



Problema de roteamento de veículos assimétrico com frota heterogênea limitada: um estudo de caso em uma indústria de bebidas

Asymmetric vehicle routing problem with heterogeneous limited fleet: a case study in a beverage industry

Raphael Harry Frederico Ribeiro Kramer¹
Anand Subramanian¹
Puca Huachi Vaz Penna²

Resumo: Este artigo aborda um estudo de caso em uma indústria de bebidas relativo ao Problema de Roteamento de Veículos Assimétrico com Frota Heterogênea Limitada (PRVAFHL). O objetivo é definir as rotas dos veículos de modo a reduzir os custos de distribuição. O PRVAFHL pertence à classe NP-difícil, isto é, sua resolução por meio de métodos exatos é uma tarefa extremamente árdua. Problemas desta natureza são geralmente tratados na prática de forma heurística. Dentre as diversas abordagens existentes, optou-se por realizar uma adaptação de uma heurística da literatura que se mostrou eficiente, sendo capaz de gerar soluções de qualidade elevada em um tempo de execução aceitável. Experimentos computacionais foram realizados em um conjunto de 7 instâncias obtidas junto à empresa em questão. Os resultados obtidos mostram que houve uma redução considerável no número de veículos utilizados e na distância total percorrida em relação às soluções adotadas pela empresa.

Palavras-chave: Otimização; Indústria de bebidas; *Iterated Local Search*; Logística.

Abstract: *This article deals with a case study in a beverage industry concerning the Asymmetric Vehicle Routing Problem with Heterogeneous Limited Fleet (AVRPHLF). The objective of the present paper is to define the vehicle routes so as to reduce distribution costs. The AVRPHLF belongs to the NP-hard class, that is, its resolution through exact methods is an extremely hard task. In practice, problems of this nature are generally treated via heuristics. Among the various solution approaches, we decided to perform an adaptation of a heuristic from the literature that has proved to be efficient, capable of generating high-quality solutions in a reasonable execution time. Computational experiments were conducted in a set of seven instances obtained from the company in question. The results obtained show that there was a considerable reduction in the number of vehicles used and in the total distance traveled with respect to the solutions adopted by the company.*

Keywords: *Optimization; Beverage industry; Iterated Local Search; Logistics.*

1 Introdução

O Problema de Roteamento de Veículos (PRV) é um dos problemas de distribuição logística mais conhecido e estudado na área de otimização combinatória em virtude da sua aplicabilidade e importância, especialmente na cadeia de suprimentos. Devido a sua complexidade de resolução, o PRV é considerado NP-Difícil, isto é, a dificuldade para encontrar a solução ótima cresce exponencialmente à medida que o número de clientes aumenta.

O PRV em sua forma clássica é definido como segue: seja um grafo não orientado $G = (V, E)$, em que $V = \{v_0, v_1, \dots, v_n\}$ é o conjunto de vértices de G ,

enquanto $E = \{(v_i, v_j): v_i, v_j \in V, i < j\}$ é o conjunto de arestas de G . O vértice v_0 representa o depósito com m veículos idênticos de capacidade Q e os restantes representam os clientes. Cada cliente v_i possui uma demanda não negativa q_i . Um custo não negativo c_{ij} de deslocamento do cliente i ao j está associado a cada aresta $\{v_i, v_j\}$, em que c_{ij} pode ser interpretado em termos de distância, tempo ou custos de viagem. O PRV consiste em determinar um conjunto de rotas (cada uma associada a um veículo), tal que cada cliente seja visitado uma única vez, por um único veículo, de modo que o custo total de viagem seja minimizado

¹ Departamento de Engenharia de Produção, Centro de Tecnologia, Universidade Federal da Paraíba – UFPB, Campus I, Bloco G, Cidade Universitária, CEP 58051-970, João Pessoa, PB, Brasil, e-mail: raphaelharry@gmail.com; anand@ct.ufpb.br

² Instituto do Noroeste Fluminense de Educação Superior, Universidade Federal Fluminense – UFF, Rua João Jazbik, s/n, CEP 28470-000, Santo Antônio de Pádua, RJ, Brasil, e-mail: ppenna@ic.uff.br

Recebido em Maio 05, 2014 – Aceito em Nov. 20, 2014

Suporte financeiro: Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Processo 471158/2012-7).

e a demanda de cada cliente seja atendida. Cada veículo deve iniciar e terminar o percurso no depósito v_0 e a demanda total de cada rota não deve exceder a capacidade do veículo. Tal problema é denominado Problema de Roteamento de Veículos Capacitado. Uma representação gráfica de uma solução de um PRV contendo 3 rotas é apresentada na Figura 1.

Quando o custo de deslocamento do cliente i para o cliente j difere do custo no sentido oposto, tem-se o PRV Assimétrico (PRVA) que, ao contrário do PRV, é definido sobre um grafo orientado $G = (V, A)$, em que $A = \{(v_i, v_j); v_i, v_j \in V, i \neq j\}$ é o conjunto de arcos de G .

Para o caso de um conjunto com K tipos de veículos com capacidades Q_k distintas, tem-se o PRV com Frota Heterogênea (PRVFH). Nesta variante, duas situações podem ocorrer: quando o número de veículos de cada tipo é conhecido, a frota é limitada, caso contrário, é ilimitada. Além disso, custos fixos e/ou variáveis associados a cada tipo de veículo podem ou não ser considerados.

Ao combinar o caso assimétrico com veículos de capacidades distintas e frota limitada, tem-se o Problema de Roteamento de Veículos Assimétrico com Frota Heterogênea Limitada (PRVAFHL).

Este trabalho trata de uma extensão ao trabalho de Kramer et al. (2012), o qual se refere a um estudo de caso relativo ao PRVAFHL em uma indústria de bebidas, considerada a maior envasadora de água mineral do País, contando com diversas unidades fabris distribuídas ao longo do território brasileiro. O objetivo consiste em determinar as rotas de distribuição e comparar os resultados obtidos com os adotados pela empresa.

PRVs, assim como muitos dos problemas de otimização combinatória (p.ex. problemas de empacotamento, sequenciamento da produção, localização, escalonamento de mão de obra, entre

outros), são geralmente tratados por métodos heurísticos, especialmente quando aplicados a casos reais e de grandes dimensões. Entende-se heurística como sendo um conjunto de métodos e regras, inspiradas em processos intuitivos, que conduzem à resolução de problemas. Ao contrário dos métodos exatos, que garantem a melhor solução possível para o problema (de acordo com o modelo matemático que o representa) ao custo de um elevado tempo de resolução, os métodos heurísticos não garantem a otimalidade das soluções, porém são capazes de gerar soluções próximas da ótima em tempos relativamente curtos. Uma explicação detalhada sobre heurísticas e meta-heurísticas pode ser encontrada em Souza (2007), Blum & Roli (2003) e Gendreau & Potvin (2010).

Para a resolução do problema proposto, decidiu-se utilizar a heurística ILS-RVND (Penna et al., 2013), devido a sua flexibilidade, simplicidade, robustez e ao seu bom desempenho computacional. Durante a coleta de dados, utilizou-se a *Application Programming Interface* (API) do *Google Maps* para a aquisição das matrizes de distância e tempo entre clientes.

O restante do artigo está organizado da seguinte forma. Inicialmente, uma breve descrição sobre os trabalhos relacionados é apresentada na Seção 2. A Seção 3 caracteriza a empresa objeto do estudo de caso, bem como as particularidades do processo de distribuição. A Seção 4 descreve o algoritmo desenvolvido para a aquisição das matrizes de distância e tempo. Na Seção 5, é apresentado o algoritmo ILS-RVND bem como os procedimentos de geração da solução inicial, de busca local e de perturbação. A Seção 6 contém os resultados obtidos, e são feitas comparações com os adotados pela empresa. Por fim, a Seção 7 conclui o trabalho.

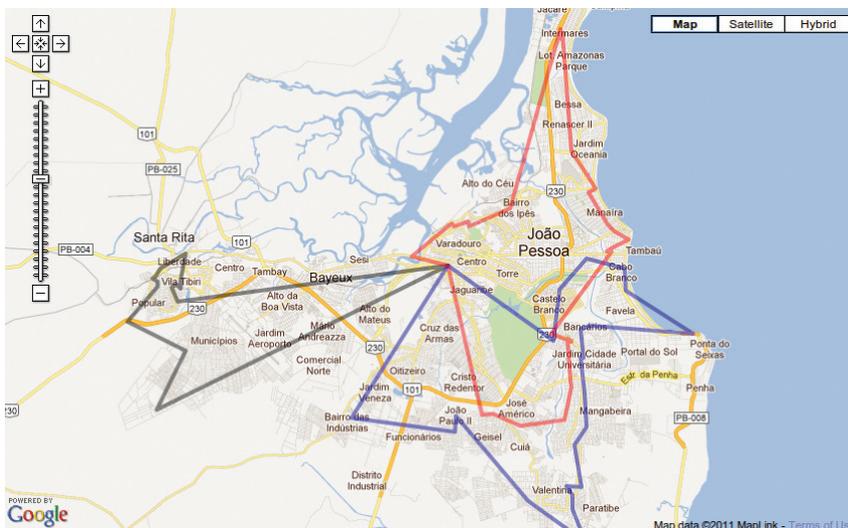


Figura 1. Solução de um PRV com 3 veículos e 56 clientes.

2 Trabalhos relacionados

O PRV foi introduzido inicialmente por Dantzig & Ramser (1959) no final dos anos 1950 para a resolução do problema de distribuição de gasolina. Nele foi apresentada a primeira formulação matemática e abordagem algorítmica para o PRV. A partir daí, o número de estudos e pesquisas relacionados ao tema cresceu significativamente. Toth & Vigo (2002) e Golden et al. (2008), por exemplo, reúnem uma quantidade razoável de aplicações nos mais diversos ramos de atuação.

Dada a elevada complexidade de resolução do problema e, dependendo das características da instância, a dificuldade de geração de soluções viáveis, a utilização de técnicas computacionais pode contribuir para uma redução nos custos de transporte na ordem de 5% a 20%, conforme afirmado por Toth & Vigo (2002).

Aplicações reais de PRVs em indústrias alimentícia, de resíduos sólidos, de bebidas, de laticínios e de distribuição de jornal podem ser encontradas em Golden et al. (2002). Os autores mostram que a utilização de técnicas de pesquisa operacional na logística de distribuição pode contribuir para a redução dos custos e aumento da produtividade, além de apresentarem diversos trabalhos práticos de PRV encontrados na literatura.

O problema de distribuição de água mineral na cidade de Itú foi estudado por Gonçalves et al. (2005). Para solucioná-lo, os autores definiram *clusters* de clientes após a determinação de medianas para, então, efetuar o roteamento. Foram utilizadas a heurística das economias de Clarke & Wright (1964), a da Inserção do Ponto Mais Próximo, e a meta-heurística Busca Tabu.

A importância prática do PRVFH pode ser verificada pela grande variedade de estudos de caso encontrados na literatura. Tarantilis & Kiranoudis (2001), Prins (2002) e Tarantilis & Kiranoudis (2007) mostram aplicações com frota limitada na indústria láctea, moveleira, alimentícia e de construção civil. Uma aplicação com frota ilimitada em uma rede varejista brasileira é descrita em Belfiore & Yoshizaki (2009), com particularidades de que cada loja possui restrições de janela de tempo e a determinados tamanhos de veículos, podendo ser atendidas por mais de um veículo. Campos et al. (2006) solucionaram o PRVFH com restrições de janela de tempo de um dos maiores grupos varejistas do Brasil por meio de algoritmos genéticos e computação paralela. A abordagem utilizada foi capaz de reduzir os custos das soluções adotadas pela empresa em cerca de 13%.

A primeira versão assimétrica do PRVFH foi inicialmente abordada por Marmion et al. (2010). Os autores desenvolveram quatro algoritmos com o objetivo principal de comparar o desempenho de dois operadores clássicos: troca e inserção.

Além da metodologia utilizada para solucionar o problema de roteamento, um fator crucial para a garantia da qualidade das soluções consiste no modo em que as informações são coletadas. Dado que a métrica utilizada para avaliação das soluções de PRVs se baseiam, em sua maioria, na distância total percorrida pelos veículos, a definição de maneira precisa da localização e da distância entre clientes são pré-requisitos fundamentais para o aumento da confiabilidade da qualidade das soluções. Neste contexto, Galvão et al. (1997) investigaram a integração de modelos de roteamento a Sistemas de Informação Geográfica (SIG) aplicados à um problema simulado de distribuição de refrigerantes, no Rio de Janeiro. Tarantilis & Kiranoudis (2002), por sua vez, propuseram um sistema de apoio à decisão que integra SIG com métodos heurísticos para solucionar PRVs em Atenas.

Devido à importância prática dos PRVs, verifica-se a existência de um elevado número de *softwares* disponíveis no mercado capazes de solucioná-los. Baker (2002) compara diferentes *softwares* atentando para os algoritmos de otimização utilizados. Uma pesquisa mais recente quanto ao preço, desempenho computacional, algoritmos utilizados, variantes solucionáveis, entre outros aspectos, pode ser encontrada em Partyka & Hall (2010).

3 Caracterização da empresa e de seu processo de distribuição

A unidade fabril analisada está sediada na Paraíba e possui milhares de clientes, desde pequenos revendedores até grandes redes de supermercados, distribuídos ao longo da região nordeste. Além do envase de água mineral, a empresa produz refrigerantes, sucos e energéticos. A distribuição dos produtos aos clientes é de responsabilidade da própria empresa que, para isso, conta com o auxílio de um *software* de roteamento.

A frota disponível para realizar a distribuição é composta por 37 veículos, sendo 36 terceirizados, com capacidades distintas, e um veículo próprio com capacidade de 25 toneladas. Esta frota é utilizada para entregar os pedidos dos clientes de todo o Estado da Paraíba e de alguns estados da região nordeste. A Tabela 1 apresenta as características dos veículos.

Tabela 1. Frota de veículos.

Veículo	Quantidade	Capacidade (Kg)
3/4	14	4000
Toco	7	8000
Truck A	9	12000
Truck B	6	13000
Carreta	1	25000

Os dados referentes aos pedidos dos clientes foram restritos à região da Grande João Pessoa (GJP), abrangendo, além desta, outras 4 cidades. Por dia, são atendidos em média 300 clientes nesta região.

Os pedidos de vendas são entregues diariamente ao setor de distribuição após serem coletados pelos vendedores. Como praticamente todos os veículos são terceirizados e o pagamento é realizado em termos da distância percorrida e utilização dos veículos, é de interesse da empresa reduzi-las.

4 Obtenção das distâncias e tempos entre clientes

Os dados utilizados na resolução do problema foram fornecidos pela empresa estudada. No entanto, durante a coleta desses dados não foi possível obter as matrizes de distância e tempo entre clientes. Estas informações pertencem à empresa desenvolvedora do *software* de roteamento, que não foi contatada por envolver questões financeiras. Diante do exposto, optou-se pela utilização do *Google Maps* para obtenção destes dados.

O *Google Maps* é um serviço do *Google* que oferece uma poderosa e amigável tecnologia de mapas, a qual permite, entre diversas funcionalidades, localizar endereços e obter rotas entre duas ou mais localidades, fornecendo uma descrição do trajeto, bem como a distância e o tempo necessário para realizá-lo. O Serviço está disponível apenas via Internet.

A *Google Maps JavaScript API* é um serviço gratuito, disponível para qualquer *site* de acesso livre. Contudo, algumas funções estão limitadas a uma determinada quantidade e frequência de requisições. Por exemplo, ao solicitar a rota de um local para outro, tem-se uma requisição. O serviço de solicitação de rotas está limitado a 15.000 requisições diárias. Ao atingir os limites diários estabelecidos, o usuário é bloqueado temporariamente a partir do seu endereço de IP (Protocolo de Internet, do inglês *Internet Protocol*), não podendo utilizar o serviço por um período de 24 horas.

Considerando que na empresa estudada, em média, 300 clientes realizam pedidos diariamente, as matrizes de distância e tempos terão, cada uma, aproximadamente, 90.000 elementos. Logo, para o preenchimento completo das matrizes, a quantidade de requisições ultrapassaria os limites estabelecidos pelo *Google*, caso fossem solicitados ponto a ponto. No entanto, ao invés de enviar uma solicitação de rota fornecendo a localização de 2 clientes por vez, foi desenvolvido um algoritmo que fornece um trajeto (limitado a 25 pontos) como requisição. Desse modo, ao se passar um trajeto contendo 25 pontos (clientes), é possível obter 24 distâncias e 24 tempos por requisição, tornando-se assim viável a obtenção das matrizes sem exceder os limites do serviço. Por exemplo, ao

se passar um trajeto com 5 pontos, A-B-C-D-E, é possível obter 4 distâncias (d) e 4 tempos (t) entre clientes: d_{AB} , d_{BC} , d_{CD} , d_{DE} e t_{AB} , t_{BC} , t_{CD} , t_{DE} .

O Algoritmo 1 apresenta o método desenvolvido para a obtenção das matrizes de distância e tempo entre clientes. As coordenadas de localização dos clientes foram fornecidas pela empresa.

5 O algoritmo ILS-RVND

O algoritmo proposto consiste em uma adaptação do ILS-RVND apresentado por Penna et al. (2013). O ILS-RVND é baseado na meta-heurística *Iterated Local Search* (ILS) (Lourenço et al., 2002) que faz uso de um procedimento *Variable Neighborhood Descent* (Hansen et al., 2010), com uma ordem aleatória de vizinhanças (RVND) na fase de busca local. Com a finalidade de contribuir para a diversificação do espaço de busca e, consequentemente, para a geração de soluções de boa qualidade, o ILS-RVND utiliza mais de um tipo de procedimento para a construção de soluções iniciais, bem como de busca local e de perturbação. O algoritmo ILS-RVND original foi adaptado de modo a considerar particularidades do estudo de caso, tais como os limites de duração das rotas e as distâncias assimétricas entre clientes.

O pseudocódigo do ILS-RVND se encontra ilustrado no Algoritmo 2. O algoritmo tem como parâmetros de entrada *MaxIter*, *MaxIterILS* e v , e este último representa o número total de veículos disponíveis. *MaxIter* representa o número de vezes que o algoritmo será executado (linhas 4-21), de modo que, a cada iteração, uma solução inicial é gerada. Depois da geração da solução inicial, o laço principal

Algoritmo 1. Obtendo matrizes de distância e tempos entre clients.

```

1:  Procedimento ObterMatrizes()
2:  i = 1;
3:  j = 1;
4:  enquanto i ≤ n faça
5:  enquanto j ≤ n faça
6:      adiciona endereço de i ao trajeto;
7:      adiciona endereço de j ao trajeto;
8:  se tamanho do trajeto = 24 ou j = n então
9:      adiciona endereço de i ao trajeto;
10:     envia trajeto como requisição de rota;
11:     esvazia trajeto;
12:  fim se
13:     j = j + 1;
14:  fim enquanto
15:     i = i + 1;
16:     j = 1;
17:  fim enquanto
18:  fim ObterMatrizes.

```

Algoritmo 2. ILS-RVND.

```

1: Procedimento ILS-RVND(MaxIter, MaxIterILS, v)
2: CarregaDados();
3:  $f^* \leftarrow \infty$ ;
4: para  $i = 1, \dots, \text{MaxIterfaça}$ 
5:  $s \leftarrow \text{GeraSoluçãoInicial}(v, \text{MaxIter}, \text{semente})$ ;
6:  $s' \leftarrow s$ ;
7: iterILS  $\leftarrow 0$ ;
8: enquanto iterILS  $\leq \text{MaxIterILSfaça}$ 
9:  $s \leftarrow \text{RVND}(s)$ ;
10: se  $f(s) < f(s')$  então
11:  $s' \leftarrow s$ ;
12: iterILS  $\leftarrow 0$ ;
13: fim se
14:  $s' \leftarrow \text{Perturbação}(s', \text{semente})$ ;
15: iterILS  $\leftarrow \text{iterILS} + 1$ ;
16: fim enquanto
17: se  $f(s') < f^*$  então
18:  $s^* \leftarrow s'$ ;
19:  $f^* \leftarrow f(s')$ ;
20: fim se
21: fim para
22: retorne  $s^*$ ;
23: fim ILS-RVND.

```

do algoritmo ILS é executado *MaxIterILS* vezes (linhas 8-16), em que se procura melhorar a solução inicial através de procedimentos de busca local (linha 9) e de perturbação (linha 14). Caso a melhor solução encontrada após a repetição (*MaxIterILS* vezes) dos procedimentos de busca local e de perturbação seja melhor que a solução inicial gerada, esta passa a ser a solução corrente e o algoritmo é reiniciado da linha 3.

5.1 Geração da solução inicial

O procedimento para construção de soluções iniciais faz uso de dois critérios de inserção, particularmente, o Critério de Inserção Mais Barata Modificado (CIMBM) e o Critério de Inserção Mais Próxima (CIMP). Além disso, duas estratégias de inserção foram empregadas: a Estratégia de Inserção Sequencial (EIS) e a Estratégia de Inserção Paralela (EIP).

O pseudocódigo do procedimento de Geração da Solução Inicial está representado no Algoritmo 3. Inicialmente, os veículos são ordenados por capacidade de maneira decrescente, de forma a designar a cada veículo uma rota contendo apenas um cliente, escolhido de acordo com a maior demanda (linhas 4-8). Posteriormente, um critério e uma estratégia de inserção são selecionados de maneira aleatória (linhas 9-10). Em seguida, uma solução inicial é gerada, utilizando o critério e estratégia selecionados (linhas 11-15). Se for gerada uma solução inviável,

o procedimento é reiniciado da linha 3. No caso do PRVAFHL, uma solução inviável é obtida quando os veículos disponíveis não são capazes de atender todos os clientes ou a duração de uma ou mais rotas ultrapassa o limite permitido. Caso não seja possível gerar uma solução inicial viável após um determinado número de tentativas consecutivas, um veículo extra é adicionado com o intuito de facilitar a geração de uma solução inicial completa (este veículo não poderá ser utilizado na solução final).

O CIMBM consiste em inserir em uma determinada rota um cliente $k \in \text{Lista de Candidatos (LC)}$ que apresente o menor custo de inserção representado pela Expressão (1), em que a primeira parcela computa o custo de inserção do cliente k entre dois clientes adjacentes i e j , enquanto a segunda é um artifício utilizado para evitar a inserção tardia de clientes localizados longe do depósito, representando uma redução no custo de inserção em função da distância ao depósito (ida e volta). O parâmetro γ pode assumir qualquer valor contido no seguinte conjunto: $\{0.00, 0.05, 0.10, \dots, 1.65, 1.70\}$ (Subramanian et al., 2010).

$$\min \left\{ (c_{ik} + c_{kj} - c_{ij}) - \gamma(c_{0k} + c_{k0}) \mid k \in LC \right\} \quad (1)$$

No CIMP, deve-se incluir à rota o cliente $k \in LC$ que apresente a menor distância a um determinado cliente i já inserido, conforme representado na Expressão (2), em que se considera que k é sempre inserido após i .

$$\min \{c_{ik} \mid k \in LC\} \quad (2)$$

Na EIS, apenas uma rota é considerada para inserção a cada iteração. Se o critério de inserção for o CIMBM, então, o valor de γ é escolhido aleatoriamente a partir do conjunto já especificado. Enquanto a *LC* não estiver vazia e existir pelo menos um cliente $k \in LC$ que possa ser adicionado à solução corrente sem violar nenhuma restrição, insere-se um cliente em cada rota (entre as possíveis) de acordo com o critério de inserção selecionado.

A EIP difere do caso anterior por considerar todas as rotas durante o cálculo dos custos de inserção. Desse modo, avalia-se a inserção do cliente k em todas as posições de todas as rotas, inserindo-o na rota que apresentar o menor custo de inserção e que não viole as restrições do problema (linhas 7-12).

5.2 Busca local

Uma das principais características da meta-heurística ILS-RVND é realizar uma ordenação aleatória das estruturas de vizinhanças (*Randomized Variable Neighbourhood Descent* – RVND), ao contrário das abordagens usuais do VND que empregam uma ordem determinística na troca de vizinhanças, evitando assim a utilização de parâmetros de ajustes, como também

Algoritmo 3. Geração da Solução Inicial.

```

1: Procedimento GeraçãoSoluçãoInicial( $v$ ,  $MaxIter$ , semente)
2:  $TentativasConsecutivas \leftarrow 0$ ;
3: Inicializa  $LC$ ;
4: Seja  $s = \{s^1, \dots, s^v\}$  o conjunto composto por  $v$  rotas vazias, ordenadas pela capacidade de forma decrescente;
5: parav  $v = 1, \dots, v-1$  faça
6:  $s^v \leftarrow k$  associado a  $\max_k \in LC\{q_k\}$ ;
7: Atualize  $LC$ ;  $\{LC \leftarrow LC - \{k\}\}$ .
8: fim para
9: CritérioInserção  $\leftarrow$  CIMBM ou CIMP (selecionado aleatoriamente);
10: EstratégiaDeInserção  $\leftarrow$  EIS ou EIP (selecionada aleatoriamente);
11: se
12:  $s \leftarrow$  InserçãoSequencial( $s$ ,  $v$ ,  $LC$ , CritérioInserção);
13: senão
14:  $s \leftarrow$  InserçãoParalela( $s$ ,  $v$ ,  $LC$ , CritérioInserção);
15: fim se
16: ses é inviável então
17:  $TentativasConsecutivas \leftarrow TentativasConsecutivas + 1$ ;
18: se  $TentativasConsecutivas = MaxIter$  então
19:  $v \leftarrow v + 1$ ;
20:  $TentativasConsecutivas \leftarrow 0$ ;
21: fim se
22: Vá para linha 3;
23: senão
24: retorne  $s$ ;
25: fim se
26: fim GeraçãoSoluçãoInicial.

```

prevenindo a convergência prematura para soluções ótimas locais de baixa qualidade.

O pseudocódigo do método RVND é mostrado no Algoritmo 4. Inicialmente uma vizinhança entre rotas é selecionada aleatoriamente em uma Lista de Vizinhanças LV (linha 4). Encontra-se então a melhor solução a partir do conjunto de movimentos da vizinhança selecionada e, caso a solução encontrada seja melhor que a solução corrente, esta passa a ser a solução corrente para, então, ser efetuada uma busca local intrarrotas (linhas 6-9). Uma vez que seja encontrada uma melhora com uma vizinhança entre rotas, todas as vizinhanças excluídas voltam a popular a LV (linha 9). Caso a solução encontrada com a vizinhança entre rotas não seja melhor que a solução corrente, tal vizinhança é excluída de LV (linha 12), e repetem-se os mesmos procedimentos (a partir da linha 4) até que a lista se torne vazia (linha 4).

O algoritmo de busca local intrarrotas, de maneira similar às entre rotas, consiste em selecionar aleatoriamente uma vizinhança contida numa lista $LV2$, encontrar a melhor solução por meio dos movimentos da vizinhança selecionada e, caso seja melhor que a solução corrente, torna-se a nova solução corrente. Caso não seja melhor, a vizinhança analisada

Algoritmo 4. RVND.

```

1: Procedimento RVND( $s$ )
2: Inicialize a Lista de Vizinhanças entre rotas  $LV$ ;
3: enquanto  $LV \neq 0$  faça
4: Selecione uma vizinhança  $N^{(n)} \in LV$  aleatoriamente;
5: Encontre o melhor vizinho de  $s^i$  de  $s \in N^{(n)}$ ;
6: se  $f(s^i) < f(s)$  então
7:  $s \leftarrow s^i$ ;
8:  $s \leftarrow$  BuscaIntrarrotas( $s$ );
9: Atualize  $LV$ ;
10: senão
11: Remova  $N^{(n)}$  de  $LV$ ;
12: fim se
13: fim enquanto
14: retorne;
15: fim RVND.

```

é excluída de $LV2$ e os mesmos procedimentos são repetidos até que se esvazie a lista. A lista $LV2$ é sempre inicializada após uma melhora por vizinhança entre rotas.

Tanto nos procedimentos de busca local entre rotas quanto nos intrarrotas, apenas movimentos que não

violem as restrições de capacidade do veículo e de duração das rotas são aceitos. O algoritmo verifica se o movimento é viável ou não, sempre após a constatação de uma melhora na solução corrente. Deste modo, obtém-se uma economia no tempo de execução do algoritmo.

A verificação da restrição de capacidade do veículo pode ser realizada em um tempo computacional constante simplesmente comparando a soma das demandas dos clientes com a capacidade do veículo de uma determinada rota. Já para a verificação da restrição de duração das rotas ser realizada em um tempo constante, necessita-se de um controle da duração de cada rota durante todo o procedimento de busca local. Algumas estruturas auxiliares são utilizadas pelo algoritmo ILS-RVND com a finalidade de se obter uma melhora de desempenho computacional. Uma descrição detalhada destas estruturas pode ser encontrada em Penna et al. (2013).

5.2.1 Estruturas de vizinhança entre rotas

Para a resolução do PRVAFHL, foram utilizadas seis estruturas de vizinhança entre rotas. Cinco dessas estruturas se baseiam no esquema λ -interchanges (Osman, 1993), que consiste em trocar até λ clientes entre duas rotas, enquanto uma se baseia no operador *Cross-exchange* (Taillard et al., 1997), que consiste em trocar dois segmentos de rotas diferentes. Para limitar o número de possibilidades, adotou-se $\lambda = 2$. Uma descrição de cada estrutura é apresentada a seguir.

1. *Shift(1,0)* – Consiste na transferência de um cliente de uma rota para outra rota.
2. *Shift(2,0)* – Consiste na transferência de dois clientes, k_1 e k_2 , de uma rota para outra. Neste caso é verificada tanto a transferência do arco (k_1, k_2) quanto a do arco (k_2, k_1) .
3. *Swap(1,1)* – Consiste na troca de um cliente de uma rota com um cliente de outra rota.
4. *Swap(2,1)* – Consiste na troca de dois clientes adjacentes, k_1 e k_2 , de uma rota com um cliente, k_3 , de outra rota. Neste caso, também verifica-se a transferência do arco (k_1, k_2) e do arco (k_2, k_1) .
5. *Swap(2,2)* – Consiste na troca de dois clientes adjacentes, k_1 e k_2 de uma rota com dois clientes adjacentes, k_3 e k_4 , de outra rota. Nesta situação, devem ser consideradas todas as 4 combinações possíveis de trocas entre os arcos formados pelos dois pares de clientes.
6. *Cross* – Consiste na troca de um segmento de uma rota por um segmento de outra rota. Um segmento de rota é determinado pelo conjunto de clientes localizados no trajeto entre dois clientes. Para o algoritmo proposto, foram considerados apenas os segmentos de rota terminando no depósito.

Uma ilustração gráfica das vizinhanças inter-rotas implementadas é apresentada na Figura 2, em que σ representa um segmento de rota, contendo um ou mais clientes, e \oplus representa um operador de concatenação entre dois segmentos de rota.

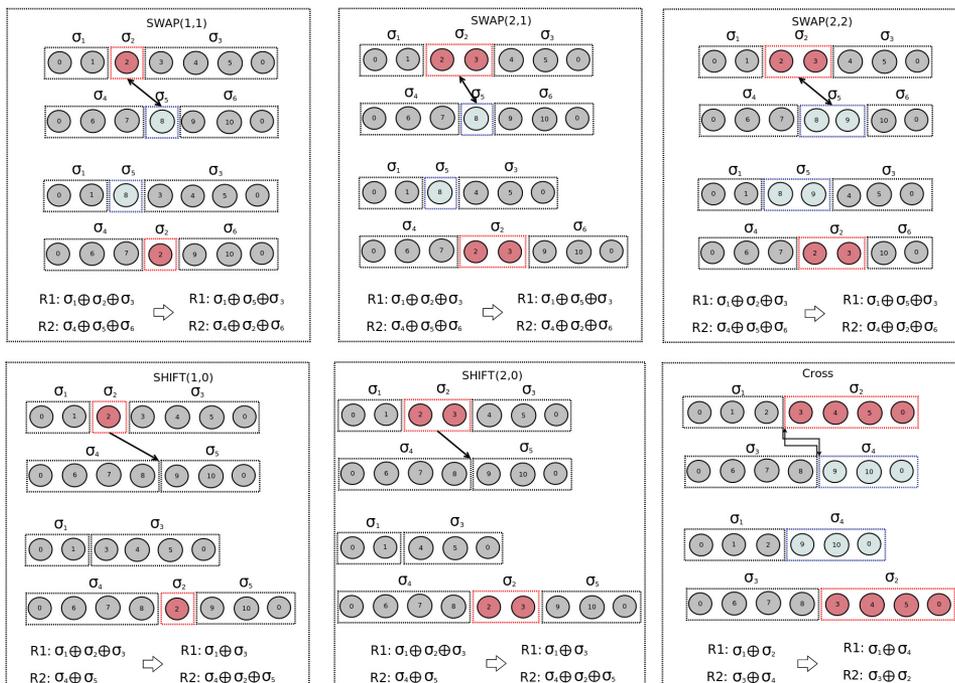


Figura 2. Vizinhanças inter-rotas.

5.2.2 Estruturas de vizinhança intrarrotas

Este conjunto de vizinhanças consiste na execução de movimentos que envolvam os clientes de uma mesma rota. Nesse caso, não é necessário verificar as restrições de capacidade dos veículos, pois considerando que a solução existente anterior seja viável, a mudança de posição de clientes em uma mesma rota não alterará sua demanda total. Todavia, a verificação do tempo de duração da rota torna-se necessária, uma vez que a mudança de posição dos clientes modificará o trajeto percorrido e, conseqüentemente, o tempo total de duração da rota. Foram utilizadas quatro estruturas de vizinhança intrarrotas, cujas descrições são apresentadas a seguir.

1. *Reinserção* – Um cliente é removido da rota e inserido em outra posição.
2. *Or-Opt2* – Dois clientes adjacentes são removidos da rota e inseridos em outra posição.
3. *Or-Opt3* – Três clientes adjacentes são removidos da rota e inseridos em outra posição.
4. *Exchange* – Troca entre dois clientes não adjacentes.

Uma ilustração gráfica das vizinhanças intrarrotas implementadas pode ser visualizada na Figura 3.

5.3 Perturbação

Segundo Lourenço et al. (2002), tal como a busca local, os mecanismos de perturbação são considerados aspectos-chave para o bom desempenho da meta-heurística ILS.

Ao contrário dos procedimentos de busca local, os mecanismos de perturbação não objetivam uma melhora na solução. Sua utilidade consiste em realizar modificações na solução corrente, de modo a escapar de ótimos locais.

Do mesmo modo que nos procedimentos anteriores, foram utilizadas duas perturbações, selecionadas aleatoriamente. Tais procedimentos são descritos a seguir:

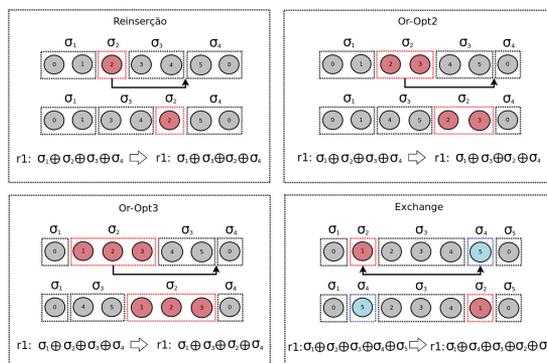


Figura 3. Vizinhanças intrarrotas.

1. *Multiple-Swap(1,1)* – Consiste na execução de vários movimentos da vizinhança *Swap(1,1)* entre diferentes pares de rotas selecionados de maneira aleatória.
2. *Multiple-Shift(1,1)* – Consiste na execução aleatória de vários movimentos da vizinhança *Shift(1,1)*.

6 Resultados computacionais

Nesta seção, é apresentada uma descrição das instâncias (pedidos diários) quanto à frota de veículos e à demanda. Posteriormente é realizada uma comparação dos resultados obtidos pela meta-heurística ILS-RVND com os adotados pela empresa analisada.

O algoritmo proposto foi implementado utilizando a linguagem de programação C++ e executado em um computador com processador Intel Core i5-2410M 2.3 GHz com 4 GB de memória RAM, sistema operacional Linux Ubuntu versão 10.10.

Depois de experimentos preliminares, o parâmetro *MaxIter* foi adotado como sendo igual a 10 para todas as instâncias. Já o parâmetro *MaxIterILS* foi configurado em função das características da instância como sendo a soma entre o número de clientes e o número de veículos disponíveis. Cada instância foi executada 10 vezes.

Embora o limite de duração das rotas adotado pelo *software* da empresa seja de 12 horas, as instâncias foram resolvidas pela metodologia ILS-RVND para um limite de 8 horas, de modo a se ajustar aos limites estabelecidos pelas leis trabalhistas.

6.1 Instâncias

A quantidade e os tipos de veículos considerados nas instâncias foram os mesmos utilizados pela empresa para atender apenas os clientes da GJP (de acordo com as rotas geradas pelo *software*). Os testes foram realizados em um conjunto de 7 instâncias, conforme apresentado na Tabela 2.

Verificaram-se nas soluções geradas pelo *software* da empresa, situações em que alguns veículos foram carregados com mercadorias além da capacidade. Esta prática é adotada nos casos em que a demanda do cliente excede a capacidade do veículo.

As instâncias KSP4, KSP5 e KSP6 se referem aos pedidos de um único dia. Tal divisão foi realizada devido à existência de pedidos de clientes localizados em uma região fora da GJP, de modo que uma das rotas geradas pelo *software* da empresa continha clientes da GJP e clientes de fora dessa região. A instância KSP4 contém todos os clientes. Na instância KSP5, foram retirados os clientes localizados fora da GJP e o veículo responsável por essa rota. Já na instância KSP6, foram desconsiderados todos os clientes dessa rota, inclusive o veículo (baseado na solução

da empresa). Outro motivo para realizar a divisão dessa instância foi tentar solucionar o PRVAFHL de modo a servir todos os clientes da GJP sem utilizar o veículo responsável pelo atendimento de clientes de outra região. As soluções obtidas pela ILS-RVND para a instância KSP5 não puderam ser comparadas por não se ter a solução da empresa para esta situação.

6.2 Comparação dos resultados

As comparações dos resultados foram feitas em relação à distância total percorrida pelos veículos e a quantidade de veículos utilizados.

As melhores soluções encontradas pela heurística ILS-RVND estão apresentadas na Tabela 3, bem como as médias e os desvios padrão referentes às 10 execuções. Os resultados estão expressos em

termos de distância percorrida e tempo de execução do algoritmo.

Verificou-se em todas as instâncias uma redução considerável na distância total percorrida, conforme pode ser observado na Tabela 4. Tais reduções podem ser encaradas como uma redução de custos variáveis, ao se considerar que os custos de combustível e depreciação dos veículos aumentam proporcionalmente com a distância percorrida.

Comparando-se ainda em termos de veículos, a utilização da heurística ILS-RVND favoreceu a redução do número de veículos em 3 instâncias das 6 comparadas, conforme pode ser verificado na Tabela 5. A redução do número de veículos implica redução de custos fixos para a empresa, uma vez que os veículos utilizados são terceirizados.

Tabela 2. Instâncias.

Instância	n	Demanda Total (Kg)	v	Tipos	Veículos		
					Quantidade (por tipo)	Capacidades (por tipo - Kg)	Capacidade Total (Kg)
KSP1	299	96736,81	17	4	13,2,1,1	4000,8000,8564,24123	100687
KSP2	313	52520,43	14	4	11,1,1,1	4000,4390,4621,8000	61011
KSP3	294	62624,06	14	2	13,1	4000,14206	66206
KSP4	341	49381,47	12	3	10,1,1	4000,8320,8696	57016
KSP5	318	46827,40	11	3	9,1,1	4000,8320,8626	53016
KSP6	291	46142,22	11	3	9,1,1	4000,8320,8626	53016
KSP7	108	23486,33	6	2	5,1	4000,4093	24093

Tabela 3. Estatísticas ILS-RVND.

Instância	Distância Total (Km)			Tempo de Execução (s)		
	Melhor Solução	Solução Média	Desvio Padrão	Melhor Solução	Solução Média	Desvio Padrão
KSP1	1394,48	1417,3	12,10	293,80	285,39	40,90
KSP2	1049,56	1054,01	2,24	360,86	335,46	31,04
KSP3	1157,02	1165,06	4,74	274,21	288,42	31,89
KSP4	1158,30	1168,89	6,79	637,06	599,74	84,05
KSP5	877,46	889,54	8,72	465,71	536,37	48,41
KSP6	856,12	861,17	3,40	277,40	318,95	33,37
KSP7	509,33	514,16	2,18	7,32	7,64	1,37

Tabela 4. Comparações dos resultados em termos de distância percorrida.

Instância	Empresa	Distância (Km)		Gap Médio (%)
		Melhor Solução ILS-RVND	Solução Média ILS-RVND	
KSP1	1519,3	1394,48	1417,3	-6,71
KSP2	1342,7	1049,56	1054,01	-21,5
KSP3	1318,9	1157,02	1165,06	-11,66
KSP4	1288,9	1158,3	1168,89	-9,31
KSP5	1082,8	856,12	861,17	-20,47
KSP6	600,2	509,33	514,16	-14,34
KSP7	1519,3	1394,48	1417,3	-6,71
			Média	-14,00%

Tabela 5. Comparações dos resultados em termos de veículos utilizados.

Instância	No. de veículos utilizados		Redução
	Empresa	ILS-RVND	
KSP1	17	17	0
KSP2	14	12	2
KSP3	14	14	0
KSP4	12	11	1
KSP6	11	10	1
KSP7	6	6	0

As melhores soluções encontradas pelo algoritmo ILS-RVND para as instâncias apresentadas foram ilustradas com o auxílio do *Google Maps*, como pode ser visualizado na Figura 4. Cada polígono representa uma rota e cada vértice do polígono representa um cliente (ou depósito).

6.3 Gráficos de convergência

Com o intuito de analisar o desempenho do algoritmo ILS-RVND, comparando os resultados obtidos pelo *software* de roteamento da empresa

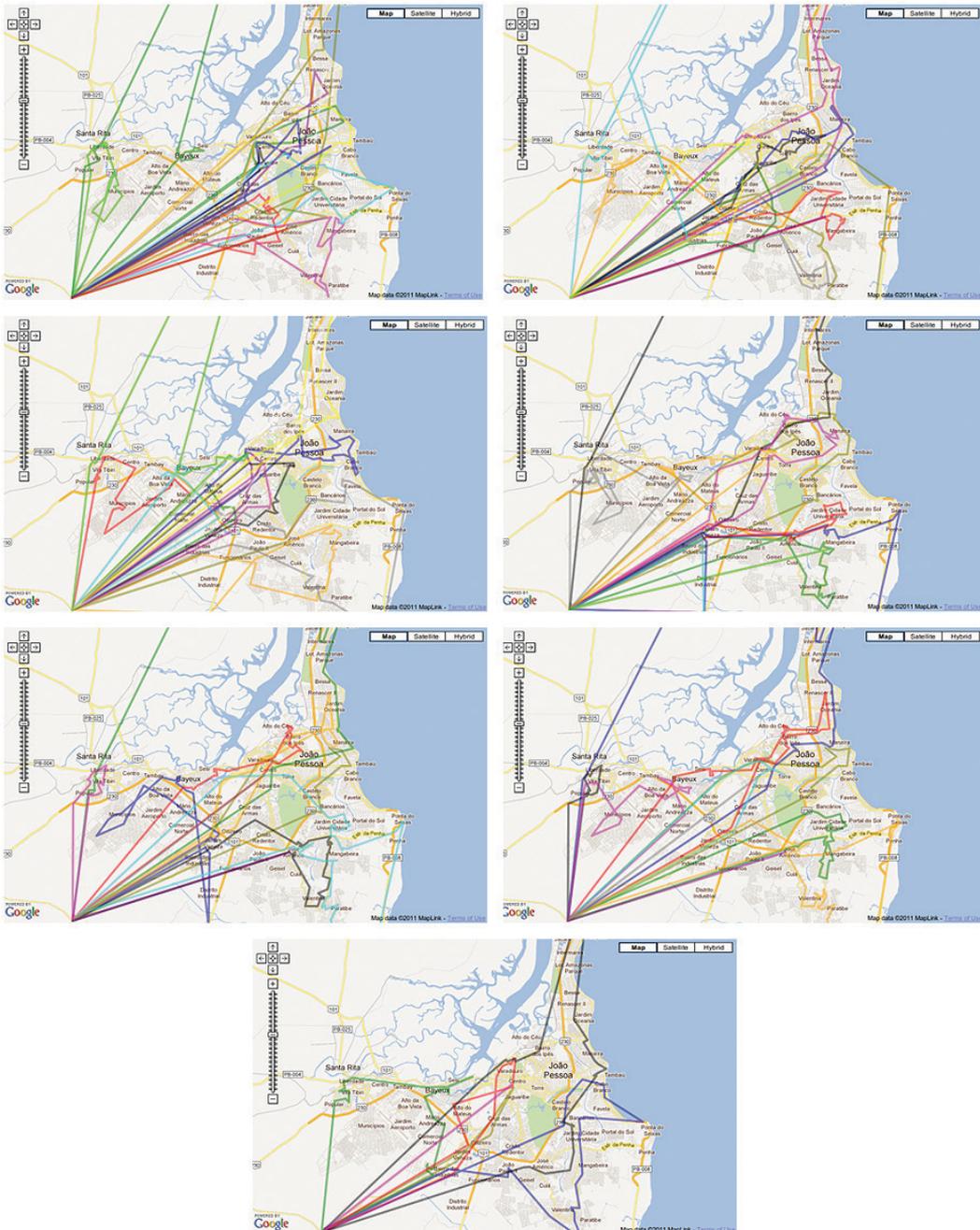


Figura 4. Melhores soluções.

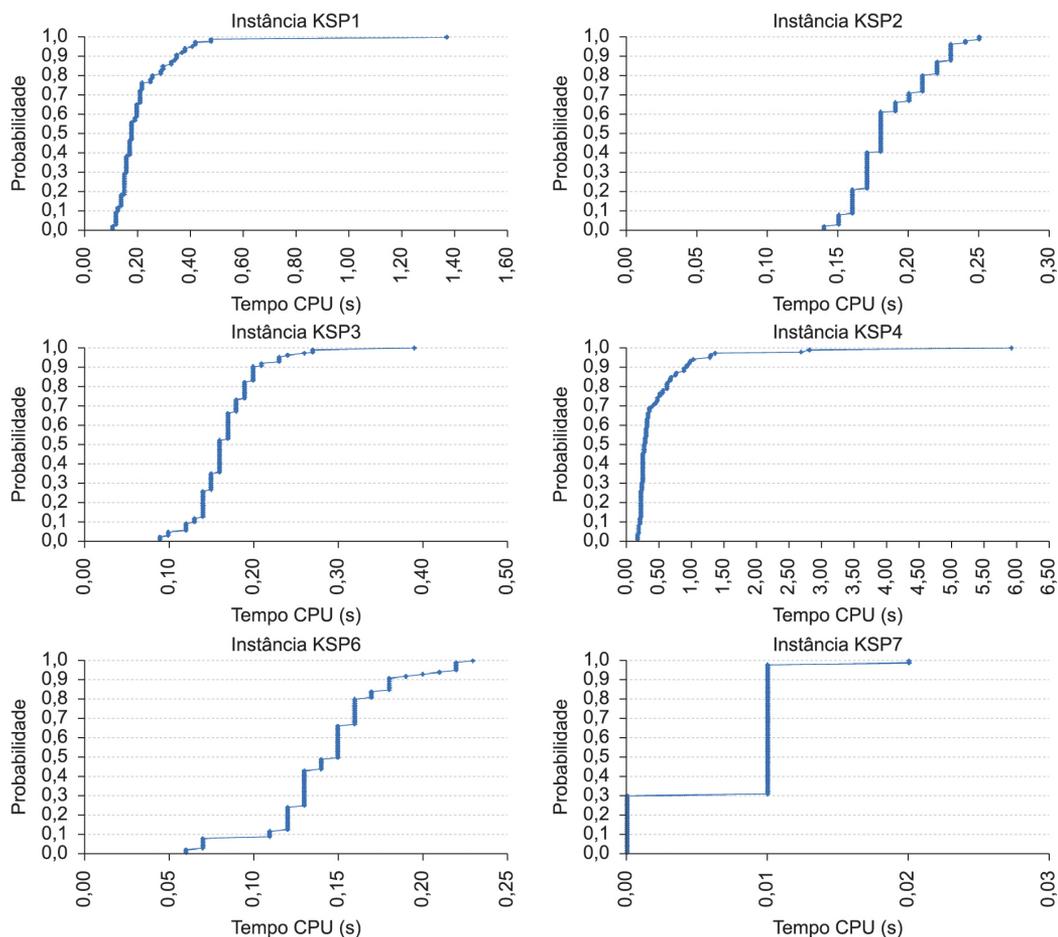


Figura 5. Gráficos de convergência.

analisada, foram elaborados gráficos de convergência para cada instância em que os resultados da empresa eram conhecidos.

Os gráficos de convergência apresentados na Figura 5 representam o tempo gasto pelo algoritmo ILS-RVND para alcançar uma solução melhor ou igual à solução da empresa. Para tanto, o algoritmo ILS-RVND foi executado 100 vezes, para cada instância, tendo como critério de parada a obtenção de uma solução melhor ou igual à obtida pelo *software* da empresa. Para cada execução, foram armazenados os tempos necessários de processamento até que o novo critério de parada fosse alcançado de modo a inferir a probabilidade de igualar ou melhorar os resultados da empresa para um dado tempo de processamento.

A partir da Figura 5, pode-se observar que o tempo máximo que o algoritmo ILS-RVND necessitou para alcançar um resultado melhor ou igual que o alcançado pelo software de roteamento foi de aproximadamente 6 segundos (para a instância de 341 clientes). No geral, tais resultados foram alcançados em menos de 1 segundo de processamento, enaltecendo o ótimo desempenho do algoritmo ILS-RVND.

7 Conclusões

Esse artigo tratou do Problema de Roteamento de Veículos Assimétrico com Frota Heterogênea Limitada (PRVAFHL) com restrições de tempo máximo de duração das rotas aplicado ao caso de uma indústria de bebidas de grande porte. Em virtude da elevada complexidade computacional do problema, o uso de métodos exatos é proibitivo, uma vez que as instâncias consideradas possuem, em média, 300 clientes. Verificou-se, então, que a utilização de métodos heurísticos foi uma estratégia que se adequou bem à proposta do trabalho, isto é, de gerar boas soluções em um tempo de execução razoavelmente curto.

Conforme esperado, a heurística ILS-RVND proposta por Penna et al. (2013) se mostrou bastante eficiente, gerando soluções de qualidade bastante elevada. O algoritmo foi testado em um conjunto de 7 instâncias referentes aos pedidos diários da empresa, no qual verificou-se melhora média de aproximadamente 14% em relação às soluções adotadas pela empresa na prática, em termos de distância percorrida. Observou-se também uma redução na

utilização de veículos para o atendimento de todos os clientes. Tais resultados podem ser encarados como possíveis reduções de custos fixos e variáveis, já que os veículos são terceirizados e a empresa paga por dia de utilização dos veículos e arca com os custos de utilização e manutenção.

Referências

- Baker, E. K. (2002). Evolution of microcomputer-based vehiclerouting software: case studies in the United States. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (SIAM Monographs on Discrete Mathematics and Applications, pp. 353-361). Philadelphia: SIAM.
- Belfiore, P., & Yoshizaki, H. T. Y. (2009). Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, 199(3), 750-758. <http://dx.doi.org/10.1016/j.ejor.2008.08.003>.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268-308. <http://dx.doi.org/10.1145/937503.937505>.
- Campos, G. G., Yoshizaki, H. T. Y., & Belfiore, P. P. (2006). Algoritmos genéticos e computação paralela para problemas de roteirização de veículos com janelas de tempo e entregas fracionadas. *Gestão & Produção*, 13(2), 271-281. <http://dx.doi.org/10.1590/S0104-530X2006000200009>.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568-581. <http://dx.doi.org/10.1287/opre.12.4.568>.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91. <http://dx.doi.org/10.1287/mnsc.6.1.80>.
- Galvão, R. D., Barros Neto, J. F., Ferreira Filho, V. J. M., & Henriques, H. B. S. (1997). Roteamento de veículos com base em sistemas de informação geográfica. *Gestão & Produção*, 4(2), 159-174.
- Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of metaheuristics*. 2 ed. Berlin: Springer. 648 p.
- Golden, B. L., Assad, A. A., & Wasil, E. A. (2002). Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy and newspaper industries. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (SIAM Monographs on Discrete Mathematics and Applications, pp. 245-286). Philadelphia: SIAM.
- Golden, B., Raghavan, S., & Wasil, E. (2008). *The vehicle routing problem: latest advances and new challenges*. New York: Springer.
- Gonçalves, S. M., & Steiner, M. T. A., & Zamboni, V. S. (2005). Uma metodologia para o problema de roteamento de veículos - um estudo de caso para a distribuição de água mineral. In *Anais do XXXVII SBPO* (pp. 1413-1423). Gramado: SOBRAPO.
- Hansen, P., Mladenović, N., & Pérez, J. A. M. (2010). Variable neighborhood search: methods and applications. *Annals of Operations Research*, 175(1), 367-407. <http://dx.doi.org/10.1007/s10479-009-0657-6>.
- Kramer, R. H. F. R., Subramanian, S., & Penna, P. H. V. (2012). Problema de roteamento de veículos assimétrico com frota heterogênea limitada: um estudo de caso em uma indústria de bebidas. In *Anais do XVI CLAIO/XLIV SBPO* (pp. 2102- 2113). Rio de Janeiro: SOBRAPO.
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2002). Iterated local search. In F. Glover & G. A. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 321-353). Norwell: Kluwer Academic Publishers.
- Marmion, M.-E., Humeau, J., Jourdan, L., & Dhaenens, C. (2010). Comparison of neighborhoods for the HFF-AVRP. In *International Conference on Computer Systems and Applications (AICCSA)* (pp. 1-7). Hammamet: IEEE. <http://dx.doi.org/10.1109/AICCSA.2010.5586979>.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4), 421-451. <http://dx.doi.org/10.1007/BF02023004>.
- Partyka, J., & Hall, R. (2010). On the road to connectivity. *ORMS Today*, 37(1), 42-49.
- Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2), 201-232. <http://dx.doi.org/10.1007/s10732-011-9186-y>.
- Prins, C. (2002). Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1(2), 135-150. <http://dx.doi.org/10.1023/A:1016516326823>.
- Souza, M. J. F. (2007). *Inteligência computacional para otimização* (Notas de aula). Ouro Preto: Universidade Federal de Ouro Preto.
- Subramanian, A., Drummond, L. M. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11), 1899-1911. <http://dx.doi.org/10.1016/j.cor.2009.10.011>.
- Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2), 170-186. <http://dx.doi.org/10.1287/trsc.31.2.170>.
- Tarantilis, C., & Kiranoudis, C. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, 50(1), 1-9. [http://dx.doi.org/10.1016/S0260-8774\(00\)00187-4](http://dx.doi.org/10.1016/S0260-8774(00)00187-4).
- Tarantilis, C., & Kiranoudis, C. (2002). Using a spatial decision support system for solving the vehicle routing problem. *Information & Management*, 39(5), 359-375. [http://dx.doi.org/10.1016/S0378-7206\(01\)00103-3](http://dx.doi.org/10.1016/S0378-7206(01)00103-3).
- Tarantilis, C., & Kiranoudis, C. (2007). A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. *European Journal of Operational Research*, 179(3), 806-822. <http://dx.doi.org/10.1016/j.ejor.2005.03.059>.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem* (SIAM Monographs on Discrete Mathematics and Applications). Philadelphia: SIAM.