

## **Apoio à decisão no carregamento de veículos: seleção e posicionamento de produtos**

**Oberlan Christo Romão**

Departamento de Informática  
Universidade Federal de Viçosa – UFV  
Viçosa – MG  
oberlan@gmail.com

**André Gustavo dos Santos**

Departamento de Informática  
Universidade Federal de Viçosa – UFV  
Viçosa – MG  
andre@dpi.ufv.br

### **RESUMO**

Nesse trabalho é tratado um problema de empacotamento para encontrar a disposição de produtos de diferentes dimensões e peso dentro de caminhões com diferentes capacidades. O problema surgiu de um caso real de uma companhia, em que a carga deve obedecer a certa restrição de peso sobre os eixos do veículo; por isso o veículo é dividido em três partições, sendo a do meio a de maior capacidade. Além dos produtos obrigatórios, a companhia pode completar a carga com outros produtos não obrigatórios, o que aumenta a natureza combinatória do problema. O problema foi dividido em duas partes, uma que seleciona produtos e caminhões, determinando qual o melhor caminhão e em qual partição cada produto deve ser colocado, e outra que determina a posição dos produtos selecionados nas devidas partições dos veículos. O método foi testado ainda com instâncias da literatura que não possuem todas essas restrições.

**Palavras-chave:** otimização combinatória, empacotamento, transporte

**Área principal:** L&T – Logística & Transportes

### **Abstract**

In this work we deal with a packing problem to find where different items with different dimensions have to be packed inside vehicles of different capacities. The problem arose from a real company, whose load has to satisfy certain weight constraints above the vehicle axle; then, the vehicle is divided in three partitions, the middle one having more capacity. Beside the obligatory items, the company may fill the vehicle with non-obligatory items, which increases the combinatory of the problem. The problem is decomposed in two parts, one selects items and vehicles, determining which is the best vehicle and in which partition each item must be located, and other that calculate the positions of the items inside the proper partition of the vehicles. The method was tested also with instances taken from the literature that does not have all these constraints.

**Keywords:** combinatorial optimization, bin packing, transport

**Main area:** L&T – Logistics & Transportes

## 1. Introdução

Este trabalho relata um método baseado em programação matemática proposto para apoiar o processo de decisão de uma companhia que precisa fazer o carregamento de diferentes produtos numa frota heterogênea de veículos. A companhia recebe pedidos de seus clientes, cada um contendo uma lista de vários produtos. Ela deve então fazer o planejamento do transporte desses produtos, escolhendo quais veículos utilizar, e que produtos colocar em cada um deles. A carga entregue não é compartilhada entre os clientes, de forma que não é necessário determinar rotas, pois cada veículo sai da companhia diretamente para o cliente. Pelo mesmo motivo não há necessidade de escolher uma ordem no carregamento dos produtos dentro do veículo, pois todos serão carregados no mesmo momento e local, e também entregues em um mesmo local. Porém, existe a necessidade de distribuição de peso dentro do veículo, sem concentrar muita carga sobre os eixos. Boa parte dos produtos são considerados obrigatórios, ou seja, devem obrigatoriamente fazer parte do carregamento. Mas a companhia pode entregar outros produtos, não obrigatórios, para complementar a carga e aproveitar o espaço não utilizado do veículo pelos produtos obrigatórios. Tais produtos geralmente fazem parte dos pedidos do cliente, mas não para uma determinada data, e podem ser entregues adiantadamente.

Percebe-se então que o problema é uma variação do problema do empacotamento: consiste em empacotar unidades menores em unidades maiores, minimizando a capacidade total não utilizada das unidades maiores. Outros critérios comuns são maximizar o número de unidades menores empacotadas, e minimizar o número de unidades maiores utilizadas. Neste problema as unidades menores são os produtos e as unidades maiores são os veículos. A capacidade dos veículos é limitada tanto em relação à área, quanto ao volume e quanto ao peso. Embora seja objetivo da companhia minimizar o número de veículos utilizados, o objetivo principal é minimizar a carga morta total, que é a capacidade não utilizada do veículo, seja em relação à área, volume ou peso. Pode-se minimizar um critério ou uma combinação destes critérios, mas todos devem ser atendidos, ou seja, os produtos colocados dentro de cada veículo devem respeitar a capacidade do veículo nos três critérios.

O problema foi dividido em duas partes: uma faz a seleção dos produtos e veículos, de tal forma que todos os produtos obrigatórios sejam selecionados, e possivelmente alguns não obrigatórios, e também sejam selecionados alguns veículos, cuja capacidade individual – peso, área e volume – seja suficiente para acomodar o total de peso, área e volume dos produtos selecionados para o veículo. Esta escolha é feita sem considerar o posicionamento dos produtos dentro do veículo. A segunda parte tenta acomodar, em cada veículo, os produtos selecionados para serem carregados no veículo. Se não for possível acomodá-los, ou uma solução não seja encontrada dentro de um prazo estipulado, volta-se à primeira parte para selecionar outros produtos ou veículos, ou mudar a alocação dos produtos aos veículos.

O problema da primeira parte é complexo. É um problema de *bin-packing* NP-difícil (Miyazawa & Wakabayashi, 1998), então ainda não se conhece método para encontrar soluções ótimas em tempo polinomial em relação ao tamanho da entrada, o que impede encontrar a solução ideal para instâncias do problema com uma grande quantidade de produtos. Assim, é recomendado o uso de heurísticas. Entretanto, em vez de meta-heurísticas, neste trabalho foi usado um modelo de programação linear inteira. Para várias instâncias a solução ótima do modelo é encontrada. Para aquelas cuja solução não é encontrada num tempo limite, usa-se a melhor solução encontrada até o momento.

O problema da segunda parte também é um problema complexo. Porém, os testes mostraram que quando os produtos escolhidos podem de fato ser posicionados dentro do veículo selecionado, uma busca exaustiva por tentativa e erro encontra tal solução rapidamente na maioria dos casos testados. Quando não há um posicionamento possível a busca não responde em tempo curto. Assim, foi estipulado um prazo limite para a busca encontrar o posicionamento dos produtos. Se ela não encontrar, entende-se que ele não existe, o que na maioria dos casos é fato.

Existem vários trabalhos semelhantes na literatura, mas a maioria considera frota de veículos homogênea: Lodi, Martello e Vigo (1999a) minimizam o número de veículos; o mesmo em Lodi e Martello e Vigo (1999b), sem considerar rotação de produtos; e Pisinger e Sigurd

(2005) minimizam custo do empacotamento. Apenas Silva (2008) aborda o mesmo problema, com produtos de diferentes status, porém mostra resultados de carga morta apenas em relação ao peso, diferentemente do trabalho aqui apresentado, que mostra resultados para outros critérios.

O método foi testado com instâncias baseadas em dados reais da companhia. Foram usados produtos com peso, área e volume semelhantes aos entregues pela companhia, bem como veículos de dimensões reais. Além disso, para efeito de validação do método, também foram usadas instâncias de um problema clássico de empacotamento, que não possui todas as restrições desse problema (por exemplo, sem distinção entre produtos obrigatórios e não obrigatórios), e os resultados encontrados foram bastante satisfatórios quando comparados aos da literatura.

O texto é organizado da seguinte forma: na seção 2 é descrito o modelo de programação matemática usado na parte de seleção de veículos e produtos; a seção 3 descreve o método usado na parte de posicionamento dos produtos nos veículos, e a seção 4 mostra a interação entre os dois métodos; a seção 5 mostra os resultados e algumas conclusões são apresentadas na seção 6.

## 2. Módulo de seleção dos produtos e dos veículos

Nesta seção descreve-se o primeiro módulo do sistema desenvolvido, responsável por resolver a primeira parte do problema. Este módulo seleciona os produtos e os caminhões de uma frota, determinando qual o melhor veículo e em qual partição cada produto deve ser colocado, a fim de minimizar a carga morta total, não importando por enquanto como será feita a montagem dos produtos. A formalização do problema e o modelo de programação matemática são baseados no trabalho de Silva (2008). Apesar dos produtos e veículos possuírem três dimensões, essa modelagem trata o problema de forma bidimensional, pois os produtos podem ser girados apenas no sentido ortogonal, e não podem ser empilhados. Logo, o único efeito da terceira dimensão, a altura, é impor limite na altura dos produtos que podem ser colocados em cada veículo.

Entretanto, aqui é acrescentada a possibilidade de empilhamento de produtos, algo que na prática pode ser feito pela companhia apenas para alguns tipos de itens. O problema continua sendo praticamente bidimensional, pois o empilhamento, quando permitido, pode ser feito apenas com itens do mesmo tipo, ou seja, produtos iguais. Logo, nenhum produto pode ser colocado com sua base tocando mais de um produto. Sua base estará exatamente em cima de outro produto de mesma dimensão, e pode, em certos casos, ser tratado como um único produto, de altura maior.

### 2.1. Conjuntos considerados no problema

- $P = \{p_1, p_2, p_3, \dots\}$  = tipos de produtos a serem alocados
- $S = \{\text{OBT} = \text{“obrigatório”}, \text{N\_OBT} = \text{“não obrigatório”}\}$  = status possíveis do produto
- $T = \{t_1, t_2, t_3, \dots\}$  = veículos disponíveis na companhia
- $B = \{1 = \text{“sobre o eixo da frente”}, 2 = \text{“entre os eixos”}, 3 = \text{“sobre o eixo de trás”}\}$  = partições da caçamba dos veículos

### 2.2. Parâmetros do problema

- $l_p, c_p, h_p, w_p$  = largura, comprimento, altura e peso dos produtos do tipo  $p$
- $ob_p, nob_p$  = quantidade de produtos do tipo  $p$  com entrega obrigatória e não obrigatória
- $e_p$  = quantidade em estoque de produtos do tipo  $p$
- $m_p$  = empilhamento máximo permitido para produtos do tipo  $p$
- $spc$  = espaçamento mínimo exigido entre os produtos a serem alocados
- $L_{tb}, C_{tb}, H_{tb}$  = largura, comprimento, altura da partição  $b$  do veículo  $t$
- $W_{tb}$  = carga máxima suportada pela partição  $b$  do veículo  $t$

Também são definidos os seguintes termos, para uma descrição mais compacta do modelo:

- $a_p, v_p$  = área e volume dos produtos do tipo  $p$ , respectivamente =  $l_p \cdot c_p$  e  $l_p \cdot c_p \cdot h_p$
- $A_{tb}, V_{tb}$  = área e volume da partição  $b$  do veículo  $t$ , respect. =  $L_{tb} \cdot C_{tb}$  e  $L_{tb} \cdot C_{tb} \cdot H_{tb}$
- $A_t, V_t, W_t$  = área, volume e peso suportados pelo veículo  $t = \sum_{b \in B} A_{tb}, \sum_{b \in B} V_{tb}, \sum_{b \in B} W_{tb}$
- $q_{ptb}$  = quantidade de produtos do tipo  $p$  suportada pela partição  $b$  do veículo  $t$  em relação ao peso =  $\frac{W_{tb}}{w_p}$
- $r_{ptb}$  = quantidade de produtos do tipo  $p$  suportada pela partição  $b$  do veículo  $t$  em relação à área, já considerando o espaçamento mínimo exigido entre os produtos =  $\frac{A_{tb}}{(c_p + spc)(l_p + spc)} = \frac{A_{tb}}{a_p + spc(c_p + l_p + spc)}$
- $u_{ptb}$  = quantidade de produtos do tipo  $p$  suportada pela partição  $b$  do veículo  $t$  em relação ao volume =  $\frac{V_{tb}}{v_p}$

### 2.3. Variáveis do modelo

- $x_{pstb}$  = quantidade de produtos do tipo  $p$  com status  $s$  na partição  $b$  do veículo  $t$
- $Z_t$  = variável binária indicando se o veículo  $t$  será usado na alocação dos produtos
- $Z_{tb}$  = variável binária indicando se a partição  $b$  do veículo  $t$  será usada na alocação
- $N_{ptb}$  = número de pilhas do produto  $p$  na partição  $b$  do veículo  $t$
- $CMP_t, CMA_t, CMV_t$  = carga morta em relação ao peso, área e volume do veículo  $t$

### 2.4. Função Objetivo

Como já dito, o objetivo é minimizar a carga morta dos veículos, que é a capacidade não utilizada, seja em relação à área, volume ou peso, ou qualquer combinação linear desses valores. Abaixo uma definição mais genérica, que pode ser usada para qualquer um desses casos.

$$\min \sum_{t \in T} [CMP_t, CMA_t, CMV_t] \quad (1)$$

### 2.5. Restrições

As restrições tratam os diversos limites estabelecidos no problema, que devem ser respeitados: peso, área e volume dos veículos, necessidade de carregamento dos produtos obrigatórios, quantidade de produtos não obrigatórios em estoque, entre outras. As restrições são apresentadas e explicadas a seguir.

As restrições (2) garantem que se qualquer partição do veículo for usada na alocação dos produtos, então o veículo com esse partição será utilizado.

$$Z_{tb} \leq Z_t, \forall t \in T, \forall b \in B \quad (2)$$

As restrições (3) a (8) garantem o limite de peso, área e volume, bem como definem a carga morta em relação a cada parâmetro. As restrições (3) garantem que a carga total em cada partição não ultrapassa o limite da partição, e as restrições (4) estabelecem a carga morta total de cada veículo em relação ao peso. As restrições (5) e (6) tem o mesmo efeito, mas sobre a área, já considerando empilhamento e um espaçamento obrigatório entre os produtos. As restrições (7) e (8) tem o mesmo efeito, mas sobre o volume.

$$\sum_{p \in P} \sum_{s \in S} w_p x_{pstb} \leq W_{tb} Z_{tb}, \forall t \in T, \forall b \in B \quad (3)$$

$$CMP_t = W_t Z_t - \sum_{p \in P} \sum_{s \in S} \sum_{b \in B} w_p x_{pstb}, \forall t \in T \quad (4)$$

$$\sum_{p \in P} (a_p + spc(c_p + l_p + spc)) N_{ptb} \leq A_{tb} Z_{tb}, \forall t \in T, \forall b \in B \quad (5)$$

$$CMA_t = A_t Z_t - \sum_{p \in P} \sum_{b \in B} (a_p + spc(c_p + l_p + spc)) N_{ptb}, \forall t \in T \quad (6)$$

$$\sum_{p \in P} \sum_{s \in S} v_p x_{pstb} \leq V_{tb} Z_{tb}, \forall t \in T, \forall b \in B \quad (7)$$

$$CMV_t = V_t Z_t - \sum_{p \in P} \sum_{s \in S} \sum_{b \in B} v_p x_{pstb}, \forall t \in T \quad (8)$$

As restrições (9), (10) e (11) fortalecem as restrições anteriores, não permitindo que a quantidade selecionada de cada produto ultrapasse um limite superior permitido a cada um, em relação ao peso, área e volume, respectivamente.

$$\sum_{s \in S} x_{pstb} \leq q_{ptb}, \forall p \in PP, \forall t \in T, \forall b \in B \quad (9)$$

$$\sum_{s \in S} x_{pstb} \leq r_{ptb}, \forall p \in P, \forall t \in T, \forall b \in B \quad (10)$$

$$\sum_{s \in S} x_{pstb} \leq u_{ptb}, \forall p \in P, \forall t \in T, \forall b \in B \quad (11)$$

Tratados os limites de peso, área e volume, é necessário tratar a quantidade de produtos selecionados de cada tipo. As restrições (12) garantem que todos os produtos obrigatórios sejam posicionados em alguma partição de algum veículo, as restrições (13) limitam a quantidade de produtos não obrigatórios àquela já pedida pelo cliente, e as restrições (14) garantem que a quantidade selecionada de cada produto, tanto obrigatórios quanto não obrigatórios, não ultrapasse o estoque presente na companhia.

$$\sum_{t \in T} \sum_{b \in B} x_{pstb} = ob_p, \forall p \in P, s = OBP \quad (12)$$

$$\sum_{t \in T} \sum_{b \in B} x_{pstb} \leq nob_p, \forall p \in P, s = N\_OBP \quad (13)$$

$$\sum_{s \in S} \sum_{t \in T} \sum_{b \in B} x_{pstb} \leq e_p, \forall p \in P \quad (14)$$

As restrições (15) e (16) estabelecem o limite mínimo e máximo de pilhas do produto  $p$  na partição  $b$  do veículo  $t$ , considerando os extremos: todas as pilhas com apenas 1 produto; e todas as pilhas com  $m_p$  produtos.

$$\sum_{s \in S} x_{pstb} \leq m_p N_{ptb}, \forall p \in P, \forall t \in T, \forall b \in B \quad (15)$$

$$\sum_{s \in S} x_{pstb} \geq N_{ptb}, \forall p \in P, \forall t \in T, \forall b \in B \quad (16)$$

As restrições (17) garantem que a pilha dos produtos não ultrapassa a altura do veículo.

$$\sum_{s \in S} x_{pstb} \leq (H_{tb} / h_p) * N_{ptb}, \forall p \in P, \forall t \in T, \forall b \in B \quad (17)$$

Por fim, as restrições (18) e (19) fazem com que as cargas estejam bem distribuídas pelo caminhão concentrando uma maior carga entre seus eixos, já que é a parte mais resistente e menos instável do mesmo.

$$\sum_{p \in P} \sum_{s \in S} w_p x_{pst2} \geq \sum_{p \in P} \sum_{s \in S} w_p x_{pst1}, \forall t \in T \tag{18}$$

$$\sum_{p \in P} \sum_{s \in S} w_p x_{pst2} \geq \sum_{p \in P} \sum_{s \in S} w_p x_{pst3}, \forall t \in T \tag{19}$$

### 3. Módulo de posicionamento dos produtos selecionados

Este módulo determina a posição onde cada produto selecionado deve ficar dentro da devida partição do veículo. Essa segunda etapa é resolvida de forma construtiva, por tentativa e erro: tentam-se todas as posições possíveis para cada item, inclusive girando ortogonalmente os produtos, já que em alguns casos só há solução se um ou mais itens forem rotacionados.

#### 3.1. Representação do empacotamento

Para determinar o empacotamento, cada partição do veículo é representada por uma matriz bidimensional, com as dimensões da partição, representação semelhante à usada por Portal *et al* (2009). Cada célula da matriz representa 1 dm<sup>2</sup> da partição e inicialmente preenchida com 0. Quando um produto *p* é adicionado na posição (*i, j*) da matriz, todas as células do “retângulo” de vértices (*i, j*) e (*i+c<sub>p</sub>, j+l<sub>p</sub>*), ou (*i+l<sub>p</sub>, j+c<sub>p</sub>*) caso o produto tenha sido girado, recebem o número do produto, identificando que estas posições estão ocupadas por tal produto.

A figura 1 apresenta um exemplo da estrutura usada. Dado uma partição B (5dm, 5dm), inicialmente a matriz está com todas as células zeradas (figura 1(a)), ou seja, nenhum produto ainda foi adicionado. Ao se adicionar o produto *p*<sub>1</sub> (3dm, 4dm) na partição, a estrutura da matriz será modificada, como é mostrada na figura 1(b). Se for adicionado *p*<sub>2</sub> (2dm, 5dm), a matriz será novamente modificada (figura 1(c)).

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figura 1(a) - Início

1	1	1	1	0
1	1	1	1	0
1	1	1	1	0
0	0	0	0	0
0	0	0	0	0

Figura 1(b) – Após a inserção do produto 1

1	1	1	1	0
1	1	1	1	0
1	1	1	1	0
2	2	2	2	2
2	2	2	2	2

Figura 1(c) – Após a inserção do produto 2

**Figura 1.** Representação do empacotamento

#### 3.2. Algoritmo para o empacotamento

Para encontrar uma solução para o empacotamento, foi usado um algoritmo de tentativa e erro (*backtracking*): os produtos são colocados um a um, e para cada posição possível de um produto tentam-se todas as posições para os demais, inclusive girando ortogonalmente. Quando um produto não pode ser posicionado, tenta-se outra posição para o produto anterior, e assim recursivamente. Depois de alguns testes, verificou-se que ordenando os produtos de forma decrescente de acordo com sua área, é obtida uma maior agilidade no tempo para encontrar uma solução. Vale lembrar que mesmo com esta estratégia, o programa pode não encontrar uma solução em tempo hábil. Na próxima seção é explicado como esse problema é tratado.

#### 4. Interação entre os dois módulos

É possível notar que a complexidade do algoritmo do módulo de posicionamento é exponencial em relação às dimensões da partição e do número de produtos selecionados para a partição. Apesar disso, nas instâncias testadas, quando há uma solução, ela geralmente é encontrada em pouco tempo. Isso acontece pois, mesmo com uma grande quantidade de produtos, a partição de um veículo não é grande, e não há muitos produtos numa mesma partição. Pelo menos não com as dimensões de produtos utilizados na empresa. O problema maior acontece quando não há solução. O método é exponencial, e testar todas as alternativas é caro computacionalmente. Algumas vezes isso acontece também quando há uma solução, e é necessário girar um dos primeiros produtos, e assim o algoritmo acaba verificando um grande número de alternativas até encontrar uma solução.

Note que a solução do primeiro módulo leva em consideração a área dos produtos, mesmo quando não é esse o critério de otimização. Um dos problemas é que mesmo respeitando a área total, ele pode gerar uma solução em que o empacotamento é inviável. Por exemplo, imagine que há dois caminhões, com  $L_1 = 4\text{dm}$  e  $C_1 = 4\text{dm}$  (área  $16\text{dm}^2$ ) e  $L_2 = 4\text{dm}$  e  $C_2 = 5\text{dm}$  (área  $20\text{dm}^2$ ). Suponha ainda dois produtos para empacotar,  $p_1$  com  $l_1 = 3\text{dm}$  e  $c_1 = 3\text{dm}$  (área  $9\text{dm}^2$ ) e  $p_2$  com  $l_2 = 2\text{dm}$  e  $c_2 = 2\text{dm}$  (área  $4\text{dm}^2$ ), e que os dois veículos suportem o peso a altura dos produtos. Pela área, os dois produtos podem ser empacotados em qualquer um dos veículos. O módulo de seleção selecionará o primeiro veículo, que tem área menor, tendo uma menor carga morta em relação à área. No entanto, pela geometria dos produtos, esse empacotamento é impossível, como mostra a figura 2.

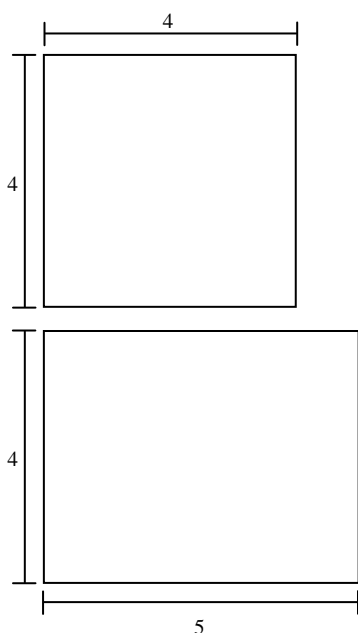


Figura 2 (a) – dois veículos

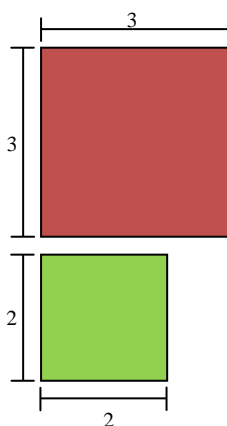


Figura 2 (b) – dois produtos a serem empacotados, que, pela área total, caberiam nos veículos

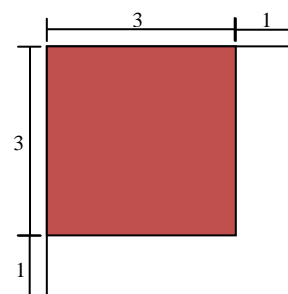


Figura 2 (c) – Veículo 1 selecionado e produto 1 empacotado, mas não há espaço para o produto 2

**Figura 2.** Problema encontrado no módulo de empacotamento

Se o módulo de posicionamento estiver demorando muito a encontrar uma solução, então ou ela não existe ou devem ser feitas muitas mudanças de posição e rotação de produtos para que ela seja encontrada. A solução utilizada para contornar esse problema foi impor um limite de tempo de execução ao módulo de posicionamento. Se nesse prazo não for encontrada uma solução, o módulo é interrompido e retorna-se ao módulo de seleção forçando-o a selecionar uma solução diferente. Isto é feito reduzindo-se ligeiramente a largura e o comprimento dos veículos selecionados anteriormente. Com isso, a distribuição dos produtos nos veículos muda e outros veículos ou outros produtos podem ser selecionados, já que a área dos veículos mudou. No



exemplo anterior, se for reduzido em 1dm a largura ou comprimento do primeiro veículo, sua nova área será de 12dm<sup>2</sup>, e como a soma das áreas dos produtos é 13dm<sup>2</sup>, o módulo de seleção, obrigatoriamente, selecionará o veículo 2 para acomodar os produtos. O módulo de posicionamento continua usando as dimensões originais dos veículos. Para o exemplo anterior, ele encontraria facilmente uma solução, que seria colocar o p<sub>1</sub> na posição (0, 0) e p<sub>2</sub> em (3, 0), como mostra a figura 3. O diagrama da figura 4 ilustra a interação entre os dois módulos.

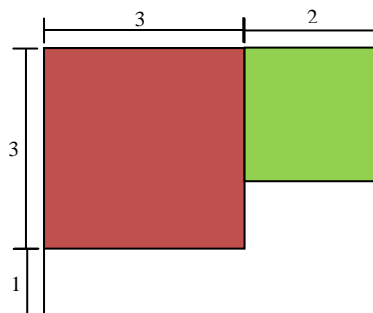


Figura 3. Solução para o problema anterior

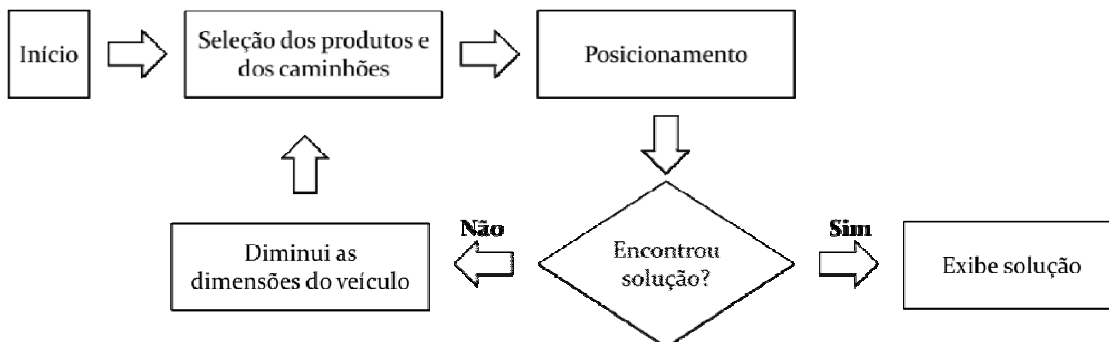


Figura 4. Diagrama de interação entre os dois módulos

Após encontrar uma solução, o software apresenta um desenho 3D interativo, mostrando visualmente onde cada produto deve ficar no interior de cada caminhão. Isto pode ser visto de duas formas: em forma de blocos sólidos ou de blocos com a identificação de cada produto. Esse resultado não é mostrado neste texto por questão de espaço e escopo do trabalho.

## 5. Resultados

O sistema foi implementado na linguagem C++, e os testes executados em uma máquina com processador Intel Core 2 Quad 2.40GHz, com 2GB RAM, e sistema operacional Windows Vista 32bits. Para resolver o modelo de programação linear inteira do módulo de seleção, o sistema usa o software Xpress® (Guéret, Prins & Sevaux, 2002).

Foram realizados testes com 20 tipos de produtos, com diferentes dimensões e peso, e todos com empilhamento (EMP) máximo 2; e 5 tipos de veículos, com capacidades diferentes, sendo 4 destes os modelos de veículos disponíveis na companhia, e 1 acrescentado para aumentar a complexidade das instâncias. As instâncias foram divididas em cinco classes, que se diferenciam pela quantidade de produtos obrigatórios (OBP) e não obrigatórios (N\_OBP) de cada produto. Os dados dos produtos e veículos são mostrados nas tabelas 1 e 2 respectivamente.

Em vários casos, tanto o módulo de seleção quanto o módulo de posicionamento podem demorar a encontrar uma solução, então foi fixado um limite de 5 minutos para cada execução dos módulos. Se o módulo de seleção não encontrar a solução ótima nesse prazo, ele retoma a melhor solução encontrada até o momento, que pode não ser a ótima. Se o módulo de posicionamento não encontrar uma solução no prazo, interrompe-se sua execução e volta-se ao módulo de



seleção reduzindo-se a largura e comprimento dos veículos, conforme mostrado na figura 4 da seção anterior. Nesse caso, inicialmente as dimensões dos veículos são reduzidas em 10%. Caso o módulo de posicionamento consiga empacotar os produtos agora escolhidos, volta-se ao primeiro módulo reduzindo apenas metade desse valor na configuração original, pois talvez haja uma solução sem tanta redução. Continua-se esse processo em estilo “busca binária” para achar um bom valor da redução (o menor valor para o qual o módulo de posicionamento encontra uma solução em tempo hábil). Se os 10% iniciais de redução não forem suficientes, as dimensões são reduzidas ainda mais 10%, e assim sucessivamente, até encontrar uma solução.

**Tabela 1** – Dados dos produtos e das classes de instâncias

ID	Dimensões (dm)			Peso (kg)	EMP	Classe 1		Classe 2		Classe 3		Classe 4		Classe 5	
	L	C	A			OBP	N_OBP	OBP	N_OBP	OBP	N_OBP	OBP	N_OBP	OBP	N_OBP
1	9	5	7	378	2	1	1	2	1	5	3	11	4	17	16
2	2	4	3	29	2	1	1	3	1	3	3	8	3	22	5
3	6	10	8	576	2	1	1	1	1	1	3	7	2	16	8
4	7	5	6	252	2	1	2	1	2	4	2	6	4	15	13
5	3	6	4	86	2	1	1	4	1	7	5	4	4	12	10
6	7	10	8	672	2	1	1	1	1	3	4	9	3	13	13
7	5	3	3	54	2	1	1	2	1	2	3	5	5	19	11
8	5	3	4	72	2	1	2	2	2	5	2	6	5	13	16
9	9	6	7	454	2	1	1	2	1	2	2	11	3	10	8
10	4	2	3	29	2	1	1	1	1	6	1	7	4	14	10
11	7	6	6	302	2	1	1	3	1	3	4	8	4	18	1
12	2	7	4	67	2	1	1	1	1	5	2	8	2	10	11
13	6	4	5	144	2	1	2	5	2	5	4	9	3	13	10
14	10	4	7	336	2	1	4	2	4	3	4	7	4	15	8
15	5	4	4	96	2	1	2	2	2	2	3	8	4	14	15
16	3	10	6	216	2	1	1	1	1	8	3	5	4	17	5
17	3	8	5	144	2	1	2	3	2	3	2	9	2	11	14
18	10	7	7	588	2	1	3	1	3	5	3	11	3	14	17
19	6	8	5	288	2	1	5	1	5	5	5	5	5	18	10
20	7	8	7	470	2	1	2	2	2	3	2	6	2	19	19
<b>Total:</b>						20	35	40	35	80	60	150	70	300	220

**Tabela 2** – Dados dos veículos disponíveis

ID	Dimensões (dm)			Quant.	Capacidade		
	L	C	H		Peso (kg)	Área (dm <sup>2</sup> )	Volume (dm <sup>3</sup> )
1	24	76	40	1	14000	1824	72960
2	24	131	30	1	27000	3144	94320
3	24	123	40	1	31000	2952	118080
4	24	148	30	1	41000	3552	106560
5	30	60	30	1	6000	1800	54000

Os resultados, mostrados na tabela 3, foram encontrados minimizando a carga morta em relação à área (CMA), área e volume (CMA+CMV), peso (CMP) e volume (CMV). A primeira

coluna refere-se à classe das instâncias e a segunda o tipo de otimização utilizado. A terceira indica a quantidade de veículos usados para ser feito o carregamento dos produtos e a quarta a quantidade total de produtos empacotados, incluindo os não obrigatórios. As próximas três colunas mostram o valor da carga morta em relação aos critérios, e em seguida este valor em termos percentuais. Por fim, o tempo total gasto, em segundos, até atingir a solução final, e o número de iterações dos dois módulos.

**Tabela 3** – Soluções encontradas para as instâncias de cada classe sob diferentes critérios de otimização.

Classe	Critério de Otimização	Selecionados		Carga morta			Capacidade não usada			Tempo(s)	Iterações
		Veículos	Produtos	CMA	CMP	CMV	Área	Peso	Volume		
1	CMA	1	49	145	578	61774	7,95%	4,13%	84,67%	354,16	4
	CMA + CMV	5	30	947	2	49001	52,61%	0,03%	90,74%	1,53	1
	CMP	1	52	365	0	61292	20,01%	0,00%	84,01%	168,25	1
	CMV	5	25	1000	3	49001	55,56%	0,05%	90,74%	1,03	1
2	CMA	1	60	284	256	61883	15,57%	1,83%	84,82%	614,40	4
	CMA + CMV	1	61	286	4	61294	15,68%	0,03%	84,01%	1048,65	4
	CMP	1	54	429	2	61293	23,52%	0,01%	84,01%	300,08	1
	CMV	1	58	358	6	61295	19,63%	0,04%	84,01%	306,47	4
3	CMA	1 e 4	149	730	19716	150171	13,58%	35,85%	83,65%	944,17*	4
	CMA + CMV	2	99	303	8	72867	9,64%	0,03%	77,26%	1004,90	4
	CMP	2 e 5	95	1791	36	129022	36,23%	0,11%	86,99%	300,12*	1
	CMV	2	107	894	8	71820	28,44%	0,03%	76,15%	1,39	1
4	CMA	1, 3 e 4	229	1024	26278	249048	12,30%	30,56%	83,69%	612,11	4
	CMA + CMV	4 e 5	180	1251	43	121427	23,37%	0,09%	75,63%	1502,70	3
	CMP	1 e 4	185	906	7	133691	16,85%	0,01%	74,47%	1936,73	4
	CMV	4 e 5	164	1625	41	121423	30,36%	0,09%	75,62%	1802,76	4
5	CMA	1, 2, 3 e 4	382	1205	6413	244966	10,50%	5,68%	62,50%	635,00	4
	CMA + CMV	1, 2 e 4	327	1685	278	207069	19,78%	0,34%	75,62%	1500,32	3
	CMP	1, 2, 3, 4 e 5	440	3669	92	347021	27,64%	0,08%	77,82%	1500,30*	3
	CMV	1, 2 e 4	323	1942	91	205568	22,79%	0,11%	75,07%	1800,45	4

\*O módulo de seleção foi interrompido antecipadamente pelo tempo limite, então pode não ter sido encontrada a melhor otimização

O sistema foi capaz de encontrar soluções para todos os 20 testes (5 classes sob 4 diferentes critérios de otimização). Em 6 deles foi necessária apenas uma iteração, caracterizando uma solução ótima. Como esperado, o critério mais difícil é minimizar a carga morta em relação à área, pois quanto menor este valor mais difícil é o trabalho do módulo de posicionamento. Em todos os casos de otimização sob esse critério foram necessárias 4 iterações dos módulos, indicando que o módulo de posicionamento não conseguiu encontrar um posicionamento viável para a seleção ótima de produtos em relação à área, ou por ser um método exponencial, ou porque tal solução não existia, já que o módulo de seleção não considera o formato dos produtos, apenas sua área. De qualquer forma, em apenas três testes o módulo de seleção gastou o tempo limite em alguma iteração.

Uma aparente incoerência ocorre na solução da classe 3 minimizando a carga morta em relação ao volume pois a solução em relação ao peso encontra uma solução 2 dm<sup>3</sup> melhor. Mas note que para encontrar uma solução em relação ao volume foram feitas 4 iterações. Isso indica

que o módulo de seleção encontrou uma solução ainda melhor em relação à carga morta por volume, mas o módulo de posicionamento não conseguiu posicionar os produtos. Apenas após algumas reduções na dimensão dos veículos o módulo de seleção conseguiu selecionar produtos de tal forma que o módulo de posicionamento conseguisse posicioná-los. Não é, portanto, a solução ótima, embora seja solução ótima para os veículos de dimensões reduzidas. Ou seja, uma solução bastante próxima da solução ótima. Algo semelhante ocorre ao minimizar a carga morta por área nas classes 3 e 4, e em relação ao peso nas classes 3 e 5.

Embora não tenham sido encontradas instâncias na literatura com as características das descritas neste trabalho, também foram realizados testes com instâncias da literatura, pois a modelagem é geral o suficiente para tratar diversas delas. Como exemplo, a tabela 4 mostra os resultados encontrados para 20 instâncias (Berkey & Wang, 1987) com diferentes quantidades e dimensões de produtos em veículos iguais para cada instância, usadas entre outros por Martello e Vigo (1998) e Lodi, Martello e Vigo (1999a). Os autores buscavam encontrar a solução com o menor número de veículos para empacotar todos os produtos. Para que o sistema aqui descrito encontrasse soluções semelhantes, todos os produtos foram considerados obrigatórios, todos os veículos foram considerados com apenas uma partição (ou seja, não há equilíbrio da carga, mesmo porque não são informados os pesos dos produtos), e a otimização foi feita minimizando a carga morta em relação à área (consequentemente busca minimizar o número de veículos).

A primeira coluna refere-se à classe das instâncias; a segunda coluna representa o número de produtos que devem ser alocados e em seguida as dimensões dos veículos em dm. A quarta e quinta colunas mostram respectivamente a solução encontrada na literatura e a solução encontrada pelo método proposto no trabalho. As demais colunas mostram informações relativas ao método aqui proposto: carga morta da solução encontrada, tempo de execução dos módulos de seleção e posicionamento, e número de vezes que cada módulo foi executado.

**Tabela 4** – Resultados para as instâncias da literatura

Classe	Produtos Usados	Dimensões (dm)	Número de Veículos		CMA	Tempo (seg)		Iterações
			Literatura	Algoritmo proposto		Seleção	Posicionamento	
2	20	90x30x20	1	1	76,00%	1,50	2,30	8
	40	90x30x20	1	1	66,85%	1,40	4,50	8
	60	90x30x20	1	1	25,67%	0,10	0,12	1
	80	90x30x21	1	1	15,78%	0,20	0,20	2
	100	90x30x22	2	2	49,40%	0,20	0,52	2
3	20	120x40x30	2	2	28,40%	1,10	1,48	5
	40	120x40x31	2	3	37,72%	1,50	3,30	6
	60	120x40x32	6	6	34,26%	21,70	8,20	7
	80	120x40x33	6	8	36,83%	14,50	13,40	9
	100	120x40x34	7	-	-	>20min	-	8
6	20	900x300x100	1	1	77,70%	0,00	4,90	1
	40	900x300x100	1	1	67,50%	0,00	6,50	1
	60	900x300x101	1	1	48,65%	0,00	6,95	1
	80	900x300x102	1	1	20,10%	21,00	5,80	12
	100	900x300x103	2	2	46,30%	5,00	3,50	5
7	20	300x100x100	2	2	26,90%	4,30	10,50	5
	40	300x100x101	4	4	30,39%	5,20	15,30	7
	60	300x100x102	5	6	32,63%	8,80	22,30	6
	80	300x100x103	8	9	29,01%	6,30	30,80	7
	100	300x100x104	10	11	26,18%	10,80	83,0	8

Analisando-se os resultados encontrados, nota-se que de uma forma geral, nos casos testados, o algoritmo encontrou soluções conhecidas (14 instâncias) ou próximas delas (5 instâncias), em um tempo bastante curto, mesmo quando há muitos produtos e veículos. Em apenas um dos casos uma solução não pôde ser encontrada num tempo limite (instância da classe

3 com 100 produtos). Vale destacar que o método aqui apresentado é mais geral e algumas restrições impostas no trabalho não foram consideradas nos trabalhos comparados da literatura.

## 6. Conclusões

Este trabalho mostra a solução do problema de seleção de produtos e veículos mais apropriados para uma companhia realizar a entrega aos clientes, reduzindo a carga morta do carregamento. O problema foi dividido em duas partes, seleção e posicionamento, resolvidas de forma iterativa, por programação matemática e enumeração.

O sistema foi testado com instâncias baseadas em dados reais de uma companhia de transporte, e encontra soluções ótimas para algumas das instâncias testadas, e soluções próximas para várias outras instâncias. Também foi avaliada sua eficácia com instâncias da literatura, que não tinham todas as restrições das instâncias reais da companhia. Os métodos encontraram as soluções da literatura em 70% dos casos apresentados.

## Agradecimentos

Este trabalho foi financiado pelo CNPq com bolsa do projeto Otimal II, e pela Funarbe – Fundação Arthur Bernardes, com o projeto Funarpeq – Métodos para Problemas de Transporte. Agradecemos também a Dash Optimization/Fair Isaac pela licença acadêmica do Xpress®.

## Referências

- Berkey, J.O.; Wang, P.Y.** (1987). Two dimensional finite bin packing algorithms. *Journal of the Operational Research Society*, vol. 38: pp. 423-429.
- Guéret, C.; Prins, C; Sevaux, M.** (2002). Applications of optimization with XpressMP. Dash Optimization Ltd.
- Lodi, A.; Martello, S.; Vigo, D.** (1999a). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, vol. 11: pp.345-357.
- Lodi, A.; Martello, S.; Vigo, D.** (1999b). Approximation algorithms for the oriented two-dimensional bin packing problem. *Journal of Operational Research Society*, vol. 112:158-166.
- Martello, S.; Vigo, D.** (1998). Exact solution of the two-dimensional finite bin packing problem, *Management Science*, vol. 44 (3), pp.388-399.
- Miyazawa, F.K.; Wakabayashi, Y.** (1998). Algoritmos de aproximação para problemas de empacotamento. In: *XVIII Congresso Nacional de Matemática Aplicada e Computacional*, Belo Horizonte, MG, pp. 335-351.
- Portal, G.M.; Rocco, R.; Ritt, M.; Buriol, L.S.** (2009). Uma busca tabu aplicada ao problema de roteamento com restrições de empacotamento tridimensionais. *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, Porto Seguro, BA.
- Pisinger, D.; Sigurd, M.** (2005). The two-dimensional bin packing problem with variable bin sizes and costs. *Discrete Optimization*, vol. 2(2): pp. 154-167
- Silva, P.T.G.C.** (2008). *Uma interface interativa e um novo algoritmo para o problema de empacotamento tridimensional*. Monografia de projeto orientado em computação II. Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, MG.