3
REVISÃO BIBLIOGRÁFICA

Problemas de Corte e Empacotamento (PCE) são, em geral, problemas de otimização combinatória cujo objetivo é achar o arranjo ótimo de unidades menores (itens) dentro de unidades maiores (objetos). Os PCE aparecem na literatura com diferentes nomes, tais como: carregamento de *pallet*, carregamento de contêiner, Problema da Mochila, *Bin Packing*, Problema de corte de materiais, entre outros (Dyckhoff, 1990).

O Problema de *Bin Packing* consiste em empacotar objetos de diferentes tamanhos ou pesos, no menor número possível de *bins* (caixas) com capacidade fixa (C). Considera-se que os *bins* são indexados (1, 2, 3, 4,...) conforme a ordem de criação dos mesmos.

Coffman *et al* (1998) considera que os algoritmos utilizados para resolver os Problemas de *Bin Packing* são classificados como:

- *On-line* – os objetos são empacotados na ordem que chegam, sem o conhecimento dos demais objetos que serão empacotados;
- *Off-line* – todos os objetos a serem empacotados são conhecidos antes de se iniciar o processo de empacotamento e são armazenados em uma lista *L*.

Existem quatro algoritmos clássicos para o Problema de *Bin Packing*, que são os seguintes:

- *Next Fit* (NF) – o primeiro objeto da lista é alocado em um *bin.* Os demais objetos são empacotados nesse mesmo *bin* até a capacidade do mesmo ser alcançada. Assim que a restrição de capacidade (C) é atingida, esse *bin* é fechado e um novo *bin* é aberto continuando-se o empacotamento dos objetos;
- *First Fit* (FF) – os *bins* são considerados parcialmente cheios e o próximo objeto da lista é alocado no primeiro *bin* que comporta o objeto sem ultrapassar a capacidade (C) do mesmo. Um novo *bin* só é aberto quando o objeto não couber em nenhum outro *bin* que está sendo utilizado;
- *Best Fit* (BF) – o objeto é colocado no *bin* mais cheio onde ele cabe. Se não houver nenhum *bin* aberto que comporte o objeto, um

novo *bin* é aberto. Se houver empate entre *bins*, o critério de desempate é feito escolhendo o *bin* de menor indexação;

- *Worst Fit* (WF) – o objeto é colocado no *bin* mais vazio onde ele cabe. O critério de desempate utilizado é feito escolhendo o *bin* de menor indexação. Se não houver nenhum *bin* aberto que comporte o objeto, um novo *bin* é aberto.

Ao invés de empacotarmos os objetos na ordem que eles aparecem na lista, podemos ordená-los em ordem decrescente de peso ou tamanho e assim, realizar o empacotamento dos mesmos. Utilizando essa regra, os procedimentos acima se transformam em NFD (*Next Fit Decreasing*), FFD (*First Fit Decreasing*), BFD (*Best Fit Decreasing*) e WFD (*Worst Fit Decreasing*).

O objetivo deste capítulo é apresentar alguns métodos utilizados em problemas de Empacotamento que nessa dissertação serão tratados como Problemas de Estivagem. Foram estudados três tipos de Estivagem:

- *Pallets*;
- Contêineres;
- Contêineres em navios.

3.1. Carregamento de *pallets*

Bischoff *et all.* (1995) consideraram o Problema de Carregamento de *Pallet* do Distribuidor, ou seja, empacotar *n* tipos diferentes de caixas retangulares com dimensões l_i (comprimento) x w_i (largura) x h_i (altura) e quantidade m_i (i = 1,..., n) em um *pallet* de dimensão *L* (comprimento) *x* *W* (largura). O objetivo do carregamento é maximizar o volume empacotado respeitando a altura limite do carregamento (H).

O procedimento proposto consiste em carregar o *pallet* em camadas horizontais, de baixo para cima e com, no máximo, dois tipos de caixas por camada.

Inicialmente, a heurística proposta gera todos os carregamentos possíveis da superfície do *pallet,* escolhendo o arranjo que apresentar maior ocupação superficial. Os carregamentos possíveis são formados por carregamentos gerados por um único tipo de caixa ou pela combinação de dois tipos de caixa. Não existe restrição quanto a orientação da caixa.

CARREGAMENTO DA SUPERFÍCIE

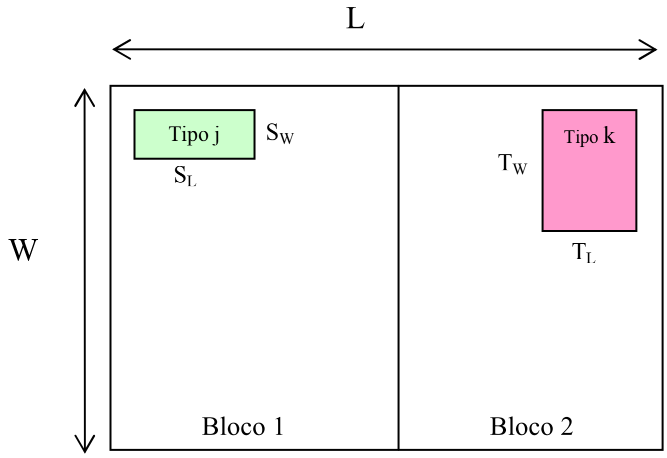

Figura 09 – Processo padrão de geração para dois blocos
Fonte: Bischoff (1995)

A figura 09 ilustra umas das possíveis formas de carregamento da superfície do *pallet* com dois tipos diferentes de caixas. O processo de geração em questão é para dois blocos que particionam o comprimento *L* do *pallet*. Os blocos são formados por caixas do tipo *j* e do tipo *k*. Os lados referentes ao comprimento das caixas (S_L e T_L comprimento das caixas *j* e *k,* respectivamente) são paralelos ao comprimento do *pallet*. Considerando que $S_L \geq T_L$, as alternativas de carregamento podem ser geradas a partir da expansão do Bloco 1 e com o preenchimento da superfície restante (Bloco 2) com caixas do tipo *k*, respeitando a quantidade existente deste tipo de caixa.

Se *l* representa a quantidade de caixas do tipo *j* que podem ser colocadas lado a lado no Bloco 1 com comprimento paralelo L (comprimento da superfície), então *l* pode assumir valores inteiros entre 0 e o menor valor entre $\left\{m_j, \left(L/S_L\right)^-\right\}$[E]

onde *mj* representa o número de caixas do tipo *j*. Sendo $l > 0$, o número de caixas que podem ser alocadas na outra direção é dado pelo menor valor entre

---

[E] $\left(L/S_L\right)^-$ retorna o próximo menor valor inteiro ao se arredondar o valor para baixo

$\left\{\left(\frac{m_j}{l}\right)^{-}, \left(\frac{W}{S_W}\right)^{-}\right\}$ [F]. Um bloco de $\left\{\left[\frac{(L - l * S_L)}{T_L}\right] * \left(\frac{W}{T_W}\right)\right\}^{-}$ [G] caixas do tipo *k* podem ser colocadas na área restante.

Exemplo 1: Carregamento com dois tipos de caixa (figura 10)

Dimensões do *Pallet*	
W	120cm
L	160cm
Caixas do tipo j	
m_j	30
S_L	35cm
S_W	30cm
Caixas do tipo k	
m_k	25
T_L	20cm
T_W	30cm

$(L/S_L)^{-} = 4$

l = min {30; 4} = 4

l pode assumir valores inteiros entre [0, 4]

Assumindo *l* = 4, ou seja, 4 caixas com comprimento paralelo a L, calcularemos a quantidade de caixas na direção W.

$(m_j/l)^{-} = 7$
$(W/ S_W)^{-} = 4$
A quantidade de caixas com a largura paralela a W é 4.

A quantidade de caixas tipo k no bloco 2 é de $[(160-4*35)/20]*[120/30]^{-} = 4$

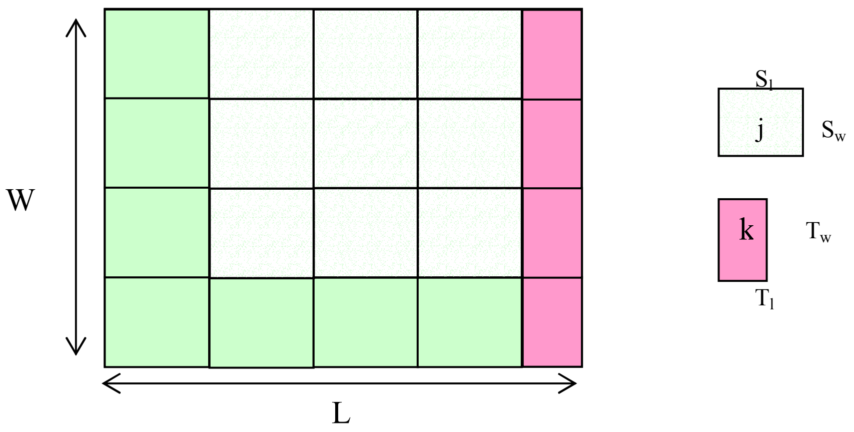

Figura 10 – Exemplo de carregamento com 2 tipos de caixa

[F] $\left(\frac{m_j}{l}\right)^{-}, \left(\frac{W}{S_W}\right)^{-}$ retorna o próximo menor valor inteiro ao se arredondar o valor para baixo

[G] $\left\{\left[\frac{(L - l * S_L)}{T_L}\right] * \left(\frac{W}{T_W}\right)\right\}^{-}$ retorna o próximo menor valor inteiro ao se arredondar o valor para baixo

No caso da camada ser carregada pelo mesmo tipo de caixa porém com orientação diferente, a única diferença em relação ao processo do carregamento com 2 caixas é que temos que subtrair de m_k a quantidade de caixas que foram alocadas no Bloco 1 e assim sucessivamente.

Exemplo 2: Carregamento com um tipo de caixa com orientação diferente (figura 11)

Dimensões do *Pallet*	
W	100cm
L	120cm
Caixas do tipo j	
m_j	50
S_L	30
S_W	20
Caixas do tipo k	
m_j’	-
T_L	20
T_W	30

$(L/S_L)^- = 4$

$l = \min\{50; 4\} = 4$

l pode assumir valores inteiros entre [0, 4]

Assumindo $l = 2$, ou seja, 2 caixas com comprimento paralelo a L, calcularemos a quantidade de caixas na direção W.

$(m_j/l)^- = 25$

$(W/ S_W)^- = 5$

A quantidade de caixas com a largura paralela a W é 5.

O total de caixas do tipo j é igual a 10.

Atualizando o nº de caixas do tipo j temos: m_j’$= 50-10 = 40$

A quantidade de caixas tipo k no bloco 2 é de $[(120-2*30)/20]*[100/30]^- = 9$

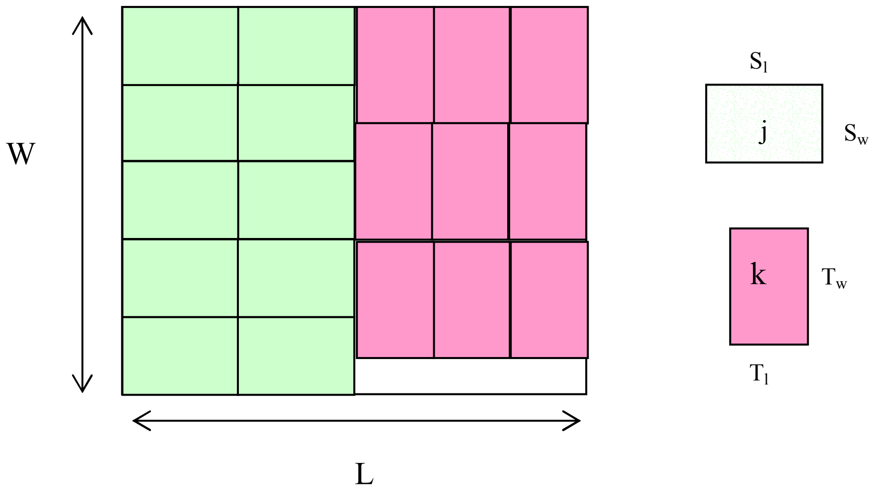

Figura 11 – Exemplo de carregamento com 1 tipo de caixa com orientação diferente

SUPERFÍCIES DE CARREGAMENTO

Novas superfícies de carregamento são criadas assim que uma camada é carregada. Essas superfícies são armazenadas na lista de superfícies potenciais de carregamento. Quando o carregamento é feito com dois tipos de caixa, podem ser geradas até 2 novas superfícies de carregamento. Essas superfícies são mais altas do que a anterior. Além disso, existe a necessidade de se gerar superfícies de carregamento a partir de áreas não-utilizadas no carregamento anterior. A figura 12 exemplifica a geração de superfícies potencias de carregamento.

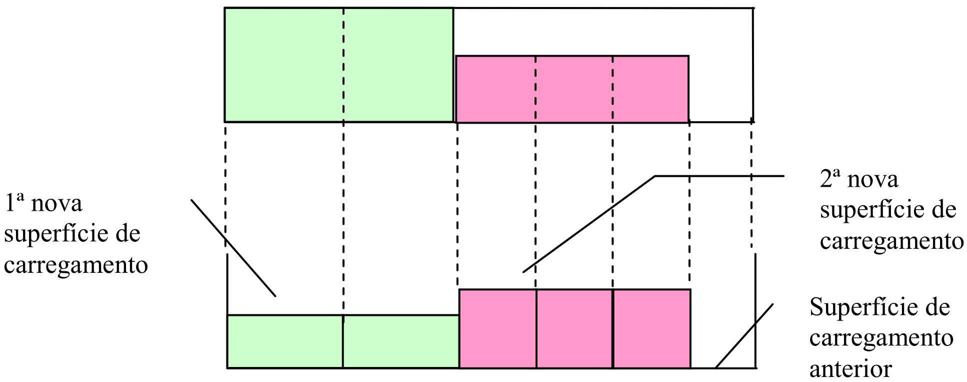

Figura 12 - Vista superior e lateral do carregamento com 2 tipos de caixa

A heurística proposta por Bischoff procura evitar a formação de superfícies longas e finas devido à dificuldade para carregar esse tipo de superfície. No caso ilustrado na figura 13, a heurística escolheria como superfície de carregamento **A** e **B** ao invés de **C** e **D** pois **C** é uma superfície longa e fina a qual a heurística tende a evitar.

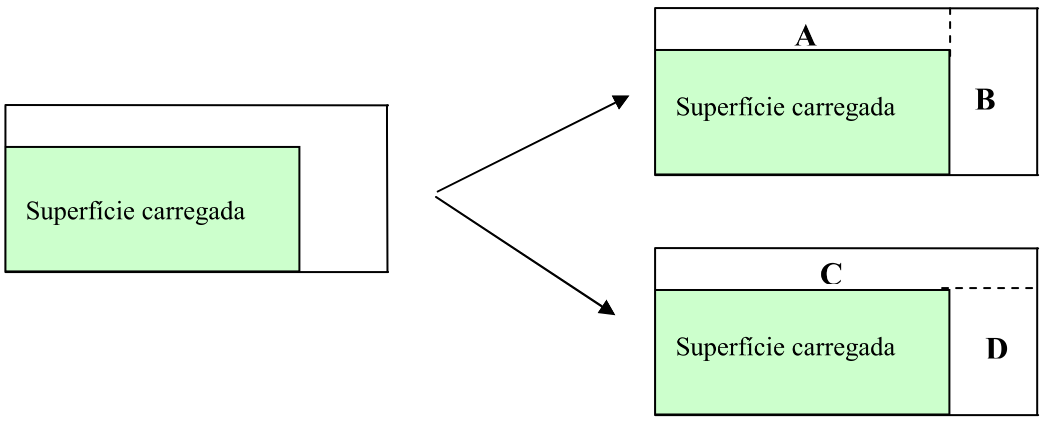

Figura 13 – Geração de superfícies de carregamento

Além de evitar a formação de superfícies longas e finas, a heurística busca a estabilidade do carregamento, priorizando o carregamento das superfícies mais baixas.

Quando a superfície selecionada para carregamento não comportar nenhuma das caixas que ainda podem ser empacotadas, essa superfície é descartada.

A desvantagem da heurística de Bischoff é a possibilidade de ocorrência da fragmentação das superfícies de carregamento nas etapas iniciais do algoritmo. Quando é utilizado o padrão de geração por dois blocos, duas novas superfícies (menores ou iguais à original) são geradas acima dos blocos atuais e assim sucessivamente, gerando uma fragmentação das superfícies de carregamento, o que dificulta o carregamento de caixas maiores.

Para minimizar o problema, o algoritmo verifica a possibilidade de unir as superfícies facilitando assim o carregamento da mesma. Duas superfícies só podem ser unidas se apresentarem a mesma altura e se forem adjacentes.

O algoritmo termina quando não existem superfícies a serem carregadas ou quando não existem caixas para serem empacotadas.

A figura 14 resume a heurística proposta por Bischoff.

.

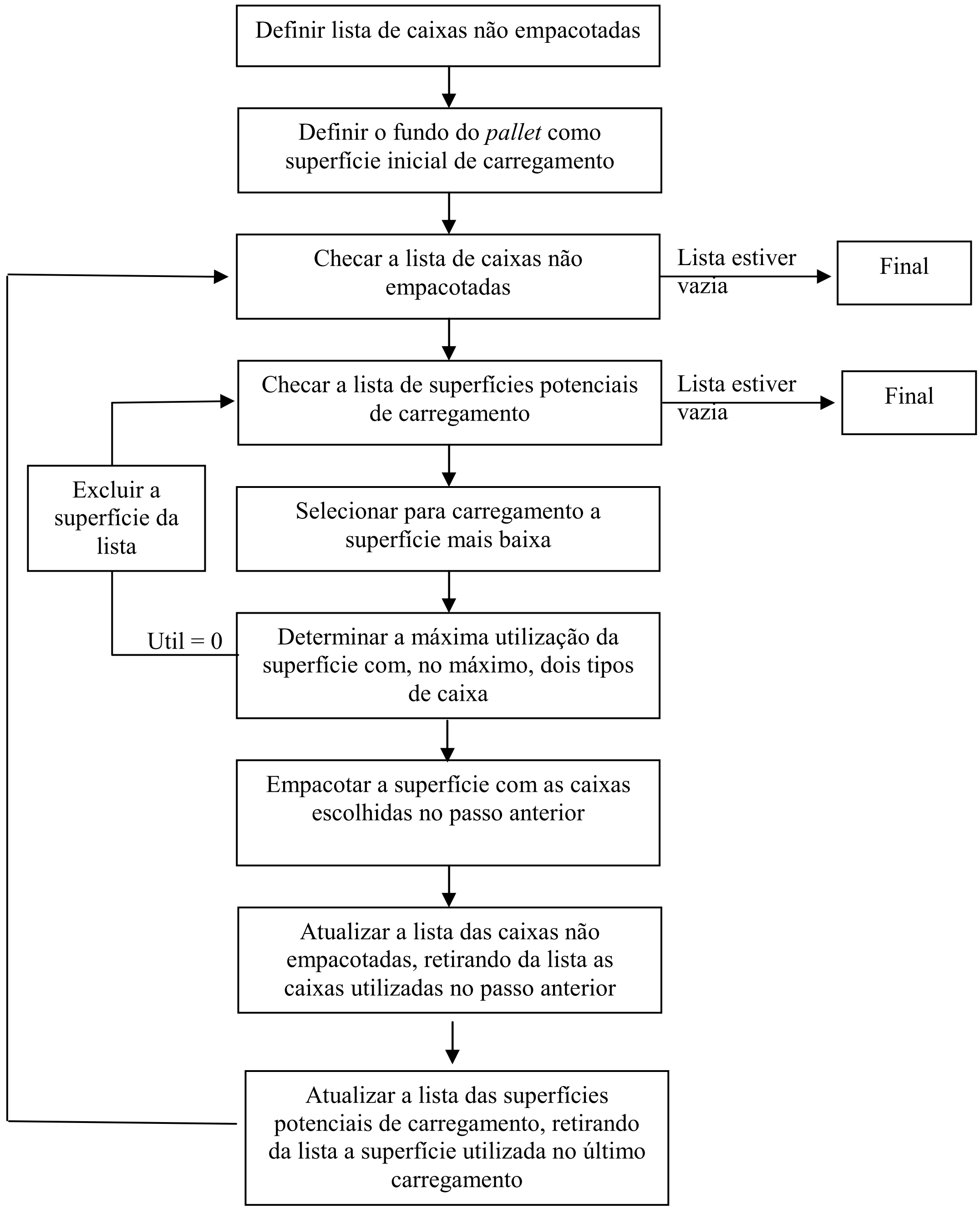

Figura 14 – Fluxograma da heurística de Bischoff
Fonte: Bischoff (1995)

Bischoff e Dowsland (1982) propuseram uma heurística para carregamento de *pallets* do produtor baseado na heurística proposta por Smith e De Cani (1980) e Steudel (1979).

O Problema de Carregamento de *Pallet* do Produtor consiste em arranjar ortogonalmente e sem sobreposição caixas com comprimento *a* e largura *b* sobre *pallet* com comprimento A e largura B.

A heurística proposta por Steudel divide o *pallet* em até quatro regiões retangulares ou blocos. Dentro de cada bloco, as caixas possuem orientação pré-fixada. A dimensão desses blocos é determinada em duas etapas. O objetivo da primeira etapa é maximizar a utilização do perímetro do *pallet* e na 2ª etapa são feitas modificações nas dimensões dos blocos a fim de tentar o preenchimento da parte central do *pallet*. A heurística escolhe o *layout* gerado com o maior número de caixas como o padrão. As etapas da heurística são ilustradas na figura 15.

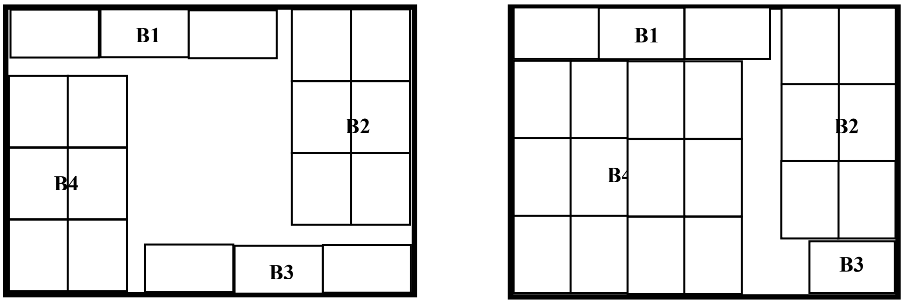

Figura 15 – 1ª e 2ª etapas da Heurística de Steudel
Fonte: Bischoff e Dowsland (1982)

A heurística proposta por Smith e De Cani também divide o *pallet* em até quatro regiões retangulares e a orientação das caixas dentro de cada região é pré-fixada. A heurística avalia todas as possibilidades de variação das dimensões dos blocos, inclusive a possibilidade de dimensão nula e retorna o padrão que tiver o maior número de caixas carregadas. As restrições existentes nessa heurística impedem a geração de *layouts* com sobreposição de caixas. A figura 16 ilustra um *layout* gerado pela heurística.

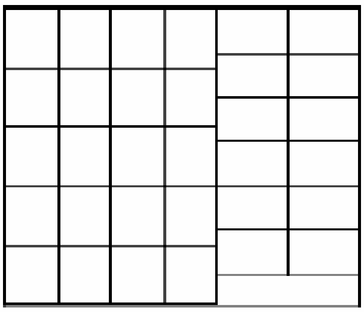

Figura 16 – *Layout* gerado pela Heurística de Smith e de Cani
Fonte: Bischoff e Dowsland (1982)

Baseado nas heurísticas descritas anteriormente, Bischoff e Dowsland propuseram uma heurística que divide o *pallet* em até 5 regiões aproveitando a região, que, pelas heurísticas anteriores, ficava vazia.

As dimensões das áreas 1, 2, 4 e 5 da figura 17 determinam a dimensão da área 3.

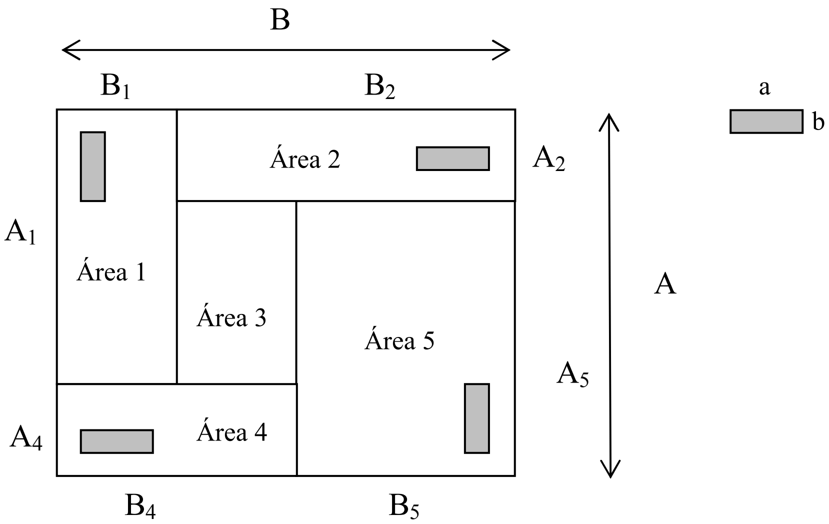

Figura 17 – Divisão do *pallet* em 5 regiões
Fonte: Bischoff e Dowsland (1982)

Na 1ª etapa da heurística, são calculadas partições eficientes do comprimento A e da largura B do *pallet*. Essas partições são definidas como uma combinação (n, m) (números inteiros não negativos) de caixas de comprimento *a* e largura *b* que devem satisfazer as seguintes condições:

$na + mb \leq S$

$S - na - mb \leq b$

onde:

- S é a dimensão do lado do *pallet* que está sendo considerada;
- *b* é a menor dimensão da caixa;
- *n* é a quantidade de caixas em uma área do *pallet* com comprimento paralelo ao lado do *pallet*;
- *m* é a quantidade de caixas em uma área do *pallet* com largura paralela ao lado do *pallet*.

Na 2ª etapa da heurística, são gerados os *layouts* a partir das partições eficientes dos quatro lados do *pallet* gerado na etapa anterior.

O autor denominou (n_L, m_L) e (n_R, m_R) as partições eficientes do lado esquerdo e do lado direito do *pallet*, respectivamente. Os lados de cima e de baixo foram denominados de (n_T, m_T) e (n_B, m_B).

As áreas 1, 2, 4 e 5 da figura 17 são alocadas nos cantos do *pallet* e tem suas dimensões definidas por:

$A1 = n_L a$ $\quad$ $B1 = m_T\, b$

$A2 = m_R\, b$ $\quad$ $B2 = n_T\, a$

$A4 = m_L\, b$ $\quad$ $B4 = n_B\, a$

$A5 = n_R\, a$ $\quad$ $B5 = m_B\, b$

A região 3 é a região restante do *pallet*. A heurística calcula quantas caixas e em que posição (horizontal ou vertical) as caixas serão dispostas na Região 3.

As seguintes condições devem ser evitadas a fim de não ocorrer a sobreposição de áreas:

$$A - A_1 - A_5 < 0 \quad \text{e} \quad B - B_1 - B_5 < 0$$

ou

$$A - A_2 - A_4 < 0 \quad \text{e} \quad B - B_2 - B_4 < 0$$

3.2
Carregamento de Contêiner

George e Robinson (1980) consideraram o Problema de Carregamento de Contêiner cujo objetivo é carregar *i* tipos de caixas diferentes com dimensões l_i (comprimento) x w_i (largura) x h_i (altura) e quantidade conhecida (b_i) dentro de um contêiner dimensão com L (comprimento) x W (largura) x H (altura). O problema consiste em achar uma ordem de carregamento e a posição das caixas dentro do contêiner de forma a otimizar o aproveitamento do espaço do contêiner. Mesmo que a soma dos volumes das caixas seja menor que o volume do contêiner, é possível que alguma(s) caixa(s) não seja(m) empacotada(s) devido ao arranjo geométrico das mesmas.

Nesta heurística, as caixas são empilhadas em camadas verticais ao longo da largura do contêiner (W) e a partir de uma das paredes do mesmo, conforme a figura 18. Não é feita nenhuma restrição em relação à quantidade de caixas que podem ser empilhadas umas sobre as outras. É desejável que caixas do mesmo tipo sejam colocadas próximas uma das outras.

O contêiner é preenchido camada por camada e uma nova camada só é criada quando a camada atual estiver carregada.

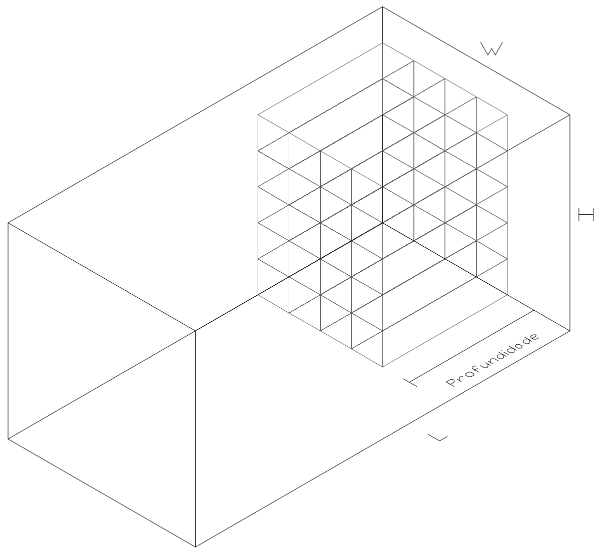

Figura 18 – Carregamento do contêiner por camada

A Heurística proposta por George e Robinson pode ser dividida em cinco etapas:

1ª etapa: Achar o primeiro tipo de caixa a ser carregada em uma nova camada e o tamanho da profundidade da camada;

2ª etapa: Definir a altura e a largura da caixa que será utilizada e a quantidade de caixas empacotadas;

3ª etapa: Criar novos espaços de carregamento;

4ª etapa: Escolher o novo espaço de carregamento;

5ª etapa: Achar o tipo de caixa para preencher os espaços vazios já existentes nas camadas e o tamanho da profundidade.

Antes de explicarmos passo-a-passo a heurística proposta, iremos introduzir o conceito de "ranqueamento das caixas" que será utilizado em alguns passos da heurística. O ranqueamento das caixas é feito da seguinte forma: primeiramente, escolhe-se a caixa com a maior entre as menores dimensões. Havendo empate neste critério, escolhe-se o tipo de caixa de maior quantidade. E como último critério de desempate, escolhe-se a caixa com maior dimensão. O exemplo ilustrado na tabela 03 esclarece os critérios de desempate:

Tabela 03 – Critérios de desempate

Caixas	Quantidade (b_i)	Comprimento (l_i)	Largura (w_i)	Altura (h_i)	Menor dimensão
1	12	7	3	4	3
2	12	6	3	3	3
3	14	5	4	5	4
4	16	5	5	6	5

Neste caso, a primeira caixa a ser empacotada é a caixa 4, seguida pela caixa 3. As caixas 1 e 2 ficaram empatadas tanto em relação ao critério caixa com maior entre as menores dimensões quanto em relação ao critério quantidade. O desempate entre as caixas ocorre em relação ao critério caixa com maior dimensão devendo a caixa 1 ser empacotada antes da caixa 2. Portanto, a ordem de carregamento
é:

4, 3, 1, 2

Passo-a-passo da heurística:

1ª etapa: Achar o primeiro tipo de caixa a ser carregada em uma nova camada e o tamanho da profundidade da camada.

Entende-se por camada de carregamento uma seção do comprimento do contêiner (L) na sua completa altura H e largura W. A seção do comprimento do contêiner utilizada é denominada profundidade da camada.

Os tipos de caixas podem ser classificados como *"aberta"* (quando um determinado tipo de caixa já foi utilizado no carregamento) ou como "*fechada*" (quando um tipo de caixa ainda não foi utilizado no carregamento).

Para iniciar o carregamento de uma nova camada, verifica-se se existe alguma caixa classificada como *"aberta"*.

- ✓ Se existir mais de um tipo de caixa classificada como "aberta", a heurística sugere dois critérios para desempate: escolher o tipo de caixa com maior quantidade de caixas ou escolher a caixa melhor ranqueada conforme os critérios descritos anteriormente;

- ✓ Caso só existam caixas classificadas como "fechadas", escolhe-se a caixa com melhor posição no ranking das caixas.

Em cada camada, a maior dimensão da primeira caixa a ser carregada determina a profundidade da camada. As outras dimensões da caixa (altura e largura) devem ser menores do que as dimensões do contêiner.

As caixas que ainda não foram empacotadas são armazenadas na "Lista de Caixas não Empacotadas".

2ª etapa: Definir a altura e a largura da caixa que será utilizada e a quantidade de caixas empacotadas.

Definido o tipo de caixa que será utilizado e a dimensão da profundidade, o algoritmo verifica a dimensão da altura e da largura e a quantidade de caixas que serão empacotadas.

- ✓ Se a quantidade de caixas for suficiente para preencher mais do que uma coluna inteira utilizando quaisquer dos dois posicionamentos possíveis (trocando a altura com a largura), então define-se o posicionamento das caixas procurando minimizar a perda na altura;
- ✓ Caso contrário, o algoritmo escolhe a maior dimensão como sendo a largura a fim de facilitar o posterior carregamento das caixas.

Definido o posicionamento da caixa, verifica-se a possibilidade de empacotar uma coluna inteira com esse tipo de caixa.

- ✓ Se não for possível o empacotamento de uma coluna inteira, carrega-se toda a quantidade disponível da caixa;
- ✓ No caso da viabilidade do carregamento em mais de uma coluna, torna-se necessário o cálculo da Largura Flexível.

George e Robinson introduziram o conceito de Largura Flexível a fim de combinar espaços não ocupados entre camadas para aumentar a utilização do mesmo. Na figura 19, após o carregamento de **1** cria-se o espaço **BCGF** como espaço de carregamento na largura do contêiner (W). Porém, verificamos que é possível unir o espaço vazio da camada anterior **ABFE** com o espaço da camada atual formando então uma região de carregamento **ACGE**. Ao criar essa nova região de carregamento, percebemos que o espaço **IDEH** ficaria vazio. Para não ocorrer esse tipo de problema, a heurística calcula a Largura Flexível **AD**. Pela figura, a próxima região a ser carregada terá sua profundidade aumentada de **BC** para **AC.**

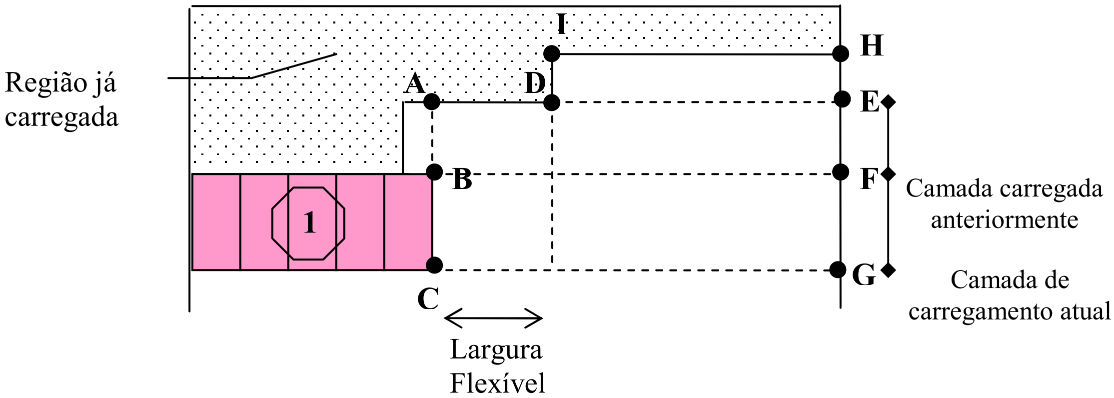

Figura 19 – Largura Flexível

Calculada a Largura Flexível, o algoritmo verifica se o número (dimensão) de colunas que serão carregadas é maior que a Largura Flexível.

- ✓ Se for, então carrega-se uma coluna a mais do que a permitida pela Largura Flexível, conforme figura 19.
- ✓ Se não, empacota-se o maior número possível de colunas completas.

Após essa etapa, o algoritmo atualiza a quantidade de caixas e a classificação (“aberta” ou “fechada”).

O carregamento da região **ACGE** é feito até exceder em no máximo uma coluna a largura flexível **AD**. Agora, a heurística tentará aumentar a profundidade do carregamento aproveitando o espaço vago **IDEH.**

3ª etapa: Criar os novos espaços de carregamento.

Os novos espaços para carregamento são criados conforme a figura 20:

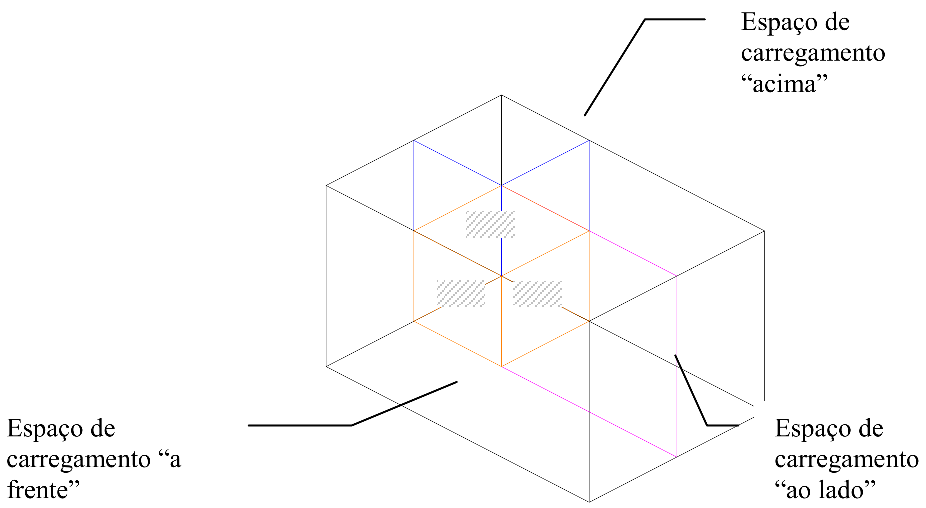

Figura 20 – Os novos espaços para carregamento

Primeiramente, gera-se os novos espaços de carregamento “a frente” e coloca-se esses espaços na “Lista de Espaços a serem Carregados”. A seguir, cria-se o espaço “ao lado” e verifica-se se a largura do espaço é maior ou igual a menor dimensão das caixas que ainda não foram empacotadas. Se a largura for maior ou igual, então adiciona-se esse espaço a “Lista de Espaços a serem Carregados”. Caso contrário, descarta-se esse espaço. O último espaço gerado é o “acima”. Gerado esse espaço, verifica-se se a altura dele é maior que a menor dimensão das caixas que ainda não foram empacotadas. Se a altura for maior, adiciona-se esse espaço à “Lista de Espaços a serem Carregados”. Caso contrário, descarta-se o espaço.

Independente da dimensão do espaço gerado “a frente”, esse nunca é descartado uma vez que o algoritmo tenta unir esses espaços através do cálculo da Largura Flexível.

Os espaços gerados são armazenados na “Lista de Espaços a serem Carregados” de forma que, carrega-se primeiro o espaço gerado “acima”, seguido do “ao lado” e finalmente do “a frente”.

4ª etapa: Escolher o novo espaço de carregamento.

Nessa etapa, o algoritmo verifica se existe alguma caixa que ainda não foi empacotada e se existir espaços não carregados.

- ✓ Verifica-se a existência de alguma caixa na “Lista de Caixas não Empacotadas”;

- Se existir, então verifica-se a existência de espaços a serem carregados na “Lista de Espaços a serem Carregados” e escolhe-se o espaço que estiver no topo da lista.
  - Verifica-se a viabilidade de unir o espaço atual de carregamento com algum espaço da “Lista de Espaços Rejeitados” definido na 5ª etapa da heurística. Para unir os espaços é necessário que a altura e a largura do espaço da “Lista de Espaços a serem Carregados” sejam menor ou igual a altura e a largura do espaço rejeitado, respectivamente.
    - Se for viável, une-se os espaços e verifica-se novamente a possibilidade de unir a esse espaço criado um outro espaço da “Lista de Espaços Rejeitados”;
    - Se não for viável unir os espaços, calcula-se a Largura Flexível.
- Se não existir significa que todas as caixas foram empacotadas.

Se o espaço a ser carregado for considerado uma nova camada o algoritmo volta para a 1ª etapa. Caso contrário, vai para a 5ª etapa.

<u>5ª etapa</u>: Achar o tipo de caixa para preencher os espaços vazios já existentes nas camadas.

Os novos espaços de carregamento possuem suas dimensões (profundidade, largura e altura) definidas. Para o carregamento desses espaços, o algoritmo verifica se algum tipo de caixa cabe no espaço.

- Se couber, verifica-se se algum tipo de caixa preenche mais de uma coluna.
  - Se sim, então escolhe-se a caixa que melhor preencha a profundidade e a heurística volta para a 2ª etapa.
  - Caso contrário, escolhe-se o tipo de caixa que melhor preencha a área da base e a heurística volta para a 2ª etapa.

- Se o algoritmo verificar que nenhuma caixa cabe no espaço vazio, ele armazena o espaço na "Lista de Espaços Rejeitados". A heurística volta para a 4ª etapa.

Cecilio e Morabito (2003) desenvolveram novas heurísticas para o problema carregamento de contêineres baseadas em refinamento e extensões da heurística de George e Robinson.

O refinamento proposto consiste em, na 5ª etapa da heurística de George e Robinson descrita anteriormente, ao invés de escolhermos o tipo de caixa que preenche a maior área da base do espaço, verificar se pode ser feita uma combinação de caixas de mesmo tamanho no comprimento desse espaço. A combinação de caixas que apresentar uma maior utilização da área da base do espaço é escolhida.

As duas extensões da heurística desenvolvidas foram denominadas de versão Arranjo e versão Camada.

Para essas novas extensões da heurística, foram acrescentados mais dois critérios para o ranqueamento das caixas, além dos critérios estabelecidos por George e Robinson.

Agora, os cincos critérios utilizados para o ranqueamento das caixas são:

- ✓ caixa com a maior entre as menores dimensões;
- ✓ caixa de maior quantidade;
- ✓ caixa com maior dimensão;
- ✓ caixa com maior volume;
- ✓ caixa com a maior razão dada por (maior dimensão/menor dimensão).

Vale ressaltar que o ranqueamento das caixas é utilizado para a escolha da primeira caixa da camada.

<u>VERSÃO ARRANJO</u>

Nessa versão são utilizados cada um dos arranjos de três dos cinco critérios descritos anteriormente, o que resulta em 60[H] possibilidades de

---

[H] $A_{m,n} = m! / (m - n)!$

$A_{5,3} = 5! / 2! = 120/2 = 60$

combinação dos critérios. Isso significa que a heurística de George e Robinson será executada 60 vezes sendo que em cada vez, um arranjo diferente de três critérios de ranqueamento das caixas será utilizado e um novo padrão de empacotamento do contêiner é gerado. O padrão que obtiver maior volume empacotado é o escolhido. O fluxograma da figura 21 resume a Versão Arranjo.

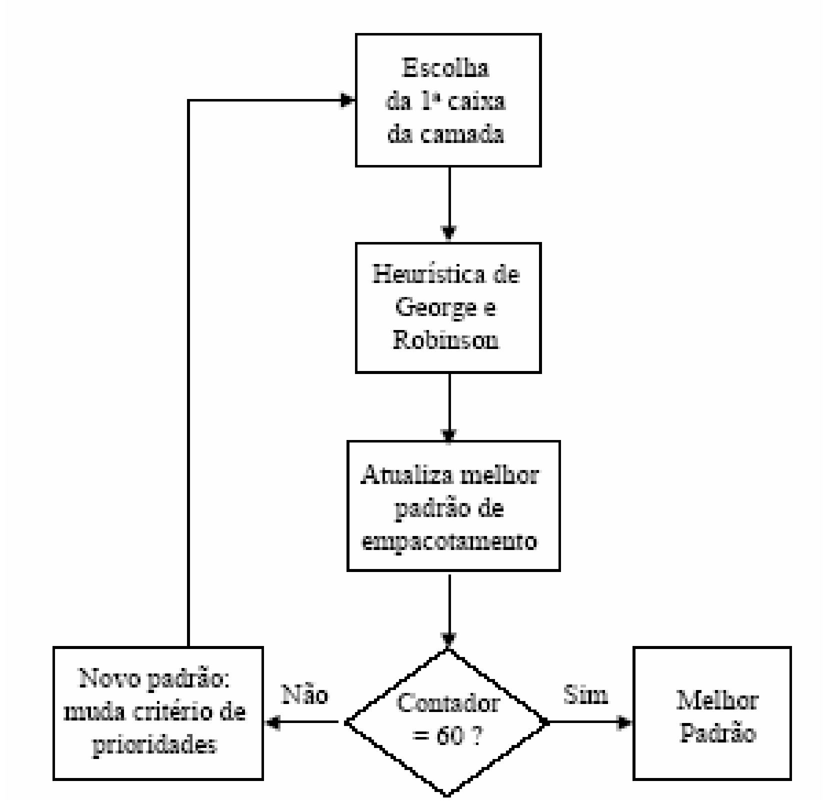

Figura 21 – Versão Arranjo da Heurística de George e Robinson
Fonte: Cecílio e Morabito (2003)

VERSÃO CAMADA

Nessa versão também são utilizados arranjo de três dos cinco critérios descritos anteriormente, o que resulta em 60 possibilidades de combinação dos critérios. Nessa versão será determinado o padrão de empacotamento de cada camada do contêiner. Isso significa que serão geradas 60 possibilidades de carregamento para cada camada do contêiner. A iteração que apresentar menos espaço vazio é escolhida como padrão para a camada. O fluxograma da figura 22 resume a Versão Camada.

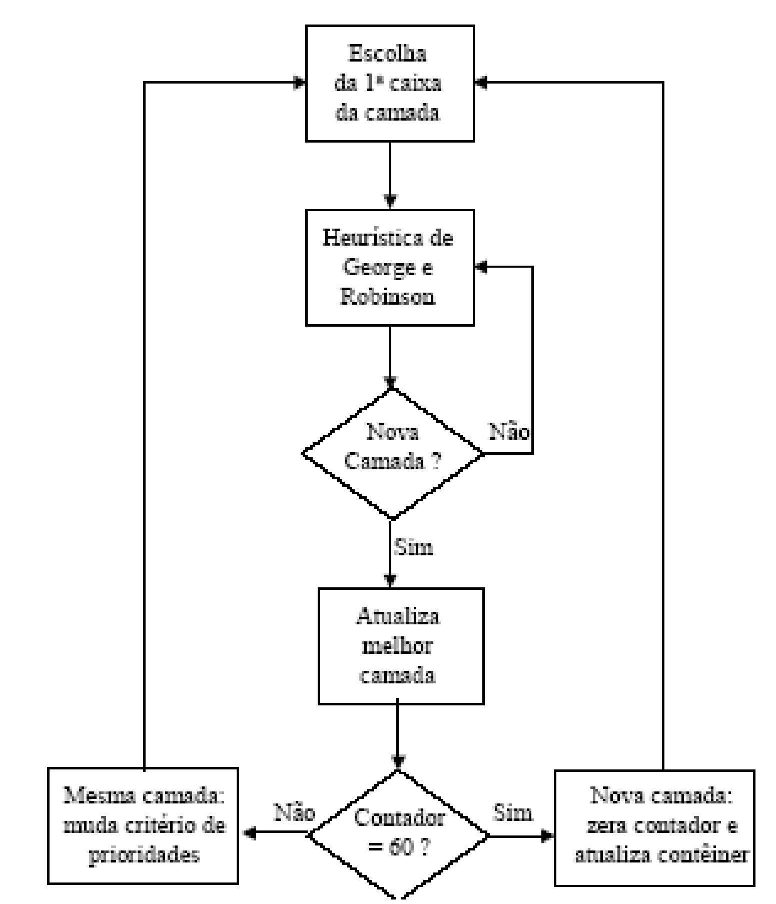

Figura 22 – Versão Camada da Heurística de George e Robinson
Fonte: Cecílio e Morabito (2003)

Com base no refinamento e nas duas novas extensões da heurística de George e Robinson, Cecílio e Morabito definiram 5 métodos de solução para o problema de carregamento de contêineres. São eles:

- ✓ Heurística de George e Robinson original + refinamento proposto;
- ✓ Heurística de George e Robinson original + versão arranjo;
- ✓ Heurística de George e Robinson original + refinamento proposto + versão arranjo;
- ✓ Heurística de George e Robinson original + versão camada;
- ✓ Heurística de George e Robinson original + versão camada + refinamento.

3.3.
Carregamento de contêineres em navio

Antes dos artigos referentes ao carregamento de contêineres em navios, serão introduzidos alguns termos técnicos importantes nesse tipo de transporte.

Os navios porta-contêineres são navios utilizados exclusivamente para o transporte de contêineres que podem ser alocados tanto nos porões dos navios quanto no convés.

Os contêineres, normalmente, seguem o padrão internacional estabelecido pela *International Standards Organization* (ISO). Eles possuem altura e largura de 8 pés e os comprimentos mais utilizados são os de 20 e 40 pés. Os contêineres de dimensão 20' x 8' x 8' são chamados de unidade padrão e adotado internacionalmente como *Twenty Feet Equivalent Unit (TEU)* ou unidade equivalente a 20 pés.

Os navios são constituídos por porões que possuem em geral, 40 pés de comprimento sendo subdividido em duas seções de 20 pés cada. Cada seção é denominada *BAY* de porão. No convés, os contêineres são armazenados sobre as tampas de escotilha de cada porão formando as *BAYS* de convés.

Pela figura 23, podemos verificar que as *BAYS* são formadas por *STACKS* (colunas) e *TIERS* (camadas), sendo que a coordenada dada pela *BAY, STACK* e *TIER* forma o chamado *SLOT/CELL*, ou seja, uma posição de contêiner. Cada *BAY* de porão ou de convés tem diversas colunas *(stack)* de 8 pés de largura. A altura das colunas é limitada pela resistência das tampas da escotilha no caso de contêineres armazenados no convés do navio e pela resistência do fundo da embarcação no caso de contêineres armazenados no porão do navio. As *BAYS* se estendem de bombordo (metade esquerda do navio olhando para a Proa) a boreste (metade direita do navio olhando para a Proa). Em geral, são construídas para a colocação de contêineres de 20' e 40', que são embarcados longitudinalmente.

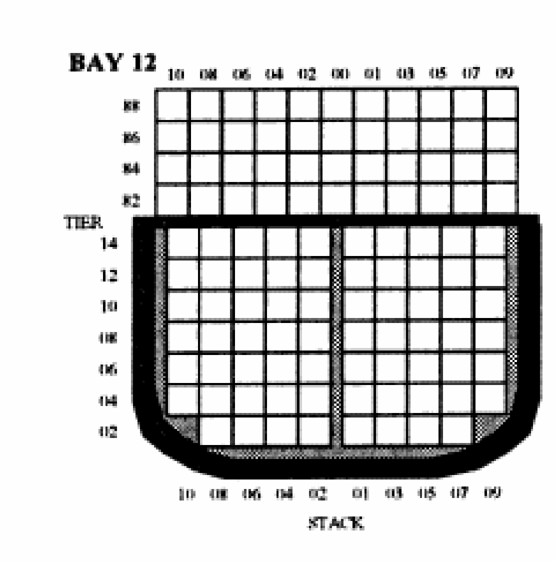

Figura 23 – Composição das *BAYS*
Fonte: Wilson *et al* (2001)

A figura 24 mostra a numeração das *BAYS* a partir da Proa (parte dianteira) para a Popa (parte traseira) em números ímpares (1, 3, 5, 7, 9, ...) que corresponde as *BAYS* ocupadas por contêiner de 20’. Quando a *BAY* for ocupada por contêiner de 40’, ela recebe uma numeração par que equivale a numeração de 2 *BAYS* ímpares. Por exemplo, se for colocado um contêiner de 40’ na *BAY* 22, entende-se que as *BAYS* que estão sendo ocupadas são as de número 21 e 23.

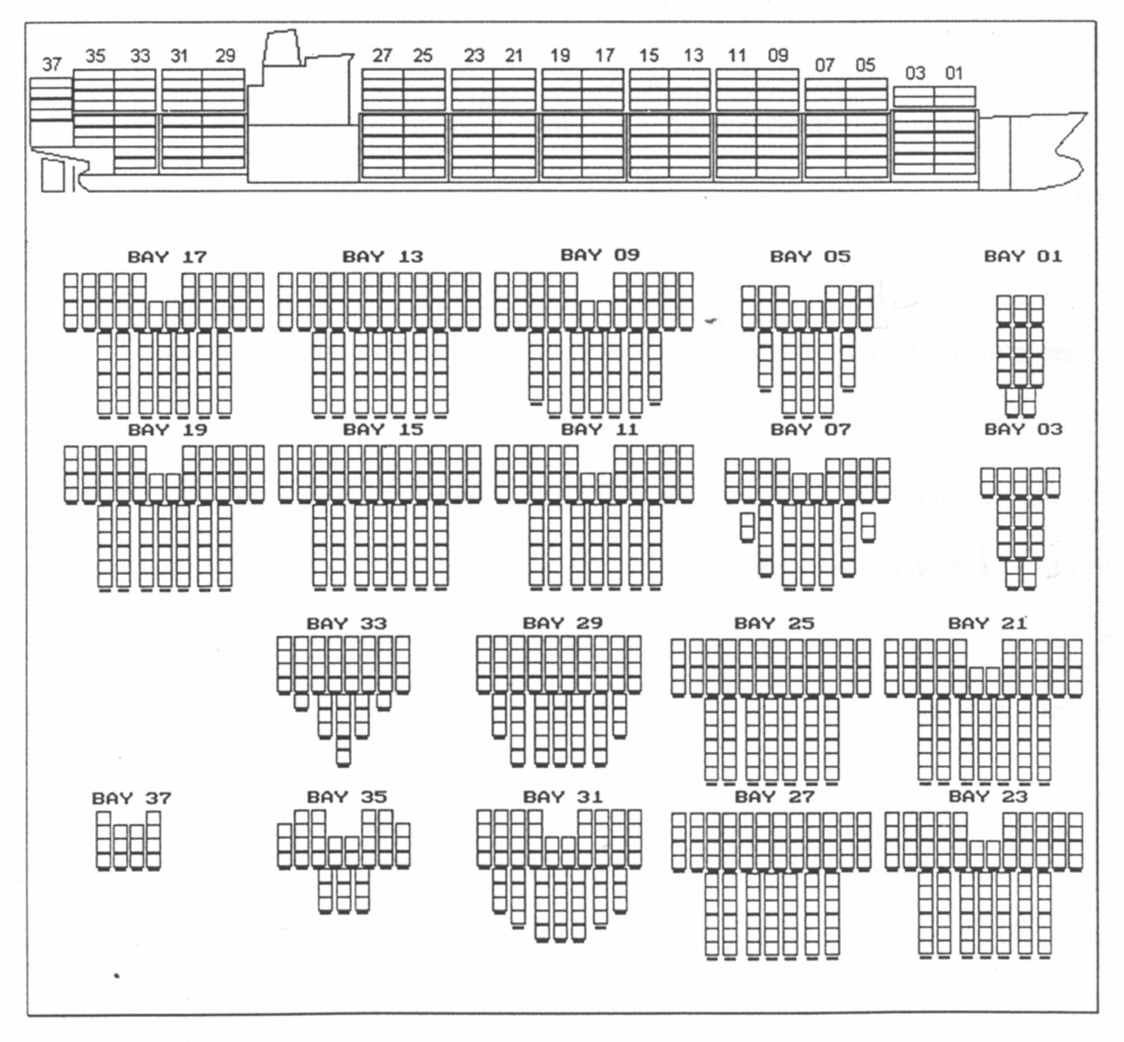

Figura 24 – Arranjo das *bays* do navio Columbus Olivos
Fonte: Hino (1999)

Outros termos são explicados a partir da figura 25.

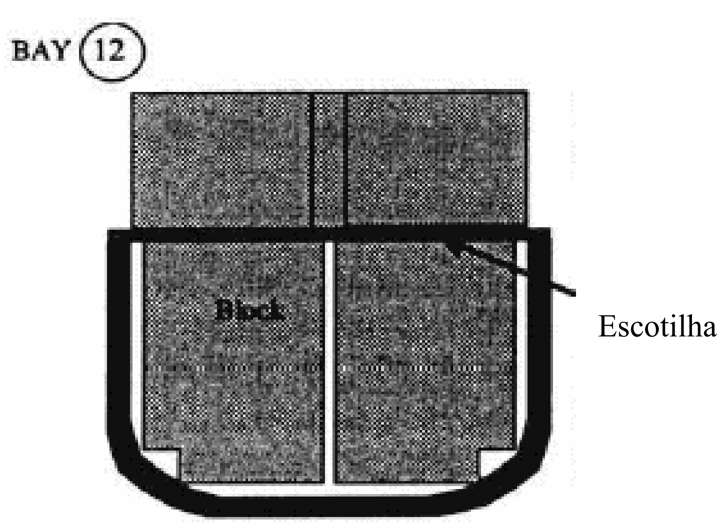

Figura 25 – Formação de Blocos de carga
Fonte: Wilson et al (2001)

Hatch – escotilha, cobertura dos porões do navio;

Hatch-lid – são as tampas das escotilhas;

Cargos spaces – espaços para alocação de carga;

Blocks – blocos de carga.

Wilson *et al* (2001) introduziram um sistema de computação que gera soluções sub-ótimas para o Problema de Planejamento de Estivagem em navios Porta- contêineres.

O Problema de Estivagem de Contêiner consiste em determinar um arranjo viável dos mesmos, de forma a facilitar as operações de carga e descarga, a um custo reduzido. O fator mais importante na otimização desse processo é minimizar o número de reestivagens pois essa atividade representa um aumento das despesas portuárias.

Esse problema é considerado combinatorial cujo tamanho depende da capacidade do navio, do abastecimento de contêiner e da demanda de cada POD (porto de destino). Este problema torna-se mais complexo pois precisamos considerar a estivagem em todos os POD e as decisões tomadas em um porto trarão conseqüências aos portos subsequentes.

Em cada POD, os contêineres com esse destino são descarregados e novos contêineres podem ser carregados para portos subsequentes.

Os autores consideram que:

- ✓ Em cada POD, ocorrem operações de carga e descarga porém o carregamento não começa até o total descarregamento dos contêineres daquele POD;
- ✓ Em cada POD, existem 2 guindastes disponíveis para as operações de carga e descarga.

Devido a complexidade computacional, o Processo de Planejamento é dividido em duas etapas:

1. Processo de Planejamento estratégico:
   - nesta etapa, os contêineres são designados a um determinado *blocked cargo-space*.

2. Processo de Planejamento tático
   - nesta etapa, os contêineres são designados aos *slots* referentes aos repectivos *blocked cargo-space.*

1. Processo de Planejamento Estratégico:

Os contêineres, classificados por classe (comprimento e POD), são alocados aos "Blocos de espaço de Carga" onde os *slots* correspondentes a cada tampa da escotilha são mantidos juntos. Os "Blocos de espaço de Carga" são ilustrados na figura 26.

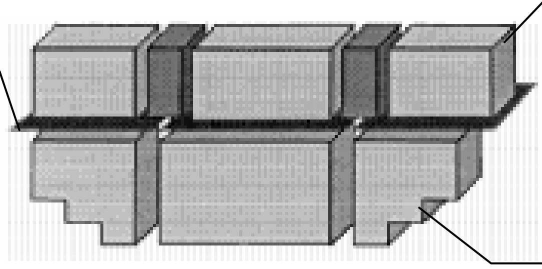

Figura 26 – Blocos de Espaço de Carga
Fonte: Wilson *et al* (2001)

Os objetivos dessa fase são:

- ✓ Minimizar o número espaços de carga ocupado por cada destino;
- ✓ Maximizar o número de guindastes utilizadas nas operações de carregamento em cada POD;
- ✓ Minimizar o número de movimentos das tampas das escotilhas;
- ✓ Minimizar o número de reestivagem;
- ✓ Minimizar o número de *cargo blocks* ocupado por contêineres.

2. Processo de Planejamento Tático:

Nessa fase, os contêineres são alocados aos *slots* aos respectivos "Blocos de espaço de carga" definidos na etapa anterior.

Os objetivos dessa fase são:

- ✓ Minimizar o número de reestivagem;

- ✓ Os contêineres devem ser armazenados do mais pesado para o mais leve;
- ✓ Minimizar o número de colunas de contêineres com POD variado.

A figura 27 ilustra a alocação de cada contêiner ao seu respectivo *slot* contendo contêineres para dois POD distintos (ROT e ILO) sendo ROT o porto mais próximo para a próxima carga e descarga de contêiner. Além disso, podemos observar que a escotilha é composta por duas tampas e que existem dois tipos de contêineres sendo carregados nessa *BAY*: o 2210 e o 4210.

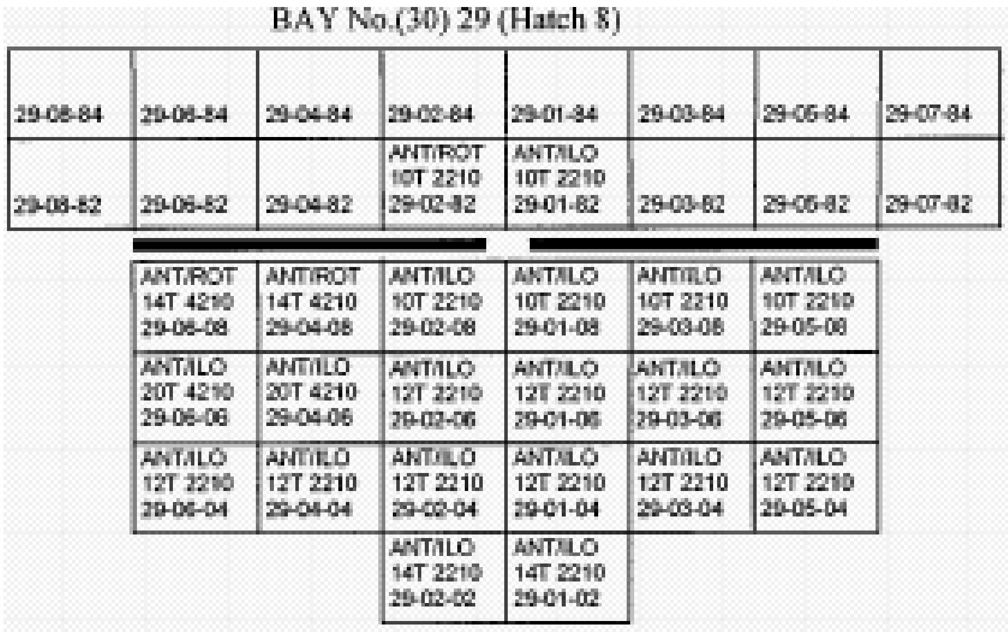

Figura 27 – Alocação de contêineres nos respectivos slots
Fonte: Wilson et al (2001)

Martin Jr. *et al* (1988) desenvolveram uma heurística para o Planejamento de Carregamento de Contêineres em navio.

Aparentemente, o processo de carregamento de contêiner em um navio é muito simples. Dado um grupo de contêineres e um grupo de posições disponíveis no navio para o carregamento destes contêineres, o problema consiste em alocar os contêineres no navio e determinar a seqüência de carregamento de forma que as restrições sejam respeitadas e o custo de movimentação dos contêineres seja minimizado.

Podemos citar como restrições para o processo de carregamento de contêiner:

- Estabilidade do navio;
- Exigências para o carregamento de cargas perigosas;
- Altura das colunas de carregamento;
- Comprimento do contêiner, entre outros.

Geralmente, no carregamento, os contêineres são movimentados do pátio de estocagem para o costado do navio por meio de caminhões. O contêiner é carregado no caminhão pelo transtainer[I]. No costado do navio, o contêiner é retirado do caminhão pelo guindaste do navio. O descarregamento é feito de maneira inversa.

O transtainer possue largura suficiente para "varrer" sete contêineres ao mesmo tempo porém, no terminal em estudo (Terminal do Porto de Portland), os contêineres são armazenadas em fileiras de seis contêineres, ficando o espaço do sétimo contêiner livre para o carregamento e passagem de caminhões. A altura de empilhamento do terminal em estudo é de 4 contêineres.

As fileiras de contêineres, no pátio de estocagem, são separadas por comprimento e sempre que possível, por porto de destino e peso. Várias fileiras de contêineres juntas formam seções sendo as seções separadas por corredores.

Para desenvolvimento da heurística, os autores utilizaram como base o Terminal do Porto de Portland (PoP) e, para simplificar o problema, foram feitas algumas considerações:

- Não foram consideradas as cargas perigosas e cargas com dimensões maiores que as dimensões padrão;
- Os contêineres são movimentados por uma combinação de transtainer, caminhão e guindaste do navio;
- Contêineres vazios não foram levados em consideração;
- Só serão carregados os navios mais novos (data de fabricação) por esses possuírem um sistema que ajusta a estabilidade do navio;

[I] Transteiners – guindastes montados sobre grandes pórticos que se movimentam sobre trilhos ou pneus, empilhando e transportando os contêineres de um ponto a outro.

- Os contêineres para a viagem já estão no pátio de estocagem e estão agrupados por porto, comprimento e peso.

O primeiro passo para resolução do problema é decidir a ordem na qual as *cells* das *bays* serão carregadas. Para isso são necessárias as seguintes informações: a ordem na qual os portos serão atendidos, número médio de contêineres classificados como leve, médio e pesado para cada combinação de porto e *bay* e o guindaste a ser usado para cada combinação de porto e *bay* quando mais de uma for utilizado.

Conforme ilustrado na figura 28, o carregamento da *bay* é feito preenchendo as *cells* no sentido bombordo-boreste caso o navio esteja atracado a boreste e no sentido boreste-bombordo caso o navio esteja atracado a bombordo. As camadas (*tier)* são preenchidas de baixo para cima

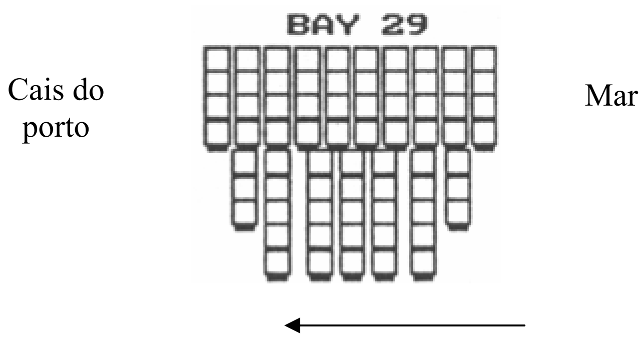

Figura 28 – Sentido de prenchimento das *cells* de uma determinada *bay*
Fonte: Wilson et al (2001)

O algoritmo inicia a procura por um contêiner que será alocado na *cell.*

Para o processo de carregamento nas bays, o peso dos contêineres das *stacks* (colunas) não pode superar um valor limite. Valor esse que, se não respeitado, afeta não só a estabilidade do navio como também a integridade do mesmo.

Dois métodos diferentes são utilizados para escolha do contêiner. O primeiro caso ou condição normal, o autor considera que o peso do contêiner não tem importância especial, ou seja, uma variação no peso afetará a estabilidade de uma forma bem branda. O outro caso é quando o último contêiner da coluna *(stack*) do convés for carregado. Neste caso, se a *stack* estiver próximo do limite de carregamento, o último contêiner pode causar uma instabilidade considerável.

O próximo passo é determinar a classe de peso (leve, médio ou pesado) do contêiner que irá preencher a *cell* além de definir o porto de destino, comprimento do contêiner e algumas vezes, altura. A definição da classe de peso do contêiner é feita pela determinação de uma variação do peso, como por exemplo: o contêiner com peso entre X ± y é classificado como leve.

O algoritmo inicia a procura por um contêiner no local onde o *transtainer* se encontra (*fileira atual*). As seguintes situações podem ocorrer:

- Se apenas um contêiner na *fileira atual* for aceito, ele é alocado na *cell;*
- Se mais de um contêiner for aceito, o algoritmo escolhe o contêiner que minimiza o re-manuseio. Desses, ele escolhe o que estiver mais próximo da pista de passagem do caminhão (*truck line*).;
- Se nenhum contêiner for aceito, o algoritmo realiza a procura em outra fileira da seção atual.

A procura não é feita em todas as *fileiras* da *seção atual*. É feita somente nas *fileiras* onde a média de peso dos contêineres é a mesma requerida pela *cell* a ser preenchida. Se nenhum contêiner for encontrado na *seção atual*, a procura é feita na seção adjacente e depois nas não adjacentes. Se nenhum contêiner para uma determinada *cell* for encontrado nas *seções,* o algoritmo muda a variação de peso que determina a classificação do contêiner em leve, médio ou pesado e volta a procura para a posição inicial do transtainer. Com essa mudança na variação de peso, poderão ser visitadas nessa nova busca fileiras que não foram "visitadas" anteriormente.

Se após a mudança na variação de peso o algoritmo ainda não aceitar nenhum contêiner para aquela determinada cell, o algoritmo muda para a próxima *cell* a ser carregada e recomeça a procura.

O outro método desenvolvido pelo autor para escolha de contêiner é quando a *cell* a ser carregada é a última da coluna (*stack*) do convés.

Para preenchimento dessa *cell* será escolhido o contêiner mais pesado que não ultrapassar o limite de resistência da escotilha. Para aumentar a flexibilidade na busca pelo contêiner, são considerados aptos a preencher a *cell* os contêineres cujo peso estiver dentro de uma variação de peso especificada em relação ao contêiner mais pesado.

Dentre esses contêineres, o que minimizar re-manuseio e/ou estiver mais próximo da passagem do caminhão (*truck line*) é escolhido para carregamento.

A busca é feita de seção por seção. A ordem de busca pelas seções é a mesma do outro procedimento: primeiro na *current section*, depois nas seções adjacentes e por último nas seções não-adjacentes. Se nenhum contêiner for encontrado para a *cell*, o algoritmo começa a procura para a próxima *cell* a ser carregada.

Uma busca cuidadosa na literatura, porém não exaustiva, mostra que a bibliografia neste assunto é relativamente escassa. No próximo capítulo será apresentada a heurística desenvolvida para a elaboração de Plano de Estivagem de placas de aço em navio.