

# Uma busca tabu aplicada ao problema de roteamento com restrições de empacotamento tridimensionais

Gabriel M. Portal<sup>1</sup>, Rafael Rocco<sup>2</sup>, Marcus Ritt<sup>1</sup> e Luciana S. Buriol<sup>1</sup>

<sup>1</sup>*Instituto de Informática – Universidade Federal do Rio Grande do Sul*

<sup>2</sup>*Engenharia de Produção e Transportes – Universidade Federal do Rio Grande do Sul*

{gmportal, marcus.ritt, buriol}@inf.ufrgs.br, rroco@producao.ufrgs.br

## Resumo

Este artigo apresenta um algoritmo de busca tabu para o problema de roteamento de veículos capacitados com restrições de empacotamento tridimensionais (*3D-Loading Capacitated Vehicle Routing Problem – 3L-CVRP*). Existem diversas propostas na literatura para solucionar o empacotamento em duas e três dimensões, assim como diversas voltadas para o roteamento de veículos. Abordar estes dois problemas conjuntamente começou a ser explorada recentemente. Mostramos que o problema pode ser resolvido de forma eficiente por uma meta-heurística busca tabu. Este método é uma adaptação de uma abordagem já sugerida na literatura. Experimentos computacionais mostram que o algoritmo implementado foi capaz de resolver de forma eficiente as instâncias testadas.

**Palavras-chave:** Busca tabu, roteamento, empacotamento.

## Abstract

This paper presents a Tabu Search algorithm to the Capacitated Vehicle Routing Problem with 3D-Loading constraints (3L-CVRP). There are several approaches in the literature to solve the packing in two and three dimensions, as well as several others to solve the Vehicle Routing problem. However, approaching these two problems together has been explored only recently. We show how this problem can be solved efficiently by a Tabu Search approach. This method is an adaption of an algorithm already suggested in the literature. Computational experiments show that the proposed algorithm was able to find good quality solutions for the tested instances.

**Keywords:** Tabu Search, routing, packing.

# 1 Introdução

Problemas de otimização combinatória surgem naturalmente nas mais comuns tarefas do cotidiano. Por exemplo, definir o empacotamento das malas no porta-malas de um carro, ou escolher a rota a ser percorrida quando deve-se visitar diversos lugares. Empresas de transporte deparam-se diariamente com instâncias complexas de problemas de otimização: é preciso encontrar o melhor roteamento para seus veículos ao mesmo tempo em que a carga a ser entregue deve ser acomodada no respectivo veículo. Além disso, em situações reais, existem diversas restrições que devem ser respeitadas: fragilidade e superfície de apoio da carga, ordem de retirada dos itens, centro de massa do veículo, veículos com características diferentes, janela de tempo para as entregas, cargas de formatos diferenciados, entre tantas outras peculiaridades do problema que podem surgir.

O problema em questão aborda dois subproblemas conhecidos na literatura: o roteamento de veículos e o empacotamento de cargas. Ambos tratados separadamente já são difíceis de serem resolvidos no caso geral, visto que pertencem à classe de problemas NP-difíceis. Juntos, eles apresentam uma complexidade ainda maior. Devido a estes fatores apenas recentemente surgiram trabalhos tratando deste problema e, ainda, modelando problemas reais de forma simplificada.

Esta proposta com a resolução combinada dos dois problemas foi inicialmente proposto por Gendreau et al. (2006). Recentemente, este mesmo problema foi abordado por Moura and Oliveira (2009), que fez um estudo bastante completo sobre a abordagem integrada do roteamento e empacotamento. Para uma descrição detalhada sobre o problema, com revisão bibliográfica completa sobre o tema, pode-se referenciar de Araújo (2006).

Este trabalho apresenta uma abordagem para resolver o problema 3L-CVRP (*3D-Loading Capacitated Vehicle Routing Problem*). O objetivo deste problema é distribuir de forma eficiente um conjunto de itens (ou cargas) para seus respectivos clientes. Toda carga está concentrada em um centro de triagem, e deve-se utilizar um conjunto de veículos para entregá-las. Deve-se respeitar restrições de fragilidade e superfície de apoio das cargas, a carga deve poder ser retirada do caminhão no momento da entrega sem necessidade de reorganização das outras cargas. Além disso, todos caminhões são homogêneos e as cargas têm um formato de paralelepípedo. O objetivo é minimizar a distância total percorrida pelos veículos.

Na próxima seção o problema é definido formalmente. A Seção 3 apresenta um algoritmo de Busca Tabu para resolver o problema. Os resultados computacionais com instâncias da literatura são relatados na Seção 4. A Seção 5 conclui e discute trabalhos futuros.

## 2 Definição do Problema

Considere um grafo não-direcionado  $G = (V, A)$ , capacitado e completo. Sem perda de generalidade, supomos que  $V = \{v_0, v_1, \dots, v_n\}$ , com o centro de triagem  $v_0$  e clientes  $v_i$ ,  $i = 1, \dots, n$ . Cada aresta  $a_{ij} \in A$  tem um custo de transporte  $d_{ij}$  associado (e.g. tempo, distância) entre  $v_i$  e  $v_j$ .

Temos uma frota homogênea de  $b$  veículos disponíveis.  $P$  representa o peso total suportado por um veículo.  $L$  representa a largura,  $H$  representa a altura, e  $C$  representa o comprimento de um veículo.

Cada cliente  $i$  tem uma demanda de  $k_i$  itens. Cada item pode ser identificado como  $M_{i,k}$  em que  $i = 1, \dots, n, k = 1, \dots, k_i$  e possui largura  $l_{i,k}$ , altura  $h_{i,k}$  e comprimento  $c_{i,k}$ . Cada item possui ainda uma indicação de fragilidade  $f_{i,k} \in \{0, 1\}$  com valor 1 se o item for frágil e 0 caso contrário. A soma do peso de todos itens do cliente  $i$  é dado por  $t_i$ .

Uma rota  $r$  é um ciclo sem repetição  $v_{r_0}, v_{r_1}, \dots, v_{r_k}, v_{r_0}$  em  $G$ , com  $v_{r_0} = v_0$ . O custo de uma rota  $r$  é definido por  $d(r) = \sum_{0 \leq i < k} d_{r_i, r_{i+1}} + d_{r_k, r_0}$ . O objetivo do 3L-CVRP consiste em encontrar no máximo  $b$  rotas factíveis disjuntas (com exceção do centro de triagem  $v_0$ ) que

contém todos clientes e que minimizam o custo total. Isto é, deve-se encontrar no máximo uma rota para cada veículo disponível.

Uma rota é factível se existe um empacotamento da demanda dos clientes desta rota que satisfaz

- i) uma restrição da dimensão e peso do veículo,
- ii) uma restrição de fragilidade,
- iii) uma restrição de superfície de apoio,
- iv) uma restrição de ordem de descarregamento (LIFO), e
- v) restrições de orientação do itens.

A restrição (i) garante que os itens de todos os clientes caibam nas dimensões do veículo, sem exceder o peso suportado  $P$ . A restrição de fragilidade exige que itens não-frágeis não podem ser posicionados em cima de itens frágeis. A restrição (iii) garante a estabilidade do carregamento. Sendo  $s \in [0, 1]$  a fração mínima permitida de superfície de apoio, para cada item  $M_{i,k}$  uma área equivalente a  $s l_{i,k} c_{i,k}$  deve estar apoiada em outro item, ou no chão do veículo. Em relação a restrição LIFO (iv), quando o veículo atender um cliente, todos os itens daquele cliente devem ser retirados do caminhão sem necessitar uma reorganização dos itens no interior do veículo. Isso implica que o item de um cliente que será visitado depois na rota não poderá ser colocado nem em cima, e nem na frente do item de um cliente que será visitado anteriormente. Por fim, há uma restrição de orientação em que cada item poderá apenas ser rotacionado horizontalmente, logo, a superfície de cima de um item deverá sempre ser mantida para cima.

### 3 Uma Busca Tabu para o 3L-CVRP

Nossa abordagem consiste em realizar empacotamento e roteamento em conjunto e baseia-se fortemente na proposta de Gendreau et al. (2006). Inicialmente é realizada uma busca local responsável pelo empacotamento de um conjunto de itens em um veículo, respeitando as restrições de empacotamento apresentadas na seção anterior. No escopo do roteamento, é realizada uma busca tabu com o objetivo de minimizar o custo total das rotas. Nessa seção, será detalhada a abordagem proposta. Primeiramente, a estrutura de dados utilizada para representar cada veículo será apresentada. Em seguida, o foco será a busca local utilizada no procedimento de empacotamento de itens. Para finalizar, a busca tabu externa, responsável pela definição das rotas, será apresentada em detalhes.

#### 3.1 Representação de um empacotamento

O procedimento de empacotamento é realizado muitas vezes e, por isso, é importante escolher uma estrutura de dados eficiente para a representação interna dos veículos. Essa estrutura de dados deve apoiar a inserção e deleção de itens, e a consulta dos itens adjacentes a um dado item. Uma representação direta do veículo como uma matriz tridimensional com células de tamanho unitário não é adequada, visto que, tanto a complexidade de atualização e consulta, quanto a complexidade de memória são dispendiosas (cúbicas no tamanho dos itens).

A estrutura de dados utilizada é uma matriz dinâmica (Ngoi et al., 1994). Esta congrega duas idéias principais: uma representação do veículo em apenas duas dimensões e o uso do menor número possível de células em cada dimensão da matriz. Cada item é representado pelas células ocupadas. Inicialmente, uma única célula representa todo o espaço livre do veículo. Na medida em que novos itens são inseridos, células são subdivididas como necessário. A matriz bidimensional ainda armazena, para cada uma de suas células, a altura correspondente. Armazena-se, complementarmente, duas listas com o tamanho das células em cada uma das dimensões.

A Figura 1 mostra um exemplo de utilização dessa estrutura. Dado um veículo vazio de dimensões  $L = 25$ ,  $H = 30$  e  $C = 60$ , a estrutura inicial é uma matriz 1x1 representando todo o veículo (Figura 1(a)). O número em cada célula indica a altura ocupada do veículo naquela porção do caminhão. Após a inserção de um elemento  $e_1$  com dimensões  $l_1 = 10$ ,  $h_1 = 20$  e  $c_1 = 25$  na posição  $p_1 = (0, 0, 0)$  ocorrerá uma modificação como mostrado na Figura 1(b). O veículo é subdividido em quatro partes retangulares: a primeira, contendo o elemento  $e_1$ , com altura 20. As outras três mantêm-se com altura 0. Inserindo-se, então, um elemento  $e_2$  com dimensões  $l_2 = 8$ ,  $h_2 = 17$  e  $c_2 = 20$  na posição  $p_2 = (10, 0, 0)$  obtém-se o resultado na Figura 1(c). Percebe-se, com esta inserção, que o veículo ficou mais subdividido e que a única célula que correspondia à altura 20 dividiu-se em duas células com altura 20. Por fim, com a inserção de um elemento  $e_3$  com dimensões  $l_3 = 10$ ,  $h_3 = 13$  e  $c_3 = 20$  na posição  $p_3 = (10, 17, 0)$ , a estrutura é atualizada, como mostrado na Figura 1(d). Neste caso, uma das dimensões do elemento inserido corresponde ao tamanho da célula onde este foi inserido e, assim, apenas uma dimensão do veículo foi subdividida.

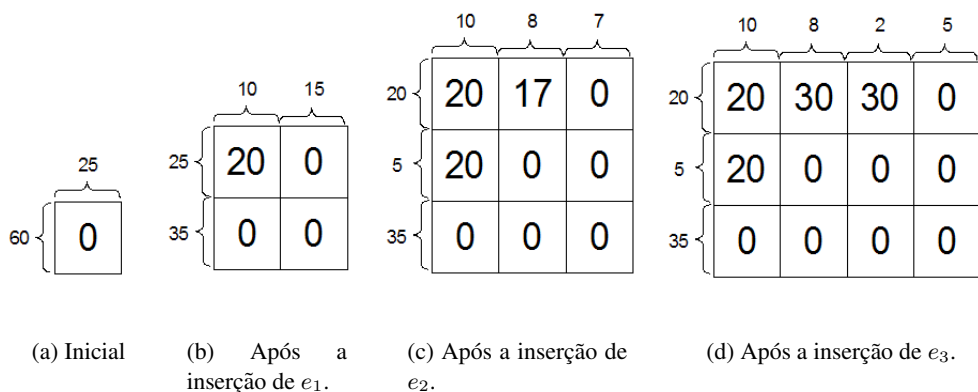


Figura 1: Representação do veículo em diferentes situações

Deve-se notar que, como tal estrutura é bidimensional, não existe informação completa em relação à terceira dimensão. Desta forma, como pode-se perceber com a inserção de  $e_3$ , se um item for inserido numa posição e ficar com uma pequena parte sem apoio, a porção de espaço abaixo da parte sem apoio ficará perdida, mesmo que esteja livre. Neste caso, um volume equivalente a 680 foi perdido devido a representação dessa forma. Isto acontece porque a representação é feita como uma visão de cima do veículo. Portanto não se sabe sobre áreas livres que estejam abaixo de itens posicionados. Sempre acontecerá alguma perda quando um item não estiver totalmente apoiado por outro (base de apoio menor que 100%). Este problema se agrava quanto menor a exigência da base de apoio dos itens. Para uma exigência total de base de apoio (o item deve estar totalmente apoiado), não há nenhuma perda de representação.

Contudo, um item deverá ter um apoio considerável e foi utilizado um valor de 75% de base de apoio. Além disso, as operações executadas na matriz dinâmica são realizadas de forma muito mais eficiente que as aplicadas à estrutura tridimensional inicial, o que foi decisivo para a sua escolha.

### 3.2 Empacotamento

Para obter uma solução inicial para o empacotamento, foi utilizada uma heurística construtiva. Dada uma sequência  $l$  de itens, a heurística fornece um empacotamento (não necessariamente viável) na ordem em que os itens são lidos, e retorna o valor deste empacotamento.

Considerando-se que o veículo esteja no plano cartesiano, cada item é inserido na posição com o menor valor da coordenada  $z$ . Em caso de empate, considera-se o menor valor na coorde-

nada  $x$ . Em caso de um segundo empate, o menor valor na coordenada  $y$ . Cada vez que um item não couber em uma dada posição, testa-se sua inserção rotacionando-o em 90 graus no plano  $x$ - $z$ . Dessa forma, em um veículo vazio, o primeiro item é sempre inserido na origem, conforme mostrado na Figura 2.

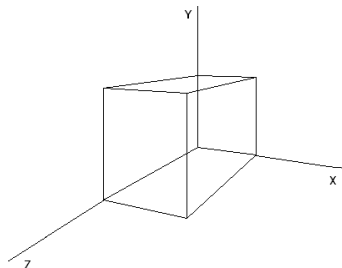


Figura 2: Representação do veículo com os eixos cartesianos.

Para o empacotamento, são consideradas a altura e a largura originais do veículo, mas com comprimento igual a duas vezes o seu comprimento. O valor utilizado para avaliar um certo empacotamento é o máximo valor utilizado na dimensão  $z$  (comprimento). Se não foi possível realizar o empacotamento, um valor infinito é atribuído a este empacotamento. Caso o empacotamento tenha sido feito com sucesso, armazena-se o possível excesso utilizado no comprimento do veículo, assim como o possível excesso de peso do mesmo. Estes valores serão utilizados para calcular a função objetivo que avalia a qualidade de uma solução. O Algoritmo 1 ilustra a heurística construtiva.

Dada uma lista ordenada de itens, estes são inseridos no veículo de acordo com a heurística acima explicada. Como esta é determinística, toda solução de empacotamento pode ser representada com esta lista. Contudo, a ordem dos itens nesta lista influencia bastante o resultado final.

---

#### Algorithm 1 Heurística Construtiva de Empacotamento

---

```

1: Inicializa  $maxX$  e  $maxZ$  com o tamanho da estrutura de dados nas dimensões  $x$  e  $z$ , respectivamente.
2: for cada item  $i$  da lista  $l$  do
3:   for  $j = 0 \dots maxZ$  do
4:     for  $k = 0 \dots maxX$  do
5:       if item  $i$  cabe na célula  $j, k$  then
6:         Modifica estrutura de dados para incluir item  $i$  na célula  $j, k$ .
7:       else
8:         Rotaciona item  $i$  90 graus no eixo  $y$ .
9:         if  $i$  cabe na célula  $j, k$  then
10:          Modifica estrutura de dados para incluir item  $i$  rotacionado na célula  $j, k$ .
11:        end if
12:      end if
13:    if  $i$  não puder ser empacotado then
14:      Desiste do empacotamento.
15:    end if
16:  end for
17: end for
18: end for
19: Retorna Ocupação máxima do veículo na dimensão  $z$ .

```

---

A solução inicial considerada foi a inserção dos itens de cada cliente na ordem inversa em que serão visitados. Muitas vezes, esta solução inicial resultará em um empacotamento infactível. Nesses casos, é aplicada uma busca local *first improvement* com o objetivo de factibilizar o empacotamento. A vizinhança consiste em todas soluções com trocas entre itens adjacentes na sequência atual.

Esta vizinhança foi escolhida principalmente devido à restrição LIFO do problema. Provavelmente não adiantará trocar itens do fundo do veículo com itens que estejam mais na frente, pois os itens mais da frente deverão ser entregues antes, enquanto os itens mais do fundo devem ser entregues aos últimos clientes da rota. Desta forma, trocas locais (entre itens adjacentes da lista) foi uma estratégia que gerou melhores resultados.

Foram realizados testes utilizando um algoritmo de busca tabu no empacotamento. Entretanto, o tempo utilizado para realizar cada empacotamento foi superior e o ganho correspondente não foi significativo. Desta forma, optamos por utilizar apenas o procedimento de busca local, visto que o mesmo é mais rápido e permite à busca do roteamento explorar uma área maior do espaço de soluções.

### 3.3 Roteamento

O roteamento consiste na definição de rotas que minimizem a distância percorrida pelos veículos. Para sua solução foi utilizado o método de busca tabu. Primeiramente, a heurística construtiva será apresentada. Em seguida, será detalhada a busca tabu e seus parâmetros.

#### 3.3.1 Solução Inicial

Obter uma boa solução inicial mostrou-se bastante importante no processo de alcançar uma melhor solução final para o problema. Foi observado que vale a pena utilizar mais tempo calculando uma solução inicial de boa qualidade e realizar as melhorias mais difíceis através da busca tabu.

Foram experimentadas diversas heurísticas para calcular a solução inicial, mas a técnica que gerou melhores resultados foi o algoritmo de economia (“savings”) de Clarke and Wright (1964), utilizada já em soluções do caixeiro viajante. A idéia é fazer inicialmente uma rota para cada cliente. Então, inicia-se um processo de unir rotas duas a duas enquanto a rota permanecer factível. Quando não for mais possível obter uma rota factível, permite-se a criação de rotas infactíveis (por não respeitar as restrições de comprimento ou capacidade do veículo), até alcançar o número de rotas igual ao número de veículos disponíveis. Para juntar duas rotas aplica-se uma heurística de vizinho mais próximo a todos clientes de cada uma das duas rotas para determinar a nova rota. Para determinar qual união de rotas realizar a cada vez, testa-se todas as combinações possíveis e escolhe-se aquela que formar uma rota combinada de menor comprimento.

#### 3.3.2 Busca Tabu

Após o cálculo de uma solução inicial, parte-se para uma busca tabu com o objetivo de melhorar esta solução. Dada uma solução, para cada cliente de cada rota tenta-se inseri-lo em todas as outras rotas. A primeira inserção que melhorar a solução torna-se a solução corrente. Caso nenhum movimento melhorar a solução corrente, a melhor solução vizinha é escolhida. Mantém-se uma lista Tabu com o inverso de todos movimentos feitos nas últimas iterações: se o cliente  $i$  passa da rota  $j$  para a rota  $k$  nas próximas iterações, é proibido mover o cliente  $i$  da rota  $k$  de volta para a rota  $j$ . Entretanto, aplicamos um critério de aspiração: caso um movimento tabu melhore a solução global obtida até o momento, este movimento é realizado.

A função objetivo para calcular a qualidade de uma solução é  $d + \alpha w + \beta l$ . O termo  $d$  consiste na soma das distâncias de cada rota;  $w$  representa a soma dos excessos de pesos de todos veículos e  $l$  representa a soma dos excessos no comprimento de todos veículos. Ainda temos os termos  $\alpha$  e  $\beta$  utilizados para penalizar uma solução infactível. Os valores dessas constantes são  $\alpha = 20 \bar{d}/P$  e  $\beta = 20 \bar{d}/C$ , em que  $\bar{d}$  é o peso médio de uma aresta do grafo.

Mesmo com penalidades para soluções infactíveis, é possível que uma solução infactível tenha uma pontuação melhor que uma solução factível. Por isso, tornou-se necessário o armazenamento da melhor solução factível obtida até o momento. Um parâmetro importante da busca

tabu é o número de iterações que um movimento permanecerá sendo tabu (*tabu tenure*). No nosso caso, este foi determinado experimentalmente: cada movimento é tabu por 15 iterações. Foram feitas algumas tentativas de definir este parâmetro com os dados do caso de entrada, mas nenhuma combinação plausível foi encontrada. O critério de parada para a busca tabu foi um tempo máximo.

### 3.4 Estudo Comparativo

Esta seção tem por objetivo comparar o algoritmo proposto neste artigo com o proposto em Gendreau et al. (2006). O objetivo inicial deste trabalho foi estudar os procedimentos e técnicas descritos por Gendreau et al. (2006), para descobrir quais elementos são essenciais para obter boas soluções. Como resultado deste estudo, percebeu-se que resultados similares podem ser obtidos com uma implementação simplificada.

Em relação à heurística de empacotamento, o algoritmo anteriormente proposto utiliza duas heurísticas construtivas distintas: a primeira é idêntica a que utilizamos e a outra visa maximizar a superfície de contato de cada item inserido no veículo. Utilizamos somente a primeira, visto que a maximização da superfície de contato de cada item exige um esforço computacional considerável.

Outra diferença está na factibilização do empacotamento. A nossa proposta optou por uma busca local com o intuito de tornar o empacotamento mais rápido. Além disso, utilizou uma vizinhança que realiza trocas entre itens adjacentes da lista de itens (objetivando aproveitar a localidade no interior do veículo). Gendreau et al. (2006) propuseram uma busca tabu para encontrar um melhor empacotamento. Em relação à vizinhança, foi proposto um esquema de classificação dos itens em dois grupos, e troca na lista entre itens classificados de maneira diferente.

Comparando o roteamento dos veículos, a abordagem geral das duas propostas é semelhante. A idéia geral da solução inicial é a mesma, visto que foi a que produziu melhores resultados. Entretanto, utilizamos uma idéia mais direta para a junção de rotas: vizinho mais próximo. Além disso, o algoritmo de Gendreau et al. (2006) possui duas formas de produzir uma solução inicial, enquanto que a nossa proposta tem sempre uma única solução inicial.

Assim sendo, o algoritmo aqui proposto possui uma idéia geral muito semelhante à de Gendreau et al. (2006), mas possui algumas adaptações em pontos específicos. Estas adaptações foram realizadas com o intuito de simplificar o algoritmo, verificando se resultados semelhantes seriam ainda alcançados.

## 4 Resultados

Todos experimentos foram realizados em um AMD Turion 64x2 com 1.6 GHz em cada processador e 2 GB de memória. Apesar do computador utilizado possuir dois cores, os experimentos foram feitos utilizando apenas um destes. Foram realizados testes com 27 instâncias da literatura (Toth and Vigo, 2002). O critério de parada do algoritmo foi um tempo máximo, ajustado levando em conta o número de clientes das instâncias. Para os primeiros nove casos (até 25 clientes), o algoritmo executou durante 30 min. Nos nove casos de teste seguintes (entre 25 e 50 clientes), o algoritmo executou durante 1h. E para os últimos nove casos (mais que 50 clientes), o algoritmo executou por 2h. Em todos os casos de teste, foi considerada uma frota de veículos iguais em que  $L = 25$ ,  $A = 30$  e  $C = 60$ . A superfície de apoio mínima considerada foi 75% da base de apoio total de cada item.

Os resultados obtidos podem ser observados na Tabela 1. As três primeiras colunas da tabela caracterizam a instância (número da instância, número de clientes e número de veículos, respectivamente). A quinta coluna apresenta o valor da solução inicial, caso a mesma seja factível. Um

Tabela 1: Resultados para as 27 instâncias propostas em Toth and Vigo (2002). As três colunas iniciais se referem aos dados da instância: nome da instância, número de clientes e número de veículos. As três colunas seguintes apresentam os resultados encontrados pelo algoritmo apresentado neste artigo, enquanto que as duas seguintes apresentam valor da solução e tempo encontrados pelo algoritmo de Gendreau et al. (2006). A última coluna relaciona os resultados alcançados por ambas abordagens.

Instância	$n$	$v$	Este artigo			Gendreau et al. (2006)		%
			Valor Inicial	Valor Final	Tempo (s)	Valor	Tempo (s)	
01	15	5	326,426	315,228	152,13	316,32	129,5	-0,35
02	15	5	-	334,964	553,56	350,58	5,3	-4,45
03	20	5	-	391,650	763,97	447,73	461,1	-12,53
04	20	6	-	464,060	25,28	448,48	181,1	3,47
05	21	7	543,91	485,639	266,96	464,24	75,8	4,61
06	21	6	-	530,657	573,04	504,46	1167,9	5,19
07	22	6	-	820,241	435,98	831,66	181,1	-1,37
08	22	8	997,695	860,276	372,67	871,77	156,1	-1,32
09	25	8	-	659,568	818,08	666,10	1468,5	-0,98
10	29	10	969,873	858,380	26,77	911,16	714,0	-5,79
11	29	9	1036,48	856,026	1693,31	819,36	396,4	4,47
12	30	9	-	702,644	3433,12	651,58	268,1	7,84
13	32	9	3126,86	2750,98	2120,31	2928,34	1639,1	-6,06
14	32	11	1970,98	1518,510	2323,92	1559,64	3451,6	-2,64
15	32	10	1710,52	1528,990	2692,76	1452,34	2327,4	5,28
16	35	11	-	738,000	350,62	707,85	2550,3	4,26
17	40	14	-	959,869	740,83	920,87	2142,5	4,24
18	44	14	1515,93	1378,490	597,43	1400,52	1452,9	-1,57
19	50	13	-	878,635	1268,20	871,29	1822,3	0,84
20	71	20	-	661,293	7152,27	732,12	790,0	-9,67
21	75	18	-	1216,540	6407,03	1275,20	2370,3	-4,60
22	75	19	-	1306,030	5823,75	1277,94	1611,3	2,20
23	75	18	-	1291,220	5855,71	1258,16	6725,6	2,63
24	75	18	-	1263,380	3517,59	1307,09	6619,3	-3,34
25	100	24	-	1613,090	7136,20	1570,72	5630,9	2,70
26	100	28	-	1856,060	7009,18	1847,95	4123,7	0,44
27	100	25	-	-	-	1747,52	7127,2	-
Média				1009,240	2388,87	1015,13	1863,9	-0,25

“-” indica que a solução inicial é infactível. Em seguida, temos o valor da melhor solução encontrada no tempo determinado. Por último, o tempo, em segundos, que o algoritmo levou para encontrar a melhor solução (e não o tempo total de execução). As duas colunas seguintes referenciam os resultados (valor da solução e tempo para obter esta solução) obtidos em Gendreau et al. (2006). A última coluna da tabela compara, na forma de porcentagem, os melhores valores alcançados pelo nosso algoritmo e pelo algoritmo de Gendreau et al. (2006). Valores negativos indicam que nossa abordagem foi superior, enquanto valores positivos indicam o contrário. Na última linha da tabela, temos a média de resultados finais. A última instância permaneceu infactível dentro do tempo determinado (7200s). Desta forma, as médias contabilizaram apenas as 26 primeiras instâncias.

Analisando os resultados obtidos, observa-se que estes são levemente melhores que os obtidos por Gendreau et al. (2006). A proximidade dos resultados obtidos pelas duas abordagens é esperada visto que ambas são bastante semelhantes. Entretanto, a nossa proposta tem diversas simplificações em relação ao trabalho anterior. Cabe destacar que com a utilização de uma busca tabu no empacotamento, os resultados obtidos foram piores em função do tempo necessário para realizar cada empacotamento. Somente quando simplificou-se o empacotamento, utilizando uma



busca local simples, que foi possível alcançar resultados um pouco superiores.

Em relação ao ambiente de testes, em Gendreau et al. (2006) foi utilizado um processador com uma frequência de relógio um pouco superior, mas com menos memória (Intel Pentium IV de 3 GHz com 500 MB de memória). Desta forma, pode-se considerar que o ambiente de teste é relativamente semelhante, não sendo decisivo para a obtenção de resultados melhores ou piores.

Um ponto negativo a ser considerado é a instância 27, uma instância com muitos clientes e muitos itens a serem entregues. A nossa abordagem não obteve uma solução factível dentro do limite de tempo de 2h para este caso. Em relação à infactibilidade deste caso, a mesma ficou muito perto de tornar-se factível: apenas um veículo ficou com uma unidade de excesso de peso.

Observamos que para a maioria das instâncias, o resultado final foi encontrado depois de muito tempo, às vezes no final do tempo limite (no caso 20, por exemplo, a solução final foi encontrada com 7152s do total de 7200s disponíveis). Contudo, logo no início da execução de muitos casos de testes, resultados relativamente bons foram encontrados. Isto se deve principalmente à busca tabu, que tem a capacidade de sair dos mínimos locais e constantemente atualizar a melhor solução corrente.

## 5 Conclusão

Os problemas encontrados na vida real relacionados com carregamento e roteamento são muitas vezes mais complexos que a definição do 3L-CVRP. Contudo, este trabalho, assim como outros que estão sendo desenvolvidos sobre o assunto, mostra que os resultados obtidos com problemas bastante complexos como este já são bastante satisfatórios. Desta forma, abrem-se novas possibilidades. Torna-se possível colocar mais restrições no problema de forma a deixá-lo mais coerente com as restrições encontradas na vida real. E, conforme isto vai sendo feito, soluções como estas já podem ser usadas por empresas transportadoras de forma a auxiliar o seu processo de escolha de rotas para os veículos e escolha das mercadorias que devem ser postas em cada veículo.

Os trabalhos futuros visam exatamente esta tendência. Pretendemos melhorar a abordagem mostrada neste artigo, mas, além disso, verificar como a mesma comporta-se na presença de mais restrições do problema. Existem algumas variações do problema que podem não ser facilmente mapeadas para a solução utilizada, como a heterogeneidade da frota de veículos. A estratégia para ocupação dos veículos neste caso deve ser bem pensada e provavelmente terá algum impacto na solução final encontrada.

## Agradecimentos

Agradecemos o apoio do CNPq ao projeto universal 481256/2009-3.

## Referências

Clarke, G. and Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research* **12**: 568–581.

de Araújo, O. C. B. (2006). *Problemas de Corte e Empacotamento Tridimensional e Integração com Roteamento de Veículos*, PhD thesis, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, Campinas, Brasil.

Gendreau, M., Iori, M., Laporte, G. and Martello, S. (2006). A tabu search algorithm for a routing and container loading problem, *Transportation Science* **40**(3): 342–350.

- Moura, A. and Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems, *OR Spectrum* .
- Ngoi, B. K. A., Tay, M. and Chua, E. (1994). Applying spatial representation techniques to the container packing problem, *International Journal of Production Research* **32**: 111–123.
- Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.