

UMA HEURÍSTICA DE BUSCA TABU SIMPLES PARA O PROBLEMA DE CARREGAMENTO DE PALETES DO PRODUTOR

Vitória Pureza

Reinaldo Morabito *

Departamento de Engenharia de Produção

Universidade Federal de São Carlos

São Carlos – SP

morabito@power.ufscar.br

* *Corresponding author*/autor para quem as correspondências devem ser encaminhadas

Recebido em 02/2003, aceito em 06/2003

Resumo

O problema de carregamento de paletes do produtor consiste em arranjar, ortogonalmente e sem sobreposição, o máximo número de caixas de dimensões (l,w) ou (w,l) , sobre um palete retangular de dimensões (L,W) . Este problema vem sendo tratado com sucesso por heurísticas de blocos, onde são gerados padrões de carregamento compostos por um ou mais blocos, cujas peças possuem a mesma orientação. Uma característica desses métodos é que os padrões gerados estão limitados aos chamados padrões não-guilhotinados de primeira ordem. Neste trabalho é proposta uma abordagem baseada na incorporação de busca tabu simples (sem estruturas de memória de longo prazo) em heurísticas de bloco. Partindo-se de uma solução inicial, são realizados movimentos de aumento de blocos, que resultam na diminuição, eliminação e criação de outros blocos. Resultados computacionais indicam que esta abordagem é capaz de gerar padrões ótimos de ordem superior em problemas de difícil resolução para outros métodos da literatura.

Palavras-chave: problema de carregamento de paletes do produtor; problemas de corte e empacotamento; busca tabu simples.

Abstract

The manufacturer's pallet loading problem consists in arranging, orthogonally and without overlapping, the maximum number of boxes with dimensions (l,w) or (w,l) onto a rectangular pallet with dimensions (L,W) . This problem has been handled successfully by block heuristics, which generate loading patterns composed by one or more blocks where the pieces have the same orientation. A feature of such methods is that the patterns are limited to the so called 1st order non-guillotine patterns. In this work we propose an approach based on the incorporation of simple tabu search (without longer term memory structures) in block heuristics. Starting from an initial pattern, the algorithm performs moves that increase the size of selected blocks in the current pattern; as a result, other blocks are decreased, eliminated or created. Computational results indicate that the approach is capable of generating superior order optimal patterns for difficult instances reported in the literature.

Keywords: manufacturer's pallet loading problem; cutting and packing; simple tabu search.

1. Introdução

O problema de carregamento de paletes (*pallet loading problem*) consiste em carregar produtos (embalados em caixas) sobre um palete retangular, de maneira a otimizar o aproveitamento do mesmo. Supõe-se que as caixas, disponíveis em grandes quantidades, devam ser arranjadas ortogonalmente, isto é, devem ter seus lados posicionados paralelamente aos lados do palete. O problema de carregamento de paletes aparece com frequência nas atividades de armazenagem, movimentação e transporte de produtos. Devido à escala de certos sistemas logísticos, um pequeno aumento do número de produtos carregados sobre cada palete pode resultar numa economia global significativa.

Hodgson (1982), ao estudar o problema, dividiu-o em dois possíveis casos: o problema de carregamento de paletes do produtor (*manufacturer's pallet loading problem*) e o problema de carregamento de paletes do distribuidor (*distributor's pallet loading problem*). No primeiro 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 tolerável do carregamento). Em cada camada, uma orientação vertical das caixas é fixada. Se nenhuma orientação for fixada (isto é, as caixas podem ser carregadas sob qualquer uma de suas faces: lw , lh e wh), tais restrições permitem 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 (i.e., faces das caixas) dentro do retângulo (L, W) (superfície do palete), e (ii) o problema unidimensional de arranjar o máximo número de camadas ao longo da altura H do palete. Note que este último é um problema da mochila (*knapsack problem*) com capacidade H .

No segundo caso, o distribuidor recebe vários bens de diversos fornecedores, embalados em caixas de diferentes dimensões (l_i, w_i, h_i) , $i=1, \dots, m$. Essas caixas são então carregadas sobre paletes. Note que o problema do distribuidor pode ser visto como um problema de carregamento de contêineres (Bischoff & Ratcliff, 1995; Morabito & Arenales, 1994). Em particular, se uma orientação vertical for fixada, por exemplo, se as caixas devem ser carregadas sob a face $l_i w_i$, então o problema do distribuidor reduz-se ao problema bidimensional de encontrar o arranjo ortogonal de retângulos (l_i, w_i) e (w_i, l_i) , $i=1, \dots, m$, dentro do retângulo (L, W) que maximize a utilização de (L, W) . Revisões dos problemas do produtor e distribuidor podem também ser encontrados nos artigos e edições especiais de Dowsland & Dowsland (1992), Sweeney & Parternoster (1992), Dyckhoff & Finke (1992), Balasubramanian (1992), Martello (1994), Bischoff & Waescher (1995), Dyckhoff *et al.* (1997), Arenales *et al.* (1999), Wang & Waescher (2002), Hifi (2002) e Martins (2002).

No presente estudo abordamos o problema do produtor, em particular, o problema bidimensional descrito em (i) de arranjar, sem sobreposição, retângulos (l, w) e (w, l) no retângulo (L, W) . Este problema também aparece no arranjo físico de paletes sobre carrocerias de caminhões e no projeto de embalagens para formar unidades de carga intermediárias (Morabito, Morales & Widmer, 2000). É importante salientar que, embora existam algoritmos polinomiais para a versão guilhotinada do problema do produtor (Tarnowski *et al.*, 1994), o caso mais geral parece ser de difícil resolução do ponto de vista da teoria de complexidade, embora isso ainda não tenha sido provado (Dowsland, 1987; Nelissen, 1995; Letchford & Amaral, 2001).

O problema do produtor pode ser modelado como um caso particular do modelo de programação linear 0-1 em Beasley (1985) proposto para o problema de corte não-guilhotinado. Outros modelos 0-1 para o problema do produtor podem ser encontrados em

Tsai *et al.* (1993) e Hadjiconstantinou & Christofides (1995) (recentemente Amaral & Letchford (2001) mostraram que esta última formulação não é válida em certos casos). Além dos métodos propostos por estes autores, outros métodos exatos para resolver o problema do produtor são reportados em Dowsland (1987) e Bhattacharya *et al.* (1998). Entretanto, todas estas abordagens são em geral computacionalmente intratáveis nas situações práticas. Mais recentemente, Lins *et al.* (2003) apresentaram uma abordagem capaz de resolver problemas considerados de difícil resolução; os autores conjecturam que tal abordagem tenha garantia de otimalidade. Pesquisas com limitantes superiores para o problema também podem ser encontradas em Nelissen (1995), Letchford & Amaral (2001) e Morabito & Farago (2002).

Diversos autores têm tratado com relativo sucesso o problema do produtor com heurísticas de construção conhecidas como *heurísticas de blocos*, onde são gerados padrões de carregamento compostos por um ou mais blocos, cujas caixas possuem a mesma orientação. Exemplos de heurísticas de blocos são reportados em Steudel (1979), Smith & De Cani (1980), Bischoff & Dowsland (1982), Nelissen (1994, 1995), Scheithauer & Terno (1996) e Morabito & Morales (1998, 1999). Uma característica desses métodos é que os padrões gerados estão limitados aos chamados *padrões não-guilhotinados de primeira ordem* (definidos na próxima seção). Parte da motivação do presente trabalho é propor uma heurística capaz de resolver problemas desta classe, não resolvidos pelas heurísticas de bloco da literatura, em tempo computacional relativamente pequeno.

Meta-heurísticas são técnicas que, quando aplicadas a métodos de busca local, permitem a superação da otimalidade local com vistas à obtenção de soluções de qualidade superior. Apesar do sucesso que estas técnicas têm alcançado em uma variedade de problemas, no que diz respeito ao problema do produtor, poucas aplicações são reportadas na literatura. Exemplos dessas aplicações são reportados em Dowsland (1993) (*simulated annealing*), Herbert & Dowsland (1996) (algoritmos genéticos), Dowsland (1996) e Yamasaki & Pureza (2001) (busca tabu). Em particular, os dois últimos trabalhos utilizam uma versão sem memória de busca tabu que parte de uma solução com o número ótimo de caixas e infactível com respeito à sobreposição, e realiza movimentos destas caixas para novas posições com vistas à sobreposição zero (padrão ótimo). Os resultados reportados indicam que estas heurísticas não têm bom desempenho em problemas de tamanho moderado (por exemplo, acima de 40 caixas).

Neste trabalho é apresentada uma abordagem baseada na incorporação de busca tabu simples a heurísticas de bloco. Ao contrário de versões avançadas que também empregam estruturas de memória de longo prazo para aplicação de estratégias de intensificação e diversificação, os elementos restritivos da busca tabu simples restringem-se a estruturas de memória de curto prazo. O papel dessas estruturas é de apenas armazenar e penalizar certas características ou atributos de soluções já visitadas, por um determinado número de iterações. Além do uso de memória, um outro aspecto que distingue a presente abordagem dos trabalhos de Dowsland (1996) e Yamasaki & Pureza (2001) é o fato da busca ocorrer no espaço factível de soluções, o que permite que um padrão esteja disponível em qualquer iteração. Além disso, movimentos ou transições entre soluções são aplicados a blocos ao invés de caixas como nos trabalhos anteriores. E, finalmente, ao contrário das heurísticas de bloco, a abordagem é capaz de tratar situações onde a solução ótima é um padrão não-guilhotinado de ordem superior.

Na próxima seção, definimos os padrões de carregamento e discutimos resumidamente heurísticas de bloco. Na seção 3, descrevemos a abordagem proposta e na seção 4, os experimentos computacionais e os resultados obtidos com a aplicação desta abordagem. As conclusões e próximos passos desta pesquisa são discutidos na seção 5.

2. Heurísticas de bloco

Os padrões de carregamento podem ser classificados em *guilhotinado*, *não-guilhotinado de 1ª ordem*, e *não-guilhotinado de ordem superior* (Arenales & Morabito, 1995). Eles são definidos da seguinte maneira:

- i. padrão guilhotinado: um corte é chamado *guilhotinado* se, ao ser produzido sobre um retângulo, produz 2 novos retângulos. Um padrão obtido por sucessivos cortes guilhotinados é chamado padrão guilhotinado (Figura 1a).
- ii. padrão não-guilhotinado de 1ª ordem: um corte é chamado *não-guilhotinado de 1ª ordem* se, ao ser produzido sobre um retângulo, produz 5 novos retângulos arranjados de maneira a não formar um padrão guilhotinado (veja corte em negrito da Figura 1b). Um padrão obtido por sucessivos cortes guilhotinados e/ou cortes não-guilhotinados de 1ª ordem é chamado padrão não-guilhotinado de 1ª ordem (Figura 1b). Note que todo padrão guilhotinado também é um padrão não-guilhotinado de 1ª ordem (mas o inverso não é verdadeiro).
- iii. padrão não-guilhotinado de ordem superior: é um padrão que não pode ser obtido por sucessivos cortes guilhotinados e/ou cortes não-guilhotinados de 1ª ordem (Figura 1c).

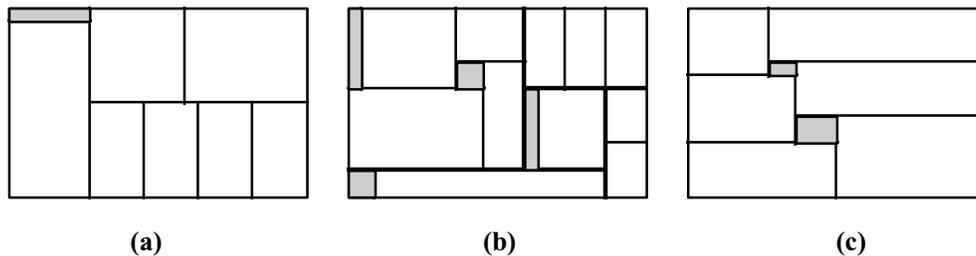


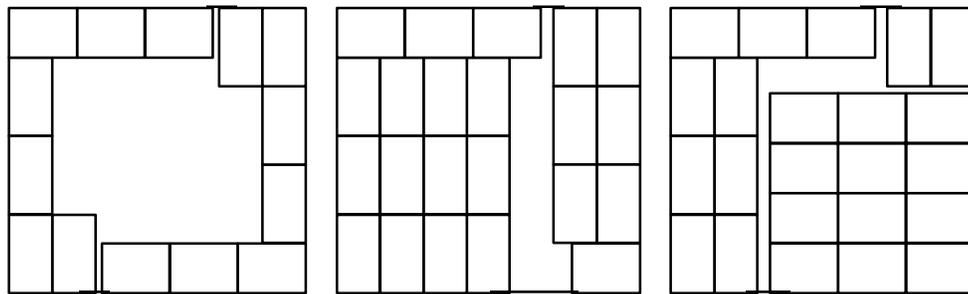
Figura 1 – (a) padrão guilhotinado, (b) corte (em negrito) e padrão não-guilhotinado de 1ª ordem, (c) padrão não-guilhotinado de ordem superior.

Dado que o problema do produtor não impõe restrições sobre o tipo do padrão, uma simplificação para abordá-lo seria impor artificialmente a restrição de padrões guilhotinados. Entretanto, esta alternativa em geral não resulta em boas soluções. Conforme mencionado na seção 1, devido às dificuldades com a aplicação de métodos exatos, diversas heurísticas têm sido propostas na literatura para resolver o problema do produtor, em especial as chamadas heurísticas de bloco (Nelissen, 1995). Cada bloco é definido como um retângulo contendo caixas arranjadas com a mesma orientação.

Um exemplo de uma heurística de bloco foi proposta em Steudel (1979). O palete é dividido em até 4 blocos e, em cada um deles, as caixas são simplesmente arranjadas conforme uma orientação pré-fixada. As dimensões destes blocos são determinadas num procedimento em dois estágios: no primeiro estágio a utilização do perímetro do palete é maximizada e no segundo, possíveis modificações nestes blocos são exploradas, variando suas dimensões, numa tentativa de preencher o bloco central. O padrão escolhido é aquele que resulta no maior número de caixas dentro dos blocos.

A Figura 2 apresenta padrões obtidos do exemplo de um palete $(L,W)=(48,40)$ com caixas $(l,w)=(11,7)$. A Figura 2a ilustra o padrão obtido no primeiro estágio e a Figura 2b, no segundo estágio. Note que 22 caixas foram arranjadas no palete. Se o padrão obtido no segundo estágio for composto por exatamente 4 blocos, então temos um padrão não-guilhotinado de 1ª ordem, conforme o ilustrado na Figura 2b (em particular, este padrão também é guilhotinado). Caso contrário, o padrão será apenas guilhotinado.

Outro exemplo de uma heurística de bloco, semelhante à anterior, foi proposta por Smith & De Cani (1980). O palete também é dividido em até 4 blocos, e mais uma vez as caixas são simplesmente arranjadas dentro dos blocos conforme uma orientação pré-fixada. O procedimento examina todas as possíveis modificações nestes blocos, variando suas dimensões (inclusive a possibilidade das dimensões serem nulas), e retorna o padrão que resultar no maior número de caixas dentro dos blocos. A Figura 2c apresenta o padrão obtido ao aplicar este procedimento para o exemplo anterior. Este padrão (com 23 caixas) é melhor do que o padrão obtido com a heurística anterior, com apenas 22 caixas. Note que os padrões das Figuras 2b e 2c são não-guilhotinados de 1ª ordem com 4 blocos, porém apenas o padrão da Figura 2b é também guilhotinado.



(a) Primeiro estágio de Steudel (b) Segundo estágio de Steudel (c) Smith e De Cani

Figura 2 – Padrões não-guilhotinados de 1ª ordem com 4 blocos para palete $(L,W)=(48, 40)$ e caixas $(l,w)=(11, 7)$.

Heurísticas de bloco baseadas na divisão do palete em cinco ou mais blocos podem ser encontradas em Bischoff & Dowsland (1982), Nelissen (1994), Scheithauer & Terno (1996) e Morabito & Morales (1998). Esta última gera padrões ótimos em problemas não-guilhotinados de 1ª ordem. Heurísticas recentes baseadas em outras técnicas são encontradas em G & Kang (2001) e Lins *et al.* (2002). Na grande maioria dos problemas, estas heurísticas encontram o padrão ótimo; nos poucos problemas em que não encontram (considerados difíceis), geram padrões de boa qualidade, em geral, a apenas uma ou duas caixas do número ótimo. Portanto, o desafio ao se propor um novo método é que este deve apresentar um desempenho superior ao resolver os problemas difíceis, em particular, os problemas cujo padrão ótimo é de ordem superior.

3. Abordagem proposta

A idéia básica da abordagem é de, partindo de uma solução gerada por uma heurística de blocos arbitrária, realizar iterativamente movimentos de aumento (ou expansão) de blocos selecionados (*blocos ativos*) em uma determinada direção (para cima, para baixo, para esquerda ou para direita). A cada movimento de expansão, apenas padrões factíveis são gerados.

Como o padrão resultante da aplicação da heurística de blocos em geral é bastante compactado (i.e., a superfície do palete é bem utilizada), o aumento de um dado bloco ativo quase sempre provoca alterações em outros blocos (*blocos passivos*), a fim de preservar a factibilidade das soluções. A estrutura de vizinhança aqui utilizada prescreve que blocos passivos possam ser diminuídos, eliminados, divididos em um ou mais blocos, ou terem suas orientações originais alteradas. Além do impacto em blocos passivos, um dado movimento de expansão pode também provocar cisões do bloco ativo em dois ou três blocos, conforme discutido a seguir. Desta forma, os padrões resultantes destes movimentos são suficientemente genéricos, incluindo padrões não-guilhotinados de ordem superior não gerados por outras heurísticas de bloco.

3.1 Estrutura de Vizinhança

Para qualquer bloco de um padrão, é possível realizar expansões para direita, para esquerda, para cima ou para baixo. A especificação da direção define a face de expansão do bloco, enquanto as demais faces permanecem inalteradas.

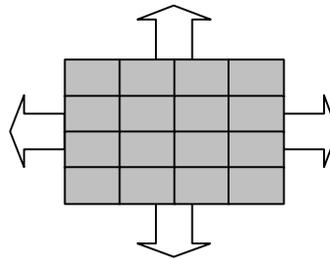


Figura 3 – Direções de expansão de um bloco do padrão (esquerda, direita, acima, abaixo).

A Figura 3 ilustra um bloco com 4 camadas verticais e 4 camadas horizontais de caixas. Uma vez especificada a face de expansão do bloco, ele pode crescer com a inclusão de caixas em uma ou mais camadas (verticais ou horizontais) adjacentes. Um caso especial é o crescimento de todas as camadas de uma face. No bloco da Figura 4, por exemplo, as possíveis camadas adjacentes em movimentos de expansão para a direita são {1}, {2}, {3}, {4}, {1,2}, {2,3}, {3,4}, {1,2,3}, {2,3,4} e {1,2,3,4} (note que {1,3}, {1,4} e {2,4} não são permitidos, uma vez que não são adjacentes). Uma vez definidas as camadas da face de expansão, o crescimento do bloco é feito adicionando-se um número de caixas a estas camadas, tal que não exceda os limites do palete. A Figura 4 ilustra dois possíveis movimentos de expansão para a direita.

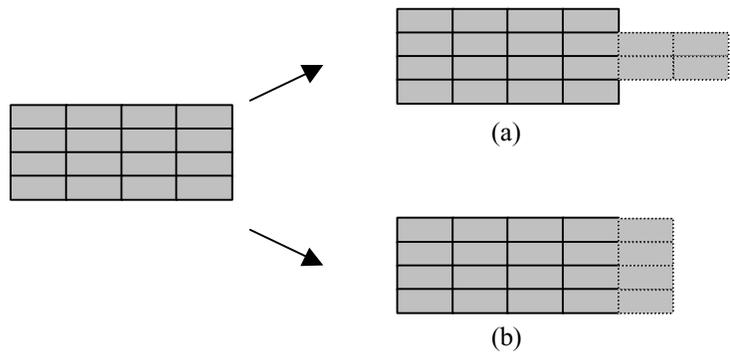


Figura 4 – Expansão do bloco para a direita com (a) adição de 2 caixas nas camadas {2, 3}; (b) adição de 1 caixa nas camadas {1,2,3,4}.

Como blocos são essencialmente estruturas retangulares, na expansão da Figura 4a, o bloco ativo original deve sofrer cisões, no caso, resultando em 3 blocos (camadas {1}, {2,3} e {4}). Na expansão da Figura 4b, o bloco ativo mantém sua integridade, simplesmente agregando uma nova coluna de caixas em sua face direita.

Uma situação especial ocorre quando a heurística de blocos gera um padrão com um único bloco (solução homogênea). Rigorosamente, a busca heurística não pode ser aplicada porque não há movimentos disponíveis para seleção. Na presente abordagem, caso não seja possível garantir a otimalidade do padrão gerado, o bloco é dividido em sub-blocos, e a partir daí prossegue-se com a busca heurística.

O crescimento do bloco ativo superpõe cada bloco passivo em maior ou menor extensão. Foram consideradas 4 circunscrições ou configurações possíveis para a área não superposta de cada bloco passivo. As configurações envolvem a partição desta área em subconjuntos de um total de 8 áreas retangulares possíveis (A, B, C, D, E, F, G e H), conforme ilustrado na Figura 5. Note que um bloco ativo pode superpor vários blocos passivos, e neste caso, cada bloco passivo é tratado separadamente conforme a Figura 5.

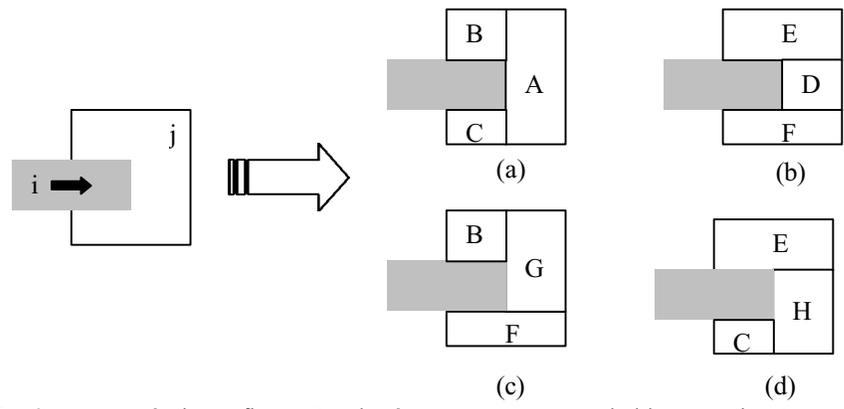


Figura 5 – Quatro possíveis configurações das áreas remanescentes do bloco passivo *j* com a expansão para a direita do bloco ativo *i*: (a) divisão nas áreas A, B e C; (b) divisão nas áreas E, D e F; (c) divisão nas áreas B, G e F; (d) divisão nas áreas E, H e C.

Para uma dada configuração e para cada área não superposta dos blocos passivos (que possam conter pelo menos uma caixa), é construído um novo bloco com orientação tal que maximize o número de caixas que nele podem estar contidas. Cada novo bloco é posicionado de tal forma que sua coordenada superior esquerda corresponde à coordenada superior esquerda da área onde ele foi criado (veja a Figura 10 adiante). A configuração selecionada para uma dada expansão de um bloco ativo é aquela que maximiza o valor da função de avaliação do movimento, definida na seção 3.4. A Figura 5 ilustra as 4 possíveis configurações resultantes de um movimento para direita do bloco ativo i (para as demais direções, rotacionar consecutivamente os desenhos em 90°), afetando o bloco passivo j . Caso haja uma tira de espaço livre entre a face direita do bloco passivo e a face esquerda de um bloco adjacente (ilustrada na Figura 6 entre o bloco passivo j_1 e o bloco adjacente j_3), a área desta tira é adicionada à área do bloco passivo (para as demais direções, rotacionar consecutivamente o desenho em 90°).

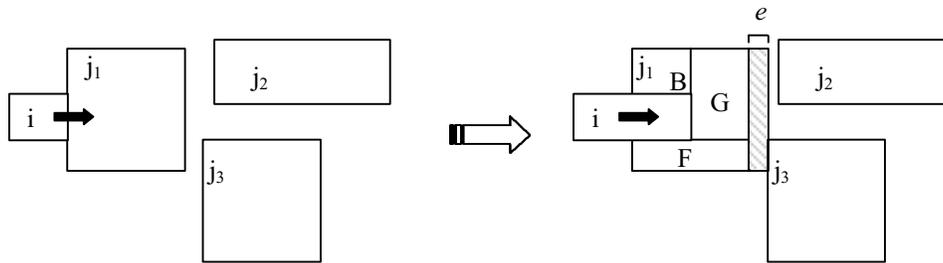


Figura 6 – Com a expansão do bloco ativo i , a área do bloco passivo j_1 é acrescida da área da tira entre o bloco j_1 e o bloco adjacente j_3 .

A espessura de uma tira de espaço livre é definida como a menor distância (horizontal para movimentos à direita e à esquerda, e vertical para movimentos acima e abaixo) entre uma face do bloco passivo e blocos adjacentes. No exemplo da Figura 6, a espessura e da tira adicionada ao bloco passivo j_1 é a distância entre j_1 e bloco adjacente j_3 . Entretanto, se incorporássemos tiras possivelmente distintas às áreas das configurações F e G do bloco j_1 (ao invés de uma única tira ao bloco j_1), a área F seria acrescida de uma tira com espessura e_1 enquanto G incorporaria uma tira correspondente à distância e_2 entre o bloco j_1 e o bloco j_2 , maior do que e_1 (veja Figura 7). É possível que tal modificação torne o processo de busca mais eficaz já que promove um maior aproveitamento dos espaços vazios do palete. Por razões de simplicidade de implementação, neste trabalho optou-se pela incorporação de tiras ao bloco.

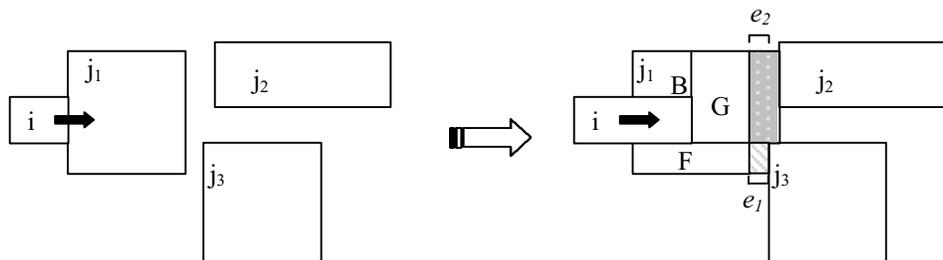


Figura 7 – Com a expansão do bloco ativo i , a área da configuração F e a área de configuração G do bloco passivo j_1 são acrescidas das áreas das tiras e_1 e e_2 , respectivamente.

Um movimento de expansão propriamente dito é definido pelos seguintes parâmetros:

- Bloco a ser expandido (1, 2, ..., número de blocos no padrão corrente).
- Direção da expansão (direita, esquerda, acima, abaixo).
- Camadas de expansão adjacentes (p.e., para cada face do bloco da Figura 4 têm-se: {1}, {2}, {3}, {4}, {1,2}, {2,3}, {3,4}, {1,2,3}, {2,3,4} e {1,2,3,4}).
- Número de caixas a serem adicionadas nas camadas de expansão (qualquer número de caixas, desde que não ultrapasse os limites do palete).
- Configuração pré-selecionada das áreas remanescentes dos blocos passivos (1, 2, 3 ou 4).

3.2 Deslizamento de Blocos

Sempre que um movimento é realizado, é comum o surgimento de espaços não aproveitados entre os blocos. Conforme mencionado na seção 3.1, procura-se incorporar alguns destes espaços na área remanescente dos blocos passivos onde são criados novos blocos (Figuras 6 e 7). Um outro procedimento que otimiza ainda mais o aproveitamento destes espaços é o deslizamento de blocos. Após cada iteração, os blocos são movidos de forma a concentrar os espaços vazios, preferencialmente, em uma única área. Como o padrão corrente geralmente possui um número de caixas já bastante próximo do ótimo (frequentemente com apenas uma caixa a menos), a soma dessas pequenas áreas livres aumenta a chance de inclusão de uma caixa adicional no padrão.

A estratégia utilizada consiste no deslizamento dos blocos em direção aos quatro cantos do palete. Inicialmente desliza-se os blocos em direção à face vertical (esquerda ou direita) mais próxima do palete. Em seguida desliza-se os blocos em direção à face horizontal (acima ou abaixo) mais próxima do palete. O procedimento é repetido até que nenhum deslizamento nas duas direções seja possível. Desta forma, os espaços vazios tendem a se concentrar no centro do palete (Figura 8). Note que os deslizamentos são restringidos pelas faces de outros blocos e pela face do palete.

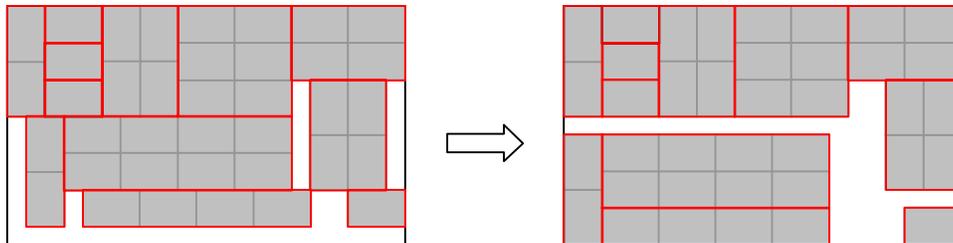


Figura 8 – Deslizamento de blocos em direção aos quatro cantos do palete.

3.3 Elementos de restrição

Em função da boa qualidade da solução inicial gerada pela heurística de blocos, o padrão obtido é invariavelmente um ótimo local com relação à estrutura de vizinhança proposta. Para que a busca heurística se desenvolva além deste ponto, torna-se necessária a utilização de mecanismos que restrinjam o retorno a padrões previamente obtidos. Para este fim, foram incorporados elementos restritivos de busca tabu com uso de memória de curto prazo (Glover & Laguna, 1997), também conhecida como busca tabu simples. Ao contrário de

versões mais avançadas que utilizam estruturas de memória de longo prazo para aplicação de estratégias de intensificação e diversificação, o papel da busca tabu simples é de apenas armazenar e restringir certas características ou atributos de soluções já visitadas por um determinado número de iterações.

Existe, entretanto, uma dificuldade intrínseca em se definir funções de memória para o problema, dado que o número de elementos que se alteram a partir da aplicação da estrutura de vizinhança é bastante variável. Cada movimento é definido por vários parâmetros, os quais podem assumir um amplo espectro de valores, além de envolver a alteração, eliminação e/ou geração de blocos em um número que não pode ser especificado previamente. Por esta razão, definir o que exatamente restringir não é uma tarefa simples.

Várias funções de memória foram investigadas. Em especial, a indexação por coordenadas de bloco apresentou os melhores resultados, sendo utilizada nos experimentos descritos na seção 4. Nesta implementação, utilizou-se períodos tabu fixos através da criação de uma lista circular L de t elementos onde são armazenadas as coordenadas dos blocos envolvidos em cada movimento. A proibição sobre coordenadas tem o propósito de procurar restringir movimentos a partir de regiões do palete mais recentemente modificadas. Em particular, são armazenadas as coordenadas originais do bloco ativo e as novas coordenadas dos blocos passivos. Não foi considerado o uso de períodos tabu dinâmicos uma vez que isso normalmente exigiria a criação de uma estrutura de memória para armazenagem de informações sobre o status (tabu-ativo ou não) de *todas* as possíveis coordenadas dos blocos (Glover & Laguna, 1997).

Sejam (x_1, y_1) e (x_2, y_2) respectivamente os pares de coordenadas do canto superior esquerdo e do canto inferior direito do bloco *ativo* do movimento candidato. Duas variantes para regra de ativação foram consideradas: (1) o movimento candidato é rejeitado quando (x_1, y_1) e (x_2, y_2) do bloco ativo estiver na lista L ; (2) um movimento candidato é rejeitado quando (x_1, y_1) ou (x_2, y_2) do bloco ativo estiver na lista L . Em ambos os casos, proíbe-se que blocos ativos e passivos sejam ativos nas próximas t iterações. Note que esta regra não os impede de serem passivos; uma conjectura para o bom desempenho apresentado (veja seção 4) é que ela permite “correções” da configuração resultante em movimentos subsequentes. A proibição de que blocos ativos e passivos sejam também passivos resultou em processos de busca altamente restritivos. Os melhores resultados foram obtidos com a regra (2), mais restritiva, o que resultou em sua seleção nos experimentos posteriores.

3.4 Avaliação dos movimentos

A função de avaliação dos movimentos inicialmente adotada para o cálculo do valor do movimento é:

$$G_0 = \text{ganho real}$$

ou seja, a diferença entre o número de caixas do padrão resultante da aplicação do movimento e o padrão anterior. G_0 pode assumir valores inteiros positivos, negativos ou zero, e para parte dos problemas tratados (veja seção 4), foi suficiente para obtenção da solução ótima.

Visando a resolução dos demais problemas, a análise do processo de busca sugeriu que dois movimentos com o mesmo valor de G_0 possam apresentar características estruturais distintas. Estas características estariam associadas a um grau maior ou menor de “qualidade estratégica” dos movimentos, de maneira que sua identificação e quantificação poderia permitir a distinção entre eles e uma possível melhoria de desempenho da busca.

Considere que padrões com alto grau de heterogeneidade sejam aqueles formados por um grande número de blocos de diferentes orientações intercalados entre si. Observou-se que em grande parte dos problemas considerados difíceis na literatura, a busca heurística é marcada pela geração de alguns poucos ótimos locais de grande influência (ou seja, existe uma grande recorrência a estas soluções, indicando baixa diversificação da busca). Estes ótimos locais são caracterizados por padrões com baixo grau de heterogeneidade (quase homogêneos). Foi também observado que movimentos de expansão que eliminam vários blocos passivos são aparentemente prejudiciais ao desempenho da heurística porque geralmente destroem uma estrutura do padrão corrente com alto grau de heterogeneidade. Com base nessas observações, concluiu-se que devem ser favorecidos movimentos com maior valor de G_0 :

- que geram padrões com maior grau de heterogeneidade.
- que modifiquem minimamente a estrutura do padrão corrente.

Com este objetivo, a função de avaliação original do movimento foi alterada com a inclusão de dois termos. Em um dado movimento, seja P o conjunto de blocos passivos afetados pela realização de um movimento de expansão do bloco ativo i , $area_e$ a área total de expansão do bloco ativo, e $area_s(j)$ a porção de $area_e$ que corresponde ao bloco passivo $j \in P$ (i.e., a área superposta do bloco passivo j). Seja também $area_n$ a área total dos novos blocos (criados a partir dos blocos passivos), e $area_h$ a porção de $area_n$ onde os blocos têm orientação diferente do bloco ativo. As Figuras 9 e 10 ilustram estas definições.

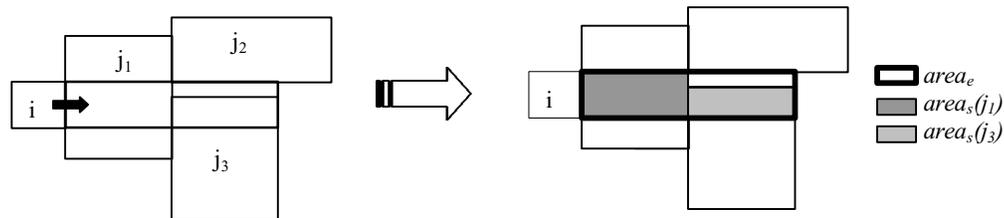


Figura 9 – Ilustração de $area_e$ e $area_s(j)$.

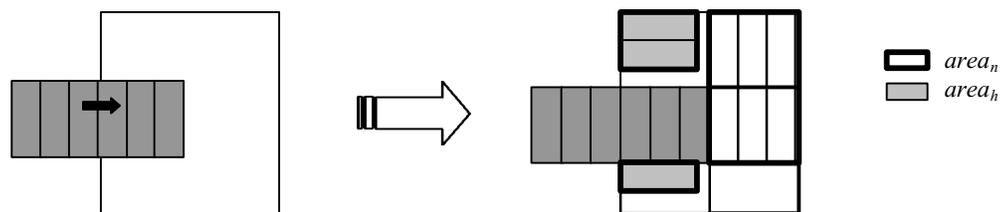


Figura 10 – Ilustração de $area_n$ e $area_h$.

A função de avaliação de movimentos foi então redefinida como:

$$G_1 = G_0 + \alpha \frac{area_h}{area_n} + (1 - \alpha) \frac{area_e - \sum_{j \in P} area_s(j)}{area_e}$$

O primeiro termo incluído ($\alpha \frac{area_h}{area_n}$) valoriza movimentos com maior proporção da área de

blocos novos com orientação diferente do bloco ativo. Note que este critério estimula mas não garante níveis maiores de heterogeneidade da solução, uma vez que a análise é local (restrita a porções do padrão) e não são consideradas intercalações das orientações dos

blocos. O segundo termo incluído ($(1-\alpha) \frac{area_e - \sum_{j \in P} area_s(j)}{area_e}$) valoriza movimentos com

menor proporção de área superposta dos blocos passivos sobre a área de expansão do bloco ativo. Menor proporção de área superposta significa que maior área livre do palete está sendo aproveitada na expansão, o que, por sua vez, implica que menores porções do padrão deverão ser efetivamente modificadas. Os dois critérios são ponderados por um fator $\alpha \in [0, 1]$ (ou seja, uma combinação convexa entre os dois critérios), permitindo o controle da dominância de um critério sobre o outro. Note que a soma destes dois termos incluídos em G_1 é um número em $[0, 1]$, funcionando como critério de desempate entre bons movimentos com o mesmo ganho real G_0 .

De acordo com a função G_1 , movimentos admissíveis com $G_0 > 0$ são prioritariamente selecionados. Na falta destes (mínimo locais), a busca selecionaria dentre os movimentos admissíveis com $G_0 = 0$, aqueles movimentos que resultem na melhor combinação (dado α) dos critérios de maior heterogeneidade (1º critério) e menor alteração da estrutura do padrão (2º critério). Somente na falta de movimentos com $G_0 = 0$ é que seriam selecionados movimentos com $G_0 < 0$, priorizando-se aqueles que maximizem a combinação dos dois critérios.

Entretanto, em vários dos problemas tratados verificou-se a ocorrência de ótimos locais de alta influência. Estes ótimos locais são caracterizados por padrões quase homogêneos, com um grande número de movimentos com $G_0 = 0$, que consistem de uma mera transferência de caixas entre blocos adjacentes com a mesma orientação. Estes movimentos não geram blocos novos com orientação diferente do bloco ativo e, por não alterarem na prática a estrutura da solução, são pouco efetivos. Como apresentam $G_0 = 0$ e $area_h = 0$, o valor de G_1 destes movimentos reduz-se ao 2º critério (menores áreas superpostas).

Nestas situações parece importante conferir maior chance de seleção a movimentos com $G_0 < 0$, uma vez que permitem que a busca escape destes ótimos locais influentes e se diversifique. Isso é feito penalizando-se movimentos que apresentem $G_0 = 0$ e $area_h = 0$ (a contribuição do 1º critério é nula). Para tal, retira-se a contribuição do 2º critério e soma-se um valor de penalidade $pen \leq 0$ à avaliação destes movimentos. Por exemplo, se $pen = -1$ e $\alpha = 0,5$, os movimentos com $G_0 = 0$ e $area_h = 0$ teriam valor igual à $-0,5$. Com isso, movimentos com $G_0 = -1$ com vantagem do 1º e 2º critérios teriam grande chance de serem selecionados. Obteve-se assim, a seguinte função de avaliação:

$$G_2 = \begin{cases} G_0 + \alpha \frac{area_h}{area_n} + (1-\alpha)pen, & \text{se } G_0 = 0 \text{ e } area_h = 0 \\ G_0 + \alpha \frac{area_h}{area_n} + (1-\alpha) \frac{area_e - \sum_{j \in P} area_s(j)}{area_e}, & \text{caso contrário.} \end{cases}$$

Note que se $G_0 = 0$ e $area_h = 0$, o valor de G_2 reduz-se a $(1-\alpha)pen$. Esta função apresentou um desempenho melhor que as anteriores e por esta razão, foi utilizada em todos os experimentos reportados na seção 4. A seguir são descritos os passos gerais do algoritmo.

3.5 Passos do Algoritmo

1. Gere um padrão inicial S_0 através de uma heurística de blocos. Faça $f(S_0)$ =número de caixas em S_0 , blk =número de blocos do padrão corrente e $ótimo$ =falso.
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.
3. Repita
 - 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 G_2 . 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. Incremente o número de iterações fazendo $it=it+1$.

Até que $mv=0$ ou it =máximo número de iterações ou $ótimo$ =verdadeiro.
4. Se $ótimo$ =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.

Caso a heurística de blocos tenha gerado um padrão homogêneo que não satisfaça o critério de otimalidade, deixa de ser possível movimentos de expansão de blocos e, conseqüentemente, a aplicação da heurística. Conforme mencionado em 3.1, uma maneira de superar esta limitação consiste em dividir o bloco em um número de sub-blocos, atualizar blk , e voltar ao passo 2.

Nos experimentos apresentados a seguir foi utilizada a heurística de blocos de Smith & De Cani (1980) no passo 1 do algoritmo, e não foram gerados padrões iniciais homogêneos (note que outras heurísticas de bloco poderiam ter sido utilizadas, por exemplo, Bischoff & Dowsland, 1992; Scheithuer & Terno, 1996; e Morabito & Morales, 1998). No passo 2, a otimalidade da solução pode ser verificada por meio de diversos limitantes superiores encontrados na literatura, conforme Nelissen (1995), Letchford & Amaral (2001) e Morabito & Farago (2002). Nos experimentos apresentados a seguir foi utilizado o número ótimo de caixas de cada problema, fornecido pela literatura. A maior dificuldade de implementação deste algoritmo diz respeito ao passo 1 (cálculo do conjunto de movimentos a cada iteração), em especial devido à necessidade do cálculo preciso das áreas não superpostas dos blocos passivos para cada uma das quatro configurações investigadas (veja seção 3.1).

4. Experimentos e Resultados Computacionais

A implementação computacional foi feita em Delphi 5, o que permitiu a elaboração de uma interface gráfica, onde os padrões gerados podiam ser observados. O uso desta interface permitiu a detecção imediata de problemas e limitações da abordagem, sendo, portanto, extremamente valiosa para o seu desenvolvimento. Os experimentos foram realizados em um microcomputador AMD Athlon 800 Mhz.

Cada execução representa uma variação paramétrica de α (peso do 1º critério em G_2) e t (tamanho da lista tabu L), ambos definidos na seção 3.4. Os valores de α foram variados em incrementos de 0,1 na faixa $[0, \dots, 0,5]$. A faixa foi restrita a esses valores pois experimentos preliminares indicaram que a dominância do 1º critério (maximização da heterogeneidade) é conflitante ao atendimento do 2º critério (minimização de área superposta). Por outro lado, a dominância do 2º critério afeta de igual forma movimentos mais ou menos heterogêneos. O parâmetro pen assumiu o valor -1 .

Os problemas foram tratados em ordem crescente do número ótimo de caixas e a faixa de valores do parâmetro t foi definida tomando como referência os valores do problema anterior (de menor porte) resolvido otimamente, admitindo a hipótese que quanto maior o número ótimo de caixas, maior é o valor adequado de t . Um limite máximo de 6000 iterações foi utilizado nos problemas com até 60 caixas, e 8000 iterações nos problemas de maior porte.

O algoritmo foi aplicado a um conjunto de 20 problemas da literatura, os 10 primeiros com solução ótima não guilhotinada de 1ª ordem (Dowland, 1984; Nelissen, 1994; Scheithauer & Terno, 1996; Battacharya *et al.*, 1998), e os demais 10 com solução ótima não guilhotinada de ordem superior (Morabito & Morales, 1998; Lins *et al.*, 2003). Estes problemas envolvem um número ótimo de 23 a 100 caixas, e são considerados difíceis uma vez que heurísticas de bloco como a de Steudel (1979), Smith & De Cani (1980) e Bischoff & Dowland (1982) não são capazes de obter seus padrões ótimos. Além disso, os últimos 10 problemas não são resolvidos nem por heurísticas de bloco mais elaboradas, como as de Scheithauer & Terno (1996) e Morabito & Morales (1998). Tais problemas são otimamente resolvidos pela abordagem de Lins *et al.* (2003), no entanto, os tempos computacionais requeridos são relativamente grandes (dezenas a centenas de minutos).

Dos 20 problemas tratados, 15 foram otimamente resolvidos, sendo 10 deles correspondentes a todo o conjunto de problemas de 1ª ordem, e 5 deles de ordem superior. A Tabela apresenta para cada problema as dimensões L, W, l e w , o número de caixas da solução inicial provida pela heurística de Smith e De Cani, o número de caixas da melhor solução provida pela heurística proposta e, nos problemas em que a heurística encontrou uma solução melhor que a inicial, os parâmetros da execução (α e t) juntamente com o tempo computacional (em segundos). Como vários dos problemas foram resolvidos otimamente com mais de um conjunto de valores de parâmetros, os resultados apresentados correspondem à execução com o valor de α que mais se aplicou a todos os problemas, ainda que em outras execuções bem-sucedidas, o tempo computacional tenha sido menor.

Tabela 1 – Resultados computacionais usando G_2 .

| Problema | 1ª Ordem | | | | | | Ordem Superior | | | | | | |
|----------|----------|---------|---------------------|-------------------|------------|-------------|----------------|----------|---------------------|-------------------|-----------------|-------------|-------|
| | (L,W) | (l,w) | # de caixas inicial | # de caixas final | α/t | Tempo (seg) | (L,W) | (l,w) | # de caixas inicial | # de caixas final | α/t | Tempo (seg) | |
| 1 | (22,16) | (5,3) | 22 | 23 [†] | 0,5/4 | 0,1 | 11 | (43,26) | (7,3) | 52 | 53 [†] | 0,5/15 | 4,9 |
| 2 | (57,44) | (12,5) | 40 | 41 [†] | 0,5/8 | 4,1 | 12 | (63,44) | (8,5) | 68 | 69 [†] | 0,5/24 | 49,7 |
| 3 | (86,82) | (15,11) | 41 | 42 [†] | 0,5/9 | 1,4 | 13 | (61,38) | (6,5) | 76 | 77 [†] | 0,2/28 | 102,2 |
| 4 | (42,39) | (9,4) | 43 | 45 [†] | 0,5/10 | 17,5 | 14 | (61,38) | (10,3) | 76 | 77 [†] | 0,5/29 | 54,5 |
| 5 | (124,81) | (21,10) | 46 | 47 [†] | 0,5/11 | 15,1 | 15 | (93,46) | (13,4) | 81 | 81 | - | - |
| 6 | (40,25) | (7,3) | 46 | 47 [†] | 0,5/12 | 21,2 | 16 | (141,71) | (13,8) | 95 | 96 [†] | 0,5/33 | 262,9 |
| 7 | (52,33) | (9,4) | 46 | 47 [†] | 0,5/12 | 28,2 | 17 | (106,59) | (13,5) | 95 | 95 | - | - |
| 8 | (56,52) | (12,5) | 46 | 48 [†] | 0,4/12 | 49,2 | 18 | (74,46) | (7,5) | 95 | 96 | 0,5/33 | 1 |
| 9 | (87,47) | (7,6) | 94 | 97 [†] | 0,2/28 | 119,92 | 19 | (86,52) | (9,5) | 98 | 98 | - | - |
| 10 | (67,44) | (6,5) | 96 | 97 [†] | 0,5/35 | 4,7 | 20 | (108,65) | (10,7) | 99 | 99 | - | - |

[†] número ótimo de caixas

A Figura 11 apresenta os padrões ótimos de ordem superior obtidos para os problemas 11, 12, 13, 14 e 16 da Tabela 1. As diferentes cores distinguem as orientações das caixas. Convém observar que se no passo 3.2 do algoritmo for utilizada a função G_0 ao invés de G_2 , apenas os problemas 1 a 7 da Tabela 1 são resolvidos otimamente, o que mostra a importância desta última função de avaliação em relação à primeira. Note também que para os problemas 11 a 20 (não guilhotinados de ordem superior), o índice de sucesso da heurística tende a decrescer com o aumento do número ótimo de caixas, o que seria esperado. Nos exemplos em que padrões ótimos não foram encontrados (problemas 15, 17, 18, 19 e 20) a heurística proposta obteve soluções a uma caixa do número ótimo (exceto pelo problema 18, a melhor solução foi gerada pela heurística de blocos de Smith e De Cani no passo 1 do algoritmo).

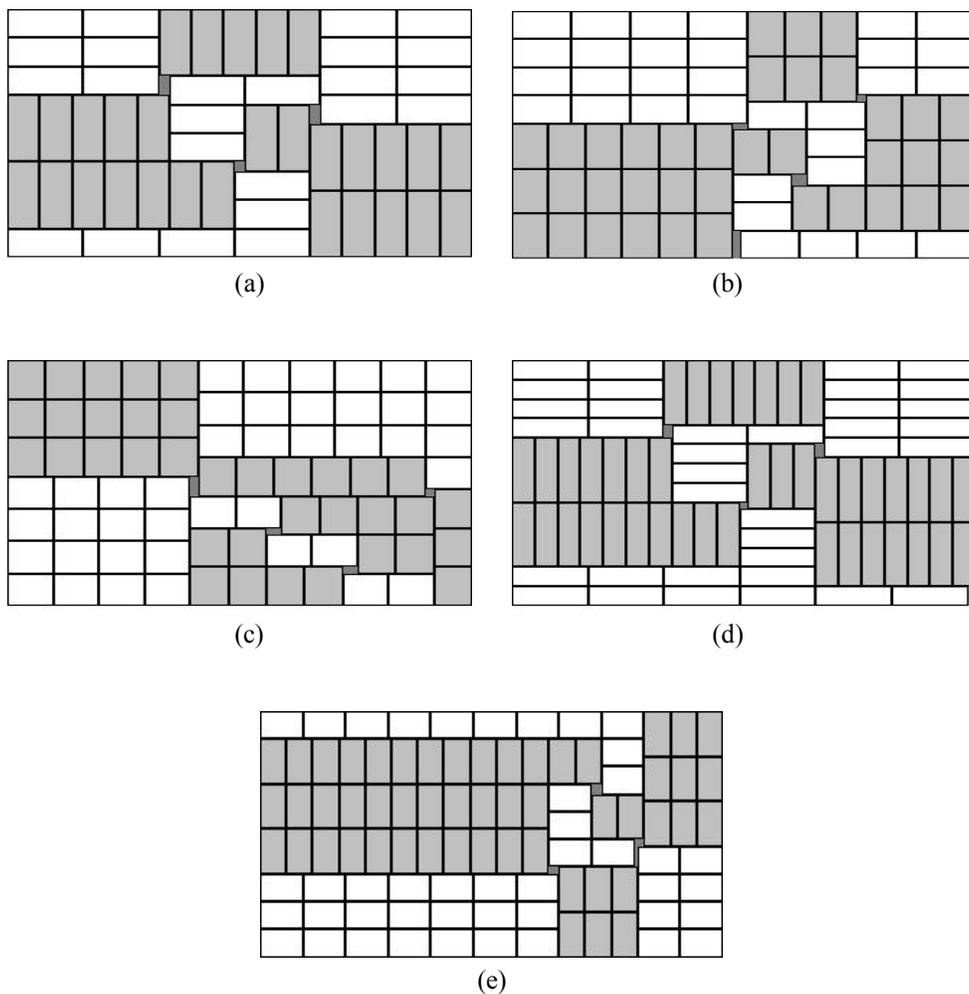


Figura 11 – Padrões não-guilhotinados de ordem superior: (a) Problema 11 (53 caixas); (b) Problema 12 (69 caixas) (c) Problema 13 (77 caixas); (d) Problema 14 (77 caixas); (e) Problema 16 (96 caixas).

4.1 Análises

Procurou-se estabelecer uma relação entre a facilidade de resolução e o tamanho da área livre do palete, ou seja, a área do palete retirando-se aquela efetivamente utilizada pelo número ótimo de caixas. Para o conjunto de problemas aqui tratados (todos com área livre não nula variando de 22% até 83% da área de uma caixa), verificou-se que aqueles com menores áreas livres são os mais sujeitos à recorrência de ótimos locais. Conforme mencionado na seção 3.4, em situações como essas é difícil encontrar caminhos de busca que gerem padrões diversificados em função da atratividade dos ótimos locais frente à qualidade inferior dos padrões intermediários.

Por outro lado, quanto maior a área livre do palete, maior é a facilidade da geração de padrões distintos com a mesma qualidade. Nossa hipótese é que nesses casos maior é a chance de se obter um padrão ótimo do problema, independentemente deste ser de 1ª ordem ou de ordem superior. Os resultados tendem a corroborar esta hipótese. Os problemas de 1ª ordem (todos resolvidos otimamente) são os que apresentam maior área livre – 9 dos 10 problemas têm área livre entre 36% e 83% da área de uma caixa. Os problemas de ordem superior, por outro lado, apresentam em geral menores áreas livres; todos os 10 problemas têm áreas livres entre 22% e 38% da área de uma caixa.

No que diz respeito aos melhores valores do parâmetro α (distribuição do peso aos critérios heterogeneidade e mínima área superposta), observa-se que em 12 dos 15 problemas resolvidos foram obtidos padrões ótimos com $\alpha = 0,5$ (Tabela 1). As exceções foram os problemas 8, 9 e 13, com $\alpha = 0,4$, $\alpha = 0,2$ e $\alpha = 0,2$, respectivamente. Este resultado sugere a importância de uma ponderação equilibrada (α em torno de 0,5) entre os dois critérios utilizados na avaliação dos movimentos.

Os melhores valores do parâmetro t (tamanho de lista), por sua vez, parecem depender do número de caixas no padrão inicial fornecido pela heurística de blocos de Smith & De Cani, (1980). Seja n^0 o número de caixas no padrão inicial. A Figura 12 apresenta a curva n^0 vs. t para os problemas tratados.

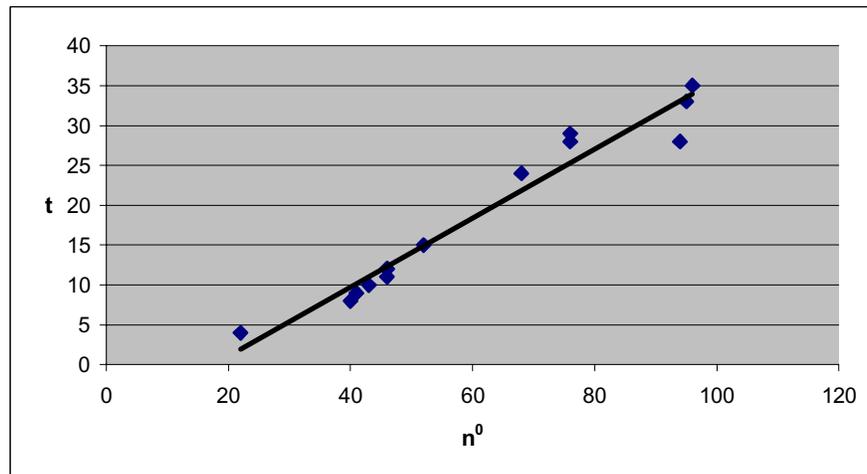


Figura 12 – Número de caixas no padrão inicial (n^0) vs. Tamanho de lista (t).

A análise de regressão destes dados resultou na seguinte função linear:

$$t = 0,4317n^0 - 7,5287$$

O valor de t assim obtido poderia ser utilizado para determinar a faixa de valores a serem testados ao se aplicar o algoritmo a outros problemas. A função sugerida tem boa aderência aos dados, exibindo alta correlação ($R^2 = 0,9563$) entre t e n^0 . De fato, outros limitantes inferiores do número de caixas, e mesmo o número ótimo, foram testados ao invés de n^0 , resultando em funções de regressão linear cujos coeficientes não diferem muito dos aqui apresentados.

Cabe ressaltar que esta análise estatística baseia-se em uma amostra de problemas razoavelmente difíceis. No caso de problemas de fácil resolução, aspectos como os parâmetros α e t tornam-se irrelevantes uma vez que a aplicação de uma boa heurística de blocos no passo 1 do algoritmo geralmente resulta em um padrão ótimo. A heurística de blocos de Morabito & Morales (1998), por exemplo, resolve otimamente mais de 99% de um conjunto de 20.000 classes de problemas.

Em termos de esforço computacional, os tempos de resolução foram considerados pequenos ou dentro de limites razoáveis, considerando-se que se tratam de problemas difíceis na literatura. Por exemplo, o problema 9 (Tabela 1), resolvido otimamente pela heurística em menos de 2 minutos, requereu de um algoritmo exato mais de 8 horas de execução em uma estação HP9000/720 (Nelissen, 1995). Heurísticas de blocos mais elaboradas são capazes de resolver os problemas 1-10 (1º ordem) em tempos computacionais bastante pequenos (p.e., Scheithauer & Terno, 1996 e Morabito & Morales, 1998); por outro lado, diferentemente da abordagem aqui proposta, tais heurísticas não são capazes de resolver nenhum dos problemas 11-20 (ordem superior). Uma alternativa é utilizar *softwares* comerciais de programação matemática para resolvê-los, por exemplo, a linguagem de modelagem GAMS com o *solver* CPLEX (Brooke *et al.*, 1992) aplicados ao modelo 0-1 de Beasley (1985); no entanto, os tempos computacionais chegam facilmente à ordem de horas de CPU.

Outro resultado interessante dos experimentos foi a geração de padrões ótimos alternativos não-guilhotinados de ordem superior em vários problemas com solução ótima de 1ª ordem conhecida (Morabito & Morales, 1998). Esse resultado é interessante na medida em que, em contextos reais, o empilhamento de camadas de caixas arranjadas em padrões não guilhotinados de ordem superior pode permitir uma melhor estabilidade do carregamento. Isto é feito melhorando-se a chamada amarração de carga (*interlock*), por sua vez obtida com a prática de empilhar as camadas rotacionadas alternadamente de 180°. Desta forma, procura-se evitar a coincidência na sobreposição vertical das caixas para não tornar o conjunto instável (Kullick, 1982).

5. Conclusões e Perspectivas

Neste trabalho foi proposto um algoritmo heurístico para resolução do problema de carregamento de paletes do produtor. O algoritmo parte de uma solução inicial (gerada por uma heurística de blocos), a qual é sucessivamente alterada pela expansão de blocos selecionados. A expansão provoca a eliminação e criação de novos blocos e conseqüente alteração do padrão de empacotamento. Foram utilizados elementos restritivos de busca tabu simples para promover a continuidade das explorações para além dos ótimos locais, e elaborada uma função de avaliação de movimentos. Experimentos computacionais contemplaram 20 problemas difíceis reportados na literatura (10 dos quais não resolvidos otimamente pelas heurísticas de bloco aqui citadas), e em 15 deles foram obtidos padrões ótimos em tempos computacionais razoáveis.

O exame dos problemas não resolvidos otimamente pode sugerir modificações na heurística proposta. Mesmo com a adoção dos critérios de maximização da heterogeneidade e minimização da área superposta, o processo de busca na maioria desses problemas ainda apresentou recorrência de ótimos locais, o que justificaria a não obtenção de padrões ótimos. Nesses casos, portanto, permanece o desafio de promover diversidade à busca, mantendo-se a qualidade das soluções geradas. Para este fim, novas funções de avaliação e a introdução de fases de intensificação e diversificação são alguns dos possíveis caminhos de pesquisa a serem investigados.

Agradecimentos

Os autores agradecem o apoio da FAPESP (processo 01/2972-2) e do CNPq (processo 522973/95-7), assim como os valiosos comentários e sugestões dos três revisores anônimos.

Referências Bibliográficas

- (1) Amaral, A. & Letchford, A. (2001). Comment on: An exact algorithm for general, orthogonal, two-dimensional knapsack problems. Submetido para *European Journal of Operational Research*.
- (2) Arenales, M. & Morabito, R. (1995). An and/or-graph approach to the solution of two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, **84**, 599-617.

- (3) Arenales, M.; Morabito, R. & Yanasse, H. (1999). Special issue: Cutting and packing problems. *Pesquisa Operacional*, **19**(2), 109-284.
- (4) Balasubramanian, R. (1992). The pallet loading problem: A survey. *International Journal of Production Economics*, **28**, 217-225.
- (5) Beasley, J.E. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, **33**, 49-64.
- (6) Bischoff, E. & Dowsland, W. (1982). An application of the micro to product design and distribution. *Journal of the Operational Research Society*, **33**, 271-280.
- (7) Bischoff, E. & Ratcliff, M. (1995). Loading multiple pallets. *Journal of the Operational Research Society*, **46**, 1322-1336.
- (8) Bischoff, E. & Waescher, G. (1995). Special issue on cutting and packing. *European Journal of Operational Research*, **84**(3), 503-712.
- (9) Bhattacharya, S.; Roy, R. & Bhattacharya, S. (1998). An exact depth-first algorithm for the pallet loading problem. *European Journal of Operational Research*, **110**, 610-625.
- (10) Brooke, A.; Kendrick, D. & Meeraus, A. (1992). *GAMS: A user's guide*. Release 2.25, The Scientific Press, EUA.
- (11) Dowsland, K. (1984). The three-dimensional pallet chart: an analysis of the factors affecting the set of feasible layouts for a class of two-dimensional packing problems. *Journal of the Operational Research Society*, **35**, 895-905.
- (12) Dowsland, K. (1987). An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, **31**, 78-84.
- (13) Dowsland, K. (1993). Some experiments with simulated annealing techniques for packing problems. *European Journal of Operational Research*, **68**, 389-399.
- (14) Dowsland, K. (1996). Simple tabu Thresholding and the Pallet Loading Problem. **In: Metaheuristics: theory & applications** [edited by I.H. Osman and J. P. Kelly], Kluwer Academic Publishers, 378-406.
- (15) Dowsland, K. & Dowsland, W. (1992). Packing problems. *European Journal of Operational Research*, **56**, 2-14.
- (16) Dyckhoff, H. & Finke, U. (1992). Cutting and packing in production and distribution: Typology and bibliography. Springer-Verlag Co, Heidelberg.
- (17) Dyckhoff, H.; Scheithauer, G. & Terno, J. (1997). Cutting and packing. **In: Annotated bibliographies in combinatorial optimisation** [edited by M. Amico, F. Maffioli and S. Martello], John Wiley & Sons, New York, NY, 393-414.
- (18) G, Y. & Kang, M. (2001). A fast algorithm for two-dimensional pallet loading problems of large size. *European Journal of Operational Research*, **134**, 193-200.
- (19) Glover, F. & Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publisher.
- (20) Hadjiconstantinou, E. & Christofides, N. (1995). An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research*, **83**, 39-56.

- (21) Herbert, A. & Dowsland, K. (1996). A family of genetic algorithms for the pallet loading problem. *Annals of Operations Research*, **63**, 415-436.
- (22) Hodgson, T. (1982). A combined approach to the pallet loading problem. *IIE Transactions*, **14**(3), 176-182.
- (23) Hifi, M. (2002). Special issue: Cutting and packing problems. *Studia Informatica Universalis*, **2**(1), 1-161.
- (24) Kullick, A. (1982). Interlocking pallet pattern simulation program. *Industrial Engineering*, September, 22-24.
- (25) Letchford, A. & Amaral, A. (2001). Analysis of upper bounds for the pallet loading problem. *European Journal of Operational Research*, **132**, 582-593.
- (26) Lins, L.; Lins, S. & Morabito, R. (2002). An n-tet graph approach for non-guillotine packings of n-dimensional boxes into an n-container. *European Journal of Operational Research*, **141**, 421-439.
- (27) Lins, L.; Lins, S. & Morabito, R. (2003). An L-approach for packing (l,w) -rectangles into rectangular and L-shaped pieces. Aceito para publicação no *Journal of the Operational Research Society*.
- (28) Martello, S. (1994). Special issue: Knapsack, packing and cutting, Part II: Multidimensional knapsack and cutting stock problems. *INFOR*, **32**(4).
- (29) Martins, G.H. (2002). Packing in two and three dimensions. PhD Dissertation, Naval Postgraduate School, Monterey, CA, EUA.
- (30) Morabito, R. & Arenales, M. (1994). An and/or-graph approach to the container loading problem. *International Transactions in Operational Research*, **1**(1), 59-73.
- (31) Morabito, R. & Farago, R. (2002). A tight Lagrangean relaxation bound for the manufacturer's pallet loading problem. *Studia Informatica Universalis*, **2**(1), 57-76.
- (32) Morabito, R. & Morales, S. (1998). A simple and effective recursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, **49**, 819-828.
- (33) Morabito, R. & Morales, S. (1999). Errata: A simple and effective recursive procedure for the manufacturers' pallet loading problem. *Journal of the Operational Research Society*, **50**, 876.
- (34) Morabito, R.; Morales, S. & Widmer, J.A. (2000). Loading optimization of palletized products on trucks. *Transportation Research*, Part E **36**, 285-296.
- (35) Nelissen, J. (1994). Solving the pallet loading problem more efficiently. Working Paper, Graduiertenkolleg Informatik und Technik, Aachen.
- (36) Nelissen, J. (1995). How to use the structural constraints to compute an upper bound for the pallet loading problem. *European Journal of Operational Research*, **84**, 662-680.
- (37) Scheithauer, G. & Terno, J. (1996). The G4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, **47**, 511-522.

- (38) Smith, A. & De Cani, P. (1980). An algorithm to optimize the layout of boxes in pallets. *Journal of the Operational Research Society*, **31**, 573-578.
- (39) Steudel, H. (1979). Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. *Management Science*, **10**, 997-1004.
- (40) Sweeney, P. & Paternoster, E. (1992). Cutting and packing problems: a categorized, application-oriented research bibliography. *Journal of the Operational Research Society*, **43**, 691-706.
- (41) Tarnowski, A.; Terno, J. & Scheithauer, G. (1994). A polynomial-time algorithm for the guillotine pallet loading problem. *INFOR*, **32**(4), 275-287.
- (42) Tsai, R.; Malstrom, E. & Kuo, W. (1993). Three dimensional palletization of mixed box sizes. *IEE Transactions*, **25**(4), 64-75.
- (43) Wang, P. & Waescher, G. (2002). Special issue on cutting and packing. *European Journal of Operational Research*, **141**(2), 239-469.
- (44) Yamassaki, C. & Pureza, V. (2001). Um refinamento do algoritmo tabu de Dowsland para o problema de carregamento de paletes do produtor. Relatório Técnico, Departamento de Engenharia de Produção, UFSCar.