



Production

ISSN: 0103-6513

production@editoracubo.com.br

Associação Brasileira de Engenharia de

Produção

Brasil

PRADO BELFIORE, PATRÍCIA; YOSHIDA YOSHIZAKI, HUGO TSUGUNOBU  
Scatter search para problemas de roteirização de veículos com frota heterogênea, janelas  
de tempo e entregas fracionadas  
Production, vol. 16, núm. 3, septiembre-diciembre, 2006, pp. 455-469  
Associação Brasileira de Engenharia de Produção  
São Paulo, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=396742028008>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe , Espanha e Portugal  
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

---

# *Scatter search para problemas de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas*

PATRÍCIA PRADO BELFIORE  
HUGO TSUGUNOBU YOSHIDA YOSHIZAKI  
POLI-USP

---

## **Resumo**

Este trabalho estuda a implementação da metaheurística *scatter search* (SS) em um problema real de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas. No problema de roteirização de veículos com entregas fracionadas, cada cliente pode ser abastecido por mais de um veículo. O problema é baseado em um único centro de distribuição, a demanda de cada cliente pode ser maior que a capacidade dos veículos e, além das restrições de janelas de tempo, há também as restrições de capacidade dos veículos e acessibilidade (alguns clientes não podem ser atendidos por alguns veículos). Os modelos foram aplicados em um dos maiores grupos varejistas brasileiros, que abastece 519 clientes distribuídos em 12 estados brasileiros. Os resultados mostraram melhorias no caso real da empresa, reduzindo em até 8% o custo total da operação.

## **Palavras-chave**

Problema de roteirização de veículos, frota heterogênea, janelas de tempo, entregas fracionadas.

# *Scatter search for heterogeneous fleet vehicle routing problems with time windows and split deliveries*

## **Abstract**

*This work studies the implementation of heuristics and scatter search (SS) metaheuristic in a real heterogeneous fleet vehicle routing problem with time windows and split deliveries (HFVRPTWSD) in Brazil. In the vehicle routing problem with time windows and split deliveries (VRPSD) each client can be supplied by more than one vehicle. The problem is based in a single depot, the demand of each client can be greater than the vehicle's capacity and beyond the time windows constraints, and there are also vehicle capacity and accessibility constraints (some customers cannot be served by some vehicles). The models were applied in one of the biggest retail market in Brazil that has 519 stores distributed in 12 Brazilian states. Results showed improvements over current solutions in a real case, reducing up to 8% the total cost of the operation.*

## **Key words**

*Vehicle routing problem, heterogeneous fleet, time windows, split deliveries.*

## INTRODUÇÃO

Este trabalho estuda a implementação de heurísticas e da metaheurística *scatter search* (SS) em um problema de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas (*heterogeneous fleet vehicle routing problem with time windows and split deliveries* – HFVRPTWSD). O HFVRPTWSD é uma combinação do problema de roteirização com frota heterogênea (*heterogeneous fleet vehicle routing problem* – HFVRP), problema de roteirização de veículos com janelas de tempo (*vehicle routing problem with time windows* – VRPTW) e problema de roteirização com entregas fracionadas (*vehicle routing problem with split deliveries* – VRPSD). O problema é baseado em um único centro de distribuição (CD), a demanda dos clientes pode ser maior que a capacidade dos veículos e, além das restrições de janelas de tempo, há também restrições de capacidade dos veículos e acessibilidade (alguns clientes não podem ser atendidos por alguns veículos).

As soluções iniciais de SS são geradas através de uma heurística construtiva (adaptação da heurística de Dullaert *et al.*, 2002).

O problema clássico de roteirização de veículos (*vehicle routing problem* – VRP) tem como objetivo encontrar um conjunto de rotas com o menor custo possível (minimizar distância total percorrida, número de veículos, etc.), iniciando e terminando no CD, de forma que a demanda de todos os nós seja atendida. A demanda é determinística. Cada nó é atendido uma única vez, por um único veículo, e cada veículo possui capacidade limitada. Algumas formulações também apresentam restrição de tempo máximo de viagem.

O VRPSD é uma variação do problema clássico de roteirização de veículos. No problema de roteirização de veículos com entregas fracionadas, cada cliente pode ser abastecido por mais de um veículo, enquanto no problema clássico de roteirização de veículos (VRP), cada cliente é atendido por um único veículo. Desta forma, para o VRPSD, além dos roteiros de entrega, deve-se determinar a quantidade entregue a cada cliente em cada veículo.

Ao contrário do problema clássico de roteirização de veículos (VRP), o objetivo do VRP com frota heterogênea é minimizar a soma dos custos fixos dos veículos e dos custos variáveis de roteirização. Os custos dos veículos são os custos de aquisição e manutenção da frota heterogênea de veículos. Os custos de roteirização são relacionadas à distância total percorrida. Neste trabalho, os custos de roteirização são dependentes do tipo do veículo.

Há duas variantes do VRP com frota heterogênea. Primeiramente, o número de veículos de cada tipo pode ser limitado ou ilimitado. Se o número de veículos for limitado, temos o problema de roteirização de veículos com frota

heterogênea fixa (*heterogeneous fixed fleet vehicle routing problem* – HFFVRP). Se o número de veículos for ilimitado, é necessário determinar o tamanho e a composição da frota. Este problema é chamado problema de dimensionamento e roteirização de uma frota heterogênea de veículos (*fleet size and mix vehicle routing problem* – FSMVRP). Neste trabalho, apesar de o número de veículos ser ilimitado, o objetivo é determinar somente a roteirização dos veículos e não o tamanho e a composição da frota. Em segundo lugar, os custos variáveis podem ser dependentes ou não do tipo de veículo.

Lenstra e Rinnooy Kan (1981) analisaram a complexidade de problemas de roteirização de veículos, e concluíram que praticamente todos os problemas de roteirização de veículos são *NP-hard* (entre eles o problema do caixeiro viajante e o problema clássico de roteirização de veículos), pois não são resolvidos em tempo polinomial.

Segundo Solomon e Desrosiers (1988), o problema de roteirização de veículos com janelas de tempo (VRPTW) também é *NP-hard* por ser uma extensão do VRP.

Gendreau *et al.* (1999), Dullaert *et al.* (2002) e Taillard (1999) afirmam que o problema de dimensionamento e roteirização de uma frota heterogênea de veículos (FSMVRP), o FSMVRP com janelas de tempo (FSMVRPTW) e o problema de roteirização com frota heterogênea fixa (HFFVRP), respectivamente, também são *NP*-completos por serem extensões do VRP.

Já o VRPSD, embora seja uma relaxação do VRP, também permanece *NP*-completo (DROR e TRUDEAU, 1990; ARCHETTI, MANSINI e SPERANZA, 2005).

Desta forma, o HFVRPTWSD, que é uma combinação do problema de roteirização de veículos com frota heterogênea (HFVRP), do problema de roteirização com janelas de tempo (VRPTW) e do problema de roteirização com entregas fracionadas (VRPSD), também é *NP*-completo, justificando o uso de heurísticas e metaheurísticas para resolução do problema.

Os algoritmos serão aplicados em um grupo varejista no Brasil que é abastecido por um CD, com o objetivo de reduzir os custos totais de roteirização. O CD atende 519 clientes distribuídos em 12 estados brasileiros. O grupo vem crescendo nos últimos anos através de uma contínua expansão da área de vendas, abrindo novas lojas e efetuando novas aquisições.

O problema consiste em determinar, a cada dia, como alocar os veículos aos clientes, a quantidade de carga em cada veículo a ser entregue para cada cliente, qual o melhor roteiro e o tempo de início de atendimento do primeiro cliente da rota, de forma a minimizar o custo total de distribuição, garantindo que a demanda dos clientes seja atendida e as demais restrições do problema sejam respeitadas.

## PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM FROTA HETEROGLÉNEA E SUAS EXTENSÕES

Taillard (1999) apresentou um método heurístico de geração de colunas para o problema de roteirização de veículos com frota heterogênea fixa (HFFVRP), onde o número de veículos é limitado e os custos variáveis diferem do tipo de veículo. Tarantilis, Kiranoudis e Vassiliadis (2004) implementaram uma nova metaheurística denominada BATA (*Back-Tracking Adaptive Threshold Accepting*) para resolução do HFFVRP.

Rochat e Semet (1994) implementaram a metaheurística busca tabu para um problema de distribuição de rações de animais com frota heterogênea fixa e restrições de janela de tempo, duração da jornada de trabalho, capacidade dos veículos e restrições de acessibilidade para atendimento de alguns clientes.

Golden *et al.* (1984) desenvolveram uma heurística de economias para o problema de dimensionamento e roteirização de uma frota heterogênea de veículos (FSMVRP), onde os custos variáveis para todos os veículos são iguais a um. Desrochers e Verhoog (1991) desenvolveram uma nova heurística de economias para o FSMVRP baseada na fusão de sucessivas rotas. O número de veículos de cada tipo é ilimitado e os custos totais de roteirização são proporcionais à distância total percorrida.

Salhi e Rand (1993) desenvolveram uma heurística de sete fases para o FSMVRP. O algoritmo é baseado nos seguintes movimentos:

- 1) combinação de rotas;
- 2) eliminação de rotas, em que os clientes eliminados são inseridos em outras rotas existentes;
- 3) inserção de clientes de uma determinada rota em outra;
- 4) fracionamento de grandes rotas em rotas menores;
- 5) troca de clientes entre rotas.

Gendreau *et al.* (1999) implementaram a metaheurística busca tabu para o FSMVRP. A solução inicial é obtida através do algoritmo GENI e, no final desta fase, aplica-se o método de melhoria denominado US. A busca tabu é baseada em um procedimento de memória adaptiva (*adaptive memory procedure – AMP*) desenvolvida por Rochat e Taillard (1995). Wassan e Osman (2002) também implementaram busca tabu para o FSMVRP, em que o número de veículos é ilimitado e os custos variáveis dependem do tipo de veículo.

Liu e Shen (1999) implementaram uma heurística de inserção seqüencial baseada no algoritmo de economias para o FSMVRPTW, onde o número de veículos é ilimitado e os custos variáveis são iguais para todos os veículos. Segundo os autores, a heurística apresentou bons resultados. Dullaert *et al.* (2002) adaptaram a heurística de inserção seqüencial I1 de Solomon (1987) para o FSMVRPTW, utilizando o

conceito de economia adaptada de Golden *et al.* (1984). O número de veículos é ilimitado e o custo de viagem de uma unidade de tempo ou distância é igual a um para todos os tipos de veículos. O algoritmo foi implementado no conjunto de problemas de Liu e Shen e, segundo os autores, os resultados foram superiores.

## PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM ENTREGAS FRACIONADAS

O problema de roteirização de veículos com entregas fracionadas (VRPSD) foi introduzido na literatura por Dror e Trudeau (1989, 1990), que apresentaram a formulação matemática do problema e analisaram as economias que podem ser geradas quando se permite que um cliente possa ser abastecido por mais de um veículo, economia relacionada ao número de veículos e à distância total percorrida. De acordo com Dror e Trudeau, o VRPSD é uma relaxação do problema clássico de roteirização de veículos, mas permanece *NP*-completo. Os autores demonstraram que, quando as distâncias satisfazem a desigualdade do triângulo, existe uma solução ótima para VRPSD, de modo que nenhum par de rotas tenha dois ou mais vértices em comum.

**A**s soluções iniciais de SS são geradas através de uma heurística construtiva (adaptação da heurística de Dullaert *et al.*, 2002).

Dror, Laporte e Trudeau (1994) apresentaram a formulação de um modelo de programação linear inteira para o VRPSD e descreveram novas classes de desigualdades válidas para o problema. Os autores também propuseram uma heurística como limitante superior e aplicaram um algoritmo *branch and bound* para o problema. Belenguer, Martinez e Mota (2000), através de uma nova classe de desigualdades válidas, propuseram um algoritmo de plano de corte e um algoritmo *branch and bound* para o VRPSD.

Frizzell e Giffin (1992) desenvolveram três heurísticas para um problema de roteirização de veículos com entregas fracionadas (VRPSD), baseadas no vizinho mais próximo, em agrupamentos de clientes e introduzindo custos no caso de entregas fracionadas. Archetti, Savelsbergh e Speranza (2006) analisaram o comportamento do pior-caso (*worst-case analysis*) para o VRPSD. Archetti, Hertz e Speranza (2006) descreveram um algoritmo tabu para resolução de um problema de roteirização de veículos com entregas fracionadas (VRPSD).

Frizzell e Giffin (1995) apresentaram a formulação matemática para um problema de roteirização de veículos com

janelas de tempo e entregas fracionadas (VRPTWSD). Os autores desenvolveram uma heurística construtiva baseada na urgência dos clientes e duas heurísticas de melhoria; uma delas permite troca de clientes da mesma rota e a outra, troca de clientes entre rotas. Ho e Haugland (2004) demonstraram que a propriedade de desigualdade do triângulo de Dror e Trudeau (1989, 1990) também é válida para o VRPTWSD. Eles implementaram busca tabu para o VRPTWSD.

Mullaseril, Dror e Leung (1997) implementaram uma adaptação da heurística proposta por Dror e Trudeau (1990) para um problema de roteirização de veículos com demanda em arcos, janelas de tempo e entregas *fracionadas* (*capacitated arc routing problem with time windows and split deliveries* – CARPTWSD).

## DEFINIÇÃO DO PROBLEMA

Este trabalho estuda um problema real de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas (HFVRPTWSD) em um dos maiores grupos varejistas brasileiros. A definição completa está detalhada a seguir.

O problema é baseado em um único centro de distribuição que atende 519 clientes em todo o Brasil, em 12 estados brasileiros. Os produtos são distribuídos através de uma frota de veículos heterogênea e ilimitada. O serviço de transporte é terceirizado. Cada tipo de veículo tem uma capacidade e um custo fixo e variável por viagem. Para um determinado dia, a demanda de cada cliente deve ser atendida. Cada cliente pode ser atendido por mais de um veículo (entregas fracionadas). Muitos clientes possuem demanda superior à capacidade do maior caminhão capaz de atendê-lo. Os clientes possuem restrições de janela de tempo e restrições de acessibilidade (alguns clientes não podem ser atendidos pelo maior veículo em função das restrições operacionais).

Os veículos saem e retornam ao CD.

As hipóteses do problema são:

- A demanda dos clientes é determinística;
- Horizonte de planejamento de um dia;
- O tempo de serviço é fixo em 1,5 hora;
- O CD trabalha 24 horas por dia.

A formulação matemática do modelo está listada a seguir.

Considere um conjunto de  $N$  clientes a serem atendidos. Em um determinado dia, a demanda  $d_i$  de cada cliente  $i \in \{1, 2, \dots, n\}$  deve ser atendida. Cada cliente possui uma restrição de janela de tempo  $[a_i, b_i]$  tal que  $a_i \leq b_i$ , que corresponde, respectivamente, ao horário inicial e final em que pode ser iniciado o atendimento. O tempo de atendimento  $s_i$  representa o tempo de descarga dos veículos (tempo médio de todos os processos administrativos, burocráticos e de operação). Um roteiro pode ser formado por um único cliente ou por vários clientes.

O deslocamento do nó  $i$ ,  $i = 0, \dots, n$  até um nó  $j$ ,  $j = 1, \dots, n+1$  requer um tempo de viagem  $t_{ij}$  e uma distância percorrida  $d_{ij}$ . Os pontos 0 e  $n+1$  representam o CD. As matrizes de tempos de viagem e de distâncias são simétricas, ou seja,  $t_{ij} = t_{ji}$  e  $d_{ij} = d_{ji}$ .

O número de tipos de veículos é representado por  $T$ . Há quatro tipos de veículos distintos: carreta, *truck*, toco e leve. Cada rota  $R_j$ ,  $j = 1, \dots, p$  será atendida por um determinado veículo  $v$ ,  $v = 1, \dots, T$  que possui uma capacidade máxima  $C_v$ , um custo fixo por viagem  $f_v$  e um custo variável  $g_v$  por unidade de distância percorrida (km), que corresponde aos custos de combustível, manutenção do veículo, pneus, lubrificantes, etc. O custo de um veículo do tipo  $v$  atender o par  $(i, j)$  é denotado  $c_{ij}^v$ , que é obtido pela multiplicação da distância  $d_{ij}$  e do custo variável  $g_v$ . Assume-se  $c_{ij}^v = \infty$  quando o cliente  $i$  ou o cliente  $j$  não podem ser servidos pelo veículo do tipo  $v$ .

As variáveis de decisão do modelo são:

$$x_{ij}^v = 1, \text{ se } j \text{ é atendido após } i \text{ pelo veículo } v; \\ 0, \text{ caso contrário}$$

$$T_i^v = \text{hora de início de atendimento do cliente } i \text{ pelo veículo } v, \quad i = 1, \dots, n \quad v = 1, \dots, T$$

$$y_i^v = \text{fração de demanda do cliente } i \text{ entregue pelo veículo } v$$

O objetivo do modelo é minimizar o custo total de transporte (soma dos custos de viagem e dos custos variáveis em função da distância), de forma que a demanda de todos os clientes seja atendida e as demais restrições do problema sejam respeitadas. A formulação completa do problema está detalhada a seguir com base nos trabalhos de Dror e Trudeau (1990), Ho e Haugland (2004) e Golden *et al.* (1984). A função objetivo pode ser escrita da seguinte forma:

$$\min \sum_{v=1}^T f_v \sum_{j=1}^n x_{0j}^v + \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^T c_{ij}^v x_{ij}^v$$

As restrições do modelo são:

$$\sum_{j=1}^n x_{0j}^v = 1 \quad v = 1, \dots, T \quad (1)$$

A restrição (1) garante que cada veículo saia do CD e chegue a um determinado cliente.

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad p = 0, \dots, n; \quad v = 1, \dots, T \quad (2)$$

A restrição (2) é a restrição de conservação dos fluxos de entrada e saída; garante que cada veículo saia de um determinado cliente e retorne ao CD.

$$\sum_{v=1}^T y_i^v = 1, \quad i = 1, \dots, n \quad (3)$$

A restrição (3) garante que a demanda total de cada cliente será atendida.

$$\sum_{i=1}^n d_i y_i^v \leq C_v \quad v = 1, \dots, T \quad (4)$$

A restrição (4) garante que a capacidade de cada veículo não será excedida.

$$y_i^v \leq \sum_{j=0}^n x_{ji}^v \quad i = 1, \dots, n; \quad v = 1, \dots, T \quad (5)$$

A restrição (5) garante que a demanda de cada cliente será atendida somente se um determinado veículo passar por aquele ponto. Nota-se que, adicionando à restrição (5) o somatório de todos os veículos, e combinando-a com a equação (3), obtém-se a restrição  $\sum_{v=1}^T \sum_{i=0}^n x_{ij}^v \geq 1 \quad j = 0, \dots, n$ , que garante que cada vértice seja visitado pelo menos uma vez, por pelo menos um veículo.

$$T_i^v + s_i + t_{ij} - M_{ij}(1 - x_{ij}^v) \leq T_j^k \quad i = 1, \dots, n; \quad j = 1, \dots, n; \quad v = 1, \dots, T \quad (6)$$

A restrição (6) impõe o horário mínimo de início de atendimento do cliente  $j$  em uma determinada rota e também garante que não haja formação de *subtours*. A constante  $M_{ij}$  é um número suficientemente grande, por exemplo,  $M_{ij} = b_i + t_{ij} - a_j$ .

$$a_i \leq T_i^v \leq b_i \quad i = 1, \dots, n \quad (7)$$

A restrição (7) garante que todos os clientes são atendidos dentro de sua janela de tempo.

$$\begin{aligned} y_i^v &\geq 0 \quad i = 1, \dots, n; \quad v = 1, \dots, T \\ T_i^v &\geq 0 \quad i = 1, \dots, n \end{aligned} \quad (8)$$

A restrição (8) garante que as variáveis de decisão  $y_i^v$  e  $T_i^v$  sejam positivas.

$$x_{ij}^v \in \{0, 1\} \quad i = 0, \dots, n; \quad j = 0, \dots, n; \quad v = 1, \dots, T \quad (9)$$

Finalmente, a restrição (9) garante que as variáveis de decisão  $x_{ij}^v$  sejam binárias.

## PRINCIPAIS CONCEITOS DE SCATTER SEARCH

A metaheurística *scatter search* (SS), também conhecida como busca dispersa (BD), será aplicada para resolução do modelo proposto neste trabalho.

*Scatter search* se baseia em combinar as soluções que aparecem no chamado conjunto de referência (*RefSet*). Este conjunto armazena boas soluções que foram encontradas durante o processo de busca. Os principais parâmetros do método *scatter search* estão descritos a seguir:

$P$  = conjunto de soluções geradas por um método de diversificação

$PSize$  = tamanho da população  $P$

$b$  = tamanho do conjunto de referência (*RefSet*)

$b_1$  = tamanho do subconjunto das soluções de alta qualidade

$b_2$  = tamanho do subconjunto de soluções diversas

$MaxIter$  = número máximo de iterações

De acordo com Glover (1998), o método consiste basicamente de cinco passos que estão descritos a seguir:

**Passo 1. Método de geração de soluções diversas:** o método consiste em gerar um conjunto  $P$  com  $PSize$  elementos. O objetivo é gerar soluções distantes entre si de forma a garantir a diversidade.

**Passo 2. Método de melhoria:** trata-se de um método de busca local para melhorar as soluções diversas. No caso de soluções infactíveis, a melhoria primeiramente consiste em torná-las factíveis. Caso a solução seja factível, aplica-se a heurística de melhoria.

**Passo 3. Método de criação e atualização do conjunto de referência:** é um método para criar e atualizar o conjunto de referência (*RefSet*). A partir do conjunto  $P$  de soluções diversas, extrai-se o conjunto de referência. É dividido em um subconjunto de soluções de alta qualidade (*RefSet*<sub>1</sub>) e um subconjunto de soluções diversas (*RefSet*<sub>2</sub>). O método SS, portanto, é baseado nas estratégias de intensificação e diversificação.

**Passo 3.1 Criação do conjunto de referência:** o conjunto de referência inicia-se com as  $b_1$  melhores soluções de  $P$ , obtidas após o método de melhoria. As  $b_2$  soluções restantes são extraídas de  $P$  com o objetivo de maximizar a distância mínima em relação às soluções incluídas no conjunto de referência.

**Passo 3.2 Atualização do conjunto de referência:** as soluções resultantes das combinações, no caso de melhoria, substituem soluções do conjunto de referência. Existem dois aspectos que devem ser considerados na atualização de *RefSet*. O primeiro aspecto considera dois tipos de atualização:

**Atualização estática (S):** Na atualização estática, o conjunto de referência não muda até que todas as combinações

de soluções de *RefSet* tenham sido realizadas. Exemplos de atualizações estáticas podem ser encontrados em Campos *et al.* (2001), Laguna e Martí (2005), Martí, Laguna e Campos (2005) e Campos, Laguna e Martí (2005).

**Atualização dinâmica (D):** Na atualização dinâmica, o conjunto de referência é atualizado sempre que uma nova solução, gerada por uma combinação, qualifica-se para entrar em *RefSet*. Nesse caso, algumas combinações do *RefSet* original não são realizadas. Em geral, esse tipo de atualização acelera a convergência do método, pois soluções de alta qualidade substituem soluções de baixa qualidade. Exemplos de atualização dinâmica encontram-se em Martí, Lourenço e Laguna (2000), Jain e Meeran (2002), Laguna e Martí (2005), Yamashita, Armentano e Laguna (2006).

O segundo aspecto considera dois critérios de atualização: por qualidade e diversidade (QD) e apenas por qualidade (Q).

**Atualização QD:** Considere uma solução gerada pelo método de combinação de soluções. Se o custo da combinação resultante for menor do que a pior solução de *RefSet*, então a nova solução resultante substitui o pior elemento de *RefSet*. Se o custo for maior do que o pior elemento de *RefSet*, esta solução é inserida em *RefSet*, se aumentar a menor distância entre as soluções deste subconjunto. Nesse caso, a solução pertencente à *RefSet* que possuir a menor distância em relação às demais é substituída pela nova solução.

**Atualização Q:** Se a nova solução gerada pelo método de combinação de soluções for melhor, em termos de qualidade, que o pior elemento de *RefSet*, então substitui-se o pior elemento de *RefSet* pela nova solução.

**Passo 4. Método de geração de subconjuntos:** este método especifica a forma pela qual são selecionados os subconjuntos para aplicar o método de combinação.

**Passo 5. Método de combinação das soluções:** busca combinar as soluções do conjunto de referência. Para isso, consideram-se os subconjuntos formados pelo Passo 4 e aplica-se o método de combinação. As soluções obtidas desta combinação podem ser imediatamente introduzidas no conjunto de referência (atualização dinâmica) ou armazenadas temporariamente em uma lista até que sejam realizadas todas as combinações e, a partir daí, verificam-se quais soluções entrarão no conjunto (atualização estática). O método de melhoria é aplicado às melhores soluções geradas pelas combinações.

Figura 1: Algoritmo implementado antes da aplicação da heurística de inserção seqüencial de Dullaert *et al.* (2002).

```

- Para cada cliente:
Enquanto a demanda remanescente for maior do que a capacidade do maior veículo capaz de atender o cliente faça
    Envie um veículo deste tipo com carga completa
    Recalcule a demanda remanescente
Fim enquanto

```

## MÉTODO DE SOLUÇÃO

Primeiramente, foi implementada uma heurística construtiva (adaptação da heurística de inserção de Dullaert *et al.*, 2002). As soluções do conjunto de referência são geradas através da incorporação de elementos aleatórios na heurística construtiva.

## ADAPTAÇÃO DA HEURÍSTICA DE INSERÇÃO SEQUENCIAL DE DULLAERT ET AL. (AHD)

Dullaert *et al.* (2002) adaptaram a heurística de inserção seqüencial de Solomon (1987) para um problema de dimensionamento e roteirização de uma frota heterogênea de veículos com restrições de janelas de tempo (FSMVRPTW).

Como mencionado anteriormente, a maioria dos clientes possui demanda superior à capacidade dos veículos. Assim, antes de aplicar a heurística de Dullaert *et al.*, um algoritmo foi implementado com o objetivo de atender a demanda de cada cliente com veículos lotados (carga completa) até que a demanda remanescente seja menor do que a capacidade do maior veículo capaz de atendê-la. Este algoritmo está descrito a seguir.

A heurística construtiva implementada no presente trabalho é similar à heurística de Dullaert, acrescentando o algoritmo descrito na Figura 1, de modo que a demanda remanescente de cada cliente seja menor que a capacidade do maior veículo capaz de atendê-lo. A partir daí, aplica-se a heurística de Dullaert *et al.* (2002), que será descrita a seguir. Os principais parâmetros utilizados são:

- $a_i$  – início da janela de tempo do cliente  $i$
- $b_i$  – fim da janela de tempo do cliente  $i$
- $i_i$  – instante de início efetivo do serviço no cliente  $i$
- $s_i$  – tempo de duração do serviço no cliente  $i$
- $t_{ij}$  – tempo de viagem do cliente  $i$  ao cliente  $j$
- $t_j$  – distância do cliente  $i$  ao cliente  $j$

O início do atendimento do cliente  $j$  é o horário máximo entre o instante de chegada ao cliente  $j$  e o início da janela de tempo ( $a_j$ ):

$$i_j = \max \{a_i, i_i + s_i + t_{ij}\} \quad (11)$$

Primeiramente, define-se o critério de início de uma rota com um determinado cliente. Os critérios de início das rotas são:

- a) cliente mais distante do CD ainda não alocado
- b) cliente com menor valor de  $b_i$  (fim da janela de tempo do cliente  $i$ )

Os critérios de início são exclusivos, ou seja, utiliza-se um ou outro. Iniciada uma rota com um determinado cliente, a próxima etapa é a utilização de um critério para inserir o próximo cliente da rota. Considera-se  $(i_0, i_1, \dots, i_m)$  a rota atual com  $i_0 = i_m = 0$  (CD) e  $m - 1$  clientes. O método de inserção utiliza dois critérios,  $c_1(i, u, j)$  e  $c_2(i, u, j)$ , a cada iteração, para inserir um novo cliente  $u$ , entre os clientes  $i$  e  $j$  da rota.

O primeiro critério tem como objetivo determinar a melhor posição de inserção de um determinado cliente  $u$ . Calcula-se, para cada cliente  $u$  ainda não alocado a nenhuma rota, a melhor posição  $p$  viável de inserção, definida por  $c_1(i(u), u, j(u))$ :

$$\begin{aligned} c_1(i(u), u, j(u)) &= \text{melhor posição de inserção do cliente } u \\ c_1(i(u), u, j(u)) &= \min[c_1(i_{p-1}, u, i_p)], \quad p = 1, \dots, m \end{aligned} \quad (12)$$

onde:

$$\begin{aligned} c_1(i, u, j) &= \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) + \alpha_3 c_{13}(i, u, j) \\ \alpha_1, \alpha_2, \alpha_3 &\geq 0 \end{aligned} \quad (13)$$

onde:

$$\begin{aligned} c_{11}(i, u, j) &= d_{iu} + d_{uj} - ud_{ij}, \quad u \geq 0 \\ c_{12}(i, u, j) &= i_{j/u} - i_j \\ c_{13}(i, u, j) &= \text{AOOS} \end{aligned} \quad (14)$$

e  $i_{j/u}$  é o novo tempo que inicia o serviço do cliente após a inserção do cliente na rota.

O critério de inserção  $c_1(i, u, j)$  leva em conta o acréscimo de distância  $c_{11}(i, u, j)$  e tempo  $c_{12}(i, u, j)$  causados pela inserção de um cliente  $u$ , além de um terceiro componente  $c_{13}(i, u, j)$  que considera um possível acréscimo no custo fixo do veículo. Porém, ao contrário da heurística II proposta por Solomon (1987), não há restrição de que a soma dos pesos ( $\alpha_1 + \alpha_2 + \alpha_3$ ) seja 1.

O terceiro componente  $c_{13}(i, u, j)$  do custo de inserção  $c_1(i, u, j)$  é uma adaptação do conceito de economia de Golden *et al.* (1984). Dullaert *et al.* (2002) implementaram 3 variações do conceito de economia: o conceito de economia combinada adaptada (*adapted combined savings* – ACS), o conceito de economia de oportunidade otimista adaptada (*adapted optimistic opportunity savings* – AOOS) e o conceito de economia de oportunidade realista adaptada (*adapted realistic opportunity savings* – AROS). O presente trabalho implementa ape-

nas o conceito de economia de oportunidade otimista adaptada (AOOS), em função dos resultados obtidos por Dullaert *et al.*

Segundo os autores, a carga atual de um veículo e sua capacidade máxima são denominadas  $Q$  e  $\bar{Q}$ , respectivamente. A nova carga do veículo e a possível nova capacidade depois da inserção de um novo cliente são representadas por  $Q^{new}$  e  $\bar{Q}^{new}$ , respectivamente.

O conceito de economia combinada adaptada (ACS) é definido como a diferença entre o custo fixo do veículo capaz de transportar a carga da rota depois e antes da inserção do cliente  $u$ ,  $ACS = F(Q^{new}) - F(Q)$ , onde  $F(q_u)$  é o custo fixo do menor veículo capaz de entregar a demanda  $q_u$ .

O conceito de economia de oportunidade otimista adaptada (AOOS) é uma extensão de ACS subtraindo  $F(\bar{Q}^{new} - Q^{new})$  que pode ser definido como o custo fixo do menor veículo capaz de transportar a carga não utilizada do novo veículo. Temos, portanto, que  $AOOS = (F(Q^{new}) - F(Q)) - F(\bar{Q}^{new} - Q^{new})$ .

O segundo critério tem como objetivo determinar o cliente  $u$  a ser inserido na rota entre os clientes  $i$  e  $j$ , que é aquele que maximiza a função  $c_2(i(u), u, j(u))$ :

$$\begin{aligned} c_2(i(u^*), u^*, j(u^*)) &= \max[c_2(i(u), u, j(u))] \\ c_2(i, u, j) &= \lambda(d_{0u} + t_{0u}) + s_u + F(q_u) - c_1(i, u, j), \quad \lambda \geq 0 \end{aligned} \quad (15)$$

onde  $s_u$  é o tempo de serviço do cliente  $u$  e  $F(q_u)$  é o custo fixo do menor veículo capaz de transportar a carga  $q_u$ . Se nenhum cliente for factível, uma nova rota é iniciada. O critério de inserção  $c_2(i, u, j)$  busca maximizar o benefício decorrente da inserção de um cliente na rota atual em relação à inserção em uma nova rota.

## PROCEDIMENTO SCATTER SEARCH IMPLEMENTADO PARA O PROBLEMA ESTUDADO

### Passo 1. Método de geração de soluções diversas

O conjunto  $P$  com  $Psize$  elementos é gerado através de um método de diversificação que incorpora elementos aleatórios na heurística construtiva, denominado método GRASP CONSTRUTIVO (GRACO).

O procedimento GRASP (*Greedy Randomized Adaptive Search Procedure*) é uma técnica desenvolvida por Feo e Resende (1989) que consiste em duas fases: fase de construção e fase de melhoria. A fase de construção gera diversas soluções de boa qualidade através de um componente aleatório. A segunda fase consiste na implementação de movimentos de melhoria, com o objetivo de melhorar as soluções iniciais obtidas. O método GRACO implementado no presente trabalho é baseado na primeira fase do procedimento GRASP.

O objetivo deste método é, portanto, incorporar elementos aleatórios nas heurísticas construtivas através de um parâmetro  $\alpha$ , gerando novas soluções.

O método GRACO foi aplicado tanto no critério de início da rota, quanto no critério de inserção. Consideremos um exemplo em que o critério de início é o cliente mais distante do CD. Desta forma, o método GRACO sorteia um elemento, entre os  $k$  elementos mais distantes do CD, para iniciar a rota. Os elementos estão ordenados em uma lista candidata, onde o primeiro elemento da lista é o cliente mais distante do CD. No critério de inserção, um cliente  $u$  será inserido entre os clientes  $i$  e  $j$ . Sorteia-se um elemento entre os  $k$  clientes mais próximos de  $i$  e  $j$  da lista candidata. O primeiro elemento da lista é o cliente mais próximo de  $i$  e  $j$ . O critério de proximidade é calculado em função do acréscimo de custo de transporte e tempo causados pela inserção de um cliente na rota.

Neste trabalho são consideradas duas variações do método GRACO, nomeadas GRACO1 e GRACO2. Em GRACO1 foi considerado  $\alpha = \frac{i}{n} \times 100$ , quando a  $i$ -ésima solução é gerada, em qualquer iteração (passagem pela etapa de diversificação). Por exemplo, na primeira iteração escolhe-se o primeiro elemento da lista candidata (solução semelhante à heurística construtiva). Já na segunda iteração, escolhe-se um elemento, aleatoriamente, entre os dois primeiros elementos da lista candidata, e assim por diante, até que sejam geradas  $PSize$  soluções.

## HFVRPTWSD é uma combinação do VRP com frota heterogênea, do VRP com janelas de tempo e do VRP com entregas fracionadas.

Em GRACO2, considerou-se na primeira iteração  $\alpha = 5\%$ , na segunda  $\alpha = 10\%$ , e assim por diante, até  $\alpha = 45\%$  (nona iteração). Nas demais iterações considerou-se  $\alpha = 50\%$ . Por exemplo, na primeira iteração escolhe-se um elemento, aleatoriamente, entre os 5% primeiros elementos da lista candidata. Na segunda iteração escolhe-se um elemento, aleatoriamente, entre os 10% primeiros elementos da lista candidata, e assim por diante, até  $\alpha = 45\%$ . A partir da décima iteração, escolhe-se um elemento, aleatoriamente, entre os 50% primeiros elementos da lista candidata, até que sejam geradas  $PSize$  soluções.

O método GRACO1 é aplicado para as três primeiras iterações da metaheurística *scatter search*. A partir da quarta iteração, utiliza-se o método de diversificação GRACO2. O objetivo de utilizar dois métodos distintos de diversificação é fazer com que a busca não estacione em um ótimo local.

### Passo 2. Método de melhoria

Para cada uma das  $PSize$  soluções obtidas a partir do método de geração de soluções diversas, aplica-se o método de melhoria que consiste em cinco fases: trocas na mesma rota, realocação de demanda, eliminação e combinação de rotas, inserção e adição de rotas. O procedimento de trocas na mesma rota é aplicado novamente no final do processo.

#### Trocas na mesma rota

As trocas na mesma rota são efetuadas através de um movimento denominado *CYC API* (*CYClic Adjacent Pairwise Interchange*), que consiste na troca cíclica de pares adjacentes de uma determinada rota. Considere uma determinada rota  $R_j$  com  $p$  clientes,  $R_j = 1, \dots, p$ . Dado  $R_j = (3, 2, 1, 4)$ , a vizinhança  $N(R_j)$  gerada pelo movimento *CYC API* é:

$$N(R_j) = \{(2, 3, 1, 4); (3, 1, 2, 4); (3, 2, 4, 1); (4, 2, 1, 3)\}$$

#### Realocação de demanda

Este movimento é aplicado para cada cliente  $i$ ,  $i = 1, \dots, n$  que está fracionado em mais de uma rota. Inicia-se com o cliente  $i$  mais distante do CD. Para um determinado cliente  $i$ , escolhe-se, inicialmente, a rota  $R_j$  que entrega a maior quantidade  $Q_i$ , pois clientes com demandas superiores têm maior probabilidade de violar a restrição de capacidade se não forem inseridos logo no início (Salhi e Rand, 1993). Assim, calcula-se o custo de realocação da demanda do cliente  $i$  da rota  $R_j$  para cada uma das rotas  $R_k$  onde o cliente  $i$  está inserido e escolhe-se aquela com maior economia. A troca só é efetuada se todas as restrições do problema forem respeitadas e o custo total reduzido. O procedimento é repetido para as demais rotas. O tipo de veículo designado é sempre o menor capaz de atender a demanda da rota.

#### Eliminação e combinação de rotas

Esta fase busca eliminar rotas com veículos menores que o segundo maior veículo da frota. O critério de eliminação também pode ser aplicado para rotas com  $n$  ou menos clientes ou rotas com capacidade ociosa. Para cada possível rota  $R_j$  eliminada (inicia-se pela rota mais longa, ou seja, a que possui a maior distância total percorrida), busca-se combiná-la com outra rota ( $R_k$ ), escolhida através de uma lista candidata. A lista candidata é baseada nas idéias de Corberán *et al.* (2002) e está descrita a seguir.

Para cada rota avaliada ( $R_j$ ), constrói-se uma lista candidata, com base nos melhores pares de rotas ( $R_j, R_k$ ). O melhor par é aquele que minimiza a distância entre duas rotas (primeiro elemento da lista candidata). Quando as rotas de um

dado par apresentam apenas um cliente, a distância entre elas é simplesmente a distância entre os dois clientes. Quando uma rota tem mais de um cliente, consideram-se os pontos extremos para calcular a distância mínima. A distância mínima entre as rotas  $R_j$  e  $R_k$  é dada por:

$$\min \{d_{R_j(1)R_k(n_k)}, d_{R_k(1)R_j(n_j)}\} \quad (16)$$

onde:

$d_{R_j(1)R_k(n_k)}$  é a distância do primeiro cliente da rota  $R_j$  com o último cliente da rota  $R_k$ ;

$d_{R_k(1)R_j(n_j)}$  é a distância do primeiro cliente da rota  $R_k$  com o último cliente da rota  $R_j$ .

No movimento de combinação, todos os clientes da rota  $R_j$  serão testados em todas as posições da rota  $R_k$ . Inicia-se com o primeiro cliente da rota  $R_j$ . A combinação de rotas é efetuada somente se houver melhoria e todas as restrições do problema forem respeitadas, após inserção de todos os clientes da rota  $R_j$ . O algoritmo pode ser descrito na Figura 2:

### Inserção

O movimento de inserção é implementado para cada rota  $R_j$ , onde todos os clientes viáveis serão testados em todas as posições de uma outra rota  $R_k$ , escolhida através de uma lista candidata, semelhante à lista descrita no movimento de eliminação e combinação de rotas. Inicia-se com a rota  $R_j$  mais longa, que é aquela que possui a maior distância total

percorrida e, para cada rota  $R_j$  escolhida, inicia-se com o cliente  $i$  mais distante do CD. A inserção de um determinado cliente é efetuada somente se houver melhoria e todas as restrições do problema forem respeitadas.

O algoritmo pode ser expresso na Figura 3.

### Adição de rotas

Da mesma forma que na realocação de demanda, consideram-se todos os clientes  $i$  cuja demanda está fracionada em mais de uma rota. Inicia-se com o cliente  $i$  mais distante do CD. Para um determinado cliente  $i$ , inicia-se com a rota  $R_j$  que entrega a menor quantidade. Neste movimento, caso a demanda do cliente  $i$  da rota  $R_j$  seja maior que a capacidade do maior caminhão, o mesmo é abastecido enquanto houver espaço e a demanda remanescente é adicionada a um novo veículo (menor veículo capaz). Para cada rota  $R_j$  avaliada, adiciona-se a demanda do cliente  $i$  a uma nova rota ou a rota existente  $R_k$  e calcula-se o acréscimo ou redução no custo total após esse movimento. No final, escolhe-se a melhor combinação, caso haja economia.

### Passo 3. Método de criação e atualização do conjunto de referência

#### Passo 3.1 Criação do conjunto de referência

O conjunto de referência é criado conforme descrito anteriormente. O critério de distância utilizado está descrito ao final do Passo 5.

**Figura 2: Algoritmo de eliminação e combinação de rotas.**

Para cada possível rota  $R_j$  a ser eliminada (inicia-se pela rota mais longa)

Escolher uma rota  $R_k$  entre as quatro primeiras rotas da lista candidata. Verificar se a rota combinada (junção dos clientes das rotas  $R_j$  e  $R_k$ ) é viável em termos de capacidade e tipo de veículo. Se a resposta for positiva, passar para a próxima etapa. Caso contrário, repetir o procedimento para as próximas quatro rotas da lista.

Inserir cada cliente da rota  $R_j$  (iniciar pelo primeiro cliente) em todas as posições da rota  $R_k$  e escolher a melhor posição viável (que minimize o acréscimo de custo e tempo e atenda a restrição de janela de tempo).

Se, após inserção de todos os clientes da rota  $R_j$  na rota  $R_k$ , houver melhoria e as restrições forem respeitadas, a combinação é efetuada.

**Figura 3: Algoritmo de inserção.**

Para cada rota  $R_j$  (inicia-se com a rota mais longa)

Escolher uma rota, aleatoriamente, entre as quatro primeiras rotas  $R_k$  da lista candidata. Verificar se pelo menos um cliente da rota  $R_j$  é viável em termos de capacidade e tipo de veículo (após inserção na rota  $R_k$ ). Se a resposta for positiva, passar para a etapa seguinte. Caso contrário, aplicar o mesmo procedimento para as próximas quatro rotas da lista.

Inserir cada cliente viável da rota  $R_j$  (iniciar pelo cliente mais distante do CD) em todas as posições da rota  $R_k$  e escolher a melhor delas, caso haja melhoria e as restrições sejam respeitadas.

### Passo 3.2 Atualização do conjunto de referência

Todos os tipos de atualização descritos anteriormente foram testados.

### Passo 4. Método de geração de subconjuntos

O método de combinação é aplicado para todos os pares de soluções do conjunto de referência (combinações do tipo 1):  $(S_1, S_2)$ ,  $(S_1, S_3)$ , ...,  $(S_1, S_{b1+b2})$ ,  $(S_2, S_3)$ , ...,  $(S_{b1+b2-1}, S_{b1+b2})$ .

### Passo 5. Método de combinação das soluções

O método de combinação de soluções implementado é dividido em dois passos. O Passo 5.1 busca combinar apenas os elementos das rotas combinadas e o Passo 5.2 atende a demanda remanescente dos clientes. O método de combinação foi baseado nas idéias de Corberán *et al.* (2002) e Rego e Leão (2000).

**N**o problema de roteirização de veículos com entregas fracionadas, cada cliente pode ser abastecido por mais de um veículo.

#### Passo 5.1

Consideremos uma solução  $A$  com  $i$  rotas e uma solução  $B$  com  $k$  rotas, onde  $A_i$  é a  $i$ -ésima rota da solução  $A$  e  $B_k$  é a  $k$ -ésima rota da solução  $B$ . As soluções  $A$  e  $B$  serão combinadas. O procedimento de combinação do Passo 5.1 é construído a partir de uma matriz  $A \times B$ , onde o componente  $(A_i, B_k)$  da matriz contém o número de elementos comuns entre a rota  $i$  da solução  $A$  e a rota  $k$  da solução  $B$ . Primeiramente definem-se quais rotas de cada solução são combinadas, ou seja, qual componente  $(A_i, B_k)$  da matriz será escolhido. Esta fase inicia-se com a combinação do componente que possui o maior número de elementos comuns. No caso de empate, escolhe-se a combinação de rotas que minimiza a função  $f$ ,  $f = \sum_{j \in A_i} |Q_{j,i}^A - Q_{j,k}^B|$ , em que  $j$  é o elemento em comum da rota  $i$  da solução  $A$  e da rota  $k$  da solução  $B$ ;  $Q_{j,i}^A$  é a quantidade entregue ao cliente  $j$  da rota  $i$  da solução  $A$  e  $Q_{j,k}^B$  é a quantidade entregue ao cliente  $j$  da rota  $k$  da solução  $B$ .

A rota combinada é formada pelos elementos comuns das rotas. A quantidade entregue para cada elemento é a menor quantidade entre as duas soluções combinadas e o tipo de veículo é o menor veículo capaz de atender a demanda da rota. O roteiro da solução combinada é semelhante ao da melhor solução e, no caso de empate, escolhe-se o roteiro de uma das soluções aleatoriamente. Cada rota combinada é excluída da lista (elimina-se a linha e coluna referente ao componente  $A_i, B_k$ ). O processo continua enquanto houver rotas (que ainda não foram combinadas) com elementos comuns.

### Passo 5.2

Esta fase tem como objetivo atender a demanda remanescente dos clientes que já foram atendidos por alguma rota no Passo 5.1, e atender a demanda total dos clientes que ainda não pertencem a nenhuma rota. Isto é feito através de um procedimento de inserção. O cliente  $i$  a ser inserido é o cliente mais distante do CD. Primeiramente, verifica-se se ele já pertence a alguma rota combinada. A partir daí, dois novos passos podem ser tomados:

#### Passo 5.2.1

Caso o cliente  $i$  já pertença a pelo menos uma rota combinada (Passo 5.1 do método de combinação), escolhe-se aquela com maior capacidade ociosa e entrega-se o mínimo entre a capacidade ociosa do veículo atual e a demanda do cliente  $i$ . Enquanto a demanda total do cliente  $i$  não for atendida e houver uma rota com capacidade ociosa em que o cliente  $i$  está inserido, o procedimento é repetido (sem mudar o veículo atual).

Caso o cliente  $i$  pertença a apenas uma rota combinada e a demanda total não tenha sido atendida com o veículo atual, o veículo é alterado para o menor capaz de atender a demanda remanescente. Caso a demanda remanescente do cliente  $i$  seja maior que a capacidade ociosa do maior veículo capaz de atendê-lo, a demanda é adicionada enquanto houver capacidade e a demanda remanescente será entregue através do Passo 5.2.2.

Se o cliente  $i$  pertencer a mais de uma rota combinada e a demanda total não for atendida com os veículos atuais, entrega-se o mínimo entre a capacidade ociosa do maior veículo capaz e a demanda remanescente do cliente  $i$ , e a rota escolhida é aquela que minimiza o acréscimo de custo fixo. O procedimento é repetido até que a demanda total do cliente  $i$  tenha sido atendida, ou até que todos os veículos (maiores capazes) no qual o cliente  $i$  está inserido estejam utilizando sua capacidade máxima.

Ao final deste passo, caso a demanda total do cliente  $i$  ainda não tenha sido atendida, aplica-se o Passo 5.2.2. Caso contrário, escolhe-se o próximo cliente mais distante do CD a ser inserido.

#### Passo 5.2.2

Caso o cliente  $i$  não pertença a nenhuma rota combinada ou a demanda total do cliente  $i$  não tenha sido atendida através do Passo 5.2.1, aplica-se o Passo 5.2.2. Com base nas soluções iniciais  $A$  e  $B$  (antes da combinação), verificam-se todas as rotas em que o cliente  $i$  está inserido e todos os arcos no qual  $x_{ij} = 1$  (o cliente  $i$  é atendido antes do cliente  $j$ ) ou  $x_{ji} = 1$  (o cliente  $i$  é atendido após o cliente  $j$ ). Assim, para cada  $j$  pertencente a uma das rotas combinadas no Passo

5.2.1, com exceção do CD ( $j = 0$ ), calcula-se o custo de inserção do cliente  $i$  (acrédito de custo fixo do veículo, custo de roteirização e tempo) antes do cliente  $j$  (quando  $x_{ij} = 1$ ) ou o custo de inserção do cliente  $i$  depois do cliente  $j$  (quando  $x_{ji} = 1$ ). Considera-se  $x_{i0} = 1$  ou  $x_{0i} = 1$  apenas para rotas com carga completa.

Para cada possível posição de inserção do cliente  $i$ , entrega-se o mínimo entre a capacidade ociosa do maior veículo capaz e a demanda do cliente  $i$ , e escolhe-se aquela com menor custo, desde que as restrições de janelas de tempo e tipo de veículo sejam respeitadas. O procedimento é repetido até que a demanda total do cliente  $i$  seja atendida ou enquanto houver uma posição de inserção do cliente  $i$ .

Verificadas todas as rotas das soluções  $A$  e  $B$  em que o cliente  $i$  está inserido, caso a demanda total não tenha sido atendida, adiciona-se uma nova rota (menor rota capaz de atender a demanda remanescente do cliente  $i$ ). O procedimento é repetido até que a demanda total do cliente  $i$  seja atendida.

O método de combinação garante que as soluções finais sejam factíveis.

O critério de distância é calculado pela soma do número de arcos não-comuns de cada solução antes da combinação. Caso um arco pertença a mais de uma rota, soma-se uma unidade para cada arco não-comum. Considera-se o arco  $x_{i0}$  ou  $x_{0i}$  apenas para rotas com carga completa ( $0 - i - 0$ ).

Para cada solução combinada de *RefSet* aplica-se o método de melhoria descrito no Passo 2. No final deste passo, o conjunto de referência é atualizado e, caso haja pelo menos uma nova solução no *RefSet*, o método de combinação é aplicado novamente. Caso contrário, completa-se uma iteração e o conjunto de referência é reconstruído, selecionando as  $b_1$  melhores soluções de *RefSet* e excluindo as  $b_2$  piores soluções. Um novo conjunto  $P$  de soluções diversas é gerado através do método de diversificação e o conjunto de soluções diversas  $b_2$  é reconstruído. O algoritmo pára ao final de cinco iterações ou ao atingir o tempo máximo de 1 hora, até que a última iteração seja finalizada. Cabe lembrar que, para as três primeiras iterações, utiliza-se o método de diversificação GRACO1 e, a partir da quarta iteração, o método GRACO2.

## ANÁLISE DE RESULTADOS

Os algoritmos foram aplicados para resolução de um problema real de distribuição, considerando um dia típico de operação.

Com base nos valores de fretes pagos às transportadoras (transporte terceirizado), em função dos clientes ou regiões e dos tipos de veículos, foi elaborado um modelo de regressão linear que determina o valor do frete do CD para os respectivos clientes, em função da distância total percorrida (ida e volta). A variável dependente utilizada no modelo é o valor do frete e a variável independente é a distância total percorrida. Para que os resultados possam ser comparados, o modelo de regressão foi aplicado à solução obtida pela empresa.

Foi escolhida uma data significativa que representa a média das operações no mês. Além desta data, foram analisados outros seis dias na mesma semana, com o objetivo de verificar o impacto na qualidade das soluções decorrente da mudança na demanda dos clientes.

A seguir são apresentados os parâmetros de calibração da metaheurística *scatter search*, para um dia típico de operação (4<sup>a</sup> feira) que representa a média das operações no mês. Com base nesses parâmetros são apresentados os resultados computacionais para os sete dias da semana.

### Parâmetros de calibração de SS

Para diferentes valores de *PSize* (10, 20, 30, 40), fixando os demais parâmetros ( $b_1 = 5$ ,  $b_2 = 5$ , atualização dinâmica e por qualidade e diversidade), os resultados obtidos podem ser vistos na Tabela 1.

De acordo com a Tabela 1, podemos verificar que o melhor resultado obtido foi *PSize* = 30. Para *PSize* = 40, o custo foi um pouco inferior, porém, o tempo computacional foi significativamente superior. Desta forma foi escolhido *PSize* = 30.

Portanto, para os demais testes do método *scatter search* será utilizado *PSize* = 30. O método de melhoria (trocas na mesma rota, realocação de demanda, combinação e eliminação de rotas, inserção e adição de rotas), o método de geração de subconjuntos (dois a dois), o método de combinação e o critério de parada (cinco iterações com um limite de tempo de 1 hora) não se alteraram. Para os demais parâmetros, foram testados os seguintes valores:

- Valores do conjunto de referência ( $b_1$ ,  $b_2$ ): (3,2), (5,5), (7,3) e (3,7)
- Critérios de atualização do conjunto de referência: QD ou Q
- Freqüência de atualização do conjunto de referência: Estática ou Dinâmica

Os resultados encontram-se na Tabela 2.

Dentre as versões do método *scatter search* testadas, o

**Tabela 1: Resultados para diferentes valores de PSize.**

<b>PSize</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>
Custo (R\$)	267.410	263.621	261.549	261.476
Tempo (s)	1021	2187	3004	4133

melhor resultado obtido foi para  $b_1 = 5$ ,  $b_2 = 5$ , critério de atualização por qualidade e diversidade (QD) e freqüência de atualização dinâmica.

#### **Resultados computacionais utilizando $PSize = 30$ , $b_1 = 5$ , $b_2 = 5$ , critério de atualização por qualidade e diversidade (QD) e freqüência de atualização dinâmica**

O custo total de abastecimento, para a heurística construtiva e para a metaheurística *scatter search* (SS), considerando os sete dias da semana, está na Tabela 3.

A economia obtida durante a semana, para a metaheurística *scatter search*, foi em torno de R\$ 51.000. Se projetarmos este resultado para os demais dias do ano, chegaremos a uma economia em torno de R\$ 2,6 milhões por ano.

As diferenças percentuais em relação à solução original, para cada dia da semana, estão no Gráfico 1.

A tabela 4 apresenta o tempo de processamento para cada um dos algoritmos e dias da semana.

Com o objetivo de verificar o tempo computacional em função do número de clientes, o algoritmo foi implementado em um determinado dia que atendia 400 clientes. Considerando esse mesmo dia de operação, o algoritmo foi resolvido para 350 clientes (eliminou-se 50 clientes aleatoriamente), e assim por diante, até chegar em 50 clientes. O gráfico 2 apresenta o tempo computacional da metaheurística *scatter search*, em segundos, para problemas com  $n = 50, 100, 150, 200, 250, 300, 350, 400$  clientes.

#### **CONCLUSÕES E FUTURAS PESQUISAS**

O presente trabalho apresentou métodos de resolução para um problema de abastecimento de clientes de um grupo varejista brasileiro. O problema é classificado como um problema de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas (*heterogeneous vehicle routing problem with time windows and fractional deliveries*).

**Tabela 2: Resultados de *scatter search* para diferentes parâmetros.**

<b>PARÂMETROS</b>			<b>CUSTO</b>
$b_1 = 3$ $b_2 = 2$	QD	Estático	115.088
		Dinâmico	115.323
	Q	Estático	115.064
		Dinâmico	114.927
$b_1 = 5$ $b_2 = 5$	QD	Estático	114.025
		Dinâmico	<b>113.717</b>
	Q	Estático	114.443
		Dinâmico	114.274
$b_1 = 7$ $b_2 = 3$	QD	Estático	114.185
		Dinâmico	114.410
	Q	Estático	114.350
		Dinâmico	114.175
$b_1 = 3$ $b_2 = 7$	QD	Estático	114.501
		Dinâmico	114.466
	Q	Estático	114.598
		Dinâmico	114.683

**Tabela 3: Custo total das soluções.**

<b>MÉTODO</b>	<b>CUSTO TOTAL (R\$)</b>						
	<b>DOM.</b>	<b>2<sup>a</sup></b>	<b>3<sup>a</sup></b>	<b>4<sup>a</sup></b>	<b>5<sup>a</sup></b>	<b>6<sup>a</sup></b>	<b>SÁB.</b>
Original	40.725	62.112	128.724	121.534	159.289	154.248	147.372
AHD	39.571	59.751	124.842	117.125	154.257	150.022	144.539
<i>Scatter search</i>	37.427	57.034	120.041	113.717	150.338	145.347	139.015

AHD: Adaptação da heurística de inserção seqüencial de Dullaert *et al.* (2002)

*fleet vehicle routing problem with time windows and split deliveries – HFVRPTWSD).*

Primeiramente, foi proposta uma heurística construtiva, que é uma adaptação da heurística de inserção seqüencial

de Dullaert *et al.* (2002). A daí, buscou-se implementar a metaheurística *scatter search* (SS).

Este trabalho permite concluir que a utilização de heurísticas construtivas é uma forma simples e rápida de se obter

Gráfico 1: Diferença percentual dos algoritmos em relação à solução original.

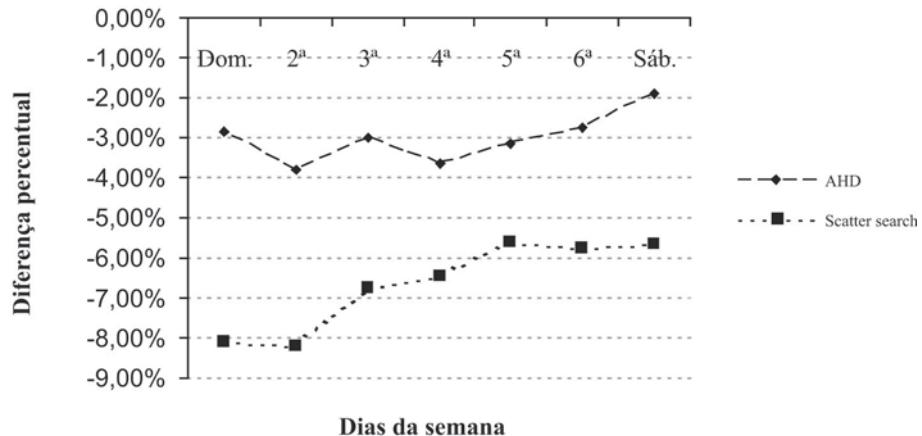
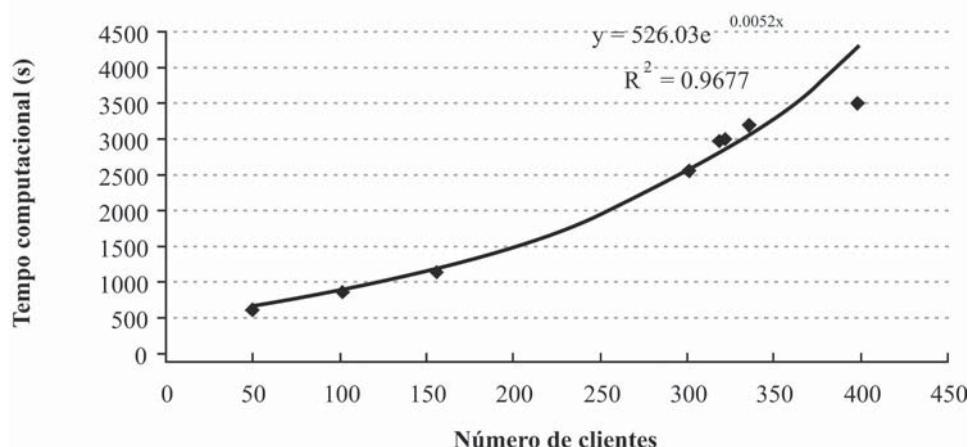


Tabela 4: Tempo de processamento.

ALGORITMO	TEMPO (S)						
	DOM.	2ª	3ª	4ª	5ª	6ª	SÁB.
AHD	22	24	42	37	49	44	40
Scatter search	860	1148	3092	3004	3494	3198	2986

Gráfico 2: Tempo computacional da metaheurística *scatter search*.



melhorias nas soluções quando comparadas àquelas obtidas empiricamente. A adoção de SS permitiu a busca de novos espaços de soluções, fazendo com que a solução não fique restrita a ótimos locais. Desta forma, foi possível uma redução ainda maior no custo total da frota, comparado com a solução atual da empresa.

Finalmente, é importante indicar que os métodos utilizados neste trabalho podem ser objetos de novas pesquisas, podendo ser adaptados a outros problemas de roteirização de veículos, como problemas com múltiplos CDs. Os algoritmos também podem ser adaptados a diferentes categorias de produtos. Alguns novos critérios e

restrições também podem ser adicionados, como a priorização de pedidos.

Sugere-se, também, como futura pesquisa, a exploração de novas alternativas de caminhos na malha logística, como o uso do *backhaul*, com o objetivo de minimizar o custo total de frete. O objetivo, com a implementação do *backhaul*, é minimizar as viagens vazias de retorno (não-produtivas) que os veículos fazem no sistema. Assim, buscam-se novos caminhos que atendam a mais de um fluxo em cada rota, por exemplo, o veículo sai cheio de um CD com destino a um cliente, deste cliente percorre vazio até um fornecedor e deste fornecedor volta cheio ao CD.

**Artigo recebido em 21/06/2006  
Aprovado para publicação em 20/10/2006**

## ■ Referências Bibliográficas

- ARCHETTI, C.; MANSINI, M.; SPERANZA, M. G. Complexity and Reducibility of the Skip Delivery Problem. *Transportation Science*, v. 39, n. 2, p. 182-187, 2005.
- ARCHETTI, C.; SAVELSBERGH, M. W. P.; SPERANZA, M. G. Worst-Case Analysis of Split Delivery Routing Problems. *Transportation Science*, v. 40, n. 2, p. 226-234, 2006.
- ARCHETTI, C.; SPERANZA, M. G.; HERTZ, A. A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*, v. 40, n. 1, p. 64-73, 2006.
- BELENGUER, J. M.; MARTINEZ, M. C.; MOTA, E. A Lower Bound for the Split Delivery Vehicle Routing Problem. *Operations Research*, v. 48, n. 5, p. 801-810, 2000.
- CAMPOS, V.; GLOVER, F.; LAGUNA, M.; MARTÍ, R. An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem. *Journal of Global Optimization*, v. 21, n. 4, p. 397-414, 2001.
- CAMPOS, V.; LAGUNA, M.; MARTÍ, R. Context-independent scatter and tabu search for permutation problems. *INFORMS Journal on Computing*, v. 17, n. 1, p. 111-122, 2005.
- CORBERÁN, A.; FERNÁNDEZ, E.; LAGUNA, M.; MARTÍ, R. Heuristic solutions to the problem of routing school buses
- with multiple objectives. *Journal of the Operational Research Society*, v. 53, n. 4, p. 427-435, 2002.
- DESROCHERS, M.; VERHOOG, T. W. A new heuristic for the fleet size and mix vehicle routing problem. *Computers & Operations Research*, v. 18, n. 3, p. 263-274, 1991.
- DROR, M.; TRUDEAU, P. Savings by Split Delivery Routing. *Transportation Science*, v. 23, n. 2, p. 141-145, 1989.
- DROR, M.; TRUDEAU, P. Split Delivery Routing. *Naval Research Logistics*, v. 37, p. 383-402, 1990.
- DROR, M.; LAPORTE, G.; TRUDEAU, P. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, v. 50, n. 3, p. 229-254, 1994.
- DULLAERT, W.; JANSENS, G. K.; SORENSEN, K.; VERNIMMEN, B. New heuristics for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Journal of Operational Research Society*, v. 53, p. 1232-1238, 2002.
- FEO, T. A.; RESENDE, M. G. C. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, v. 8, n. 2, p. 67-71, 1989.
- FRIZZELL, P. W.; GIFFIN, J. W. The bounded split delivery vehicle routing problem with grid network distances. *Asia Pacific Journal of Operational Research*, v. 9, n. 1, p. 101-106, 1992.
- FRIZZELL, P. W.; GIFFIN, J. W. The Split Delivery Vehicle Scheduling Problem with Time Windows and Grid Network Distances. *Computers & Operations Research*, v. 22, n. 6, p. 655-667, 1995.
- GENDREAU, M.; LAPORTE, G.; MUSARAGANYI, C.; TAILLARD, E. D. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, v. 26, n. 12, p. 1153-1173, 1999.
- GLOVER, F. A Template for Scatter Search and Path Relinking. In: *Artificial Evolution*, Lecture Notes in Computer Science 1363, HAO, J.-K., E. LUTTON, E. RONALD, M. SCHOENAUER, D. SNYERS (Eds.), Springer-Verlag, p. 13-54, 1998.
- GOLDEN, B. L.; ASSAD, A.; LEVY, L.; GHEYSENS, F. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, v. 11, n. 1, p. 49-65, 1984.
- HO, S. C.; HAUGLAND, D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, v. 31, n. 2, p. 1947-1964, 2004.
- JAIN, A. S.; MEERAN, S. A multi-level hybrid framework applied to the general flow-shop scheduling problem.
- LAGUNA, M.; MARTÍ, R. Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions. *Journal of Global Optimization*, v. 33, n. 2, p. 235-255, 2005.
- LENSTRA, J. K.; RINNOOY KAN, A. H. G. Complexity of Vehicle and Scheduling Problems. *Networks*, v. 11, n. 2, p. 221-227, 1981.
- LIU, F. H.; SHEN, S. Y. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, v. 50, n. 7, p. 721-732, 1999.
- MARTÍ, R.; LOURENÇO, H.; LAGUNA, M. Assigning proctors to exams with scatter search. In: *Computing Tools for Modeling, Optimization and Simulation*, LAGUNA, M.; VELAVERDE, J.L.G. (Eds.), Kluwer Academic Publishers, p. 215-227, 2000.
- MARTÍ, R.; LAGUNA, M.; CAMPOS, V. Scatter search vs genetic algorithms: an experimental evaluation permutation problems. In: *Metaheuristic Optimization via Adaptive memory and evolution: Tabu search and scatter search*, REGO, C.; ALIDAEE, B. (Eds.), Kluwer Academic Publishers, p. 263-285, 2005.

## ■ Referências Bibliográficas

- MULLASERIL, P. A.; DROR, M.; LEUNG, J. Split-delivery Routing Heuristics in Livestock Feed Distribution. *Journal of the Operational Research Society*, v. 48, n. 2, p. 107-116, 1997.
- REGO, C.; LEÃO, P. A Scatter Search Tutorial for Graph-Based Permutation Problems. Hearin Center for Enterprise, University of Mississippi, HCES-10-00, USA, 2000. In *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*, REGO, C; ALIDAEE, B. (Eds.), Kluwer Academic Publishers, p. 1-24, 2005.
- ROCHAT, Y.; SEMET, F. A Tabu Search Approach for Delivering Pet Food and Flour in Switzerland. *Journal of the Operational Research Society*, v. 45, n. 11, p. 1233-1246, 1994.
- ROCHAT, Y.; TAILLARD, E. D. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, v. 1, n. 1, p. 147-167, 1995.
- SALHI, S.; RAND, G. K. Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research*, v. 66, n. 3, p. 313-330, 1993.
- SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows Constraints. *Operations Research*, v. 35, n. 2, p. 254-265, 1987.
- SOLOMON, M. M.; DESROSIERS, J. Time Window Constrained Routing and Scheduling Problem. *Transportation Science*, v. 22, n. 1, p. 1-13, 1988.
- TAILLARD, É. D. A heuristic column generation method for the heterogeneous fleet vehicle routing problem. *RAIRO - Operations Research*, v. 33, n. 1, p. 1-14, 1999.
- TARANTILIS, C. D.; KIRANoudis, C. T.; VASSILIAidis, V. S. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, v. 152, n. 1, p. 148-158, 2004.
- YAMASHITA, D. S.; ARMENTANO, V. A.; LAGUNA, M. Scatter Search for Project scheduling with resource availability cost. *European Journal of Operational Research*, v. 169, n. 2, p. 623-637, 2006.
- WASSAN, N. A.; OSMAN, I. H. Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, v. 53, p. 768-782, 2002.

## ■ Sobre os autores

### Patrícia Prado Belfiore

Doutora em Engenharia de Produção – POLI-USP  
Professora do Departamento de Engenharia de Produção do Centro Universitário da FEI  
End.: Av. Humberto de Alencar Castelo Branco, 3972 – Assunção – São Bernardo do Campo – SP  
CEP: 09850-901  
Tel. / Fax: (55) (11) 4353-2900  
E-mail: patricia.belfiore@labfin.com.br

### Hugo Tsugunobu Yoshida Yoshizaki

Professor Livre-Docente do Departamento de Engenharia de Produção – POLI-USP  
End.: Av. Prof. Almeida Prado, Trav. 2, 128 – Cidade Universitária – São Paulo – SP  
CEP: 05508-900  
Tel. / Fax: (55) (11) 3091-5363  
E-mail: hugo@usp.br