

# Apoio à decisão no carregamento de veículos com empacotamento tridimensional

**Oberlan Christo Romão**

Departamento de Informática – Universidade Federal de Viçosa  
Av. P.H. Rolfs, s/n, DPI - Campus UFV - 36570-000, Viçosa – MG  
oberlan.romao@ufv.br

**André Gustavo dos Santos**

Departamento de Informática – Universidade Federal de Viçosa  
Av. P.H. Rolfs, s/n, DPI - Campus UFV - 36570-000, Viçosa – MG  
andre@dpi.ufv.br

## RESUMO

Nesse trabalho é tratado um problema do empacotamento tridimensional que tenta encontrar uma disposição ótima para produtos de diferentes dimensões e peso em uma frota heterogênea de caminhões, de forma a minimizar a carga morta (não utilizada) total. Cada produto apresenta um status de acordo com a urgência de entrega dos mesmos, ganhando uma maior prioridade no momento de alocação aqueles com maior urgência na entrega. O problema é dividido em duas partes. A primeira seleciona os produtos e os veículos, determinando qual o melhor veículo cada produto deve ser colocado, minimizando a carga morta total. Para essa parte é proposto um modelo de programação linear inteira. A segunda calcula o posicionamento dos produtos selecionados, determinando a posição onde cada item deve ficar no veículo. Esta parte é resolvida por um método construtivo auxiliado por algoritmo genético. O método é testado com instâncias específicas e com dois conjuntos de instâncias da literatura, mostrando sua eficiência e superando outros métodos em alguns casos.

**PALAVRAS CHAVE:** Otimização Combinatória, Empacotamento Tridimensional, Carregamento de Contêineres.

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

## ABSTRACT

In this work we deal with a three-dimensional bin packing problem trying to find an optimal disposition for products of different dimensions and weight inside an heterogenous fleet of vehicles, to minimize the total unused capacity. Each product has a status according to the urgency of delivering, having a higher priority to allocate those with greater urgency. The problem is divided into two parts. The first selects products and vehicles, determining the best vehicle where each product must be placed, minimizing the total unused capacity. In this part we propose an integer linear programming model. The second part calculates the positioning of the selected products, determining the position where each item must be inside the vehicle. The method is tested with specific instances and with two sets of instances from the literature, showing its effectiveness and improving the results of other methods in some cases.

**KEYWORDS:** Combinatorial Optimization, Three-dimensional Bin Packing, Containers loading.

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

## 1. Introdução

Este trabalho apresenta um método híbrido que combina programação matemática e meta-heurística 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 deles contendo uma lista de vários produtos. Ela deve então fazer o planejamento do transporte desses produtos, escolhendo quais veículos utilizar, e quais 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. 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 antecipadamente.

O problema é uma variação do problema do empacotamento tridimensional: empacotar unidades menores em unidades maiores. No caso em questão, as unidades menores são os produtos e as unidades maiores os veículos. O problema consiste então em encontrar a posição de cada produto dentro dos veículos. Mais que isso, deve selecionar quais os veículos serão utilizados, e os produtos opcionais para compor o carregamento. Embora minimizar o número de veículos utilizados seja um dos objetivos da empresa, o objetivo principal é minimizar a carga morta total, que é a capacidade não utilizada do veículo, seja em relação ao volume ou peso, ou uma combinação deles, mas os produtos colocados dentro de cada veículo devem respeitar a capacidade do veículo em ambos critérios. O diferencial do problema aqui tratado é a existência dos produtos opcionais que podem ser usados para compor a carga dos veículos.

O método proposto resolve o problema em duas partes: *i*) módulo de seleção: seleciona produtos e veículos, de tal forma que todos os produtos obrigatórios sejam selecionados, e possivelmente alguns não obrigatórios, e que a capacidade individual dos veículos - peso e volume - seja suficiente para acomodar o total de peso e volume dos produtos selecionados para cada um (esta escolha é feita sem levar em conta o posicionamento dos produtos dentro do veículo); *ii*) módulo de posicionamento: tenta acomodar, em cada veículo, os produtos selecionados para serem carregados nele. Se não for possível acomodá-los volta-se ao módulo de seleção para selecionar outros produtos ou veículos, ou mudar a alocação previamente feita.

Determinar a posição de cada produto dentro do veículo com o critério de minimizar o volume não utilizado no carregamento de um veículo é um problema NP-Difícil (Martello et al., 2000), ou seja, ainda não se conhece método para encontrar soluções ótimas em tempo polinomial em relação à entrada, o que impede encontrar a solução ideal para instâncias do problema com uma grande quantidade de produtos e veículos em tempo eficiente. Assim, o uso de heurísticas é o mais indicado. Entretanto, para o módulo de seleção, foi usado um modelo de programação linear inteira. No módulo de posicionamento é usada uma heurística construtiva auxiliada por um algoritmo genético.

## 2. Referencial teórico

Em geral, problemas de empacotamento tridimensional (do Inglês *Three-dimensional Bin Packing*) podem assumir as seguintes classificações:

- Problema de Empacotamento Tridimensional ou *3D-BPP (Three-dimensional Bin Packing)*: Dado um conjunto de itens retangulares de tamanhos variados e um número de contêineres, o objetivo é empacotar todas os itens no menor número de contêineres. Geralmente os produtos não podem ser rotacionados. Em Martello et al. (2000) é apresentado um método exato para se resolver esse problema;
- Problema de Carregamento de Contêiner ou *CLP (Container Loading Problem)*: Os itens devem ser empacotados em um único contêiner, com o objetivo de maximizar o volume utilizado do mesmo. Uma variante desse problema é o caso onde todos os itens devem ser acomodados em um único contêiner que possui altura infinita, com objetivo de encontrar uma solução viável de tal forma que a altura seja mínima. O primeiro trabalho sobre o *CLP* foi apresentado por George and Robinson (1980). Outros trabalhos sobre o problema são encontrados em Lim et al. (2005), Bischoff and Ratcliff (1995a), Bischoff and Ratcliff (1995b), Baltacioglu et al. (2006) e Bortfeldt and Gehring (1997);
- Problema da Mochila Tridimensional ou *3D-KLP (Three-dimensional Knapsack Loading Problems)*: Cada item possui um valor associado, o objetivo é escolher um subconjunto de itens, que caibam dentro de uma única mochila, no caso o contêiner, de modo que o lucro total seja máximo. Considerando que o lucro de cada item seja o seu volume, então o *3D-KLP* corresponde à minimizar o volume não utilizado, como no *CLP*. Trabalhos sobre *3D-KLP* foram apresentados por Pisinger (2002) e Gehring et al. (1990).

Nesse trabalho é desenvolvido um algoritmo para um problema que une características do Problema do Empacotamento Tridimensional e do Problema de Carregamento de Contêiner. Trabalhos semelhantes podem ser encontrados na literatura em Ivancic et al. (1989) e Eley (2002), mas estes trabalhos consideram veículos (contêineres) homogêneos, diferentemente do trabalho aqui apresentado. Em relação ao carregamento com produtos opcionais, encontramos apenas Romão and Santos (2010) e Silva and Mateus (2009) que abordam o mesmo problema, porém nesses trabalhos o empacotamento é praticamente bidimensional, já que produtos podem ser empilhados apenas sobre produtos idênticos e em quantidade bastante limitada, diferentemente do trabalho aqui apresentado, que permite empilhamento tridimensional de produtos diferentes.

### 3. Método Híbrido Proposto

O problema é dividido em duas partes: a primeira faz uma seleção de produtos e veículos, e a segunda o posicionamento desses produtos nos veículos selecionados. A seguir são descritos os métodos de solução de cada uma dessas partes e a interação entre elas.

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

Este módulo seleciona os produtos e os caminhões de uma frota, determinando qual o melhor veículo para cada produto ser colocado, a fim de minimizar a carga morta total, não importando por enquanto como será feito posicionamento dos produtos no interior do veículo.

Para se resolver essa parte foi proposto um modelo de programação matemática baseado no trabalho de Romão and Santos (2010), cujo objetivo é minimizar a carga morta dos veículos, que é a capacidade não utilizada, seja em relação ao volume ( $CMV_t$ ) ou peso ( $CMP_t$ ), ou qualquer combinação linear desses valores; pode-se ainda minimizar o número de veículos usados ( $NT$ ) no transporte.

Nós consideramos os seguintes conjuntos para o problema:

- $\mathcal{P} = \{p_1, p_2, \dots\}$ : tipo de produtos a serem alocados;
- $\mathcal{S} = \{OBT = \text{obrigatório}, N\_OBT = \text{não obrigatório}\}$ : possíveis status do produto;
- $\mathcal{T} = \{t_1, t_2, \dots\}$ : veículos disponíveis na companhia;

e os seguintes dados de entrada:

- $l_p, c_p, h_p, w_p$ : largura, comprimento, altura e peso do produto  $p, \forall p \in \mathcal{P}$ ;
- $ob_p, nob_p$ : quantidade de produtos do tipo  $p \in \mathcal{P}$  com entrega obrigatória e não obrigatória;
- $L_t, C_t, H_t, W_t$ : largura, comprimento, altura e peso suportado do veículo  $t, \forall t \in \mathcal{T}$ ;

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

- $v_p = l_p c_p h_p$ : volume do produto  $p, \forall p \in \mathcal{P}$ ;
- $V_t = L_t C_t H_t$ : volume do veículo  $t, \forall t \in \mathcal{T}$ ;
- $q_{pt} = \lfloor V_t / v_p \rfloor$ : quantidade máxima de produtos do tipo  $p$  suportado pelo veículo  $t$  em relação ao volume;
- $u_{pt} = \lfloor W_t / w_p \rfloor$ : quantidade máxima de produtos do tipo  $p$  suportado pelo veículo  $t$  em relação ao peso;

A formulação proposta usa o seguinte conjunto de variáveis:

- $x_{pst}$ : quantidade de produtos do tipo  $p$  com status  $s$  no veículo  $t$ ;
- $Z_t$ : variável binária indicando se o veículo  $t$  será usado na alocação dos produtos;
- $CMV_t, CMP_t$ : carga morta em relação ao volume e ao peso do veículo  $t$ ;
- $NT$ : número de veículos utilizados no carregamento.

Como dito anteriormente, o objetivo é minimizar a carga morta (não utilizada) dos veículos selecionados. Isto pode ser medido pelo volume ou peso, ou qualquer combinação linear deles. Pode-se também minimizar o número de veículos utilizados. A função objetivo genérica (1) pode ser usada para minimizar qualquer uma destas medidas dependendo do peso dos parâmetros  $\alpha, \beta, \gamma$ .

As restrições (2) e (3) garantem o limite de peso e volume, bem como definem a carga morta em relação a cada quesito. As restrições (2) garantem que a carga total de cada veículo mais a carga morta em relação ao peso corresponda o limite do mesmo. As restrições (3) tem o mesmo efeito, mas sobre o volume. As restrições (4) fortalecem as restrições anteriores, não permitindo que a quantidade selecionada de cada produto ultrapasse um limite superior inteiro, em relação ao peso e volume. As restrições (5) garantem que todos os produtos obrigatórios sejam alocados em algum veículo, e as restrições (6) limitam a quantidade de produtos não obrigatórios àquela já pedida pelo cliente. A restrição (7) conta o número de veículos usados. Finalmente, as restrições (8) e (9) definem o tipo das variáveis de decisão.

$$\min Z = \alpha \sum_{t \in \mathcal{T}} CMV_t + \beta \sum_{t \in \mathcal{T}} CMP_t + \gamma NT \quad (1)$$

$$\sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} w_p x_{pst} + CMP_t = W_t Z_t, \forall t \in \mathcal{T} \quad (2)$$

$$\sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} v_p x_{pst} + CMV_t = V_t Z_t, \forall t \in \mathcal{T} \quad (3)$$

$$\sum_{s \in \mathcal{S}} x_{pst} \leq \min(u_{pt}, v_{pt}), \forall p \in \mathcal{P}, \forall t \in \mathcal{T} \quad (4)$$

$$\sum_{t \in \mathcal{T}} x_{pst} = ob_p, \forall p \in \mathcal{P}, s = OBT \quad (5)$$

$$\sum_{t \in \mathcal{T}} x_{pst} \leq nob_p, \forall p \in \mathcal{P}, s = N\_OBT \quad (6)$$

$$\sum_{t \in \mathcal{T}} Z_t = NT \quad (7)$$

$$x_{pst} \in \mathbb{Z}^+, \forall p \in \mathcal{P}, \forall s \in \mathcal{S}, \forall t \in \mathcal{T} \quad (8)$$

$$Z_t \in \{0, 1\}, \forall t \in \mathcal{T} \quad (9)$$

### 3.2. Módulo de posicionamento dos produtos selecionados

Este módulo determina a posição onde cada produto selecionado deve ficar dentro do devido veículo. Essa segunda fase é resolvida por um método construtivo combinado com algoritmo genético: dada uma fila de produtos, o método construtivo tenta encontrar uma posição para cada produto, na ordem que estão presentes na fila. A principal dificuldade nesse caso é encontrar a melhor fila. Para isso, foi proposto um algoritmo genético que se encarrega de encontrar uma boa ordem em que os produtos devem ser colocados no interior do veículo.

Cada produto possui seis faces, mas para efeito de posicionamento no veículo apenas três são distintas, pois as faces opostas são idênticas. Cada uma dessas três faces pode ser rotacionada ortogonalmente para obter uma nova configuração.

#### 3.2.1. Algoritmo Genético

Os Algoritmos Genéticos, AGs, são métodos de otimização e busca inspirados nos mecanismos de seleção natural e sobrevivência dos mais aptos. No AG existe uma população de indivíduos, cada um sendo representado como os genes de um cromossomo, representando uma possível solução do problema. Cada cromossomo é avaliado para gerar um valor que é usado para serem comparados com os outros cromossomos. Esses valores representam a adaptação (*fitness*) de cada indivíduo e os melhores tem mais chances de se reproduzirem para gerar novos indivíduos com características semelhantes. Geralmente dois indivíduos são selecionados e seus genes são combinados gerando dois novos indivíduos. Alguns cromossomos podem sofrer mutação, que modifica alguns de seus genes, introduzindo novas características à população. Apenas os melhores indivíduos são selecionados para se reproduzirem e gerar novos indivíduos para a próxima população, e assim eles tendem a terem valores cada vez mais próximo da solução ótima.

Neste trabalho, cada gene representa um produto a ser empacotado. Um cromossomo codifica a ordem em que os produtos serão colocados no interior do veículo. Como cada gene representa um produto, a quantidade de vezes que cada produto aparece no cromossomo corresponde à quantidade daquele produto no carregamento. Desta forma, todos os cromossomos da população de um veículo tem o mesmo tamanho, que é a soma das quantidades de cada produto, como pode ser visto na Figura 1, que ilustra exemplos de representação de cromossomo para os dados de carregamento da Tabela 1.

Produto	Quantidade
1	3
2	5
3	4

Tabela 1. Exemplo de carregamento

C1	1	1	1	2	2	2	2	2	3	3	3	3
C2	2	2	2	2	2	3	3	3	3	1	1	1
C3	3	3	1	2	2	2	1	3	1	3	2	2

Figura 1. Exemplo de representação: diferentes cromossomos para os dados da Tabela 1

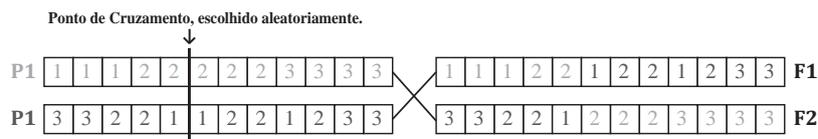
A população inicial contém no mínimo 15 cromossomos, formados de acordo com os 5 critérios abaixo, cada um gerando 3 cromossomos:

- Ordem decrescente de volume de cada produto;
- Ordem crescente de volume de cada produto;
- Ordem decrescente de área do produto;
- Ordem decrescente de quantidade a ser empacotada de cada produto;
- Ordem decrescente de volume total, ou seja, volume vezes a quantidade.

O restante da população é gerado de forma aleatória. A população é modificada pelos operadores de *seleção*, *crossover* e *mutação* formando sucessivas populações. No final, o melhor cromossomo gerado é retornado.

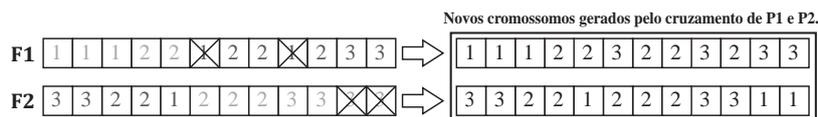
Na fase de seleção é usado o método de torneio ternário. São escolhidos três indivíduos (cromossomos) aleatoriamente e o melhor é adicionado na população na qual será aplicado o crossover. O melhor é escolhido de acordo com sua adaptação (*fitness*) que é a porcentagem do volume ocupado pelos produtos ou o número de produtos que o algoritmo construtivo consegue alocar no veículo.

Os cromossomos da nova população, escolhidos pela fase de seleção, são recombinados (*crossover*). Nessa fase são escolhidos dois cromossomos aleatoriamente e eles são recombinados a partir de uma posição aleatória. Assim, temos dois novos indivíduos, que são adicionados a uma nova população. A Figura 2 ilustra essa fase.



**Figura 2. Exemplo de cruzamento entre dois cromossomos**

Como dito anteriormente, a quantidade de vezes que cada produto aparece no cromossomo deve ser igual à quantidade de cada produto no carregamento. De acordo com a Tabela 1, a Figura 2 exemplifica a geração de dois filhos que não atendem a essa restrição. No filho 1 (F1) o produto do tipo 1 aparece cinco vezes, quando deveria aparecer três vezes e o produto 3 aparece duas vezes ao invés de quatro. Já no filho 2 (F2) o produto do tipo 3 aparece seis vezes quando deveria aparecer quatro vezes e o produto 1 aparece uma vez ao invés de três. Para corrigir esse problema, após ser feito o cruzamento, é verificado se algum produto excedeu ou não atingiu a quantidade exigida no carregamento, caso exista, os produtos que excederam a quantidade são removidos e os que não atingiram a quantidade são adicionados na posição dos que foram removidos. A Figura 3 ilustra a correção do cruzamento anterior.



**Figura 3. Correção do cromossomo após o crossover**

Por fim, o operador de mutação é aplicado com uma determinada probabilidade. Ele simplesmente troca a posição de quatro genes do cromossomo. Dessa forma aumenta-se a variabilidade genética na população e impede-se que a busca fique estagnada em um mínimo local. A Figura 4 ilustra essa fase.



Figura 4. Exemplo da mutação em um cromossomo

### 3.2.2. Cálculo do posicionamento

Para determinar a posição dos produtos, cada veículo é representado por uma matriz bidimensional com as dimensões da base do veículo. Cada célula da matriz representa  $1u.m.^2$  (unidade de medida ao quadrado), inicialmente preenchida com 0. Quando um produto  $p$  de dimensões  $(c, l, h)$  é adicionado na posição  $(i, j)$  todas as células do “retângulo” de vértices  $(i, j)$  e  $(i + c, j + l)$  recebem a altura  $h$  do produto, indicando que esta área está ocupada até a altura  $h$ . Quando um produto  $p'$  de dimensões  $(c', l', h')$  é adicionado sobre o produto  $p$ , todas as células da área de  $p'$ , na matriz, recebem  $h + h'$ . O produto pode ser posicionado apenas se  $(h + h') \leq H$ .

Os produtos são colocado na ordem codificada no cromossomo. Para cada opção de rotação, tenta-se todas as posições, a partir do canto do veículo, inclusive sobre outros produtos, desde que não sobre espaço vago sob o produto. Ele é colocado na primeira posição possível. Produtos não posicionados neste primeiro momento são tentados novamente, agora permitindo espaço vago sob o produto, desde que não seja superior à metade da área da base, por questão de equilíbrio.

A Figura 5 apresenta um exemplo da estrutura usada e uma lista de produtos a serem posicionados. Dado um veículo  $V(10, 7, 5)$ , inicialmente a matriz está com todas as células zeradas, nenhum produto foi adicionado. Ao ser adicionado o produto  $P_1(5, 5, 3)$  a estrutura da matriz será modificada, como mostra a Figura 6. Se for adicionado o produto  $P_2(2, 3, 5)$ , este deverá ficar em uma área cujas células estejam zeradas, para não ultrapassar a altura do veículo (Figura 7, matriz mais à esquerda). Ao se adicionar o produto  $P_3(4, 4, 2)$ , este poderá ficar em cima do produto  $P_1$ , pois suas alturas somadas não resultam em altura maior que a altura do veículo, como ilustrado pela matriz à direita na Figura 7. Por fim, é adicionado o produto  $P_4(5, 5, 5)$  no veículo, conforme mostra a Figura 8, que exhibe também a solução de forma 3D.

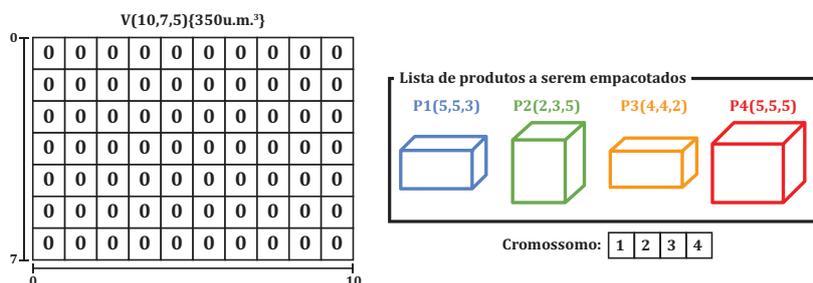


Figura 5. Exemplo de representação por matriz e uma lista de produtos do carregamento

Nesse exemplo, a função de adaptação (*fitness*) do Algoritmo Genético será 74,9%, se for em relação ao volume ocupado no veículo, ou 100%, se for em relação ao número de produtos alocados no veículo.

### 3.3. Interação entre os dois módulos

Como dito anteriormente, a solução do módulo de seleção respeita o volume do veículo, mesmo quando não é esse o critério de otimização. Um problema é que mesmo assim

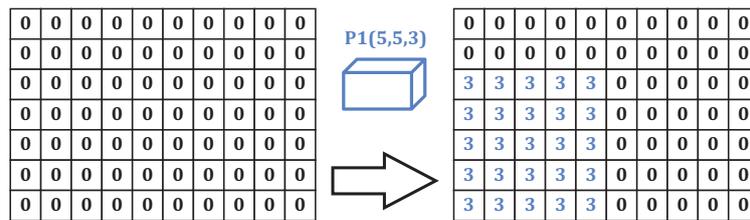


Figura 6. Adicionando o primeiro produto

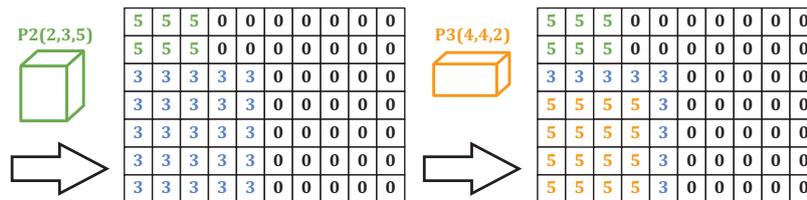


Figura 7. Adicionando o segundo produto (à esquerda), que foi girado, e adicionando o terceiro produto, que pode ficar em cima de outro produto (P1)

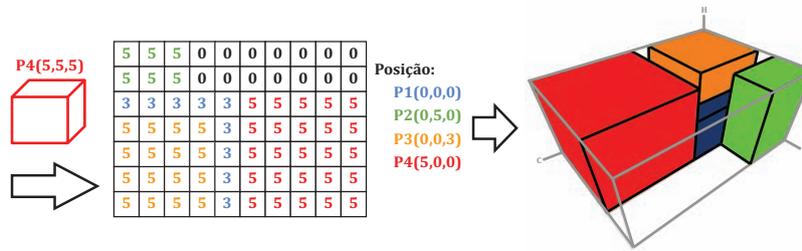


Figura 8. Adicionando o ultimo produto e exibindo, em 3D, a solução encontrada

ela pode ser inviável no módulo de posicionamento. Considere, por exemplo, dois veículos,  $V_1(4dm, 4dm, 5dm)\{80dm^3\}$  e  $V_2(4dm, 5dm, 5dm)\{100dm^3\}$ , e suponha dois produtos a serem carregados  $p_1(3dm, 3dm, 5dm)\{45dm^3\}$  e  $p_2(2dm, 2dm, 5dm)\{20dm^3\}$ , e que os dois veículos suportem o peso dos produtos. Pelo total de volume, ambos podem ser empacotados em qualquer um dos veículos. Nesse caso, o módulo de seleção selecionará o veículo  $V_1$ , que tem menor volume, tendo, portanto, menor carga morta em relação ao volume. No entanto, a geometria dos produtos não permite carregá-los neste veículo.

A solução proposta para contornar esse problema é reduzir ligeiramente uma ou mais dimensões dos veículos anteriormente selecionados. Isso pode forçar uma redistribuição dos produtos nos veículos. No exemplo anterior, se for reduzido 1dm no comprimento de  $V_1$ , seu volume será de  $60dm^3$ . Como a soma dos volumes dos produtos é  $65dm^3$ , o módulo de seleção, obrigatoriamente, selecionará  $V_2$  para acomodar os produtos. Vale ressaltar que essa redução só é válida no módulo de seleção, ou seja, o módulo de posicionamento considera as dimensões originais dos veículos.

O diagrama da Figura 9 ilustra essa interação entre os dois módulos.

#### 4. Resultados

Os algoritmos foram implementados na linguagem C++ usando como compilador o Microsoft Visual Studio C++ 2010, e os testes executados em uma máquina Intel i7 Quad Core 2.93GHz, com 4GB de RAM e sistema operacional Windows 7 64bits. Os modelos de programação linear foram resolvidos com chamadas de funções do *Xpress-MP* (FICO, 2009). Não foram encontradas instâncias na literatura com as características des-

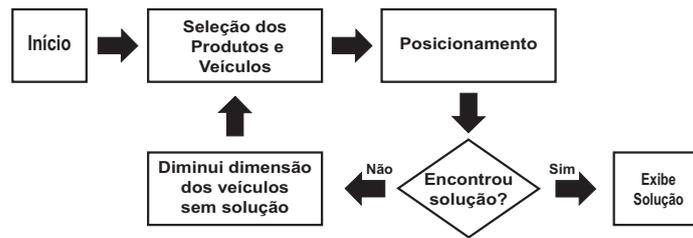


Figura 9. Diagrama de interação entre os dois módulos

critas nesse trabalho, principalmente em relação à distinção entre produtos obrigatórios e não obrigatórios, então foram geradas várias instâncias, com diferentes tamanhos e características, para testar os algoritmos. Entretanto, a modelagem é geral o suficiente para tratar diversas instâncias da literatura, então foram realizados testes e comparações com outros tipos de instâncias. Apresentamos os resultados para dois tipos, aqui denominados: *Classe BR1–BR7* (Bischoff and Ratcliff, 1995a) e *Classe IMM* (Ivancic et al., 1989).

Como o cálculo do posicionamento pode ser muito demorado, foi estipulado um tempo máximo de 20min para a execução do Algoritmo Genético ou 30 iterações. Além disso, se mais de 5 iterações não apresentarem melhora o algoritmo é interrompido. Após alguns testes, foi concluído que uma taxa de mutação bem alta, em torno de 75%, gera soluções melhores, isso se dá ao fato dele aumentar a variabilidade genética na população. Assim, a taxa de mutação foi definida como 75% e o tamanho da população como 25.

#### 4.1. Instâncias geradas

Foram criadas instâncias com 10 tipos de produtos, com diferentes dimensões e peso, todos podendo ser rotacionados em todas as seis possíveis rotações; e 5 tipos de veículos, com diferentes capacidades, baseadas em veículos reais utilizados nas transportadoras brasileiras. As instâncias foram divididas em quatro classes, que se diferenciam pela quantidade de produtos obrigatórios ( $ob_p$ ) e não obrigatórios ( $nob_p$ ) de cada produto. Os dados dos produtos e veículos são mostrados nas Tabela 2 e 3, respectivamente.

Tabela 2. Dados dos produtos e das classes de instâncias

$ID_p$	Dimensões (dm)			Peso (kg)	Classe 1		Classe 2		Classe 3		Classe 4	
	$c_p$	$l_p$	$h_p$		$w_p$	$ob_p$	$nob_p$	$ob_p$	$nob_p$	$ob_p$	$nob_p$	$ob_p$
1	3	5	11	85	37	0	45	5	67	19	88	34
2	12	4	4	65	45	0	53	12	72	23	109	30
3	7	4	10	140	31	0	40	7	53	21	81	38
4	4	13	7	295	39	0	43	3	58	13	85	34
5	6	8	11	110	28	0	30	9	39	22	83	43
6	6	9	12	225	17	0	23	5	42	18	97	36
7	5	15	9	75	19	0	15	0	38	17	107	33
8	12	13	8	145	25	0	25	4	40	12	94	37
9	10	15	10	175	23	0	29	3	45	16	89	19
10	10	14	11	235	14	0	21	10	48	15	117	26
<b>Total:</b>					278	0	324	58	502	176	950	330

A Tabela 4 mostra o resultado para todas as classes para alguns critérios de otimização. A primeira coluna refere-se à classe dos produtos; a segunda ao critério de otimização, que pode ser minimizar a carga morta em relação ao volume ( $CMV$ ), em relação ao peso ( $CMP$ ), em relação ao volume e peso ( $CMV + CMP$ ) e o número de veículos utilizados ( $NV$ ); a terceira coluna indica as ID's dos veículos utilizados; a quarta a quantidade de produtos empacotadas. Observe que essa quantidade deve ser, no mínimo, a quantidade total de produtos obrigatórios para cada classe; a quinta e a sexta coluna indicam a quantidade de

**Tabela 3. Dados dos veículos disponíveis**

$ID_t$	Dimensões (dm)			Quantidade	Peso Suportado (kg)
	$C_t$	$L_t$	$H_t$		$W_t$
1	76	24	40	3	14000
2	131	24	30	3	27000
3	123	24	40	3	31000
4	148	24	30	3	41000
5	60	30	30	3	6000

carga morta, respectivamente, em relação ao volume (em  $dm^3$ ) e peso (em kg); a sétima e a oitava representam a porcentagem de carga morta, também em relação ao volume e peso; por fim, a última coluna exibe o tempo gasto, em segundos, para se resolver o problema.

Analisando-se os resultados encontrados da Tabela 4, nota-se que o algoritmo encontrou solução para todas as instâncias testadas. Em alguns casos encontrou-se uma solução ótima para o critério de otimização, que é o caso da classe 2 e 3 quando são otimizadas a carga morta em relação ao peso (*CMP*). Podemos observar que se obteve o melhor resultado para cada critério de otimização em cada classe. Além disso, quando minimizamos o número de veículos (*NV*) apenas produtos obrigatórios são selecionados para fazerem parte do empacotamento, já que nesse caso não há interesse na carga morta.

**Tabela 4. Solução encontrada para cada classe sob diferentes critérios de otimização**

Classe	Crit. Otim.	Selecionados		Carga Morta		Cap. Não Usada		Tempo(s)	
		ID Veículos	#Prod.	CMV	CMP	Volume	Peso		
1	CMV		2, 2	278	25134	12815	13,32%	23,73%	122,30
	CMP		1, 1, 1	278	55374	815	25,30%	1,94%	145,06
	CMV + CMP		1, 3	278	27534	3815	14,41%	8,48%	184,38
	NV		3, 3	278	72654	20815	30,76%	33,57%	296,37
2	CMV		3, 2	333	23342	21765	10,39%	30,23%	330,72
	CMP		1, 3, 5	338	43479	0	17,74%	0,00%	64,58
	CMV + CMP		1, 1, 2	366	27417	500	11,41%	0,91%	284,51
	NV		3, 3	324	43798	13210	18,55%	21,31%	269,08
3	CMV		2, 4, 4, 5	507	34871	38070	9,65%	33,10%	276,63
	CMP		1, 1, 1, 2, 5, 5, 5	578	104495	0	21,99%	0,00%	412,03
	CMV + CMP		3, 3, 5, 5, 5	529	58357	415	14,66%	0,52%	317,19
	NV		3, 3, 3, 4	502	135171	57550	29,33%	42,95%	229,18
4	CMV		1, 2, 2, 3, 3, 4, 4, 5, 5, 5	960	169100	82085	19,37%	35,69%	546,61
	CMP		1, 1, 1, 2, 2, 3, 3, 3, 5, 5, 5	1064	190985	40300	20,67%	19,47%	826,32
	CMV + CMP		1, 1, 1, 2, 3, 3, 3, 5, 5, 5	967	124459	28555	15,01%	15,86%	404,09
	NV		1, 2, 2, 3, 3, 3, 4, 4, 4	950	236555	137340	25,29%	48,36%	377,24

#### 4.2. Classe BR1-BR7

Este grupo contém 700 instâncias que são subdivididas em 7 classes contendo 100 cada uma, nomeadas de “BR1” a “BR7”, do Problema de Carregamento de Contêiner disponíveis na *OR-Library* (Beasley) e definidas por Bischoff and Ratcliff (1995a), cujo objetivo é descobrir o volume máximo que um conjunto de produtos pode ocupar em um único veículo (contêiner). Estas são resolvidas usando apenas o módulo de posicionamento, considerando que todos os produtos são obrigatórios, assim podemos comprovar a eficiência desse módulo. Essas instâncias já foram usadas entre outros por Bischoff and Ratcliff (1995a), Bischoff and Ratcliff (1995b), Gehring and Bortfeldt (1997) e Bortfeldt and Gehring (1997). O gráfico da Figura 10 compara os resultados obtidos por todos os métodos, e mostra o bom desempenho do algoritmo proposto, que é superior a três dos métodos, competitivo com os outros dois, e em média encontra o segundo melhor conjunto de soluções.

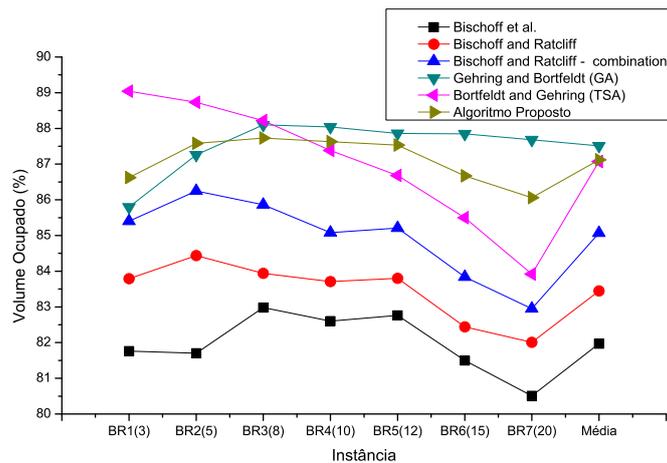


Figura 10. Gráfico do resultado para as instâncias de Bischoff and Ratcliff (1995a)

### 4.3. Classe IMM

Este grupo possui instâncias do Problema de Empacotamento Tridimensional que também podem ser encontradas na *OR-Library* (Beasley) e definidas por Ivancic et al. (1989), cujo objetivo é descobrir o número mínimo de veículos para se empacotar todos os produtos de um carregamento. Tais instâncias foram usadas entre outros por Ivancic et al. (1989), Bischoff and Ratcliff (1995a) e Bischoff and Ratcliff (1995b). Os resultados encontrados são mostrados na Tabela 5. Note que o algoritmo aqui proposto (AP) obteve melhor desempenho que um dos métodos, e só não obteve melhor desempenho que os demais por causa da instância 14. Além disso, encontrou novas soluções para as instâncias 1, 2, 5 e 12, utilizando menos veículos que qualquer um dos outros métodos.

Tabela 5. Quantidade de veículos das soluções obtidas para as instâncias da Classe IMM

Inst.	#Prod.	Ivancic et al. (1989)	Bischoff & Ratcliff (1995a)	Bischoff & Ratcliff (1995b)	AP	Inst.	#Prod.	Ivancic et al. (1989)	Bischoff & Ratcliff (1995a)	Bischoff & Ratcliff (1995b)	AP
1	70	26	27	27	25	25	72	6	5	5	6
2	70	11	11	11	10	26	72	3	3	4	3
3	180	20	21	26	22	27	95	5	5	5	5
4	180	27	29	27	30	28	95	10	11	12	11
5	180	65	61	59	51	29	118	18	17	23	21
6	103	10	10	10	10	30	118	24	24	26	29
7	103	16	16	16	16	31	118	13	13	14	14
8	103	5	4	4	4	32	90	5	4	4	4
9	110	19	19	19	19	33	90	5	5	5	5
10	110	55	55	55	55	34	90	9	9	8	9
11	110	18	19	25	19	35	84	3	3	3	3
12	95	55	55	55	53	36	84	18	19	14	14
13	95	27	25	27	25	37	102	26	27	23	23
14	95	28	27	28	40	38	102	50	56	45	45
15	95	11	11	15	12	39	102	16	16	18	16
16	95	34	28	29	28	40	85	9	10	11	10
17	95	8	8	10	9	41	85	16	16	17	24
18	47	3	3	2	2	42	90	4	5	5	5
19	47	3	3	3	3	43	90	3	3	3	3
20	47	5	5	5	6	44	90	4	4	4	4
21	95	24	24	26	26	45	99	3	3	3	3
22	95	10	11	11	10	46	99	2	2	2	2
23	95	21	22	22	24	47	99	4	3	4	4
24	72	6	6	7	6	<b>Total</b>		<b>763</b>	<b>763</b>	<b>777</b>	<b>768</b>

## 5. Conclusão

Este trabalho apresenta um método para se resolver um problema de empacotamento tridimensional com frota heterogênea, minimizando a carga morta total ou o número de veículos, considerando produtos obrigatórios e também não obrigatórios, que podem compor a carga. O problema é dividido em dois módulos, seleção e posicionamento, resolvidos respectivamente por programação matemática e algoritmo genético, de forma interativa. Além de instâncias específicas do problema, foi testado com várias instâncias da literatura, sem produtos opcionais e veículos homogêneos, apresentando bons resultados tanto para um problema de carregamento de contêiner quanto para outro de empacotamento tridimensional, conseguindo inclusive soluções com menor número total de veículos.

**Agradecimentos.** Trabalho financiado pelo CNPq e parcialmente por Gapso e Sydle.

## Referências

- Baltacioglu, E., Moore, J. T., and Hill Jr., R. R.** (2006). The distributor's three-dimensional pallet-packing problem: a human intelligence-based heuristic approach. *International Journal of Operational Research*, 1(3):249–266.
- Beasley, J. E.** OR-Library. <http://people.brunel.ac.uk/mastjbj/jeb/orlib/thpackinfo.html>.
- Bischoff, E. E. and Ratcliff, M. S. W.** (1995a). Issues in the development of approaches to container loading. *Omega*, 23(4):377–390.
- Bischoff, E. E. and Ratcliff, M. S. W.** (1995b). Loading multiple pallets. *Journal of the Operational Research Society*, 46(11):1322–1336.
- Bortfeldt, A. and Gehring, H.** (1997). Applying tabu search to container loading problems. *Operations Research Proceedings*, 533–538.
- Eley, M.** (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141(2):393–409.
- FICO, X. O. S.** (2009). XPRESS-BCL Reference Manual.
- Gehring, H. and Bortfeldt, A.** (1997). A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research*, 4:401–418.
- Gehring, H., Menschner, K., and Meyer, M.** (1990). A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research*, 44(2):277–288.
- George, J. A. and Robinson, D. F.** (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3):147–156.
- Ivancic, N., Mathur, K., and Mohanty, B. B.** (1989). An integer-programming based heuristic approach to the three-dimensional packing problem. *Journal of Manufacturing and Operations Management*, 2:268–298.
- Lim, A., Rodrigues, B., and Yang, Y. P.** (2005). 3-D Container Packing Heuristics. *Applied Intelligence*, 22(2):125–134.
- Martello, S., Pisinger, D., and Vigo, D.** (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2):256–267.
- Pisinger, D.** (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2):382–392.
- Romão, O. C. and Santos, A. G. d.** (2010). Apoio à decisão no carregamento de veículos: seleção e posicionamento de produtos. *XLII SBPO - Simpósio Brasileiro de Pesquisa Operacional*. Bento Gonçalves - RS, 1353–1364.
- Silva, P. d. T. G. C. and Mateus, G. R.** (2009). Um modelo de programação inteira mista para uma especialização do problema de empacotamento tridimensional. *XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional*. Porto Seguro - BA, 1302–1313.