i, q_i, r_i) : comprimento, largura e altura da caixa i ($i=1..N$).
 (L_j, W_j, H_j) : comprimento, largura e altura do contêiner j ($j=1..m$).
 M : número arbitrariamente grande.

Variáveis Binárias

- s_{ij} : igual a 1 se a caixa i é colocada no contêiner j e 0, caso contrário.
 n_j : igual a 1 se o contêiner j é utilizado e 0, caso contrário.
 (l_{xi}, l_{yi}, l_{zi}) : indicam se o comprimento da caixa i é paralelo ao eixo x , y ou z . Por exemplo, l_{xi} é 1 se o comprimento da caixa i é paralelo ao eixo x e 0, caso contrário.

(w_{xi}, w_{yi}, w_{zi}) : indicam se a largura da caixa i é paralela ao eixo x, y ou z . Por exemplo, w_{xi} é 1 se a largura da caixa i é paralela ao eixo x e 0, caso contrário.

(h_{xi}, h_{yi}, h_{zi}) : indicam se a altura da caixa i é paralela ao eixo x, y ou z . Por exemplo, h_{xi} é 1 se a altura da caixa i é paralela ao eixo x e 0, caso contrário.

$a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik}$: indicam a localização das caixas entre si. O valor de a_{ik} é 1 se a caixa i está no lado esquerdo da caixa k . Similarmente, as variáveis $b_{ik}, c_{ik}, d_{ik}, e_{ik}$ e f_{ik} representam se a caixa i está a direita, atrás, em frente, abaixo ou acima da caixa k , respectivamente. As variáveis são necessárias e definidas somente quando $i < k$.

Variáveis contínuas

(x_i, y_i, z_i) : coordenadas do canto inferior frontal esquerdo (FLB) da caixa i .

$$\text{Minimizar } \sum_{j=1}^m L_j \cdot W_j \cdot H_j \cdot n_j - \sum_{i=1}^N p_i \cdot q_i \cdot r_i$$

sujeito a:

$$x_i + p_i \cdot l_{xi} + q_i \cdot w_{xi} + r_i \cdot h_{xi} \leq x_k + (1 - a_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (1)$$

$$x_k + p_k \cdot l_{xk} + q_k \cdot w_{xk} + r_k \cdot h_{xk} \leq x_i + (1 - b_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (2)$$

$$y_i + q_i \cdot w_{yi} + p_i \cdot l_{yi} + r_i \cdot h_{yi} \leq y_k + (1 - c_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (3)$$

$$y_k + q_k \cdot w_{yk} + p_k \cdot l_{yk} + r_k \cdot h_{yk} \leq y_i + (1 - d_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (4)$$

$$z_i + r_i \cdot h_{zi} + q_i \cdot w_{zi} + p_i \cdot l_{zi} \leq z_k + (1 - e_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (5)$$

$$z_k + r_k \cdot h_{zk} + q_k \cdot w_{zk} + p_k \cdot l_{zk} \leq z_i + (1 - f_{ik}) \cdot M \quad \forall i, k, i < k \quad (i, k = 1..N) \quad (6)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \geq s_{ij} + s_{kj} - 1 \quad \forall, k, j, i < k \quad (i, k = 1..N) \quad (7)$$

$$\sum_{j=1}^m s_{ij} = 1 \quad \forall i \quad (i = 1..N) \quad (8)$$

$$\sum_{i=1}^N s_{ij} \leq M \cdot n_j \quad \forall j \quad (j = 1..m) \quad (9)$$

$$x_i + p_i \cdot l_{xi} + q_i \cdot w_{xi} + r_i \cdot h_{xi} \leq L_j + (1 - s_{ij}) \cdot M \quad \forall i, j \quad (i = 1..N, j = 1..m) \quad (10)$$

$$y_i + q_i \cdot w_{yi} + p_i \cdot l_{yi} + r_i \cdot h_{yi} \leq W_j + (1 - s_{ij}) \cdot M \quad \forall i, j \quad (i = 1..N, j = 1..m) \quad (11)$$

$$z_i + r_i \cdot h_{zi} + q_i \cdot w_{zi} + p_i \cdot l_{zi} \leq H_j + (1 - s_{ij}) \cdot M \quad \forall i, j \quad (i = 1..N, j = 1..m) \quad (12)$$

$l_{xi}, l_{yi}, l_{zi}, w_{xi}, w_{yi}, w_{zi}, h_{xi}, h_{yi}, h_{zi}, a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik}, s_{ij}, n_j = 0$ ou 1

$x_i, y_i, z_i \geq 0$

A função objetivo minimiza o espaço não utilizado nos contêineres selecionados. As restrições (1) a (6) asseguram a não sobreposição das caixas e são necessárias apenas para pares de caixas que estejam no mesmo contêiner, o que é verificado pela restrição (7). A restrição (8) assegura que cada caixa será colocada apenas em um contêiner. A restrição (9) assegura que caixas sejam atribuídas a um contêiner somente se este for utilizado. As restrições (10) a (12) asseguram que todas as caixas carregadas respeitem as dimensões físicas do contêiner a que foram atribuídas.

Note que as seguintes relações entre as variáveis binárias l_{xi} , l_{yi} , l_{zi} , w_{xi} , w_{yi} , w_{zi} , h_{xi} , h_{yi} e h_{zi} precisam ser satisfeitas:

$$l_{xi} + l_{yi} + l_{zi} = 1$$

$$w_{xi} + w_{yi} + w_{zi} = 1$$

$$h_{xi} + h_{yi} + h_{zi} = 1$$

$$l_{xi} + w_{xi} + h_{xi} = 1$$

$$l_{yi} + w_{yi} + h_{yi} = 1$$

$$l_{zi} + w_{zi} + h_{zi} = 1$$

Utilizando-se estas relações, pode-se eliminar 5 variáveis (l_{yi} , w_{xi} , w_{zi} , h_{xi} e h_{yi}) e reduzir significativamente o modelo. Com a eliminação, as restrições 1-6 e 10-12 são reescritas como:

$$x_i + p_i \cdot l_{xi} + q_i \cdot (l_{zi} - w_{yi} + h_{zi}) + r_i \cdot (1 - l_{xi} - l_{zi} + w_{yi} - h_{zi}) \leq x_k + (1 - a_{ik}) \cdot M \quad (1')$$

$$x_k + p_k \cdot l_{xk} + q_k \cdot (l_{zk} - w_{yk} + h_{zk}) + r_k \cdot (1 - l_{xk} - l_{zk} + w_{yk} - h_{zk}) \leq x_i + (1 - b_{ik}) \cdot M \quad (2')$$

$$y_i + q_i \cdot w_{yi} + p_i \cdot (1 - l_{xi} - l_{zi}) + r_i \cdot (l_{xi} - l_{zi} - w_{yi}) \leq y_k + (1 - c_{ik}) \cdot M \quad (3')$$

$$y_k + q_k \cdot w_{yk} + p_k \cdot (1 - l_{xk} - l_{zk}) + r_k \cdot (l_{xk} - l_{zk} - w_{yk}) \leq y_i + (1 - d_{ik}) \cdot M \quad (4')$$

$$z_i + r_i \cdot h_{zi} + q_i \cdot (1 - l_{zi} - h_{zi}) + p_i \cdot l_{zi} \leq z_k + (1 - e_{ik}) \cdot M \quad (5')$$

$$z_k + r_k \cdot h_{zk} + q_k \cdot (1 - l_{zk} - h_{zk}) + p_k \cdot l_{zk} \leq z_i + (1 - f_{ik}) \cdot M \quad (6')$$

$$x_i + p_i \cdot l_{xi} + q_i \cdot (l_{zi} - w_{yi} + h_{zi}) + r_i \cdot (1 - l_{xi} - l_{zi} + w_{yi} - h_{zi}) \leq L_j + (1 - s_{ij}) \cdot M \quad (10')$$

$$y_i + q_i \cdot w_{yi} + p_i \cdot (1 - l_{xi} - l_{zi}) + r_i \cdot (l_{xi} - l_{zi} - w_{yi}) \leq W_j + (1 - s_{ij}) \cdot M \quad (11')$$

$$z_i + r_i \cdot h_{zi} + q_i \cdot (1 - l_{zi} - h_{zi}) + p_i \cdot l_{zi} \leq H_j + (1 - s_{ij}) \cdot M \quad (12')$$

Note que o resultado destas modificações é ainda um modelo de programação linear inteira mista 0-1. Ele contém $(\frac{1}{2} mN(N-1) + 3N(N+m-1) + N+m)$ restrições e $((3N+m+4)N+m)$ variáveis, das quais $3N$ variáveis são contínuas e $((3N+m+1)N+m)$ são binárias (CHEN *et al.*, 1995).

Modificações adicionais no modelo podem ser feitas considerando especificidades da instância. Por exemplo, se um único contêiner deve ser selecionado para carregar o conjunto de caixas, as restrições (8) e (9) são excluídas, a variável s_{ij} é eliminada, a restrição (7) tem seu lado direito da alterado para 1, as restrições (10'), (11') e (12') têm seus lados direitos respectivamente alterados para $\sum_{j=1}^m L_j \cdot n_j$, $\sum_{j=1}^m W_j \cdot n_j$ e $\sum_{j=1}^m W_j \cdot n_j$, e a restrição $\sum_{j=1}^m n_j = 1$ é incluída.

Nas duas seções seguintes deste capítulo, apresenta-se uma revisão (não exaustiva) de métodos exatos e heurísticos propostos para resolução de variantes do PCC comumente encontradas. Em particular, e dada sua relevância, critérios de estabilidade do carregamento utilizados por vários dos métodos descritos são discutidos separadamente na última seção.

3.2 Métodos Exatos

Em Chen *et al.* (1995), o modelo apresentado na seção anterior é validado por meio de um exemplo com 3 contêineres de tamanhos diferentes e 6 tipos de caixas. O modelo para este exemplo contém 198 restrições e 153 variáveis (18 contínuas e 135 binárias) é resolvido com o *software* LINGO. Outro exemplo de modelo de programação inteira mista 0-1 é proposto em TSAI *et al.* (1993) e endereçado ao problema de carregamento de paletes tridimensional com caixas heterogêneas. A dimensão da altura das caixas é fixa, de forma que os carregamentos estão restritos a duas possíveis orientações para as caixas. Soluções ótimas são obtidas utilizando-se o algoritmo proposto por Kochenberger e Richard (1982), e baseado em um procedimento *branch-and-bound*. O maior problema resolvido pelos autores possui 4 tipos de caixas.

Os trabalhos a seguir, originalmente propostos para a resolução de problemas bidimensionais, podem ser estendidos para o caso tridimensional. Em Beasley (1985), o problema de corte bidimensional não-guilhotinado é formulado por um modelo de programação inteira 0-1 e um procedimento de busca em árvore é aplicado para resolvê-lo. Hadjiconstantinou e Christofides (1995) apresentam um procedimento exato de busca em árvore para resolver problemas da mochila bidimensional. O tamanho da busca é limitado por

limitantes derivados a partir de uma relaxação lagrangiana de uma formulação de programação inteira 0-1 do problema. Martello *et al.* (2000) propõem algoritmos enumerativos exatos para o carregamento em caixas com orientação fixa em um único contêiner. Em particular, um algoritmo *branch-and-bound* é proposto.

Deve-se ressaltar que, além das dificuldades intrínsecas envolvidas em sua resolução, nenhum dos modelos citados nesta seção possui restrições que garantam estabilidade, mesmo em termos triviais, nos carregamentos. Assim, é possível obter padrões ótimos que possuam caixas cuja porcentagem de suporte de suas bases suportadas é igual a 0%.

3.3 Métodos Heurísticos

De acordo com Pisinger (2002), as heurísticas mais comumente utilizadas para resolução do problema de carregamento de contêiner podem ser classificadas de acordo com a forma de geração dos padrões. Especificamente, estes podem ser criados por meio da construção de *paredes virtuais*, *pilhas de caixas*, *cortes guilhotinados* ou *cubóides*.

A construção de paredes virtuais define camadas horizontais ou verticais do carregamento. As dimensões de camadas verticais são determinadas pela área (W , H) do contêiner, a largura da camada é definida, em geral, pela largura da primeira caixa nela empacotada (Figura 3.1), e o empacotamento é feito do fundo para a frente do contêiner (GEHRING *et al.*, 1990). As dimensões de camadas horizontais, por sua vez, são definidas pela área (L , W) do contêiner e pelo comprimento da primeira caixa empacotada. A primeira caixa empacotada é definida segundo algum critério como, por exemplo, a caixa que possui o maior volume (GEHRING *et al.*, 1990). Davies e Bischoff (1999) alternam a seqüência das camadas ao longo do comprimento L do contêiner como forma de melhorar a distribuição de peso no contêiner.

Um caso particular de carregamento em camadas, no qual a demanda é irrestrita, pode ser visto como ocorrendo em duas etapas (MORABITO e ARENALES, 1994, 1997). A primeira etapa consiste em resolver várias vezes o caso bidimensional para a área (L , W) (camadas horizontais, com altura h_i $i=1..j$ em que j são os tipos de caixas) ou para a área (W , H) (camadas verticais, com altura l_i). Cada camada possui caixas de mesma altura. Na segunda etapa é resolvido o problema da mochila unidimensional, ou seja, a escolha das camadas que compõem o empacotamento resultante, sendo o valor de cada camada a

quantidade de caixas que ela possui. Note que na construção de camadas horizontais, a somatória das alturas das camadas selecionadas está limitada à altura H do contêiner, enquanto que no caso de camadas verticais, a somatória das larguras das camadas selecionadas está sujeita ao comprimento L do contêiner.

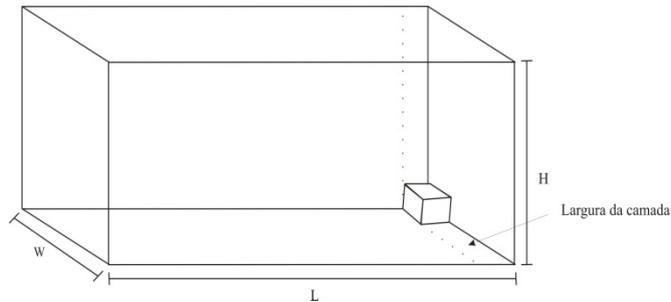


FIGURA 3.1 - Empacotamento da primeira caixa de uma camada vertical.

No primeiro passo na construção de pilhas de caixas, pilhas com altura máxima H são formadas (Figura 3.2) neste passo, vários problemas da mochila unidimensional são resolvidos. No segundo passo, é resolvido o problema de empacotamento bidimensional de arranjar as pilhas sobre a área (L, W) do contêiner (MORABITO e ARENALES, (1997)

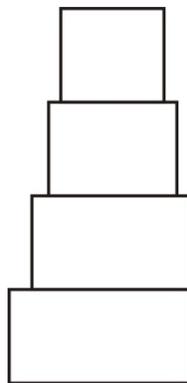


FIGURA 3.2 - Pilha de caixas.

Conforme discutido na Seção 2.1, cortes guilhotinados são aqueles que quando realizados sobre um paralelepípedo produzem dois paralelepípedos. Neste tipo de padrão de carregamento, todas as caixas podem ser obtidas por uma seqüência de cortes guilhotinados. A Figura 3.3 apresenta um exemplo deste tipo de padrão.

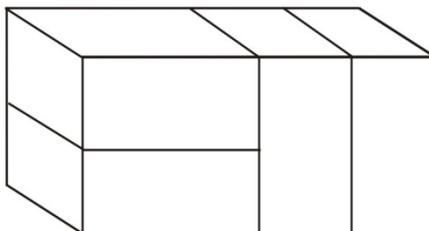


FIGURA 3.3 – Exemplo de padrão de corte guilhotinado.

Em heurísticas que constroem cubóides, blocos tridimensionais são formados por caixas do mesmo tipo com a mesma orientação. As caixas que formam um cubóide devem ser dispostas ao longo das três dimensões do contêiner de modo a formarem um paralelepípedo. A quantidade de caixas que compõem um cubóide é definida pela quantidade de caixas ao longo do eixo X multiplicado pela quantidade de caixas ao longo do eixo Y multiplicado pela quantidade de caixas ao longo do eixo Z . O volume de um cubóide é dado pelo produto do volume de uma caixa e o número de caixas que compõem o cubóide. A Figura 3.4 é um exemplo de cubóide.

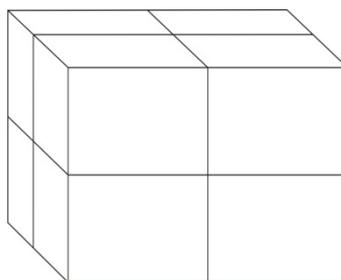


FIGURA 3.4 – Cubóide com 8 caixas.

Métodos heurísticos para problemas de carregamento de contêineres são geralmente projetados levando em consideração características do sortimento das unidades pequenas. Em outras palavras, a literatura aponta métodos diferenciados para o tratamento de carregamentos heterogêneos e para o tratamento de carregamentos homogêneos. A razão para esta discriminação se deve ao fato de procedimentos para carga heterogênea utilizarem informações do valor relativo de dimensões das caixas em suas decisões (Seção 3.3.1), o que geralmente resulta em carregamentos deficientes quando as caixas são idênticas (GEORGE, 1992; CECÍLIO, 2003).

3.3.1 Métodos Heurísticos para Carregamentos Heterogêneos

George e Robinson (1980) abordam o carregamento de uma carga heterogênea em um único contêiner. Os autores propõem uma heurística que constrói camadas ao longo da largura do contêiner e agrupa espaços vagos entre camadas para aumentar a utilização do volume. Não são consideradas restrições de empilhamento máximo e orientação das caixas.

O comprimento da camada é definido pelo comprimento da primeira caixa colocada nela, e algumas restrições são postas na escolha dessa caixa. Uma caixa que já tenha sido utilizada é chamada de tipo *open*. Para que caixas iguais sejam colocadas próximas umas as outras, caixas tipo *open* tem maior prioridade de seleção em relação a caixas do tipo “*closed*” (caixas ainda não utilizadas). Caso existam mais de um tipo *open* de caixas, duas regras são utilizadas para seleção: o tipo de caixa com maior quantidade disponível de caixas, e o tipo de caixa com a maior das menores dimensões. Caso não existam tipos *open*, seleciona-se o tipo de caixa *closed* de acordo com os três critérios a seguir:

1. Aquele cuja menor dimensão é maior do que a menor dimensão dos demais tipos;
2. Aquele com a maior quantidade disponível de caixas;
3. Aquele cuja maior dimensão é maior que a maior dimensão dos demais tipos.

O algoritmo proposto coloca as caixas em pilhas feitas de caixas de um mesmo tipo. Caso a camada não seja preenchida completamente pelo tipo de caixa que a iniciou, o espaço vago é preenchido com pilhas de um tipo de caixa que proporciona a menor perda no comprimento do espaço. Este espaço, porém, não é considerado uma nova camada. Os autores testaram o método em um conjunto de instâncias, e um exemplo com 784 caixas distribuídas em oito tipos é detalhado. A melhor solução obtida para este exemplo carrega 783 caixas, deixando somente uma caixa fora do contêiner.

Posteriormente, Cecílio e Morabito (2004) propuseram um refinamento para a heurística de George e Robinson, alterando o procedimento de escolha do tipo de caixa para os espaços que sobram nas camadas. Ao invés de simplesmente escolher o tipo de caixa que preenche a maior área da base do espaço como prescrito na heurística original, verifica-se se pode ser feita uma combinação de caixas iguais no comprimento desse espaço. A partir desta combinação de caixas iguais, é escolhida aquela que resulta em uma maior utilização da área da base do espaço. Com base neste refinamento foram desenvolvidas duas versões da heurística de George e Robinson (1980), a versão “Arranjo” e a versão “Camada”. Na versão

Arranjo é utilizado um arranjo de três dos cinco critérios de classificação. Esta versão consiste em executar a heurística 60 vezes, utilizando um arranjo de classificação de caixas diferente em cada execução. A solução final corresponde ao melhor resultado obtido dentre todas as iterações em termos de volume empacotado. A versão Camada, por sua vez, executa 60 vezes os procedimentos para carregar cada camada do contêiner (esgotando todas as possibilidades de arranjos), e a camada com melhor avaliação é incluída no padrão de empacotamento. Combinando o algoritmo de George e Robinson e os procedimentos propostos, cinco métodos são propostos neste trabalho:

- Heurística de George e Robinson mais refinamento;
- Heurística de George e Robinson mais versão arranjo;
- Versão Arranjo mais refinamento;
- Heurística de George e Robinson mais versão camada;
- Versão camada mais refinamento.

Para o exemplo tratado em George e Robinson, foi obtida uma solução em que todas as 784 caixas são carregadas.

Bischoff e Marriot (1990) tratam o carregamento em um único contêiner quando o objetivo é obter o comprimento mínimo necessário do contêiner para acomodar uma dada carga. Os autores propõem 14 heurísticas correspondentes a modificações da heurística de George e Robinson (1980) e duas adaptações do método de Bischoff e Dowsland (1982) para o carregamento de paletes.

Em Gehring *et al.* (1990), o carregamento tem como objetivo minimizar o espaço não utilizado do contêiner. As caixas são empacotadas em camadas verticais isoladas, que podem ser alternadas para balancear o peso no contêiner ao longo do comprimento do contêiner. Segundo os autores, iniciar o empacotamento com caixas de alto volume tende a resultar em uma boa utilização do volume do contêiner. Seguindo essa idéia, é criada uma lista de caixas em ordem decrescente de volume, o que por sua vez, define a ordem de empacotamento das caixas.

A largura de cada camada é definido pela primeira caixa empacotada, denominada LDB (*Layer determining box*), na camada. O algoritmo prevê interação com o usuário de modo a permitir a escolha da orientação da primeira caixa da camada (1º critério) e alteração na ordenação das caixas (2º critério) antes que o procedimento de resolução seja iniciado. O primeiro critério consiste em enumerar as possíveis orientações (no máximo 6,

dependendo das dimensões das caixas) que a LDB da primeira camada pode ter. O segundo critério consiste em utilizar a segunda, terceira, quarta, etc., caixa da lista ordenada para iniciar o empacotamento. Não há restrições de proximidade de caixas (por exemplo, produtos que não podem ficar próximos) e as caixas podem ser posicionadas em qualquer lugar do contêiner.

Morabito e Arenales (1994, 1997) abordam o caso especial de carregamento de um único contêiner com caixas de baixa densidade, visando maximizar o volume da carga e satisfazendo restrições de estabilidade do carregamento. São revistas algumas abordagens conhecidas da literatura, tais como os procedimentos em duas etapas de carregar caixas em camadas horizontais e em pilhas verticais, a aplicação de técnicas de programação dinâmica e, em particular, métodos de busca baseados na representação do espaço de soluções num grafo-E/OU.

Ngoi *et al.* (1994) abordam o carregamento de contêineres em que, diferentemente da maioria dos algoritmos encontrados na literatura que empacotam caixas em camadas, utilizam uma técnica de representação espacial. Esta técnica foi desenvolvida por Ngoi e Whybrew (1993), e representa objetos e espaços disponíveis por meio de uma estrutura simples de matriz. É utilizada uma matriz tridimensional para representar os espaços vazios e as posições e dimensões de todas as caixas. Esta matriz é composta por uma cadeia de matrizes bidimensionais que representam camadas horizontais do contêiner. Os elementos das matrizes bidimensionais representam as caixas empacotadas a partir de um número de identificação. Espaços disponíveis são representados por células da matriz com valor 0. Devido à complexidade e a grande quantidade de combinações do problema algumas simplificações foram feitas, tais como:

- Todas as caixas têm o mesmo destino;
- As caixas têm a mesma altura.
- O centro de gravidade de uma caixa é seu centro geométrico;
- As caixas não se deformam devido a outras caixas empacotadas sobre delas.

O algoritmo é iterativo por natureza; uma caixa é selecionada e empacotada em cada ciclo de empacotamento. Há três passos principais em cada ciclo de empacotamento:

Passo 1. Todos os espaços disponíveis e caixas não empacotadas são identificados.

Passo 2. De acordo com algumas regras do algoritmo são avaliadas as possíveis combinações de caixas não empacotadas e espaços disponíveis. A combinação que minimiza o espaço não utilizado é selecionada.

Passo 3. A combinação selecionada é atualizada na representação do espaço.

O ciclo de empacotamento termina quando todas as caixas são empacotadas ou não há nenhum espaço disponível que possa ser aproveitado.

Bischoff e Ratcliff (1995a) tratam o problema de carregamento tridimensional de paletes. O padrão deve ter altura máxima H e o objetivo é minimizar o número de paletes requeridos para uma dada carga. Os autores propõem uma heurística em que os padrões são construídos em camadas horizontais, admitindo-se 2 tipos de caixas a serem empacotados em cada camada. O carregamento visa um arranjo eficiente e considera a estabilidade da carga. São considerados procedimentos sequenciais e paralelos para o carregamento dos paletes. Este problema é chamado na literatura de Problema de Carregamento de Paletes do Distribuidor, discutido no próximo capítulo.

Bischoff e Ratcliff (1995b) apresentam 2 heurísticas aplicadas ao problema de carregamento de contêiner. A primeira foi especificamente projetada para produzir carregamentos estáveis, enquanto a segunda heurística possui características que a tornam particularmente adequada para situações em que a carga tem vários destinos (*multi-drop situations*). Para mensurar a estabilidade do carregamento os autores usaram 2 medidas, posteriormente adotadas por diversos autores, e discutidas na Seção 3.4.

Gehring e Bortfeldt (1997) tratam do carregamento de carga fortemente heterogêneo em um único contêiner. É proposta uma heurística composta de três passos principais:

Passo 1. Um dado número de caixas é arranjado em um conjunto de torres de caixas disjuntivas. As caixas nas torres geradas precisam ter sua base totalmente circunscrita à área da caixa abaixo dela (estabilidade vertical) e a base da torre em contato com o chão do contêiner deve consistir de uma única caixa. Este passo é gerado por meio de um algoritmo guloso que minimiza o espaço vazio sobre as caixas.

Passo 2. As torres são arranjadas no chão do contêiner por meio de um algoritmo genético que maximiza a área utilizada da base do contêiner.

Estes passos são repetidos um certo número de vezes. Cada repetição inclui a geração de uma variante nova do conjunto de torres e a cobertura do chão do contêiner com o conjunto de torres encontrado no passo 1. Ao final, a melhor solução obtida é retornada.

Gehring e Bortfeldt (2002) apresentam um algoritmo genético paralelo para o carregamento de uma carga fortemente heterogênea em um único contêiner. Várias populações separadas são sujeitas a um processo evolucionário independente, e os melhores indivíduos são trocados entre as subpopulações.

Eley (2002) estudam o empacotamento de caixas em um ou múltiplos contêineres. O método proposto utiliza uma heurística de construção de cubóides, o que, de acordo com o autor, apresenta várias vantagens práticas, dentre as quais ressalta-se:

- As instruções para a execução do carregamento tendem a ser mais simples, possibilitando uma diminuição no tempo de execução da operação.
- Restrições de carregamento de caixas de um mesmo tipo próximas umas as outras é naturalmente satisfeita, pois os cubos são homogêneos.
- A estrutura cubóide diminui o risco de deslizamento das caixas.

No primeiro passo do procedimento proposto, uma heurística gulosa seleciona tipos de caixas em ordem decrescente de volume. Para um dado tipo de caixa, a heurística avalia todos os espaços vazios no contêiner e orientações possíveis para a formação de cubóides. O melhor cubóide obtido com aquele tipo de caixa é, então, empacotado. Antes que estas operações sejam aplicadas ao próximo tipo de caixa, espaços vazios no contêiner são atualizados (um passo adicional é a junção de espaços vazios adjacentes que formem cubos). Em seguida a solução gerada é melhorada por um procedimento de busca em árvore. O nó raiz representa o contêiner vazio, e cada nó seguinte representa o contêiner parcialmente preenchido. Se há m tipos de caixas, cada uma dessas soluções parciais podem ser ramificadas em $6m$ novas soluções parciais caso todas as caixas possam ser empacotadas em qualquer uma das 6 possíveis orientações. Como o número de nós aumenta consideravelmente com o tamanho do problema, apenas um conjunto de nós dentre os melhores avaliados é expandido. A função de avaliação considera não somente o volume utilizado, mas também o potencial de preencher os espaços vazios com caixas ainda não empacotadas, derivados do limitante inferior obtido com a heurística gulosa. Uma limitação do método diz respeito ao fato de que um dado padrão pode ser obtido por diferentes seqüências de empacotamento. Em vista disso, algumas regras simples foram implementadas para eliminar ramos da árvore associados a

estas seqüências. Duas estratégias foram implementadas para o problema de múltiplos contêineres: seqüencial (um contêiner é empacotado após o outro) e a estratégia simultânea (um dado número de contêineres é empacotado simultaneamente).

Lodi *et al.* (2002a) abordam o problema de empacotamento de *bins*, em que os itens têm orientação fixa, isto é, não podem ser rotacionados. O algoritmo proposto empacota os itens em camadas; os itens são posicionados com suas bases sobre o piso de um *bin*, sendo que a primeira camada coincide com ele. Os pisos das camadas subseqüentes são definidos pelo item mais alto da camada imediatamente inferior. Para produzir um empacotamento “efetivo” em camadas devem ser considerados dois aspectos possivelmente conflitantes: (i) obter um bom preenchimento vertical, empacotando itens com altura semelhante na mesma camada, e (ii) obter um bom preenchimento horizontal, resolvendo o problema bidimensional. Um algoritmo de busca tabu desenvolvido por Lodi *et al.* (1999) é proposto para resolver o problema. A principal característica deste algoritmo é uma estrutura de vizinhança parametrizável que varia de tamanho dinamicamente durante a busca e é independente das especificidades do problema considerado. O algoritmo não satisfaz restrições de estabilidade vertical.

Moura e Oliveira (2005) tratam o problema de carregamento por meio de uma abordagem GRASP. A heurística proposta é baseada no algoritmo de George e Robinson (1980) e inclui considerações de estabilidade da carga como restrição na fase construtiva do algoritmo, e a maximização do volume do contêiner como objetivo, tanto na fase de construção como na busca local em que melhorias são obtidas. A busca local seleciona aleatoriamente um ponto na seqüência do empacotamento e remove todas as caixas desta posição até o fim da seqüência. A partir da solução parcial, durante a reconstrução, a caixa da posição selecionada é proibida de ser escolhida na primeira iteração e todas as decisões utilizam um critério guloso, sem aleatoriedade.

O desempenho do algoritmo proposto é avaliado em termos da estabilidade do carregamento (utilizando as duas medidas de estabilidade propostas em Bischoff e Ratcliff, 1995b) e do volume ocupado. O método é comparado com nove algoritmos da literatura, utilizando instâncias de Bischoff e Ratcliff (1995b). Na média geral, o algoritmo proposto obtém o quarto melhor resultado.

Bischoff (2006) aborda o problema com um único contêiner em que o peso suportável da carga é um fator chave. É determinado o tipo de caixa e onde e como ela será posicionada no contêiner, levando em consideração combinações de espaços disponíveis,

caixas não empacotadas e suas orientações permitidas. Para implementar essas combinações o autor propõe uma adaptação no procedimento de Ngoi *et al.* (1994). As combinações viáveis são avaliadas levando em consideração 5 diferentes fatores: (i) relação entre tamanho da caixa e posição da superfície da carga; (ii) relação entre as dimensões da caixa e do espaço; (iii) geração de espaço não utilizado; (iv) potencial para construir colunas com caixas idênticas, e (v) perda relativa da capacidade que a carga pode suportar, ou seja, a diferença entre a capacidade de carga da área sobre a qual a caixa seria colocada e capacidade de peso mínimo de permanência em sua superfície superior. Somente uma caixa é empacotada em cada iteração. Também foi estudado o efeito de cada fator baseado na escolha de cada fator com distribuição uniforme no intervalo (0,1).

Araújo e Armentano (2007) estudam o problema com um único contêiner sujeito às restrições de orientação e de estabilidade. Os autores desenvolveram uma heurística construtiva com múltiplos reinícios com vistas à construção de cubóides máximos em espaços vazios, visando a maximização de ocupação destes espaços. Em cada passo do processo construtivo, um cubóide é selecionado probabilisticamente de uma lista restrita de candidatos. Segundo os autores, procedimentos de melhoria das soluções geradas não foram considerados devido à dificuldade de criação de vizinhanças.

3.3.2 Métodos Heurísticos para Carregamentos Homogêneos

Um dos primeiros estudos reportados é apresentado em Han *et al.* (1989), em que um algoritmo de programação dinâmica, semelhante à Steudel (1979), é proposto para o carregamento em um contêiner com vistas à maximização da utilização do volume. O contêiner é preenchido em camadas, as quais, diferentemente das abordagens encontradas na literatura, são construídas por prismas (sólidos geométricos delimitados por faces planas cujas bases se situam em planos paralelos) com formato em L (Figura 3.5). Apenas uma instância (com $(L, W, H) = (48, 42, 40)$ e $(l_i, w_i, h_i) = (11, 6, 6)$) é utilizada para avaliar o algoritmo. A solução obtida utiliza 95,76% do volume do contêiner, correspondente a 16,07% de melhoria em relação ao resultado obtido em *General Services Administration* (1966).

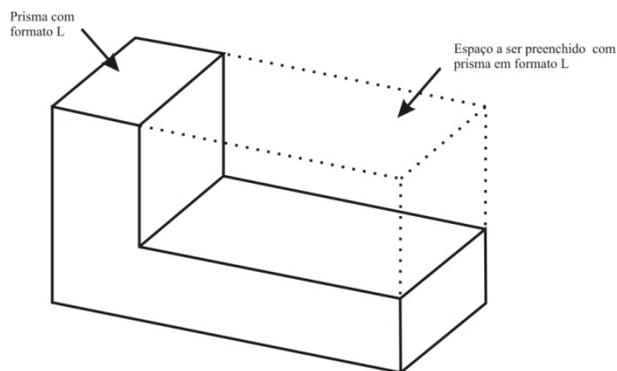


FIGURA 3.5 – Prisma.

George (1992) propõe um algoritmo de construção em camadas, as quais podem ser criadas a partir de qualquer lado do contêiner. A carga a ser transportada pode ser suficientemente grande para necessitar de mais de um contêiner. Os principais passos do método são apresentados a seguir:

Passo 1. Considere o carregamento, iniciando-se do fundo para frente do contêiner:

- a. Escolha cada dimensão da caixa, em ordem, para definir a dimensão de profundidade da camada.
- b. Use um algoritmo para o problema bidimensional com vistas a encontrar o melhor padrão (melhor camada) dentre os 3 possíveis padrões de carregamento para as dimensões de largura (W) e altura (H). O algoritmo empacota as caixas em pilhas, assegurando que as caixas tenham a base totalmente suportada. São feitas mudanças de orientações nas caixas, a fim de mudar as dimensões das pilhas.

Passo 2. Use a melhor camada encontrada no passo anterior para carregar o contêiner, preenchendo-o com o maior número de camadas possíveis.

Passo 3. Modifique o número de camadas, desempacotando as caixas, camada por camada, e substituindo por camadas que melhor utilizam o espaço disponível, entre a profundidade (comprimento) disponível do contêiner e a largura da camada.

Passo 4. Combine qualquer espaço disponível para empacotar caixas na camada.

Passo 5. Reorienta o contêiner para empacotar as caixas de uma lateral a outra e a seguir, do chão ao teto do contêiner. Repita os passos anteriores para essas duas orientações.

Passo 6. Escolha a melhor das 3 orientações.

Para facilitar o empacotamento, o carregamento é mostrado do fundo para frente do contêiner. Segundo o autor, o algoritmo de George e Robinson (1980) desenvolvido para

carregamento heterogêneo requer uma heurística complexa, que não apresenta um bom desempenho quando o carregamento é homogêneo. Para comprovar esse comportamento, o autor compara as duas heurísticas para 7 faixas de variações de tamanhos das caixas. A heurística de George (1992) apresentou resultados superiores em todas as faixas. Foi desenvolvido também um conjunto de limitantes superiores na solução.

Liu e Hsiao (1997) consideram o carregamento em um palete de dimensões (L, W) , sujeito à altura máxima H e à carga máxima M . Os autores propõem um algoritmo de carregamento em camadas para obtenção do padrão de maior estabilidade de carga dentre aqueles com maior utilização do volume do contêiner. O método consiste dos seguintes passos:

Passo 1. Para cada face da caixa (l, w) , (l, h) e (h, w) é aplicada uma versão modificada da heurística de Smith e De Cani (1980) para resolução do problema bidimensional. As soluções obtidas com as faces (l, w) , (l, h) e (h, w) correspondem respectivamente às camadas denominadas como de tipo B , S e E .

Passo 2. Cada um dos 3 padrões de camadas obtidos no passo anterior é rotacionado para criação de 3 outros padrões de camadas, totalizando um máximo de 12 padrões distintos. Apesar dos padrões de camadas resultantes da rotação serem de mesmo tipo da camada original, cada um destes, juntamente com o padrão da camada original, caracteriza um diferente subtipo.

Passo 3. Um programa linear sujeito a restrições de altura máxima H , carga máxima do palete M e limitantes superiores triviais é aplicado para determinação do máximo número de caixas no carregamento. O programa possui 3 variáveis inteiras que indicam o número de camadas de tipo B , S e E no padrão. Em outras palavras, é resolvido um problema da mochila multidimensional (DYCKHOFF e FINKE, 1992). Deve-se ressaltar que as restrições de peso se aplicam somente ao palete, não restringindo o peso que cada caixa pode suportar.

Passo 4. São calculados os coeficientes de estabilidade Ω_{ij} entre camadas do subtipo i quando posicionadas sobre camadas do subtipo j . Os coeficientes são calculados segundo critérios apresentados no trabalho de Carpenter e Dowsland (1985). Dentre estes critérios, incluem-se o número de caixas na camada que é suportada por mais de uma caixa e a porcentagem da base da caixa que é suportada por outra. Como há no máximo 12 padrões de camadas distintos, são calculados 144 coeficientes.

Passo 5. Um programa linear computa o subtipo de cada camada do padrão com vistas à maximização da estabilidade do carregamento. Para tal, são considerados os valores dos coeficientes Ω_{ij} obtidos no passo 4.

Lins *et al.* (1999) propõem um algoritmo recursivo para o problema de carregamento de paletes tridimensional ou de contêineres, correspondente a uma generalização da heurística de Morabito e Morales (1998) para o problema de carregamento de paletes. O método representa empacotamentos factíveis como designações factíveis de profundidade em uma tríade (3 “fios” representando as dimensões X , Y e Z). Cada designação corresponde a uma partição do contêiner em subcontêineres específicos. Cada subcontêiner pode ser recursivamente dividido em subcontêineres adicionais, de acordo com as especificações na tríade. Um subcontêiner que não é dividido é carregado com caixas de mesma orientação. Não há restrições de orientação das caixas, no entanto o algoritmo pode ser facilmente adaptado para o caso em que a orientação é fixada.

Posteriormente, Lins *et al.* (2002) estenderam o algoritmo anterior para o tratamento do problema de empacotamento de caixas com n dimensões em um n -contêiner. Para o caso especial de $n=3$, soluções obtidas com um conjunto de 14 exemplos são reportadas para 4 versões denominadas $2-D$, lk , gui , e $3-D$. O algoritmo $2-D$ admite que a orientação vertical das caixas é fixada, e consiste em resolver o problema bidimensional e adicionar camadas idênticas enquanto não se ultrapasse o teto do contêiner. O algoritmo gui é o algoritmo recursivo com $n=3$ e nível de recursividade $k=2$ ou 3 , sem fixação da orientação vertical das caixas e limitado a padrões guilhotinados. O algoritmo lk admite que a orientação vertical das caixas não é fixada e resolve o problema dividindo o padrão de carregamento em camadas, em que cada camada é um padrão bidimensional com caixas de mesma altura. Para três opções de empilhamento das camadas no contêiner, são resolvidos problemas da mochila para obter o número de camadas de cada tipo, e o melhor padrão tridimensional resultante é selecionado. Finalmente, o algoritmo $3-D$ é o algoritmo recursivo com $n=3$ e nível de recursividade $k=2$ ou $k=3$, sem fixação da orientação vertical das caixas. Dentre os 4 algoritmos, $3-D$ é o que apresenta os melhores resultados em número de caixas para todas as instâncias do conjunto.

3.4 Estabilidade do Carregamento

A estabilidade da carga é uma consideração de grande relevância prática no carregamento tridimensional, uma vez que o deslocamento das caixas dentro do contêiner pode levar a danos e à perda da carga. Vários autores (BISCHOFF e RATCLIFF, 1995b; GEHRING e BORTFELDT, 1997; MOURA e OLIVEIRA, 2005; ARAÚJO e ARMENTANO, 2007) têm considerado esta questão em seus trabalhos, com o objetivo de definir e aplicar critérios quantitativos que assegurem carregamentos com um grau aceitável de estabilidade. Esta seção descreve como a estabilidade estática da carga é abordada nos trabalhos revisados nas seções anteriores. Deve-se ressaltar, entretanto, que não existe um consenso sobre estes critérios, e no tocante à estabilidade dinâmica (capacidade da carga manter sua posição com o contêiner em movimento), poucos trabalhos são encontrados na literatura.

3.4.1 Carregamento heterogêneo

Conforme mencionado na Seção 3.3.1, Bischoff e Ratcliff (1995b) propõem 2 medidas para mensurar a estabilidade do carregamento.

M1. Número médio de caixas que dão suporte a cada caixa posicionada acima do piso do contêiner (quanto maior o valor, melhor). Esta medida procura medir a estabilidade vertical do carregamento.

M2. Porcentagem média de caixas que não possuem pelo menos três faces laterais em contato com outras caixas (quanto menor o valor, melhor). Esta medida procura medir a estabilidade horizontal do carregamento.

De acordo com Araújo (2006), os critérios de estabilidade de Bischoff e Ratcliff (1995b) implicam que para garantir que o centro de gravidade das caixas seja amparado, a porcentagem da base de cada caixa apoiada por outras caixas deva ser igual ou superior a 55%.

Gehring e Bortfeldt (1997) calculam a estabilidade vertical de cada caixa i do carregamento como a porcentagem da área da base em contato com as caixas abaixo de i , dividida pela área da base total de i . Para a caixa ser estável, o resultado não pode estar abaixo

de um valor dado. No método proposto por estes autores, a restrição de estabilidade das caixas é naturalmente satisfeita pela lei de formação das pilhas. Somente uma caixa do carregamento, denominada caixa base, deve ser a base de uma pilha. Qualquer outra caixa deve ter a sua base totalmente suportada pela caixa abaixo (Figura 3.6(a)).

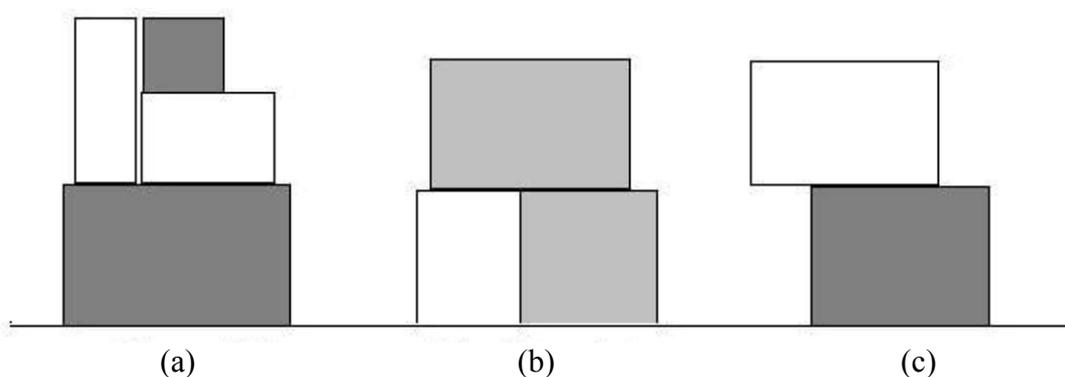


FIGURA 3.6 – Estabilidade vertical segundo Gehring e Bortfeldt (1997): (a) Pilha aceitável (estável), (b), e (c) Pilhas não aceitáveis (instáveis).

Moura e Oliveira (2005) e Araújo e Armentano (2007) utilizam critérios idênticos para mensurar a estabilidade da carga. Segundo estes autores, a carga é considerada estável se:

1. Todas as caixas são totalmente suportadas;
2. Várias outras caixas suportam uma caixa;
3. Todas as caixas têm pelo menos 3 faces laterais em contato com outras caixas.

Os autores utilizam as medidas M1 e M2 de Bischoff e Ratcliff (1995b) para mensurarem os critérios 2 e 3, respectivamente.

3.4.2 Carregamento homogêneo

Liu e Hsiao (1997) consideram os seguintes dois critérios de estabilidade vertical, por sua vez, baseados em Carpenter e Dowsland (1985):

1. Uma caixa deve ser suportada por no mínimo 2 caixas da outra camada; os autores não mencionam como é o padrão de carregamento nas paredes do palete, pois para que todas as caixas sejam suportadas por no mínimo

2 caixas, há uma perda de espaço nas paredes do palete. Além disso, mais de $x\%$ da área da base de cada caixa precisa estar em contato com cada caixa que a suporta. Segundo os autores, o valor de x deve pertencer ao intervalo $[5,15]$. A Figura 3.7 ilustra um carregamento verticalmente estável segundo este critério, em que a linha em vermelho indica a menor área da base da caixa que está sendo suportada por uma das caixas abaixo.

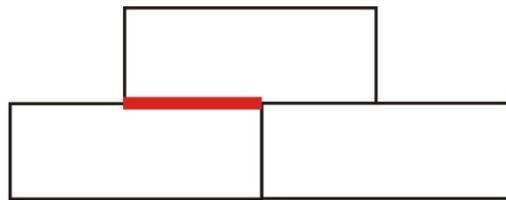


FIGURA 3.7 – Carregamento estável verticalmente segundo critério 1 de Liu e Hsiao (1997).

2. Cada caixa deve ter no mínimo $y\%$ da área da base em contato com as caixas da camada abaixo. Segundo os autores, o valor de y deve pertencer ao intervalo $[75,95]$. A Figura 3.8 ilustra este critério, em que as linhas em vermelho indicam a área da base das caixas suportadas por outras caixas abaixo.

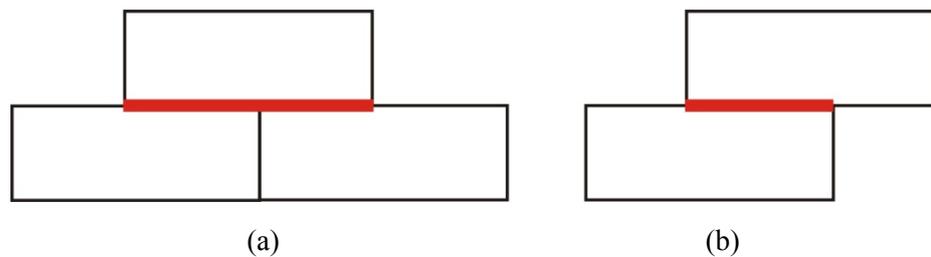


FIGURA 3.8 – Estabilidade vertical segundo critério 2 de Liu e Hsiao (1997): (a) carregamento estável; (b) carregamento instável.

4 UM ALGORITMO DE BUSCA TABU PARA O PROBLEMA DE CARREGAMENTO DE CONTÊINER COM CAIXAS IDÊNTICAS

Dentre os vários trabalhos na literatura que tratam do problema de carregamento de contêineres, a grande maioria aborda o caso mais geral do empacotamento de caixas heterogêneas. Em outras palavras, o caso particular em que a carga consiste de caixas idênticas é relativamente pouco explorado na literatura. De acordo com a classificação de Dyckhoff (1990), este problema é classificado como 3/B/O/C. Em Wascher *et al.* (2007), esse problema é classificado como IIPP (veja Tabela 2.2).

Não obstante, a idéia de elaborar uma heurística específica para o carregamento de contêiner com caixas idênticas (PCCCI) é justificada pela necessidade comercial de vários produtores que possuem cargas de tamanho considerável de um único tipo de produto. Apesar de ser possível estabelecer um carregamento satisfatório utilizando uma heurística geral como, por exemplo, via os métodos apresentados na seção 3.2.1.1, a qualidade destas soluções pode ser melhorada a partir da observação de que vários destes métodos utilizam critérios de decisão que não fazem sentido no PCCCI (GEORGE, 1992; CECILIO 2003). Um exemplo destes critérios é a escolha do tipo de caixa com o maior valor de uma determinada dimensão para iniciar uma camada ou pilha (GEORGE e ROBINSON, 1980; GEHRING *et al.*, 1990).

Neste capítulo é descrito um procedimento heurístico para o PCCCI. O procedimento corresponde a uma extensão do algoritmo de busca tabu de Pureza e Morabito (2006), originalmente elaborado para problema bidimensional de carregamento de paletes do produtor (PCPP). Dada a sua relevância para o presente trabalho, a Seção 4.1 apresenta uma descrição detalhada do PCPP e um resumo da literatura em métodos heurísticos. Na Seção 4.2, a heurística de Pureza e Morabito é descrita, seguindo-se à extensão proposta para o PCCCI (Seção 4.3).

4.1 O Problema do Carregamento de Paletes

O problema de carregamento de paletes (PCP) consiste em arranjar ortogonalmente e sem sobreposição o máximo número de caixas retangulares sobre um palete retangular. Hodgson (1982), ao estudar o problema, dividiu-o em dois possíveis casos: o

problema de carregamento de paletes do distribuidor (PCPD) e o problema de carregamento de paletes do produtor (PCPP).

No primeiro caso, o distribuidor recebe caixas de diferentes dimensões (l_i, w_i, h_i) $i=1, \dots, m$, de diversos fornecedores. As caixas são então carregadas sobre paletes de dimensões (L, W, H) (Figura 4.1a). Este problema possui, portanto, três dimensões relevantes, e pode ser reduzido ao problema bidimensional de se obter o arranjo ortogonal de retângulos, por exemplo (l_i, w_i) , dentro do retângulo (L, W) que maximize a utilização da área de (L, W) caso a orientação vertical de caixas seja fixada.

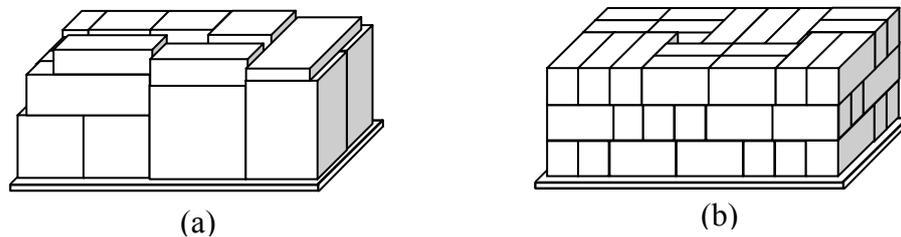


FIGURA 4.1 – Exemplos de padrões de carregamento para (a) PCPD, e (b) PCPP (Fonte: MORABITO e PUREZA, 2008).

No segundo caso, o produtor fabrica um bem que é embalado em caixas iguais de dimensões (l, w, h) . Estas caixas devem ser então carregadas em camadas horizontais sobre paletes de dimensões (L, W, H) (onde H é a altura máxima do carregamento). Se as caixas puderem ser arranjadas sobre quaisquer de suas faces $(lw, lh$ e $wh)$, pode-se decompor o problema do produtor em dois subproblemas: (i) para cada face das caixas, o problema bidimensional de arranjar ortogonalmente o máximo número de retângulos dentro do retângulo (L, W) , e (ii), o problema unidimensional de arranjar camadas horizontais ao longo da altura H do palete com o máximo número de caixas. Se uma orientação vertical das caixas for fixada, por exemplo, se as caixas devem ser carregadas sobre sua face lw , então o máximo número de camadas horizontais (com dimensões (L, W, h)) empilhadas sobre o palete é dado por $\lfloor H/h \rfloor$ (em que $\lfloor x \rfloor$ denota o maior inteiro menor ou igual a x). Caso contrário, o problema pode ser visto como um problema da mochila com capacidade H .

O problema de carregamento de paletes do produtor do tipo (i) é tratado com relativo sucesso por heurísticas construtivas conhecidas como *heurísticas de bloco* (BISCHOFF; DOWSLAND, 1982; MORABITO; MORALES, 1998, 1999; NELISSEN, 1994, 1995; SCHEITHAUER; TERNO, 1996; STEUDEL, 1979, BIRGIN *et al.*, 2009). Tais procedimentos pesquisam “*layouts*” de padrões que resultam em um bom ajuste em torno das

bordas do palete, e estendem estes padrões ao centro do palete de maneira a formar um ou mais blocos retangulares em que as caixas possuem a mesma orientação pré-fixada (Figura 4.2). Cada *layout* resultante corresponde a uma solução, e aquela com o maior número de caixas é selecionada. Conforme discutido na Seção 2.1, este problema é chamado de bidimensional, pois considera duas dimensões relevantes - comprimento e largura do palete (L, W), e comprimento e largura das caixas (l, w) - e duas possíveis orientações de caixas em relação ao comprimento L do palete - dimensão l e dimensão w . Note também que estas orientações definem os retângulos (l, w) e (w, l) .

Quanto maior o número de blocos considerado, maior tende a ser a qualidade do padrão de carregamento gerado. Um refinamento do conceito de heurísticas de bloco é proposto em Morabito e Morales (1998, 1999). Neste trabalho, um algoritmo aplica recursivamente a heurística de 5 blocos de Bischoff e Dowsland (1982) na área de cada bloco gerado.

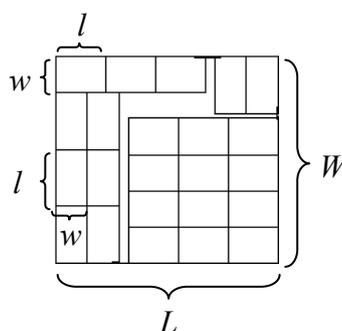


FIGURA 4.2 – Um padrão de carregamento com 4 blocos para a instância $(L, W, l, w) = (61, 44, 9, 7)$.

Padrões de carregamento podem ser classificados em guilhotinados, não-guilhotinados de 1ª ordem, e não-guilhotinado de ordem superior (ARENALES e MORABITO 1995). Um padrão guilhotinado é obtido por sucessivos cortes guilhotinados, por sua vez definidos como cortes que quando aplicados a um retângulo, produzem 2 novos retângulos (Figura 4.3a). Um padrão não-guilhotinado de 1ª ordem é obtido por sucessivos cortes guilhotinados e/ou cortes não-guilhotinados de 1ª ordem. Um corte é chamado não-guilhotinado de 1ª ordem se, quando aplicado a um retângulo, produz 5 novos retângulos arranjados de maneira a não formar um padrão guilhotinado (veja corte em negrito na Figura 4.3b). Finalmente, um padrão não-guilhotinado de ordem superior é aquele que não pode ser obtido por sucessivos cortes guilhotinados e/ou cortes não-guilhotinados de 1ª ordem (Figura 4.3c).

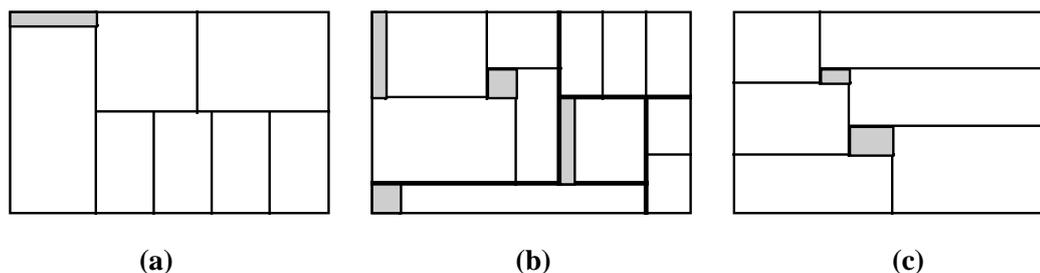


FIGURA 4.3 - (a) padrão guilhotinado, (b) corte (em negrito) e padrão não-guilhotinado de 1ª ordem, (c) padrão não-guilhotinado de ordem superior. (Fonte: PUREZA e MORABITO, 2006)

Em várias heurísticas de bloco, padrões de carregamento gerados estão limitados a padrões não-guilhotinados de 1ª ordem. O algoritmo recursivo de Morabito e Morales (1998, 1999), por exemplo, foi capaz de resolver otimamente mais de 99,9% dentre 20.000 exemplos analisados. Os exemplos não resolvidos correspondem a situações cuja solução ótima é um padrão não guilhotinado de ordem superior. Posteriormente, esta heurística foi generalizada em Lins *et al.* (2002) para o tratamento do problema de empacotamento de caixas com n dimensões em um n -contêiner. Assim como sua versão bidimensional, o algoritmo provê soluções ótimas limitadas a padrões não-guilhotinados de 1ª ordem.

Em vista desta limitação, Pureza e Morabito (2006) propõem um algoritmo de busca tabu, em que movimentos de expansão de blocos são aplicados a partir da solução gerada pela heurística de 4 blocos proposta em Smith e De Cani (1980). Esta abordagem, descrita na próxima seção, é capaz de tratar satisfatoriamente situações em que a solução ótima represente qualquer tipo de padrão.

4.2 O Algoritmo de Pureza e Morabito

O algoritmo tabu de Pureza e Morabito (PM) é um procedimento heurístico iterativo no qual a busca por melhores padrões de empacotamento ocorre no espaço factível, Isso permite que uma solução esteja disponível em qualquer iteração. O algoritmo consiste nos passos descritos na Figura 4.4:

Passo 1. Gere um padrão inicial S_0 . Faça $f(S_0)$ =número de caixas em S_0 , blk =número de blocos da solução corrente e $ótimo$ =falso.

Passo 2. Faça $it=0$, $S=S_0$, $f(S)=f(S_0)$, $S^*=S$, $f(S^*)=f(S)$ e lista tabu $L=\emptyset$. Se $f(S_0)$ atender critérios de otimalidade, faça $ótimo$ =verdadeiro e vá para o passo 4.

Passo 3. Repita até que $mv=0$ ou it =máximo número de iterações ou $ótimo$ =verdadeiro

3.1 Para cada bloco $i=1,\dots,blk$, calcule os movimentos possíveis nas quatro direções. Faça mv =número total de movimentos de todos os blocos.

3.2 Se $mv>0$, selecione o movimento não tabu com a maior avaliação. Seja S' o padrão resultante da aplicação deste movimento. Se todos os movimentos estão proibidos, elimine o elemento mais antigo de L (t -ésimo elemento) e repita o passo. Caso contrário, efetive o movimento, gerando um novo padrão e faça $S=S'$. Se $f(S) > f(S^*)$, faça $S^*=S$ e $f(S^*)=f(S)$. Se S^* atende critérios de otimalidade, faça $ótimo$ =verdadeiro e vá para o passo 4. Caso contrário, faça $it=it + 1$.

3.3 Aplique o procedimento de deslizamento de blocos para a extremidade do palete mais próxima a eles até que nenhum deslizamento seja verificado.

Passo 4. Se $ótimo$ =verdadeiro, pare pois S^* corresponde a um padrão ótimo. Caso contrário, se $mv=0$ e $0 < it < \text{número máximo de iterações}$, o padrão final convergiu para um único bloco; se $mv=0$ e $it=0$, o padrão inicial S_0 é homogêneo.

FIGURA 4.4 – Passos do algoritmo PM para o PCPP.

O padrão inicial (passo 1) é provido pela heurística de blocos proposta em Smith e De Cani (1980). Esta heurística gera até 4 blocos de caixas arranjadas conforme uma orientação pré-fixada das caixas. Note que existem duas possíveis orientações de arranjo de caixas: com a face l ou face w paralela à dimensão L do palete. O procedimento varia as dimensões dos blocos (inclusive com a possibilidade de serem nulas) examinando todas as possíveis modificações nestes blocos e retornando o padrão que resulta no maior número de caixas dentro dos blocos. Um exemplo de solução inicial com 4 blocos é ilustrada na Figura 4.2.

Seguindo-se à geração do padrão inicial, inicia-se a fase de busca de padrões de melhor qualidade. Padrões candidatos à seleção resultam de *movimentos* aplicados aos blocos do padrão corrente (passo 3.1). Cada movimento é definido por 5 parâmetros, os quais são

discutidos a seguir:

1) Bloco a ser expandido

Cada bloco a ser expandido, denotado por *bloco ativo*, é identificado por um número $i \in \{1, 2, \dots, blk\}$ em que *blk* é o número de blocos no padrão corrente.

2) Direção de expansão

São consideradas 4 direções de expansão aplicadas a cada face do bloco, a saber, “à direita”, “à esquerda”, “acima”, e “abaixo” (Figura 4.5).

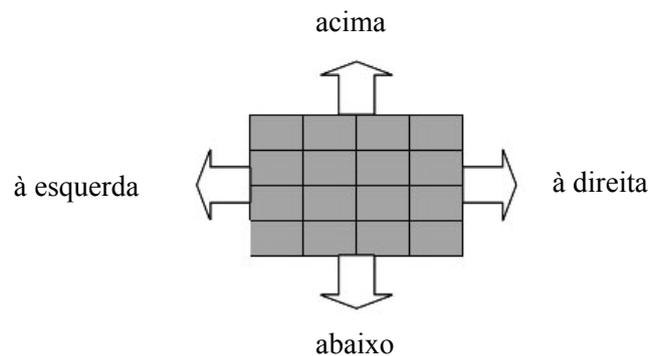


FIGURA 4.5 – Direções de expansão.

3) Camadas a serem expandidas

A expansão de um dado bloco ativo é feita tomando-se quaisquer camadas da face de expansão, desde que adjacentes. Na Figura 4.6, as expansões para a direita das camadas 1 e 3, 1 e 4, e 2 e 4 não são permitidas, uma vez que as camadas em cada par são não adjacentes. Quando a expansão contempla todas as camadas da face do bloco (Figura 4.6b), o resultado é o mesmo bloco aumentado. Caso apenas algumas camadas da face se expandam, o bloco ativo é dividido em até 3 outros blocos (Figura 4.6c).

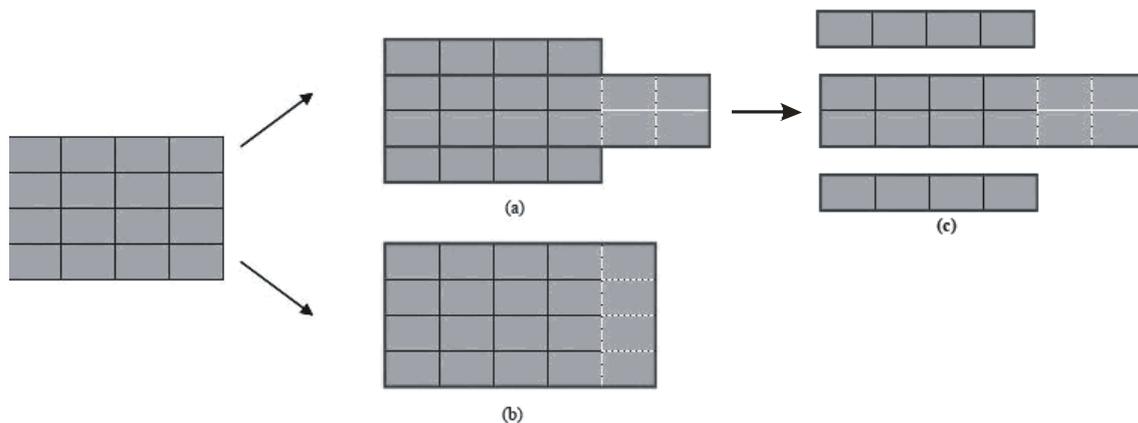


FIGURA 4.6 – Expansão de um bloco para a direita: (a) 2 camadas da face expandidas; (b) face inteira expandida; (c) bloco ativo é dividido em até 3 outros blocos.

4) Número de caixas adicionadas a cada camada de expansão

Este número pode ser qualquer desde que os limites do palete não sejam ultrapassados. Na Figura 4.6a, 2 caixas foram adicionadas às camadas 2 e 3 da face direita do bloco, enquanto na Figura 4.6b, 1 caixa foi adicionada às camadas 1-4 da mesma face.

5) Configuração das áreas remanescentes dos blocos passivos

Movimentos de expansão de blocos ativos geralmente resultam na superposição de blocos nas adjacências, denotados de *blocos passivos*. Quando há superposição, blocos passivos são eliminados e novos blocos são criados nas áreas não superpostas. São consideradas 4 configurações candidatas para geração destes novos blocos. As configurações envolvem a divisão da área em até 3 áreas retangulares. A Figura 4.7 ilustra estas configurações para movimentos para direita (para as demais direções basta rotacionar consecutivamente os desenhos em 90°). Se a expansão do bloco ativo superpor mais de um bloco passivo, áreas remanescentes de cada um destes blocos são tratadas separadamente.

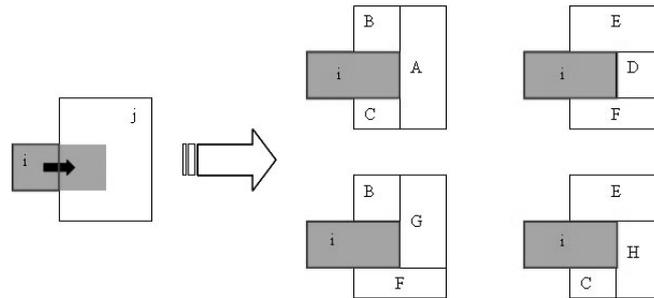


FIGURA 4.7 – Quatro possíveis configurações das áreas remanescentes do bloco passivo j com a expansão para a direita do bloco ativo i . (FONTE: PUREZA e MORABITO, 2006).

A orientação das caixas selecionada em cada área de uma dada configuração é tal que maximiza o número total de caixas. Uma vez calculadas as 4 possíveis configurações, a opção selecionada para o movimento é aquela que possui o maior valor segundo a função de avaliação dos movimentos candidatos, definida a seguir.

4.2.1 Avaliação de movimentos candidatos

Cada movimento é avaliado segundo a seguinte função:

$$G = \begin{cases} G_0 + \alpha \frac{area_h}{area_n} + (1 - \alpha) pen & \text{se } G_0 = 0 \text{ e } area_h = 0 \\ G_0 + \alpha \frac{area_h}{area_n} + (1 - \alpha) \frac{area_e - \sum_{j \in P} area_s(j)}{area_e}, & \text{caso contrário} \end{cases}$$

A função G é composta por 3 termos associados a diferentes critérios. O 1º critério (G_0) corresponde à diferença do número de caixas no padrão resultante do movimento e no padrão anterior (ganho real). G_0 é inteiro, podendo assumir valores positivos (aumento no número de caixas), negativo ou zero. No 2º critério $\left(\frac{area_h}{area_n}\right)$, $area_n$ é a área total dos novos blocos criados, e $area_h$ é a porção de $area_n$ em que os blocos têm orientação diferente do bloco ativo. Este critério estimula a geração de padrões com alto grau de heterogeneidade, ao privilegiar movimentos que geram padrões formados por blocos de diferentes orientações, intercalados entre si. Finalmente, no 3º critério $\left(\frac{area_e - \sum_{j \in P} area_s(j)}{area_e}\right)$, $area_e$ é a área total de expansão do bloco ativo e $area_s(j)$ é a porção de $area_e$ que superpõe um dado bloco passivo j .

Este critério privilegia movimentos que modificam minimamente a estrutura do padrão corrente. Caso o movimento tenha ganho real e grau de heterogeneidade ambos iguais a zero ($G_0=0$ e $area_h=0$), substitui-se o critério 3 por um termo de penalização.

G_0 é geralmente preponderante na escolha do movimento, enquanto os outros critérios são relevantes em caso de empates no ganho real. Os critérios 2 e 3 são utilizados por incorporarem uma medida de “qualidade estratégica” do movimento. Note que ambos são ponderados por um fator $\alpha \in [0,1]$, permitindo o controle da dominância de um sobre o outro. Os experimentos reportados pelos autores indicam o uso de $\alpha=0,5$.

4.2.2 Elementos de restrição tabu

Devido à boa qualidade da solução inicial, o padrão obtido é invariavelmente um ótimo local com relação à estrutura da vizinhança proposta. Diante disto torna-se necessário a utilização de mecanismos que restrinjam o retorno a padrões previamente obtidos para que a busca heurística se desenvolva além desse ponto. Para este fim foram incorporados elementos restritivos de busca tabu simples, cujo papel é armazenar e restringir certas características de soluções já visitadas por um determinado número de iterações.

Nesta heurística foi utilizada uma lista tabu de tamanho fixo *tsize* em que após cada movimento, são armazenadas: as coordenadas originais do bloco ativo e as novas coordenadas dos blocos passivos. Proíbe-se, então, que blocos com estas coordenadas sejam ativos nas próximas *tsize* iterações, no entanto não os impedem de serem passivos.

Nos experimentos reportados, *tsize* é calculado por uma função linear do número de caixas do padrão inicial.

4.2.3 Recursos adicionais

Sempre que um movimento é realizado, é comum o surgimento de espaços vazios entre os blocos do padrão corrente. Estes espaços são aproveitados de duas maneiras:

- Deslizando-se os blocos para as extremidades do palete mais próximas a eles. O deslizamento dos blocos é feito repetidamente a partir das caixas mais próximas às extremidades do contêiner, até que nenhum deslizamento seja verificado (Passo 3.3 da Figura 4.4). Conforme observado na Figura 4.8, ele permite que espaços não utilizados se

concentrem no centro do palete, favorecendo a inclusão de mais caixas no padrão em movimentos posteriores.

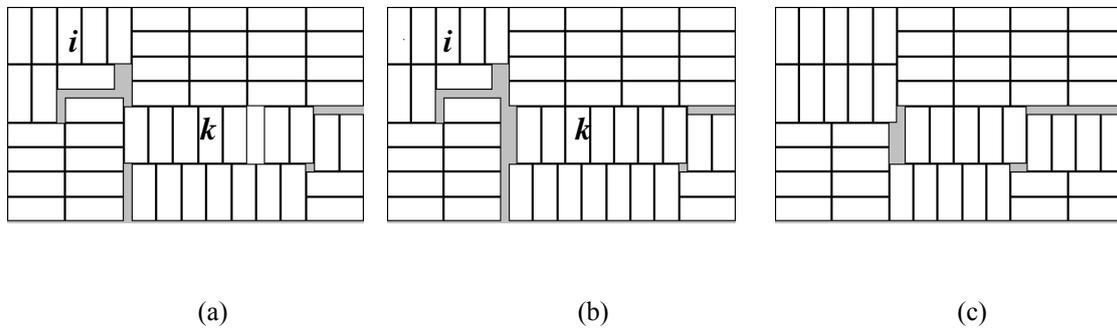


FIGURA 4.8 – (a) Padrão de carregamento com 51 caixas (iteração it); (b) aplicação de deslizamento no padrão em (a); (c) padrão de carregamento com 52 caixas após dois movimentos (iteração $it + 2$). (Adaptada de PUREZA e MORABITO, 2006).

- Adicionando-se tiras de espaços vazios à área de blocos passivos. A incorporação de tiras de espaço vazios aumenta a área de blocos passivos, aumentando também a chance de inclusão de um número maior de caixas durante a reconfiguração da área não superposta (Figura 4.9).

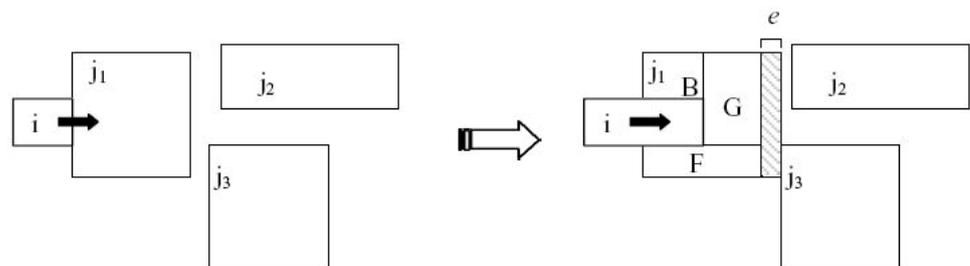


FIGURA 4.9 – Adição da área da tira de espessura e entre os blocos j_1 e j_3 à área do bloco j_1 . (FONTE: PUREZA e MORABITO, 2006).

4.3 A Extensão Proposta

Em nossa extensão da heurística de Pureza e Morabito (2006) para o tratamento do problema de carregamento de contêineres, os passos da heurística descritos na seção anterior são mantidos, e são efetuadas modificações que levam em conta as diferenças entre os dois problemas e refinamentos adicionais. Primeiramente, note que o problema de

carregamento de paletes considera duas dimensões relevantes (comprimento e largura do palete e das caixas) e duas possíveis orientações de caixas em relação ao comprimento L do palete (retângulos lw e wl). No problema de carregamento de contêineres passam a ser consideradas três dimensões (comprimento, largura e altura do contêiner e das caixas), e seis possíveis orientações de caixas em relação ao comprimento L do contêiner (retângulos lw , wl , lh , hl , wh e hw).

A seguir, são descritas as modificações em relação ao algoritmo PM.

4.3.1 Passo 1 – Geração do Padrão Inicial

Para a geração do carregamento inicial, a heurística proposta em Smith e De Cani (1980) também é utilizada. Note que no chão do contêiner (definido pelas dimensões L e W), as caixas podem ser posicionadas sobre 3 faces distintas: lw , lh ou wh . Para cada uma dessas 3 opções é aplicada a heurística para o preenchimento do chão do contêiner. Em seguida, o padrão resultante é estendido na direção da altura H do contêiner (adicionando camadas idênticas àquela gerada pela heurística de Smith e De Cani (1980)) enquanto não se ultrapasse o teto do contêiner. Dentre as três possibilidades, o padrão de carregamento que resulta no maior número de caixas é selecionado para compor a solução inicial. A Figura 4.10 ilustra um exemplo com os 3 padrões de preenchimento do chão do contêiner obtidos com a heurística. Note que apesar da heurística de Smith e De Cani gerar padrões de carregamento de até 4 blocos, neste exemplo os melhores padrões consistiram de apenas 2 blocos.

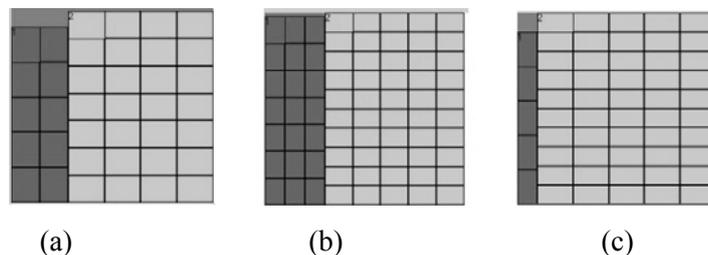


FIGURA 4.10 – Opções de carregamento do chão do contêiner para a instância $(L,W,H,l,w,h)=(50,50,50,9,7,5)$: (a) caixas sobre a face (l,w) , (b) caixas sobre a face (w,h) , (c) caixas sobre a face (l,h) .

Na Figura 4.10a, o padrão de carregamento do chão do contêiner com as caixas arranjadas sobre a face (l,w) apresenta uma camada com 38 caixas. Estendendo este padrão até o teto do contêiner (total de 10 camadas), é possível empacotar 380 caixas. Na Figura

4.10b (caixas arranjadas sobre a face (w,h)), obtêm-se um total de 355 caixas divididas em 5 camadas. Finalmente, na Figura 4.10c, o arranjo das caixas arranjadas sobre a face (l,h) resulta no empacotamento de 385 caixas divididas em 7 camadas. Neste exemplo, portanto, o padrão da Figura 4.10c é utilizada como solução inicial da extensão da heurística PM. Note que as camadas são adicionadas aos blocos existentes, ou seja, o número de blocos do padrão de carregamento não é alterado.

4.3.2 Passo 3.1 - Movimentos de expansão

A extensão da heurística do PCPP para o PCCCI implica necessariamente na consideração de uma dimensão adicional (H). Como blocos possuem 6 faces, têm-se 6 possíveis direções de expansão. Portanto, além das direções “à direita”, “à esquerda”, “acima” e “abaixo”, presentes no caso bidimensional, foram incluídas as direções “atrás” e “à frente” (Figura 4.11). Note também que para cada face de expansão, têm-se ainda 2 dimensões que indicam as camadas horizontais e verticais de caixas a serem expandidas. Observa-se assim que o aumento de única dimensão provoca um incremento considerável de movimentos de expansão.

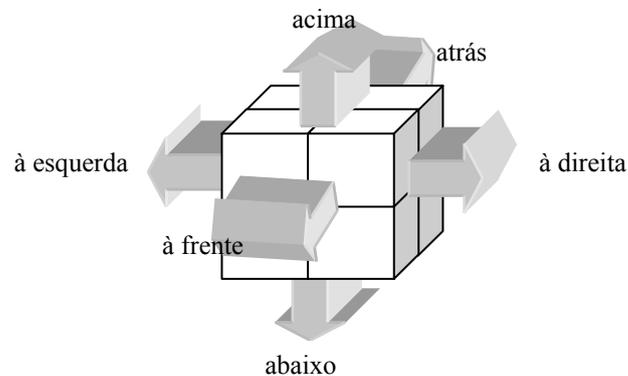


FIGURA 4.11 – As 6 direções de expansão em um carregamento tridimensional.

Nos exemplos tratados neste trabalho, o tamanho da vizinhança não trouxe maiores dificuldades do ponto de vista de memória e tempo computacional. Cabe ressaltar que foram encontradas instâncias para as quais a busca na vizinhança chega a investigar milhões de movimentos em apenas uma única iteração. Nestas situações, caso o espaço de memória requerido apresente limitações, apenas uma parcela dos melhores movimentos poderia ser armazenada para seleção. Em relação a tempos computacionais excessivos para a

aplicação, a geração de movimentos poderia ser restringida a um percentual do número total. A definição deste percentual deve considerar o *trade-off* tempo vs qualidade da solução.

4.3.3 Passo 3.1b - Movimentos adicionais

Além de movimentos de expansão, são propostos outros dois tipos de movimento. No primeiro destes, são investigadas possíveis mudanças de orientação das caixas de cada bloco. São desconsideradas mudanças de orientação que resultem na sobreposição do bloco em questão a outros blocos do padrão, ou seja, o movimento pode ser visto como um operador unário.

Caso haja mais de uma orientação factível, é selecionada aquela que resulta no maior ganho de acordo o primeiro termo (G_0) da função de avaliação G , descrita na Seção 4.2.1 (empates são decididos aleatoriamente). A função G não é utilizada, uma vez que não existem blocos passivos para aplicação do critério de heterogeneidade (segundo termo de G). Pela mesma razão, o critério de mínima alteração da estrutura do padrão corrente (terceiro termo de G) também não é aplicado. Cabe ressaltar que antes da investigação do impacto das mudanças de orientação, tiras de espaços vazios em torno das faces do bloco são adicionados ao volume do mesmo para possibilitar a inclusão de caixas adicionais.

O segundo tipo de movimento consiste em aplicar uma mudança de orientação de bloco seguido a um movimento de expansão (movimento híbrido). Neste caso, a avaliação do movimento utiliza a função G , somando-se as contribuições do 1º e 2º movimentos para cada um dos termos de G .

4.3.4 Reconfiguração de volumes remanescentes

No problema tridimensional, há uma vasta gama de possibilidades para a utilização de volumes remanescentes de blocos passivos com vistas à criação de novos blocos. Tornou-se necessária, portanto, a redução destas possibilidades. Especificamente, decidiu-se utilizar as configurações do caso bidimensional para definir configurações das *faces superiores* dos volumes. A Figura 4.12 ilustra as extensões para as configurações tridimensionais, as quais são somente aplicáveis a movimentos de expansão e híbridos.

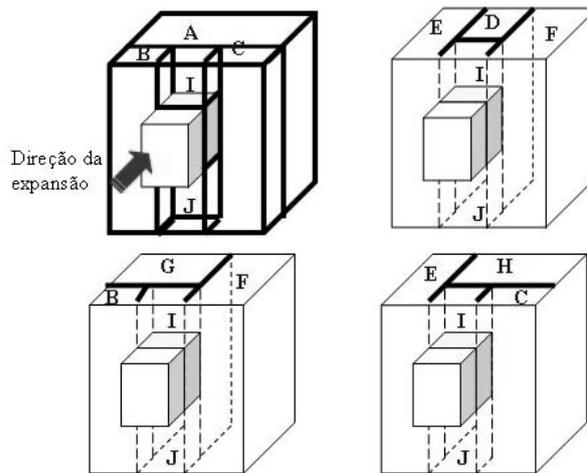


FIGURA 4.12 – As 4 configurações adotadas dos volumes remanescentes de blocos passivos.

Note que foi adotado o número de 4 configurações, sendo que em cada uma delas, dois blocos adicionais (I e J) foram criados em relação à versão bidimensional da heurística (Figura 4.6). Os blocos I e J correspondem respectivamente aos volumes acima e abaixo da expansão (volume superposto). Desta maneira para cada configuração no problema tridimensional, podem ser criados no máximo 5 blocos.

4.3.5 Recursos Adicionais

Divisão de Blocos

Este recurso foi criado com duas finalidades. A primeira consiste em reforçar o impacto do deslizamento de blocos (Passo 3.3 - Figura 4.4) na concentração de espaços não utilizados no centro geométrico do contêiner. Para tal, investiga-se para cada bloco do padrão se em sua extensão em qualquer uma das 3 dimensões está contido o ponto médio da dimensão associada do contêiner. Considere, por exemplo, a dimensão L (comprimento). Se o bloco i tem coordenadas de comprimento inicial e final $(x_1^i, x_2^i) = (0, \frac{3}{4}L)$, então em sua dimensão de comprimento está contida a coordenada central do comprimento do contêiner ($\frac{1}{2}L$). Neste caso, o bloco em questão é dividido o mais próximo possível da coordenada $\frac{1}{2}L$, criando-se assim um bloco adicional. Este procedimento é aplicado para as três dimensões, de forma que um mesmo bloco pode ser dividido em dois ou mais blocos. Os blocos resultantes são então deslizados para as extremidades do contêiner mais próximas a eles (Figura 4.13).

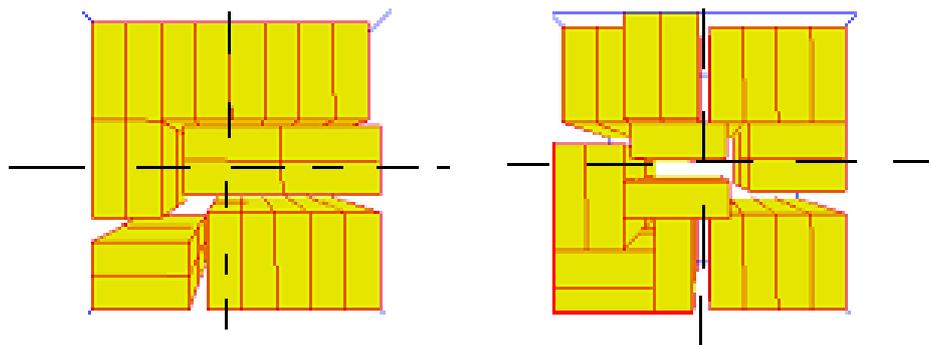


FIGURA 4.13 – Vista superior de um carregamento, antes e durante a aplicação dos procedimentos de divisão e deslizamento de blocos.

A segunda finalidade diz respeito às dificuldades da aplicação de movimentos quando o problema é tridimensional. O fato dos blocos possuírem alturas diferentes faz com que movimentos de expansão sobreponham blocos passivos situados em diferentes camadas do contêiner e destruam o padrão corrente em muito maior escala. Experimentos preliminares realizados antes da implementação deste procedimento, mostraram que os padrões obtidos apresentam baixa qualidade. A aplicação da divisão de blocos seguida de deslizamento minimiza este efeito uma vez que procura nivelar os blocos em camadas, controlando-se assim, o nível de alterações no padrão corrente.

Deve-se ressaltar, entretanto, que a aplicação da divisão de bloco seguida de deslizamento não pode ser feita a cada iteração. Dado que são proibidos movimentos de blocos ativos cujas coordenadas estejam armazenadas na lista tabu (Seção 4.2.2), alterações constantes nas coordenadas dos blocos tornam ineficaz o efeito restritivo da lista (ciclagem). Assim, este procedimento é aplicado com uma probabilidade dada por um parâmetro exógeno.

União de Blocos

Este recurso foi proposto visando especificamente instâncias cujo carregamento contempla um grande número de caixas. Nestes exemplos, é comum a formação de poucos blocos com muitas caixas. Ao serem divididos, estes blocos resultam em um conjunto de blocos menores com a mesma orientação e, conseqüentemente, é gerado um número razoável de movimentos que consistem de uma mera transferência de caixas entre blocos adjacentes de mesma orientação ($G_0 = 0$). Caso a solução corrente seja um ótimo local, o usual é que se verifique uma sucessão de movimentos do tipo acima; como o número de coordenadas para posicionamento dos blocos é grande em exemplos de maior porte, a lista

tabu teria de ser muito longa para forçar um movimento com $G_0 < 0$, ou o parâmetro que penaliza estes movimentos na função G (pen) teria de ser muito negativo.

A união de blocos visa minimizar este efeito danoso à exploração do espaço de soluções. Ela também potencializa o papel do movimento de mudança de orientação, uma vez que blocos maiores permitem um maior número de orientações factíveis e, portanto, maiores possibilidades de ganhos em número de caixas. Este procedimento é aplicado após a divisão de blocos e deslizamento, e consiste em agrupar cada par de blocos que exibam certas características em cada dimensão do problema. Como exemplo, considere os blocos i e j e a dimensão L . Os blocos são agrupados se tiverem a mesma orientação e satisfizerem as seguintes restrições:

- a coordenada final do comprimento de i é igual à coordenada inicial do comprimento de j ($x_2^i = x_1^j$).
- as coordenadas inicial e final da largura de i são respectivamente iguais às coordenadas inicial e final da largura de j ($y_1^i = y_1^j$; $y_2^i = y_2^j$).
- as coordenadas inicial e final da altura de i são respectivamente iguais às coordenadas inicial e final da altura de j ($z_1^i = z_1^j$; $z_2^i = z_2^j$).

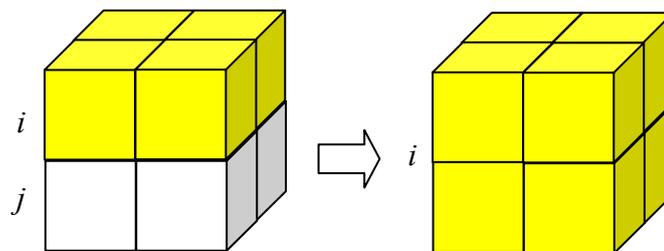


FIGURA 4.14 – União dos blocos i e j .

A aplicação do procedimento de união de blocos é repetida para cada dois blocos distintos e para cada dimensão até que nenhuma alteração no número de blocos seja verificada. Note, entretanto, que a união de blocos pode, em certos casos, gerar um número tão pequeno de blocos a ponto das restrições tabu proibirem qualquer movimento disponível. O critério de aspiração que elimina o elemento mais antigo da lista tabu nestes casos (passo 3.2 do algoritmo original) não é suficiente para garantir um fluxo suave das explorações caso a união de blocos seja realizada com muita frequência. Por esta razão, este último

procedimento também é aplicado com uma probabilidade dada por um parâmetro exógeno. Além disso, se houver a redução do número de blocos além de certo limite, a lista tabu é reinicializada.

O passo 3.3 do algoritmo original descrito na Figura 4.4 é substituído pela utilização da divisão de blocos e deslizamento. Além disso, são adicionados dois passos extras que prescrevem respectivamente, a aplicação do procedimento de união de blocos e a reinicialização da lista tabu.

A Figura 4.15 sumariza os passos da adaptação proposta, denominada *TSI*. Nos passos 3.3 e 3.4, *divip* e *unip* são parâmetros com valores entre 0 e 1 que regulam respectivamente a frequência de aplicação dos procedimentos de divisão e união de blocos. *Reinip* é um número entre 0 e 1 que ajuda a determinar a condição para reinicialização da lista tabu. Experimentos computacionais e resultados associados obtidos com a aplicação de *TSI* a um conjunto de instâncias da literatura são apresentados e discutidos no próximo capítulo.

Passo 1. Gere um padrão inicial S_0 . Faça $f(S_0)$ =número de caixas em S_0 , blk =número de blocos da solução corrente e LB (limitante inferior do valor da solução)=falso.

Passo 2. Faça $it=0$, $S=S_0$, $f(S)=f(S_0)$, $S^*=S$, $f(S^*)=f(S)$ e lista tabu $L=\emptyset$. Se $f(S_0)$ atender critérios de otimalidade, faça LB =verdadeiro e vá para o passo 4.

Passo 3. Repita até que $mv=0$ ou it =máximo número de iterações ou LB =verdadeiro

3.1 Para cada bloco $i=1, \dots, blk$, calcule os movimentos de expansão (6 direções), de troca de orientação de caixas, e híbridos. Faça mv =número total de movimentos de todos os blocos.

3.2 Se $mv>0$, selecione o movimento não tabu com a maior avaliação. Seja S' o padrão resultante da aplicação deste movimento. Se todos os movimentos estão proibidos, elimine o elemento mais antigo de L (t -ésimo elemento) e repita o passo. Caso contrário, efetive o movimento, gerando um novo padrão e faça $S=S'$. Se $f(S) > f(S^*)$, faça $S^*=S$ e $f(S^*)=f(S)$. Se S^* atende critérios de otimalidade, faça LB =verdadeiro e vá para o passo

4. Caso contrário, faça $it=it + 1$.

(Divisão de blocos e Deslizamento)

Passo 3.3 Repita até que nenhum deslizamento de blocos seja verificado

3.3.1 Gere um número aleatório r entre 0 e 1. Se $r < divip$, aplique o procedimento de divisão de blocos.

3.3.2 Aplique o procedimento de deslizamento de blocos para a extremidade do contêiner mais próxima a eles até que nenhum deslizamento seja verificado.

(União de blocos)

Passo 3.4 Gere um número aleatório r entre 0 e 1. Se $r < unip$, aplique o procedimento de união de blocos até que nenhuma união seja verificada.

Passo 3.5 Se o número de blocos $< reinip \times$ (Número de caixas do padrão corrente), reinicialize a lista tabu.

Passo 4. Se LB =verdadeiro, pare pois S^* corresponde a um padrão ótimo. Caso contrário, se $mv=0$ e $0 < it <$ número máximo de iterações, o padrão final convergiu para um único bloco; se $mv=0$ e $it=0$, o padrão inicial S_0 é homogêneo.

FIGURA 4.15 – Passos da adaptação proposta (algoritmo *TSI*) para o PCCCI.

5 EXPERIMENTOS COMPUTACIONAIS

O algoritmo tabu proposto, aqui denominado TS1, foi programado a partir da implementação do algoritmo PM em Borland Delphi 7, na linguagem Pascal. O uso deste compilador permitiu a elaboração de uma interface gráfica, em que os padrões tridimensionais gerados podiam ser observados sob 7 diferentes vistas: superior, inferior, frontal, traseira, esquerda, direita, e isométrica (visão em perspectiva de 3 vistas) e dentre outras opções, a de realce de blocos selecionados. A Figura 5.1 ilustra a interface com vista isométrica de um carregamento e realce do bloco 56.

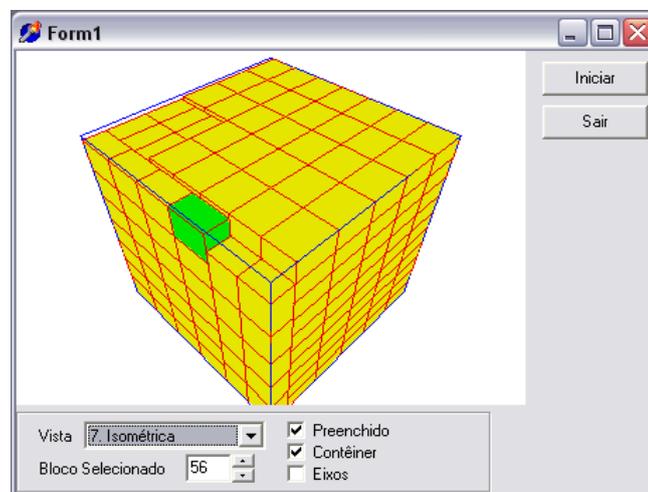


FIGURA 5.1 – Interface gráfica da implementação.

Os valores dos parâmetros α e pen utilizados no algoritmo PM foram mantidos em TS1 ($\alpha=0,5$ e $pen=-1$). Após experimentos preliminares com um conjunto selecionado de instâncias teste, os parâmetros $divip$ e $unip$ foram fixados em 0,3, o que significa que tanto a divisão como a união de blocos são aplicadas em 30% das iterações. O parâmetro $reinip$, por sua vez, assumiu o valor 0,05, ou seja, a lista tabu é reinicializada sempre que o número de blocos se reduz a 5% do número de caixas do padrão corrente.

Como critérios de paradas, foram adotados: (i) obtenção da melhor solução reportada, e (ii) 50.000 iterações sem melhoria. Este valor se fez necessário para problemas maiores, uma vez que a configuração de volumes remanescentes, responsável pela geração de novos blocos, pode ser considerada muito simples para um problema tridimensional. Como

resultado, a convergência para soluções de qualidade requer um número maior de movimentos.

No tocante ao tamanho da lista tabu (*tlista*), os experimentos contemplaram faixas de 11 valores (veja o apêndice para maiores detalhes). Assim como no caso bidimensional, a determinação destas faixas parte do pressuposto de que valores adequados de *tlista* crescem com o número de caixas do padrão inicial (s_0).

O algoritmo TS1 foi aplicado a um conjunto de 14 instâncias do PCCCI propostas em LINS *et al.* (2002). Neste conjunto de exemplos, um mesmo contêiner com dimensões $(L, W, H) = (50, 50, 50)$ é utilizado, enquanto as dimensões das caixas são variáveis. Com base nos resultados obtidos, o algoritmo foi analisado em termos de qualidade da solução, esforço computacional, robustez e estabilidade do carregamento.

Todos os experimentos foram conduzidos em um computador Pentium D, 3.0 GHz e 800 MHz. As próximas seções descrevem as análises realizadas.

5.1 Qualidade da Solução e Tempo Computacional

A qualidade das soluções foi inferida comparando-se o número de caixas das soluções incumbentes de *TSI* com o obtido pelos 4 algoritmos recursivos *2-D*, *lk*, *gui*, e *3-D* propostos em LINS *et al.* (2002), descritos na Seção 3.3.2. Cabe lembrar que o algoritmo *3-D* tem a garantia de prover padrões ótimos do tipo não-guilhotinado de 1ª ordem.

A Tabela 5.1 apresenta os resultados para os 5 algoritmos. As colunas 2-4 descrevem as dimensões das caixas em cada exemplo, e as colunas 5-8 apresentam para cada algoritmo da literatura, o número de caixas das soluções obtidas e os tempos computacionais (em segundos) requeridos para sua obtenção, quando reportados. A coluna 9 reporta o número de caixas das soluções incumbentes geradas pelo algoritmo *TSI* e os tempos computacionais para sua obtenção (em segundos). Também é apresentado o número de caixas das soluções iniciais de cada instância (entre parêntesis). A última coluna (TSI_p) é explicada na seção 5.2. A penúltima linha (SOMA) mostra o número acumulado de caixas nos carregamentos, enquanto a última linha apresenta a média dos desvios percentuais (MDP) da qualidade das soluções obtidas em relação às soluções com maior número de caixas. As melhores soluções de cada exemplo são apresentadas em negrito. Como os melhores resultados de *TSI* foram em geral obtidos com mais de um tamanho de lista tabu (veja o apêndice para maiores

detalhes), em caso de empate em número de caixas, foi escolhida a solução com menor tempo computacional.

Tabela 5.1 – Resultados computacionais: Número de caixas no carregamento.

Instância	Dimensões das caixas			Algoritmo								
	l	w	H	2D*	lk*	gui*	3-D*		TS1⁺		TS1_p⁺	
1	13	14	23	18	24	22	26	2	26 (24)	4,1	26	4,1
2	17	20	12	16	22	22	26	1	26 (20)	233	26	411
3	11	22	15	24	27	27	29	1	29 (26)	69	29	69
4	17	21	6	32	47	46	54	148	54 (42)	141	53	218
5	22	8	11	48	54	54	58	17	58 (52)	96	58	150
6	18	21	5	40	54	54	57	523	57 (48)	246	57	347
7	9	11	19	48	60	57	62	81	62 (50)	49	61	6
8	13	8	18	48	64	60	64	19	64 (48)	0,6	64	0,6
9	9	16	11	64	74	72	75	40	75 (72)	132	75	254
10	13	15	7	63	79	79	84	347	83 (72)	62	83	62
11	7	9	18	78	102	100	106	163	106 (90)	35	106	59
12	7	6	18	118	158	156	161	1977	161 (144)	3261	161	3561
13	7	9	11	156	176	170	176	1419	176 (168)	146	174	0,5
14	9	7	5	390	394	394	396	377	396 (385)	396	396	511
SOMA				1143	1335	1313	1374		1373		1369	
MDP				16,8%	2,8%	4,4%	0,0%		0,1%		0,2%	

Ambiente computacional:

♦ Pentium II, 400 MHz

† Windows - Pentium D, 3.0 GHz e 800 MHz

A Tabela 5.1 mostra que dentre os 5 algoritmos, *3-D* é o que apresenta os melhores resultados em número de caixas para todas as instâncias do conjunto. Não obstante, o desempenho do algoritmo *TSI* só é inferior no caso da instância 10 (uma caixa a menos). Isso corresponde a menos de 1% de desvio percentual médio. O fato de *TSI* não ter superado *3-D* em nenhum dos exemplos é um indicativo de que sua vizinhança não é suficientemente poderosa. Em Junqueira *et al.* (2009) as instâncias 1 a 3 foram resolvidas otimamente com o método *branch&bound*, tendo sido obtido o número ótimo de 27 caixas para as instâncias 1 e 2. O número de caixas ótimo da instância 3 é, por sua vez, igual ao obtido por *3-D* e *TSI*, ou

seja, 29 caixas. Note que em relação aos demais algoritmos, a superioridade de *TS1* é evidente, o que sugere que o desenvolvimento desta pesquisa pode ser considerado promissor.

A coluna 6 da Tabela 5.2 apresenta a frequência com que cada melhor solução de *TS1* foi obtida para as faixas de lista adotadas. Nota-se que a melhor solução de algumas instâncias é frequentemente obtida apesar da variação do parâmetro, enquanto em outros casos, ela é obtida uma única vez.

Em particular, as instâncias 1 e 11 obtiveram o número máximo de caixas em todas as execuções, sendo que a instância 1 é uma das instâncias que possuem os maiores tamanhos de caixas, enquanto a instância 11 tem caixas relativamente pequenas. Para as instâncias 2 e 4, o maior número de caixas foi obtido com apenas um tamanho de lista. No entanto, para todas as instâncias, a diferença entre a melhor e a pior solução não foi maior do que 1 caixa, o que explica o valor relativamente pequeno das variâncias.

Tabela 5.2 – Frequência, média e variância de obtenção da solução provida por *TS1* com maior número de caixas.

Instância	Dimensões das caixas			Algoritmo			
	L	W	H	TS1			
				Nº de caixas	Frequência	Média	Variância
1	13	14	23	26	11/11	26	0,00
2	17	20	12	26	1/11	25	0,09
3	11	22	15	29	6/11	29	0,27
4	17	21	6	54	1/11	53	0,09
5	22	8	11	58	6/11	57	0,40
6	18	21	5	57	2/11	56	0,17
7	9	11	19	62	2/11	61	0,26
8	13	8	18	64	11/11	64	0,21
9	9	16	11	75	4/11	74	0,25
10	13	15	7	83	6/11	82	0,09
11	7	9	18	106	11/11	106	0,00
12	7	6	18	161	2/11	160	0,41
13	7	9	11	176	2/11	175	0,61
14	9	7	5	396	2/11	395	0,17

Ambiente computacional:

Windows - Pentium D, 3.0 GHz e 800 MHz

A Figura 5.2 ilustra vistas isométricas das soluções obtidas para alguns dos exemplos tratados.

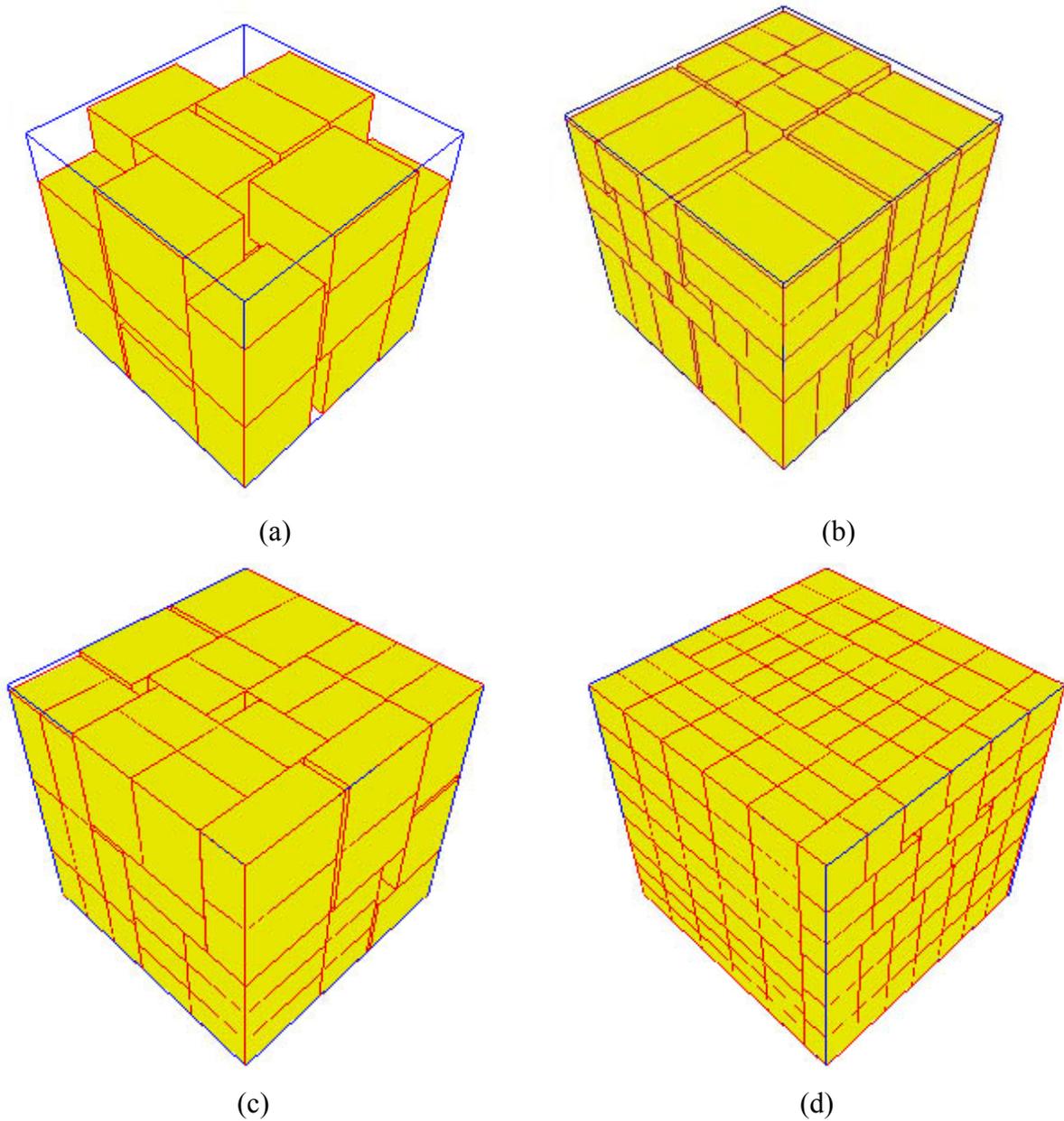


FIGURA 5.2. – Vistas isométricas de carregamentos obtidos pelo algoritmo *TSI*: (a) Instância 2, (b) Instância 5, (c) Instância 8 e (d) Instância 14.

Devido aos diferentes ambientes computacionais, os tempos de CPU de *3-D* e *TSI* não são diretamente comparáveis (apesar de indicarem que *TSI* requer maior tempo de execução que *3-D*). O intuito é na realidade o de expor o esforço computacional das melhores

execuções de *TSI* ao resolver este conjunto de exemplos tão diversificado. Em média, requereu-se 406 segundos de CPU. Nota-se também que os tempos requeridos não são proporcionais ao tamanho da instância. De fato, à parte da instância 10, para a qual não se obteve a melhor solução reportada, os exemplos 2 e 4 são os que apresentaram maior dificuldade de resolução.

É importante ressaltar que os procedimentos de união de blocos e reinicialização da lista se mostraram apenas importantes para a instância 14, de maior porte. Para as demais instâncias, este procedimento não trouxe benefícios e, de forma geral, atrasou a obtenção da incumbente. Nosso propósito em não excluir estes passos nestes casos é o de manter ao máximo a simplicidade da extensão proposta, reduzindo o método a uma única variação.

5.2 Robustez e o tamanho da lista tabu

Assim como no caso bidimensional, buscou-se obter uma função que fornecesse o tamanho da lista tabu *tlista* em função do número de caixas do padrão inicial (s_0). Como as melhores soluções de cada instância foram obtidas com diferentes tamanhos de lista, a função foi obtida aplicando-se o Método de Diferenças Divididas de Newton (Burden *et al.*, 2003). Este método foi utilizado por fornecer polinômios interpoladores de uma amostra de pontos em que parte destes pontos é previamente fixada. Dentre os pontos (s_0 , *tlista*) que forneceram as melhores soluções para o conjunto de instâncias tratado, foram testadas fixações de pontos associados a instâncias para as quais o número de soluções ótimas era pequeno. O polinômio que forneceu o maior número de valores de *tlista* que resultaram em soluções ótimas é dado por:

$$tlista(s_0) = 7 + 0,5*(s_0 - 20) - 0,001602564(s_0 - 20)*(s_0 - 48) + 0,000002536(s_0 - 20)*(s_0 - 48)*(s_0 - 72)$$

A Figura 5.3 apresenta a curva s_0 vs. *tlista* definida por este polinômio (de grau 3), e a última coluna da Tabela 5 (TSI_p) mostra os resultados obtidos com o valor calculado de *tlista*.

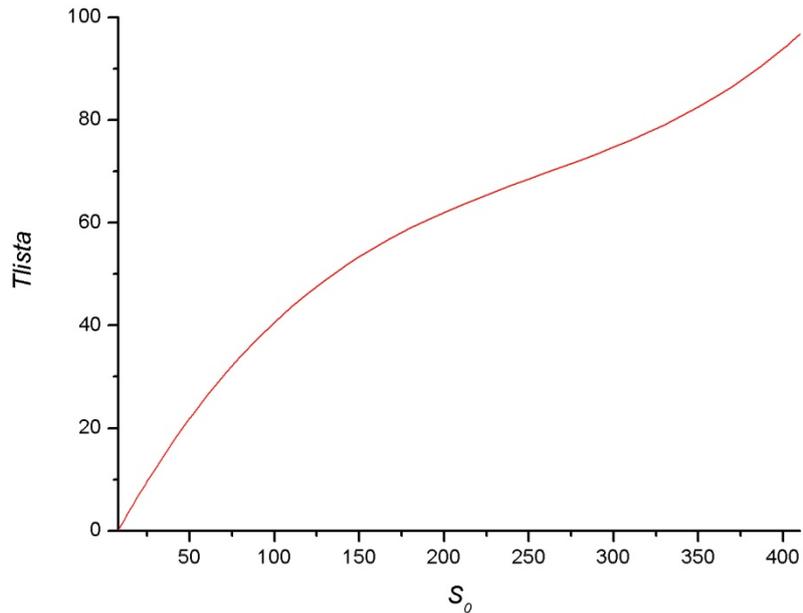


FIGURA 5.3 – Número de caixas do padrão inicial (s_0) vs. tamanho da lista (t_{lista}).

Observa-se que em 11 dos 14 exemplos, TSI_p apresenta soluções com o mesmo número de caixas que o obtido com TSI , o que sugere que a aplicação do polinômio é relativamente eficaz para a obtenção de tamanhos adequados de lista. Além disso, nos 3 casos em que os números de caixas não foram equivalentes, as soluções resultantes apresentam uma degradação razoavelmente pequena da qualidade (de 1 a 2 caixas). A Tabela 5.3 apresenta os resultados obtidos como a porcentagem de utilização do volume do contêiner.

Tabela 5.3 – Resultados computacionais: Porcentagem de utilização do volume do contêiner.

Instância	Dimensões das caixas			Porcentagem de utilização do volume do contêiner		
	l	W	H	3-D*	TS1 ⁺	TS1 _p ⁺
1	13	14	23	87,07%	87,07%	87,07%
2	17	20	12	84,86%	84,86%	84,86%
3	11	22	15	84,22%	84,22%	84,22%
4	17	21	6	92,53%	92,53%	90,82%
5	22	8	11	89,83%	89,83%	89,83%
6	18	21	5	86,18%	86,18%	86,18%
7	9	11	19	93,30%	93,30%	91,79%
8	13	8	18	95,85%	95,85%	95,85%
9	9	16	11	95,04%	95,04%	95,04%
10	13	15	7	91,73%	90,64%	90,64%
11	7	9	18	96,16%	96,16%	96,16%
12	7	6	18	97,37%	97,37%	97,37%
13	7	9	11	97,57%	97,57%	96,47%
14	9	7	5	99,79%	99,79%	99,79%
MÉDIA				92,25%	92,17%	90,83%

Ambiente computacional:

* Pentium II, 400 MHz

+ Windows - Pentium D, 3.0 GHz e 800 MHz

5.4 Sensibilidade à solução inicial

Os experimentos com o algoritmo TS1 descritos na Seção 5.1 foram repetidos, desta vez, a partir de um padrão inicial homogêneo. O objetivo foi o de analisar o papel da solução inicial no desempenho do algoritmo, em particular, quando s_0 tem qualidade geralmente inferior à da solução gerada pela extensão da heurística de Smith e De Cani. Para tal, adotou-se o melhor padrão homogêneo obtido dentre as três possibilidades de posicionamento de faces das caixas (lw , lh ou wh) sobre o chão do contêiner.

Os resultados sugerem que a qualidade da solução incumbente é razoavelmente preservada ao se utilizar o padrão inicial homogêneo. Tomando o mesmo conjunto de tamanhos de lista, os valores das incumbentes apresentados na Tabela 5.1 foram obtidos em 13 dos 14 exemplos (a única exceção foi verificada para o exemplo 4, com uma caixa a

menos). Observa-se também que dependendo da instância, a frequência com a qual estes valores foram obtidos pode tanto diminuir, como aumentar ou manter-se constante. Em particular, é interessante notar que em uma das instâncias para a qual o algoritmo original apresentou maiores dificuldades de resolução (instância 2), a frequência obtida a partir do padrão homogêneo foi triplicada. Para a instância 5, por outro lado, a frequência caiu pela metade.

Em relação aos tempos computacionais, padrões homogêneos iniciais quase sempre provocam um aumento considerável do tempo médio de obtenção da solução incumbente. Em relação à instância 1, por exemplo, o tempo médio observado foi de 736 segundos a partir do padrão homogêneo, enquanto com a solução inicial provida pela extensão da heurística de Smith e De Cani requereu-se cerca de 30 segundos.

5.5 Estabilidade

Nesta seção, a estabilidade da carga nas soluções obtidas com o tamanho de lista fornecido pelo polinômio (veja coluna 10 da Tabela 5.1) foi mensurada segundo Bischoff e Ratcliff (1995a). Cabe ressaltar que a estabilidade do carregamento não é o foco deste trabalho, e que o algoritmo não inclui procedimentos que visam melhorar a estabilidade do carregamento durante as iterações.

Especificamente, foram calculados o número médio de caixas que suportam cada caixa posicionada acima do piso do contêiner (medida M1), e a porcentagem média de caixas que não possuem pelo menos três lados em contato com outras caixas (medida M2). Estes valores são apresentados nas colunas 4 e 5 da Tabela 5.4. Também é mostrada na coluna 3, a porcentagem da menor área da base de uma caixa do carregamento em contato com outras caixas (medida M0). Seguindo a interpretação de Araújo (2006) acerca dos critérios de estabilidade de Bischoff e Ratcliff (1995b), consideramos como verticalmente estáveis aquelas soluções cuja porcentagem da base de cada caixa apoiada por outras caixas (M0) é igual ou superior a 55%. Note que o critério M1 é também uma medida de estabilidade vertical, sem, no entanto, definir um valor mínimo que caracterize um carregamento estável. Considerando o critério M0, observa-se que apenas 2 das 14 melhores soluções são verticalmente estáveis. Estes resultados revelam que a estabilidade não é uma propriedade obtida naturalmente pelo algoritmo.

A fim de tentar melhorar de forma pontual a estabilidade vertical da solução incumbente, foi desenvolvido e aplicado um procedimento adicional (denominado PE). PE

consiste na aplicação de deslizamentos de caixas consideradas críticas à medida M_0 , ou seja, caixas cuja porcentagem de área de contato de suas bases é inferior a 55%. PE é aplicado à solução incumbente obtida com o algoritmo TSI_p segundo os seguintes passos:

Passo 1. Calcule a porcentagem de área de contato da base de cada caixa da. Seja B_c , o conjunto de caixas do carregamento com porcentagem de área inferior a 55%.

Passo 2. Para cada caixa $b \in B_c$:

2.1. Deslize temporariamente a caixa b nas 4 direções horizontais (quando possível) na seguinte ordem: direita, esquerda, frente e atrás. O deslizamento em cada direção é realizado até que b atinja a face de outra caixa do carregamento.

2.2 Se a porcentagem de área de contato da caixa b com o deslizamento for maior que o valor calculado no passo 1, aceite e aplique permanentemente o deslizamento da caixa b .

FIGURA 5.4 – Passos do procedimento PE.

A coluna 6 apresenta os resultados obtidos de M_0 após a aplicação de PE. Os valores de M_1 e M_2 de todas as instâncias não foram alterados, no entanto, as soluções das instâncias 3 e 9 se tornaram verticalmente estáveis (totalizando 4 instâncias) segundo o critério utilizado. Para as demais instâncias, a aplicação de PE melhorou a estabilidade das soluções incumbentes, exceto nos exemplos 11, 12 e 13. Nestes casos, as caixas com menor área de base em contato possuíam os quatro lados em contato com outras caixas e não puderam ser deslizadas.

Tabela 5.4 – Resultados computacionais: Estabilidade do carregamento.

Instância	Algoritmo TS1 _p								
	Melhor solução					Melhor solução verticalmente estável			
	Número de Caixas	M0	M1	M2	M0 após PE	Número de Caixas	M0	M1	M2
1	26	11	1,1	0	17	25	56,5	1,1	4
2	26	10	1,3	42,3	12	25	58,3	1,2	44
3	29	49,6	1,5	34,5	55	29	55	1,5	34,5
4	53	47,1	1,4	1,9	53	53	57,1	1,4	0
5	58	36,4	1,4	29,3	38	56	62,5	1,1	26,8
6	57	20	2	15,8	20	56	55,6	1,5	17,9
7	61	11,1	1,4	3,3	17	61	57,9	1,6	8,2
8	64	100	1,5	0	100	64	100	1,5	0
9	75	54,5	1,4	5,3	64	75	64	1,4	5,3
10	83	5,7	1,7	2,4	23	76	100	1	11,8
11	106	44,4	1,6	0,9	44,4	90	100	1,6	0,9
12	161	11,1	1,4	2,5	11,1	160	66,7	1,6	1,3
13	174	14,3	1,3	2,3	14,3	168	55,6	1,1	11,9
14	396	55,6	1,7	0	55,6	396	1,7	0	0

Nas colunas 7-10, são mostrados os resultados, desta vez, associados à melhor solução verticalmente estável gerada ao longo das iterações. Note que estas soluções representam uma degradação de 35 caixas, ou 2,55% do total de caixas das melhores soluções obtidas com a aplicação de PE. Esta degradação seria maior caso tivesse sido imposto um valor mínimo para M0 superior a 55%. Em apenas seis dos exemplos (instâncias 3, 4, 7, 8, 9 e 14) foram obtidas soluções estáveis que mantiveram o mesmo número de caixas. Em contrapartida, note que as instâncias 8, 10 e 11 apresentam todas as caixas do carregamento com 100% da área de suas bases suportadas.

Em relação à estabilidade vertical mensurada pelo critério M1, não parece haver uma associação entre esta medida e M0 (Figura 5.4). Em algumas instâncias, o valor de M1 apresentado pela melhor solução é maior do que o obtido pela melhor solução verticalmente estável ($M0 \geq 0,55$). Em outros casos, o oposto foi verificado.

Em relação à estabilidade vertical mensurada pelo critério M1, não parece haver uma associação entre esta medida e M0 (Figura 5.4). Em algumas instâncias, o valor de M1 apresentado pela melhor solução é maior do que o obtido pela melhor solução verticalmente estável ($M0 \geq 0,55$). Em outros casos, o oposto foi verificado.

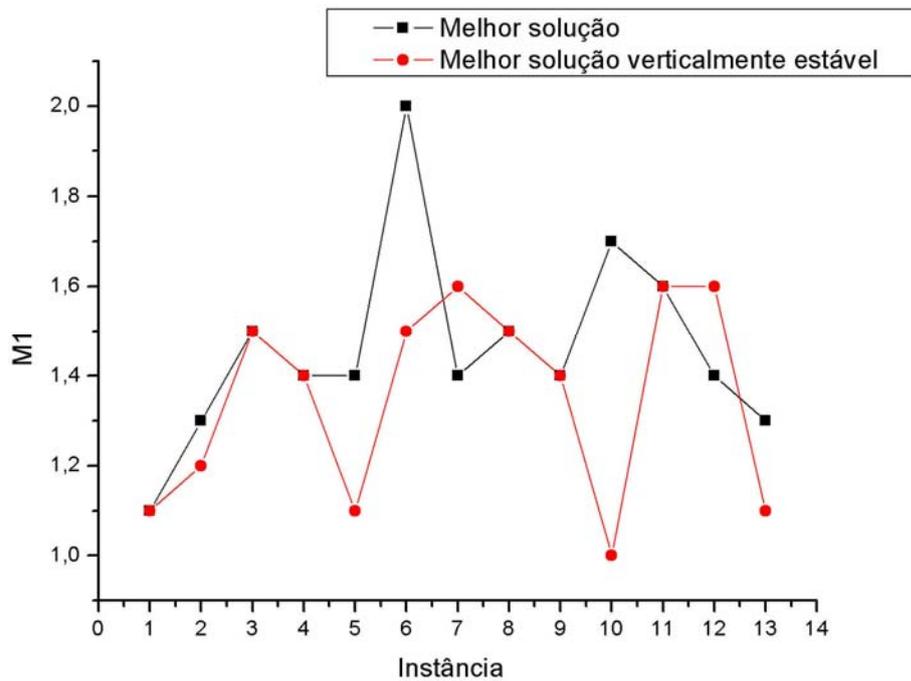


FIGURA 5.5 – Estabilidade critério M1.

Em relação à estabilidade horizontal, mensurada pelo critério M2 (Figura 5.5), cerca de 70% das melhores soluções apresentam menos de 10% das caixas do carregamento com menos de 3 lados suportados. No que tange às melhores soluções verticalmente estáveis, cerca de 60% destas apresentam menos de 10% das caixas do carregamento com menos de 3 faces suportadas. Este resultado sugere que melhorias na estabilidade horizontal acompanham soluções consideradas verticalmente estáveis segundo o critério M0.

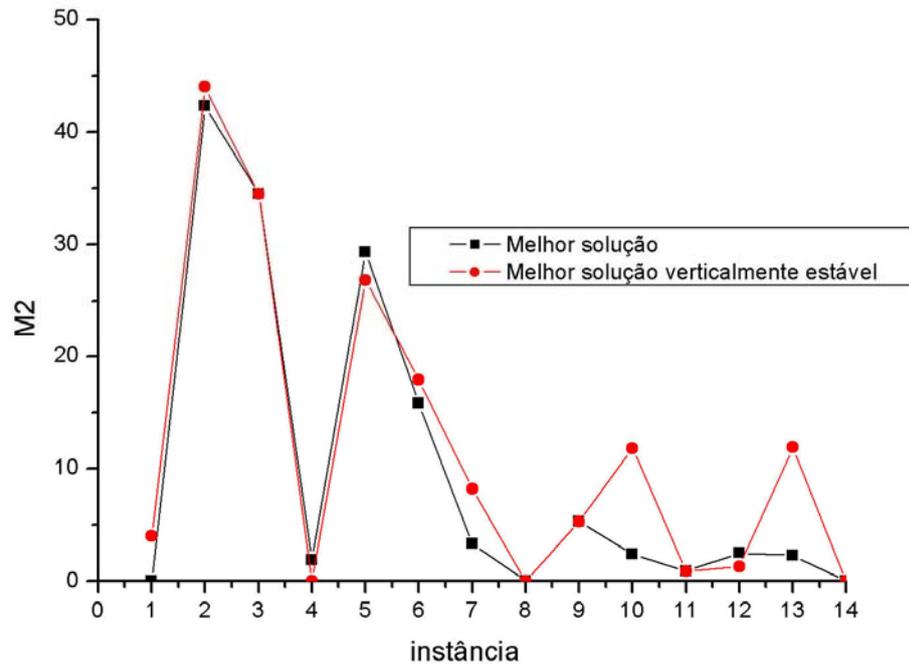


FIGURA 5.6 – Estabilidade critério M2.

6 CONCLUSÕES E PERSPECTIVAS DE PESQUISA FUTURA

Este trabalho apresentou uma extensão do algoritmo de busca tabu de Pureza e Morabito (2006), originalmente proposto para o problema do carregamento do paletes, com vistas ao tratamento do problema de carregamento de contêineres com caixas homogêneas. Além da adequação à diferença de dimensionalidade entre os dois problemas, foram incluídos dois novos tipos de movimentos (troca de orientação de caixas e híbrido) na geração da vizinhança de busca, e incorporados procedimentos de união, divisão e deslizamento de blocos. Os resultados obtidos com um conjunto de instâncias da literatura demonstram que em termos de utilização do volume, a abordagem proposta é bastante promissora.

A mensuração de critérios de estabilidade de carga, por outro lado, indicou que o método não gera padrões estáveis na maioria dos casos. Desta forma, como perspectivas de pesquisa futura, sugere-se que tais critérios sejam incorporados ao processo de seleção da solução da vizinhança. Uma idéia inicial é a de adicionar um termo que quantifica a estabilidade da carga à função de avaliação dos movimentos. O peso deste termo variaria conforme a fase da busca, de maneira a priorizar a estabilidade ou a maior utilização do volume do contêiner. É de se esperar que os carregamentos resultantes exibam um menor número de caixas em relação ao aqui obtido. Além da estabilidade, outras restrições práticas, como orientação das caixas, e peso e empilhamento máximo, também poderiam ser incorporadas ao método.

Outra linha de pesquisa que naturalmente advém deste trabalho é a extensão da abordagem para o tratamento de carga heterogênea para os casos de demanda irrestrita e de demanda restrita, sendo esta última a mais abordada em aplicações. Nestes casos, uma decisão adicional do método é a seleção do tipo de caixa na criação dos blocos. Acreditamos que mesmo neste caso, a organização atual do algoritmo seria mantida, e que seriam necessários poucos procedimentos adicionais e pequenas modificações nas estruturas de dados.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ABDOU, G., YANG, M. A Systematic Approach for the Three Dimensional Palletization Problem. **International Journal of Production Research**, v.32, n.10, p.2381-2394, 1994.
- ALVAREZ-VALDES R., PARREÑO F., TAMARIT J. M. A tabu search algorithm for the pallet loading problem, **OR Spectrum**, vol 27, n.1, p.43-61, 2005.
- ANAND, S., MCCORD, C., SHARMA, R., An Integrated Machine Vision Based System for Solving the Nonconvex Cutting Stock Problem Using Genetic Algorithms. **Journal of Manufacturing Systems**, v.18, p.396-415, 1999.
- ARAÚJO, O.C.B. **Problemas de Corte e Empacotamento Tridimensional e Integração com Roteamento de Veículos**. Campinas, 2006. Tese de Doutorado, Faculdade de Engenharia e de Computação, Unicamp.
- ARAÚJO, O.C.B., ARMENTANO, V.A. A Multi Start Random Constructive Heuristic for the Container Loading Problem, **Pesquisa Operacional**, v.27, n.2, p.311-331, 2007.
- ARENALES, M., MARQUES, F.P. O Problema da Mochila Compartilhada e Aplicações. **Pesquisa Operacional**, v.22, n.3, p.285-304, 2002.
- ARENALES, M., MORABITO, R. An and/or-graph Approach to the Solution of Two-Dimensional Non-Guillotine Cutting Problems. **European Journal of Operational Research**, v.84, p.599-617, 1995.
- ARENALES, M., MORABITO, R., YANASSE, H. Problemas de Corte e Empacotamento. Livro-texto de Mini curso, **XXVI Simpósio Brasileiro de Pesquisa Operacional**, São João Del Rei, MG – Brasil, 2004.
- ARMBRUSTER, M. A Solution Procedure for a Pattern Sequencing Problem as Part of a One-Dimensional Cutting Stock Problem in the Steel Industry. **European Journal of Operational Research**, v.141, p.328-340, 2002.
- ARNTZEN, B. C.; BROWN, G. G.; HARRISON, T. P.; TRAFTON, L. L. Global Supply Chain Management at Digital Equipment Corporation. **Interfaces**, v.25, n.1, p.69-93, 1995.
- BALLOU, R.H. **Logística Empresarial**, Atlas, São Paulo, 1993.
- BEASLEY, J.E. An Exact Two-Dimensional Non Guillotine Cutting Tree Search Procedure. **Operations Research**, v.33, p.49-64, 1985.
- BELLUZZO, L., MORABITO, R. Otimização nos Padrões de Corte de Chapas de Fibra de Madeira Reconstituída: Um Estudo de Caso. **Pesquisa Operacional**, v.25, n.3, p.391-415, 2005.
- BELOV, G., SCHEITHAUER, G. Setup and open stacks minimization in one dimensional stock cutting. **INFORMS Journal on Computing**, v.19, p.27-35, 2007.

BENNELL, DOWSLAND, A tabu thresholding implementation for the irregular stock cutting problem. **International Journal of Production Research**, v.37, n.18, p.4259-4275, 1999.

BIRGIN, E.G., MARTÍNEZ, J.M., RONCONI, D.P., Optimizing the packing of cylinders into a rectangular container: A nonlinear approach. **European Journal of Operational Research**, v.160, p.19-33, 2005.

BIRGIN, E. G., LOBATO, R. D. e MORABITO, R., An effective recursive partitioning approach for the packing of identical rectangles in a rectangle, **Journal of the Operational Research Society**, doi:10.1057/jors.2008.141, 2009.

BISCHOFF, E., DOWSLAND, W. An application of the micro to product design and distribution. **Journal of the Operational Research Society**, v.33, p.271-280, 1982.

BISCHOFF, E.E., MARRIOTT, M.D. A Comparative Evaluation of Heuristics for Container Loading. **European Journal of Operational Research**, v.44, p.267-276, 1990.

BISCHOFF, E., RATCLIFF, M.S.W. Loading Multiple Pallets. **Journal of the Operational Research Society**, v.46, p.1322-1336, 1995a.

BISCHOFF, E., RATCLIFF, M.S.W. Issues in the Development of Approaches to Container Loading. **Omega**, v.23, n.4, p.377-390, 1995b.

BISCHOFF, E.E., Three-Dimensional Packing Of Items With Limited Load Bearing Strength. **European Journal of Operational Research**, v.168, p.952-966, 2006.

BORTFELDT, A. A genetic algorithm for the container loading problem. In: Proceedings of the Conference on Adaptive Computing and Information Processing, London, v. 2, p.749-757, 1994.

BURDEN, R.L. FAIRES, J.D, **Análise Numérica**, Thomson Learning (Pioneira), São Paulo, 2003.

CAMM, J. D., CHORMAN, T. H., DILL, F. A., EVANS, J. R., SWEENEY, D. J. WEGRYN, G. W. Blending OR/MS, Judgment, and GIS: restructuring P&G's Supply Chain. **Interfaces**, v.27, n.1, p.128-142, 1997.

CARPENTER, H., DOWSLAND, W.B. Practical considerations of the pallet loading problem. **Journal of the Operational Research Society**, v.36, p.489-497, 1985.

CECILIO, F. O. **Heurísticas Para o problema de carregamento de carga dentro de contêineres**. São Carlos, 2003. Dissertação de Mestrado, Departamento de Engenharia de Produção, UFSCar.

CECILIO, F. O., MORABITO, R. Refinamentos na heurística de George e Robinson para o problema de carregamento de caixas dentro de contêineres. **Revista Transportes**, v. 12, p. 32-45, 2004.

CHEN, C.S., LEE, S.M., SHEN, Q.S. An Analytical Model for the Container Loading Problem. **European Journal of Operational Research**, v.80, p.68-76, 1995.

CHOONG, S.T., COLE, M.H., KUTANOGLU, E. Empty container management for intermodal transportation networks. **Transportation Research**, Part E v.38, p.423–438, 2002.

CORMEN, T., LEISERSON, C., RIVEST, R., STEIN, C. **Algoritmos (Teoria e Prática)** - Tradução da 2ª edição Americana. 2002.

DAVIES, A.P., BISCHOFF, E.E. Weight Distribution Considerations in Container Loading. **European Journal of Operational Research**, v.114, p.509-527, 1999.

DIAS, M.A.P. **Administração de Materiais: Uma Abordagem Logística**. 4ª edição. São Paulo: Atlas, 1995.

DOWSLAND, W.B., Improving palletisation efficiency – The Theoretical Basis and Practical Application, **International Journal of Production Research**, vol 33, n.8, p.213-222, 1995.

DYCKHOFF, H. A typology of cutting and packing problems, **European Journal of Operational Research**, v.44, p.145-159, 1990.

DYCKHOFF, H., FINKE, U. **Cutting and Packing in Production and Distribution: Typology and Bibliography**. Heidelberg: Springer-Verlag, 1992.

DYSON, R.G., GREGORY, A.S. The cutting stock problem in the flat glass industry. **Operational Research Quarterly**, v.25, p.41-53, 1974.

ELEY, M. Solving container loading problems by block arrangement. **European Journal of Operational Research**, v.141, n.2, p.393-409, 2002.

FAROE, O., PISINGER, D., ZACHARIASEN, M. Guided local search for the three-dimensional bin packing problem, **INFORMS Journal on Computing**, v.11, p.345–357, 1999.

GAREY, M., JOHNSON, D. **Computers and intractability: A guide to the theory of NP-completeness**. W. H. Freeman and Company, San Francisco, 1979

GEHRING, H., BORTFELDT, A. A Genetic Algorithm for Solving the Container Loading Problem. **International Transactions in Operations Research**, v.4, n.5/6, p.401-418, 1997.

GEHRING, H., BORTFELDT, A. A Parallel Genetic Algorithm for Solving the Container Loading Problem. **International Transactions in Operations Research**, v.9, p.497-511, 2002.

GEHRING, H., MENSCHNER, K., MEYER, M. A Computer-Based Heuristic for Packing Pooled Shipment Containers. **European Journal of Operational Research**, v.44, p.277-288, 1990.

GENERAL SERVICES ADMINISTRATION, **Warehouse Operations**, U.S. Government Printing Office, Washington DC, 1966.

GEORGE, J.A. A Method for Solving Container Packing for a Single Size of Box. **Journal of Operational Research Society**, v.43, n.4, p.307-312, 1992.

GEORGE, J.A., ROBINSON, D.F. A Heuristic for Packing Boxes into a Container. **Computers and Operations Research**, v.7, p.147-156, 1980.

GRADISAR, M., RESINOVIC, G., KLJAJIC, M., Evaluation of algorithms for one-dimensional cutting. **Computers & Operations Research**, v.29, p.1207-1220, 2002.

HADJICONSTANTINO, E., CHRISTOFIDES, N. An exact algorithm for general, orthogonal, two dimensional knapsack problem. **European Journal of Operational Research**, v.83, p.39-56, 1995.

HAESSLER, R.W., A procedure for solving the 1.5: dimensional coil slitting problem. **AIIE Transactions**, New York, v.10, p.70-75, 1978.

HAN, C.P., KNOTT, K., EGBELU, P.J. A Heuristic Approach to the Three-Dimensional Cargo Loading Problem. **International Journal of Production Research**, v.27, n.5, p.757-774, 1989.

HINXMAN, A. The Trim-Loss and Assortment Problems: A Survey. **European Journal of Operational Research**, v.5, p.8-18, 1980.

HODGSON, T., A combined Approach to the Pallet Loading Problem **IIE Transaction** v.14, p. 176-182, 1982.

JUNQUEIRA, L., MORABITO, R. Modelos de Otimização para Problemas de Carregamento de Contêineres com Considerações de Estabilidade e de Empilhamento. **Aceito para publicação na revista Pesquisa Operacional**, 2009.

KOCHENBERGER, G.A., RICHARD, V. H. A Simple, All Primal Branch and Bound Approach to Pure and Mixed Integer Binary Problems. **Operational Research Letters**, vol.1, n.5, p.393-403, 1982.

HOTO, R., MACULAN, N., MARQUES, F., ARENALES, M. Um Problema de Corte com Padrões Compartimentados. **Pesquisa Operacional**, v.23, n.1, p.169-187, 2003.

LEACHMAN, R. C., BENSON, R. F., LIU, C., RAAR, D. J. IMPReSS: an automated production-planning and delivery-quotation systems at Harris Corporation semiconductor sector. **Interfaces**, v.26, n.1, p.6-37, 1996.

LEE, H. L., BILLINGTON, C., CARTER, B. Hewlett-Packard Gains Control of Inventory and Service through Design for Localization. **Interfaces**, v.23, n.4, p.1-11, 1993.

LIN, G.; Ettl, M.; BUCKLEY, S.; BAGCHI, S.; YAO, D. D.; NACCARATO, B.L.; ALLAN, R.; KIM, K.; KOENIG, L. Extended Enterprise Supply Chain Management at IBM Personal Systems Group and Other Divisions. **Interfaces**, v.30, n.1, p.7-25, 2000.

LINS, L., LINS, S., MORABITO, R. A 9-Fold Partition Heuristic for Packing Boxes into a Container. **Investigación Operativa**, v.7, n.3, p.69-82, 1999.

LINS, L., LINS, S., MORABITO, R. An n -tet graph approach for non-guillotine packing of n -dimensional boxes into an n -container. **European Journal of Operational Research**, v.141, p.421-439, 2002.

LIU, F.F., HSIAO, C. A Three-Dimensional Pallet Loading Method for Single-Size Boxes. **Journal of the Operational Research Society**, v.48, p.726-735, 1997.

LODI A., MARTELLO S., VIGO D. Heuristic and Metaheuristic Approaches for a Class of Two-Dimensional Bin Packing Problems, **INFORMS Journal on Computing**, v.11, p.345-357, 1999.

LODI, A., MARTELLO, S., MONACI, M. Two-dimensional packing problems: A survey. **European Journal of Operational Research**, v.141, p.241-252, 2002.

LODI A., MARTELLO S., VIGO D. Heuristic Algorithms for the Three-Dimensional Bin Packing Problem, **European Journal of Operational Research**, v.141, p. 410-420, 2002a.

MADSEN, O. Glass cutting in a small firm. **Mathematical Programming**, v.17, p.85-90, 1979.

MADSEN, O. An application of travelling-salesman routines to solve patternallocation problems in the glass industry. **Journal of the Operational Research Society**, v.39, p.249-256, 1988.

MAGEE, J.F. **Logística Industrial: Análise e Administração dos Sistemas de Suprimento e Distribuição**. São Paulo: Pioneira, 1977.

MARTELLO, S., TOTH, P., **Knapsack Problems: Algorithms and Computer Implementations**. John Wiley & Sons, Chichester, 1990.

MARTELLO, S., PISINGER, D., VIGO, D. The Three Dimensional Bin Packing Problem. **Operations Research**. v. 48, p.256-267, 2000.

MORABITO, R. **Uma Abordagem em Grafo-E/OU para o Problema do Empacotamento: Aplicação ao Problema do Carregamento de Paletes e Contêineres**. São Carlos, 1992. Tese de Doutorado, Escola de Engenharia de São Carlos, USP.

MORABITO, R., ARENALES, M. Um exame dos Problemas de Corte e Empacotamento. **Pesquisa Operacional**, v.12, n.1, p.1-20, 1992.

MORABITO, R., ARENALES, M. An And/Or-graph Approach to the Container Loading Problem. **International Transactions in Operations Research**, v.1, n.1, p.59-73, 1994.

MORABITO, R., ARENALES, M. Abordagens para o Problema do Carregamento de Contêineres. **Pesquisa Operacional**, v.17, n.1, p.29-56, 1997.

MORABITO, R., ARENALES, M. Optimizing the cutting of stock plates in a furniture company. **International Journal of Production Research**, v.38, n.12, p.2725-2742, 2000.

MORABITO, R. Problemas de Corte e Empacotamento, Livro-texto de Mini curso, **Elavio**, Montevidéo – Uruguai, 2004.

MORABITO, R., MORALES, S. A Simple Effective Recursive Procedure for the Manufacturer's Pallet Loading Problem. **Journal of the Operational Research Society**, v.49, p.819-828, 1998.

MORABITO, R., MORALES, S. Errata: A simple and effective recursive procedure for the manufacturers' pallet loading problem. **Journal of the Operational Research Society**, v.50, p.876, 1999.

MORABITO, R., PUREZA, V., **“Loading optimization”, in: Optimization in Food Engineering**, Ferruh Erdogdu (ed), Taylor & Francis CRC Press, 2008.

MOURA, A., OLIVEIRA, J. F. A GRASP Approach to the Container-Loading Problem, **IEEE Intelligent Systems**, v.4, p.50-57, 2005.

NGOI, B. K. A., WHYBREW, K., A Fast Spatial Representation Method. **Journal of International Advance Manufacturing Technology**, v.8, p.71-77, 1993.

NGOI, B. K. A., TAY, M. L., CHUA, E. S. Applying spatial representation techniques to the container packing problem, **International Journal of Production Research**, v.32, p.111-123, 1994.

NELISSEN, J. Solving the pallet loading problem more efficiently. Working Paper, **Graduiertenkolleg Informatik und Technik, Aachen**. 1994.

NELISSEN, J. How to use the structural constraints to compute an upper bound for the pallet loading problem. **European Journal of Operational Research**, v.84, p.662-680, 1995.

OLIVEIRA, L.K. **Métodos Exatos Baseados em Relaxação Lagrangiana e Surrogate para o Problema de Carregamento de Paletes do Produtor**. São Carlos, 2004. Tese de Doutorado, Departamento de Engenharia de Produção, UFSCar.

PILEGGI, C. F., MORABITO, R., ARENALES, M. Abordagens para otimização integrada dos problemas de geração e sequenciamento de padrões de corte: caso unidimensional. **Pesquisa Operacional**, v. 25, n. 3, p. 417-448, 2005.

PILEGGI, C. F., MORABITO, R., ARENALES, M. Heurísticas para os problemas de geração e sequenciamento de padrões de corte bidimensionais. **Pesquisa Operacional**, v. 27, n. 3, p. 549-568, 2007.

PINTO, M.J. **Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões**. São José dos Campos, 2004. Tese de Doutorado, Computação Aplicada, Instituto Nacional de Pesquisas Espaciais.

PISINGER, D. Heuristics for the container loading problem, **European Journal of Operational Research**, v.141, p.382-392, 2002.

POLDI, K.C., ARENALES, M., Heurísticas para o Problema de Corte de Estoque Unidimensional Inteiro. **Pesquisa Operacional**, v.23, n.3, p.473-492, 2006.

PUREZA, V., MORABITO, R. Some Experiments With A Simple Tabu Search Approach For The Manufacturer's Pallet Loading Problem. **Computers & Operations Research**, Estados Unidos, v. 33, n. 3, p. 804-819, 2006.

SCHEITHAUER, G. A three dimensional bin packing algorithm. **Journal of Information Processing and Cybernetics**, v.27, p.263-271, 1991.

SCHEITHAUER, G. Algorithms for the container loading problem. **Operations Research Proceedings** 1991, Springer-Verlag Berlin, Heidelberg 1992.

SCHEITHAUER, G., TERNO, J. The G4-heuristic for the pallet loading problem. **Journal of the Operational Research**, v.84, p.662-680, 1996.

SILVA, J., SOMA, N. Um Algoritmo Polinomial para o Problema de Empacotamento de Contêineres com Estabilidade Estática da Carga. **Pesquisa Operacional**, v.23, n.1, p.79-98, 2003.

SMITH, A., DE CANI, P. An algorithm to optimize the layout of boxes in pallets, **Journal of the Operational Research Society**, v.31, p.573-578, 1980.

STEUDEL, H. J. Generating pallet loading patterns: a special case of the two dimensional cutting stock problem. **Management Science**, v.10, p.997-1004, 1979.

TSAI, R.D., MALSTROM, E.M., KUO, W. Three-Dimensional Palletization of Mixed Box Sizes. **I.I.E.Transactions**, v.25, n.4, p.64-75, 1993.

WÄSCHER, G., HAUBNER, H., SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v.183, p.1109-1130, 2007.

YANASSE, H.H., SOMA, N.Y., MACULAN N. An Algorithm for Determining the K-Best Solutions of the One-Dimensional Knapsack Problem. **Pesquisa Operacional**, v.20, n.1, p.117-134, 2000.

YANASSE, H.H., MORABITO, R., "Linear models for one-group two-dimensional guillotine cutting problems", **International Journal of Production Research**, v.44 n.17, p.3471-3491, 2006.

YANASSE, H.H. Minimization of open orders – polynomial algorithms for some special cases. **Pesquisa Operacional**, v.16, p.1-26, 1996.

YANASSE, H.H. On a pattern sequencing problem to minimize the maximum number of open stacks. **European Journal of Operational Research**, v.100 n.3, p.454-463, 1997.

YANASSE, H.H., LAMOSAS, M.J.P.. An integrated cutting stock and sequencing problem. **European Journal of Operational Research**, v.183, n.3, p.1353-1370, 2007.

APÊNDICE A

8.1 Tamanhos de lista tabu

A tabela abaixo mostra para cada instância, os valores de tamanho de lista tabu (*tlist*) utilizados nas execuções de *TSI* e *TSI_p* e aqueles com os quais foi obtido o melhor resultado.

Tabela A.1 – Valores de tamanho de lista tabu (*tlist*) utilizados.

Instância	Tamanhos de lista testados	Tamanhos de lista que geraram o melhor resultado
1	5 a 15	5 a 15
2	5 a 15	7
3	6 a 16	6, 7, 9, 11, 12 e 14
4	15 a 25	17
5	15 a 25	15, 19, 20, 23, 24 e 25
6	15 a 25	16 e 21
7	15 a 25	18 e 25
8	15 a 25	15 a 25
9	24 a 34	26, 27, 31 e 33
10	25 a 35	25, 28, 31, 32, 34 e 35
11	35 a 45	35 a 45
12	53 a 63	53 e 63
13	53 a 63	54 e 56
14	90 a 100	90 e 91

8.2 Estruturas de dados

Esta seção descreve as estruturas de dados utilizadas pelo algoritmo. São apresentadas as estruturas de dados envolvidas nas principais funções do algoritmo, ou seja, aquelas envolvidas com:

- Armazenagem de dados;
- Cálculo e realização de movimentos;
- Deslizamento de blocos.

8.2.1 Armazenagem

Parâmetros de entrada (*input*): Comprimento, largura e altura do contêiner, e comprimento, largura e altura das caixas. Estes 6 dados inteiros e positivos são armazenados nas constantes L, W, H, ll, ww e hh , respectivamente.

Variáveis que armazenam características dos blocos: Fornecem as seguintes informações:

1. Quantidade de caixa em cada camada

As variáveis $numx, numy$ e $numz$ inteiras e positivas armazenam a quantidade de caixas de cada bloco ao longo do eixo x , eixo y e eixo z , respectivamente.

2. Orientação dos blocos

A variável ox armazena esta informação como um número inteiro e positivo entre 1 e 6 (número de possíveis orientações de caixas). As seis possíveis faces inferiores da caixa são ilustradas abaixo. Por exemplo, $ox=1$ representa a situação em que as caixas do bloco em questão tem hl como face inferior e o lado de dimensão h é paralelo ao eixo y do contêiner.

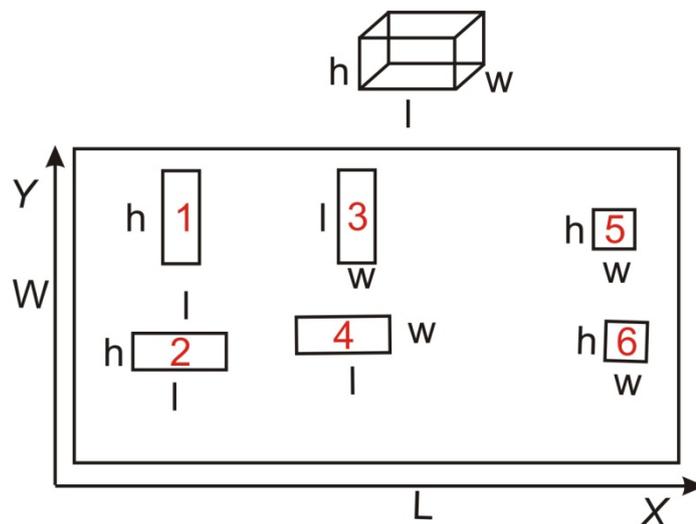


FIGURA A.1 – Orientação dos blocos

3. Posição do bloco no contêiner

Seis variáveis inteiras positivas $(x_1, y_1, z_1, x_2, y_2, z_2)$ são utilizadas para armazenar as coordenadas inferior esquerda (x_1, y_1, z_1) e superior direita (x_2, y_2, z_2) para

identificar a posição do bloco e definir todo o espaço ocupado pelo mesmo no contêiner. A figura abaixo ilustra estas coordenadas em um bloco genérico.

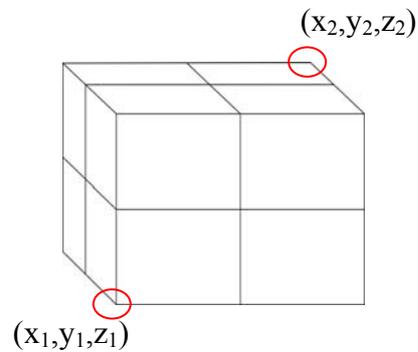


FIGURA A.2 – Posição do bloco no contêiner.

As variáveis acima descritas são, na realidade, campos de um “*record*” (também denominado estrutura por outras linguagens computacionais) denominado *attrib*:

attrib=record

numx : integer;

numy : integer;

numz : integer;

ox : integer;

x1 : integer;

x2 : integer;

y1 : integer;

y2 : integer;

z1 : integer;

z2 : integer;

end;

struct2=array[1..maxblock] of attrib;

A estrutura *bl_att* é do tipo *struct2* e consiste de um *array* de *attrib*. Ou seja, ela armazena para cada bloco do padrão corrente (cada posição do *array*), os atributos do bloco. O número de blocos no padrão corrente (variável inteira) é armazenado na variável *blk*.

Exemplo de padrão de carregamento armazenado:

Instância: $(L,W,H,ll,ww,hh)=(50,50,50,9,7,5)$.

$blk=2$.

Bloco 1	Bloco 2
$bl_att[1].numx = 1$	$bl_att[2].numx = 5$
$bl_att[1].numy = 5$	$bl_att[2].numy = 10$
$bl_att[1].numz = 7$	$bl_att[2].numz = 7$
$bl_att[1].x1 = 0$	$bl_att[2].x1 = 5$
$bl_att[1].y1 = 5$	$bl_att[2].y1 = 0$
$bl_att[1].z1 = 0$	$bl_att[2].z1 = 0$
$bl_att[1].x2 = 5$	$bl_att[2].x2 = 50$
$bl_att[1].y2 = 50$	$bl_att[2].y2 = 50$
$bl_att[1].z2 = 49$	$bl_att[2].z2 = 49$
$bl_att[1].ox = 3$	$bl_att[2].ox = 4$

As informações acima representam o seguinte padrão de carregamento:

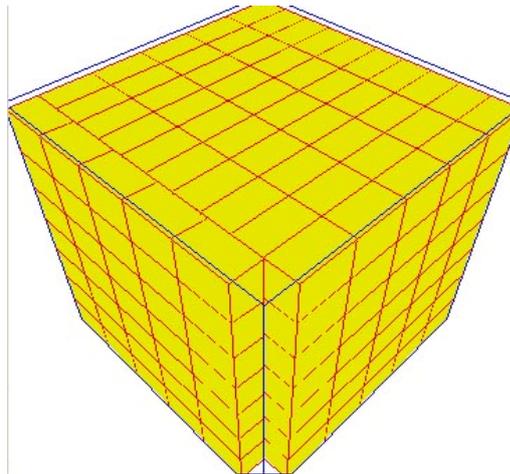


FIGURA A.3 – Padrão de carregamento.

8.2.2 Cálculo e realização de movimentos

Os movimentos de expansão para cada uma das 6 direções são calculados para cada bloco do padrão de carregamento corrente. Como os movimentos em qualquer direção são similares, o procedimento é ilustrado com um exemplo de expansão à direita.

Exemplo de aplicação de um movimento de expansão à direita:

Na expansão do bloco i (ilustrado em azul na figura abaixo), movimentos são computados adicionando-se uma caixa por vez em cada camada ($numy$ e $numz$). Note que o bloco i possui $numx=4$, $numy=3$ e $numz=2$.

Passos:

1. O primeiro movimento calculado consiste na adição de uma caixa (em amarelo) na camada inferior do bloco (camada 1), de acordo com a figura abaixo:

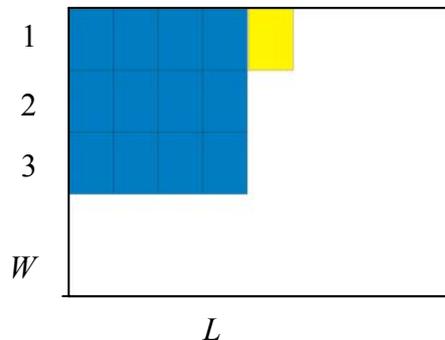


FIGURA A.4 – Aplicação do movimento à direita: $numx=1$.

2. Os próximos movimentos calculados são a adição de caixas, uma a uma, acima da(s) caixa(s) que foram expandidas na camada 1 até atingir o valor de $numz$ (neste exemplo, $numz=2$). Desta forma, movimentos à direita incluem também a adição de caixas (expansões) na direção do eixo perpendicular associado (z). Note que cada adição de caixa(s) numa camada de $numz$ corresponde a um movimento distinto.
3. Em seguida é adicionada uma caixa ao lado da(s) caixa(s) que foram adicionadas no passo 1, conforme figura abaixo:

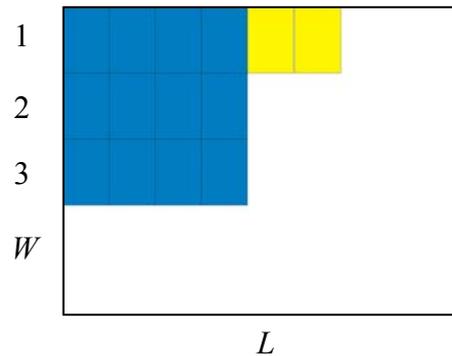


FIGURA A.5 – Aplicação do movimento à direita: $numx=2$.

4. Os passos 2 e 3 são repetidos enquanto a adição de caixas na camada 1 não ultrapassar a borda direita do mesmo.
5. Os passos seguintes consistem em se repetir os passos 2 e 3 para as demais camadas do bloco individualmente, ou seja, camadas 2 e 3.
6. A seguir, são consideradas expansões de duas ou mais camadas adjacentes simultaneamente. No exemplo acima são consideradas as expansões de:
 - camadas 1 e 2
 - camadas 1, 2 e 3
 - camadas 2 e 3

Sempre que um movimento é calculado, seus atributos são armazenados na estrutura abaixo juntamente com seu valor associado:

storingp=record

value: integer; // ganho real em numero de caixas (G_0)

block: integer; // identificador do bloco ativo

co: integer; // novo valor da coordenada da face do bloco expandida

layer: integer; // número da camada inicial de expansão do bloco

layerz: integer; // número da camada inicial de expansão do bloco no eixo perpendicular à expansão principal

vert: integer; // numero de camadas de expansão adjacentes

vertz: integer; // numero de camadas de expansão adjacentes em no eixo perpendicular à expansão principal

num: integer; // numero de caixas adicionadas em cada camada de expansão

```

split: integer; // identificador da configuração aplicada aos blocos passivos
hetearea: real; // componente do valor do movimento referente a proporção
                de área nos blocos criados com orientação diferente da do bloco
                ativo
minarea: real; // componente do valor do movimento referente a proporção
                de área superposta dos blocos passivos sobre a área de expansão
                do bloco ativo.
fo: real; // valor do movimento.
kind: char; // tipo de movimento: expansão à direita (r), esquerda (l), acima
                (u), abaixo (d), frente (f), atrás (a) e troca de orientação (m)
newox:integer; // nova orientação do bloco ativo se houver troca
end;
struct1=array[1..maxmov] of storingp;

```

Note que a estrutura *struct1* consiste de um *array* de *storingp*. Ou seja, cada posição deste *array* armazena os atributos de um dado movimento. Note também que em cada posição, *value*, *hetearea*, *minarea* e *fo* armazenam os dados para cálculo da função de avaliação dos movimentos (seção 4.2.1).

A estrutura abaixo armazena o valor e o identificador dos movimentos calculados com a finalidade de ordenação e identificação do movimento mais valioso:

```

sortingp=record
    move: integer; // identificador do movimento, correspondente à
                    posição do movimento no array struct1
    fo: real; // valor do movimento
end;
struct7=array[1..maxmov] of sortingp;

```

Após a seleção do movimento (movimento não tabu com maior valor de *fo*), as informações armazenadas na posição do movimento em *struct1* são utilizadas para a geração do novo padrão.

Em movimentos envolvendo troca de orientação das caixas do bloco (movimentos simples ou híbridos), a nova orientação do bloco é armazenada em *newox*. Em movimentos híbridos, o padrão corrente é armazenado em uma estrutura similar a *bl_att*, seguido da aplicação de mudança de orientação e expansão. Calculado o movimento, e antes do cálculo do movimento seguinte, o carregamento corrente é recuperado.

8.2.3 Deslizamento de blocos

Os blocos são deslocados para as extremidades do contêiner como prescrito na seção 4.2.3. As estruturas de dados responsáveis pelo deslizamento são apresentadas a seguir.

- Armazenagem das coordenadas (unitárias) de posicionamento das caixas:

```
struct4=array[1..2000] of integer;
Xset, Yset, Zset: struct4;
```
- Armazenagem das coordenadas dos blocos que precedem o sucedem um dado bloco *i* em relação aos eixos *x,y* e *z*:

```
ord=record
  x1,x2,y1,y2,z1,z2: integer; // coordenadas x1, x2, y1,y2, z1 e z2 do bloco i
  predx1,sucx1: integer; // coordenada x1 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada x1
  predx2,sucx2:integer; // coordenada x2 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada x2
  predy1,sucy1:integer; // coordenada y1 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada y1
  predy2,sucy2:integer; // coordenada y2 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada y2
  predz1,sucz1: integer; // coordenada z1 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada z1
  predz2,sucz2: integer; // coordenada z2 do bloco que precede e do bloco
  que sucede o bloco i segundo a coordenada z2
end;
struct5=array[1..maxblock] of ord;
```

order: struct5;

- Estrutura para ordenação dos blocos segundo coordenadas, utilizada no cálculo de espaços vazios entre blocos passivos e blocos a eles adjacentes

```
pord=record //
    co,blk:integer; // coordenada relevante, numero do bloco
    end;
struct6=array[1..maxblock] of pord;
orderx1, orderx2, ordery1, ordery2, orderz1, orderz2: struct6;
```

- Variáveis de suporte

```
headx1, headx2, heady1, heady2, headz1, headz2, tailx1, tailx2, taily1,
taily2,tailz1, tailz2: integer;
```

Foi utilizado o procedimento *heapify*, vastamente abordado em livros de estruturas de dados (CORMEN *et al.* 2002), para ordenar os blocos segundo suas coordenadas. Para cada bloco corrente, são armazenadas suas coordenadas $(x_1, x_2, y_1, y_2, z_1, z_2)$ e o identificador do bloco nas estruturas *orderx1*, *orderx2*, *ordery1*, *ordery2*, *orderz1* e *orderz2*, respectivamente.

Aplicação de heapify para ordenação de blocos em ordem de x_1 :

```
headx1:=orderx1[1].blk;
```

Headx1 recebe o numero do bloco com menor valor de x_1 , ou seja, o bloco cujo início é o mais próximo do limite esquerdo do contêiner.

```
order[headx1].x1:=orderx1[1].co;
```

Na estrutura *order[.].x1*, é armazenada a coordenada do bloco cujo numero está armazenado em *headx1*.

```
v:=headx1;
```

A variável *v* recebe o numero do bloco em *headx1*, o bloco "cabeça" da ordenação.

```
order[v].predx1:=0;
```

Order[v].predx1 armazena o bloco que precede *v* na ordenação em x_1 ; como *v* é o bloco com a menor coordenada segundo x_1 , ele não tem predecessores e esta estrutura recebe o valor 0.

Reaplicação de heapify para ordenação de blocos em ordem de x_1 :

$order[v].sucx1 := orderx1[1].blk;$

Em $order[v].sucx1$, armazena-se a informação de que o sucessor em x_1 do ultimo bloco já ordenado (v) é o bloco na primeira posição de *heapify*.

$order[orderx1[1].blk].predx1 := v;$

Em $order[orderx1[1].blk].predx1$, armazena-se a informação de que o predecessor em x_1 do bloco na primeira posição de *heapify* é ultimo bloco já ordenado (v).

$order[orderx1[1].blk].x1 := orderx1[1].co;$

Em $order[orderx1[1].blk].x1$, armazena-se a coordenada x_1 do bloco na primeira posição de *heapify*.

$v := order[v].sucx1;$

Como o bloco na primeira posição de *heapify* foi ordenado, ele se torna o último bloco já ordenado (v).

Esse procedimento também é aplicado para x_2, y_1, y_2, z_1 e z_2 .

Somente após a ordenação ser realizada e as posições relativas dos blocos entre si serem computadas, é aplicado o movimento de deslizamento nas 6 direções do contêiner. Como os deslizamentos em qualquer direção são similares, o procedimento é ilustrado com um exemplo de deslizamento à direita:

Exemplo de aplicação de deslizamento à direita:

Instância: $(L, W, H, ll, ww, hh) = (8, 7, 5, 1, 1, 1)$.

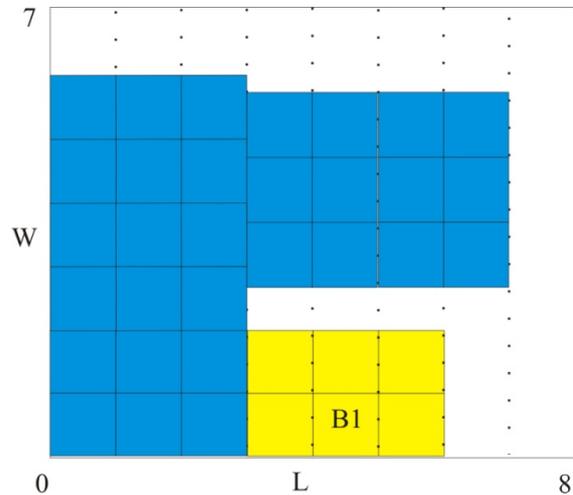


FIGURA A.6 – Aplicação do movimento de deslizamento à direita: *iteração 0*.

Passos:

1. Inicializa-se o procedimento, fazendo v o último bloco segundo a coordenada x_2 ($v := tailx_2$).
2. Repete-se até que ($v=0$) ou ($order[v].x_2 < L/2$) :
 - 2.1. Verifica-se se v está na última posição de x ($order[v].x_2 = L$). (Exemplo: bloco amarelo B1 da figura acima). Se verdadeiro, vai-se para o passo 4. Caso contrário, enquanto $Xset[index] > order[v].x_1$ faz-se $index := index - 1$;

A variável *index* marca a posição na estrutura **Xset** (conjunto de coordenadas x_1 no problema) cuja coordenada associada é a primeira que sucede a coordenada x_1 do bloco v e, portanto, a primeira posição x_1 considerada para deslocar o bloco v .

3. Verifica-se se $order[v].x_1 < L - order[v].x_2$ (bloco v mais próximo da aresta esquerda do contêiner). Se verdadeiro, vai-se para o passo 4. Caso contrário, verifica-se se $Xset[index] + order[v].x_2 - order[v].x_1 > L$ (o deslocamento do bloco ultrapassa a aresta do contêiner). Se verdadeiro, vai-se para o passo 4. Caso contrário:

Repete-se até que $(order[v].x2 = L)$ ou $(index \geq cardx)$ ou $(exit1=true)$:

3.1. $k:=order[v].sucx2$

Neste último caso, fazer k o sucessor de k e repetir o passo considera que v possa ter mais de um sucessor em x_2 , ou seja, mais de um bloco com a mesma coordenada x_2 de k ainda que as coordenadas y seja diferentes. Tais blocos precisam passar no teste de viabilidade do deslizamento.

3.2. Faz-se $exit1:=false$.

3.3. Enquanto $(k \neq 0)$ e $(exit1=false)$:

Verifica-se se $(order[k].x1 < Xset[index] + order[v].x2 - order[v].x1)$ e $(order[k].x2 > Xset[index])$ e $((order[k].y1 < order[v].y2)$ e $(order[k].y2 > order[v].y1))$, ou seja, se a mudança na coordenada de $x1$ causa superposição de blocos. Se verdadeiro faça $exit1:=true$. Caso contrário, faz-se $k:=order[k].sucx2$.

3.4. Atualiza-se as coordenadas do bloco v :

$order[v].x2:=Xset[index] + order[v].x2 - order[v].x1$

$order[v].x1:=Xset[index];$

$bl_att[v].x1:=order[v].x1$

$bl_att[v].x2:=order[v].x2$

3.5. Faz-se $index:=index + 1$

4. Faz-se $v:=order[v].predx2$ (v passa a ser o seu predecessor, cujo deslizamento será investigado)

Para o bloco B1 da Figura A.6, o procedimento executou o laço do passo 2 três vezes. Nas duas primeiras vezes, o bloco foi deslizado para a direita em duas unidades de comprimento. Na 3ª vez, a verificação do passo 2.1 foi confirmada (B1 ocupa a última posição de x), provocando a saída do laço. O deslizamento do bloco predecessor de B1 em x é, então, analisado, dando-se continuidade ao procedimento.

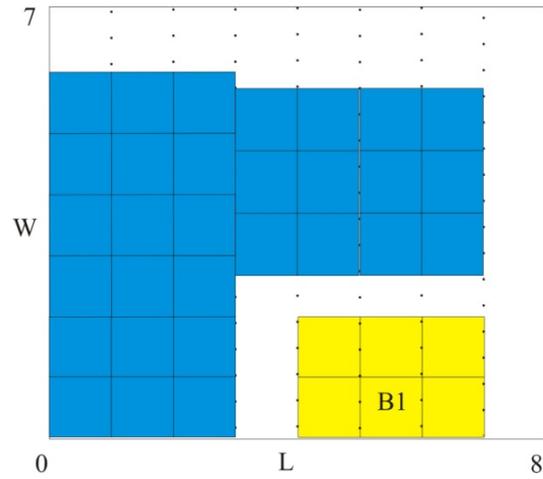


FIGURA A.7 – Aplicação do movimento de deslizamento à direita: *iteração 1*.

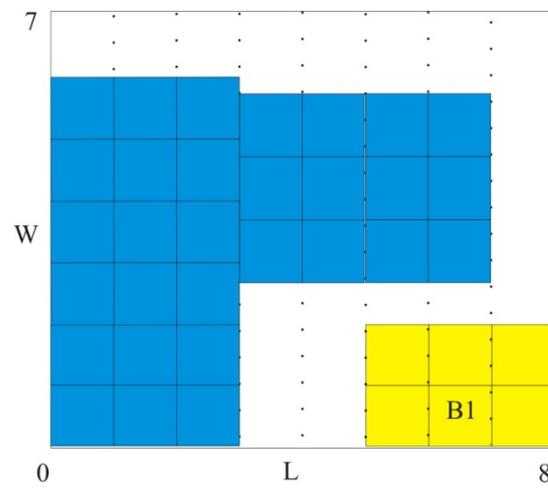


FIGURA A.8 – Aplicação do movimento de deslizamento à direita: *iteração 2*.