CLAIO SBPO
Congreso Latino-Iberoamericano de Investigación Operativa
Simpósio Brasileiro de Pesquisa Operacional
24 a 28/09
Rio de Janeiro
Brasil - 2012

HEURÍSTICA GRASP APLICADA AO PROBLEMA DE ROTEAMENTO DE VEÍCULO COM BACKHAULS E FROTA HETEROGÊNEA FIXA

Marcelus Xavier Oliveira[1], Marcone Jamilson Freitas Souza[2], Sérgio Ricardo de Souza[1], Dayanne Gouveia Coelho[1], Puca Huachi Vaz Penna[3]

[1]Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte - MG, Brasil

[2]Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto - MG, Brasil

[3]Inst. do Noroeste Fluminense de Educação Superior – Universidade Federal Fluminense (UFF)
CEP 28.470-000, Santo Antônio de Pádua, Rio de Janeiro, Brasil

marcelusxavier@gmail.com, dayagc@gmail.com, ppenna@ic.uff.br
marcone@iceb.ufop.br, sergio@dppg.cefetmg.br

Resumo. *Este artigo apresenta uma aplicação do algoritmo GRASP para a solução do Problema de Roteamento de Veículos com Backhauls e Frota Heterogênea Fixa (PRVBFHF). Neste problema, os clientes são divididos em dois conjuntos: clientes linehauls e clientes backhauls. Os primeiros são aqueles que recebem uma quantidade de produto oriunda de um depósito, enquanto os segundos são aqueles em que se coleta uma quantidade de produto para ser transportada até o único depósito. O problema consiste em atender a todos os clientes, começando pelos clientes linehauls e depois pelos clientes backhauls, respeitando-se as capacidades dos veículos que os atendem. Para resolver o PRVBFHF, é proposto um algoritmo que combina os procedimentos heurísticos GRASP, Randomized Variable Neighborhood Descent, Sweep Method e PFIH. O algoritmo foi avaliado a partir de experimentos computacionais em um conjunto de instâncias da literatura e os resultados encontrados validam sua eficiência.*

PALAVRAS-CHAVE: *Problema de Roteamento de Veículos com Bakhauls, GRASP, PFIH, Sweep Algorithm.*

Abstract. *This paper presents an application of the GRASP algorithm to solve the Heterogeneous Fixed Fleet Vehicle Routing Problem with backhauls (HFF-VRPB). In this problem, the customers are divided into two groups: linehauls customers and backhauls customers. Linehauls customers are those who receive an amount of products coming from a depot, while in backhauls clients a quantity of products to be transported to the depot are collected. The problem consists in to serve all customers, starting with the linehauls customers and, in the sequel, by the backhauls customers, respecting the vehicle capacities that serve them. To solve the HFFVRPB, it is proposed an algorithm that combines the heuristic procedures GRASP, Randomized Variable Neighborhood Descent, Sweep Method and PFIH. The algorithm was evaluated from computational experiments with a set of instances from the literature and the results have showed its effectiveness.*

KEYWORDS: *Vehicle Routing Problem with Bakhauls, GRASP, PFIH, Sweep Algorithm.*

1 Introdução

O Problema de Roteamento de Veículos com *Backhauls* (PRVB) é uma extensão do Problema de Roteamento de Veículos (PRV) envolvendo pontos de entrega e coleta. Os pontos *Linehaul* (de entrega) são clientes que recebem uma determinada quantidade de produto oriunda de um único depósito. Os pontos *Backhaul* (de coleta) são aqueles associados aos clientes que têm mercadorias com destino ao depósito. A restrição principal indica que todos os clientes de entregas (*linehaul*) devem ser atendidos obrigatoriamente antes dos clientes de coleta (*backhaul*) em uma determinada rota.

O presente trabalho estuda o Problema de Roteamento de Veículos com *Backhauls* e Frota Heterogênea Fixa (PRVBFHF). Diferentemente do PRVB, neste caso considera-se que a frota de veículos é heterogênea, ou seja, os veículos disponíves possuem capacidades distintas e, além disto, o número de veículos da frota é limitado. As quantidades de produtos para entrega e coleta são previamente conhecidas. Uma solução viável para o problema consiste em um conjunto de rotas em que todos os clientes *linehauls* de cada rota são visitados antes que se atenda qualquer cliente *backhaul* e a capacidade do veículo atribuído a cada rota não é violada pelo conjunto de clientes *linehual* ou *backhaul* contidos na mesma.

Na literatura pode-se encontrar vários trabalhos que estudam e desenvolvem técnicas para resolver o PRVB. Métodos heurísticos, baseados no algoritmo desenvolvido por Clarke e Wright (1963), têm sido propostos nos trabalhos de Casco et al. (1988), Deif e Bodin (1984) e Goeschalckx e Jacobs-Blecha (1989) para resolvê-lo. Casco et al. (1988) resolvem o PRVB criando, primeiramente, rotas *linehauls* pelo método de *savings*, para, em seguida, inserir rotas *backhauls*. Deif e Bodin (1984) foram um dos primeiros autores a propor uma heurística para resolver o PRVB. Na heurística proposta por eles, que é uma extensão da heurística de Clarke e Wright (1963), os clientes *backhauls* só são visitados quando todas as rotas dos clientes *linehauls* forem traçadas. Dessa forma, as rotas são divididas nos conjuntos dos clientes *backhauls* e dos clientes *linehauls* e organizadas separadamente. No trabalho de Goeschalckx e Jacobs-Blecha (1989), o algoritmo desenvolvido contém duas fases, sendo que, na primeira, é gerada uma solução inicial viável e, na segunda, é feita uma melhora na solução inicial gerada por meio de heurísticas de busca local.

Toth e Vigo (1996) propõem uma heurística *Cluster-first-route-second*, baseada em uma relaxação lagrangeana de uma formulação do PRVB, em que é construído um conjunto de rotas viáveis usando movimentos de vértices e trocas de arcos. O algoritmo desses autores supera as heurísticas propostas por Goeschalckx e Jacobs-Blecha (1989). Nos trabalhos de Toth e Vigo (1997) e Mingozzi et al. (1999), o PRVB é resolvido por meio de métodos exatos. Toth e Vigo (1997) fizeram um estudo de dois casos do PVRB (simétrico e assimétrico), e utilizam o método exato *Branch-and-Bound* para resolver o problema. Mingozzi et al. (1999) utilizaram o algoritmo *Branch-and-Price* para resolver o PVRB simétrico. É descrita uma nova formulação de programação inteira para o PRVB, baseada em uma abordagem de particionamento de conjuntos. O método se baseia na escolha de um menor limite inferior para o custo da solução ótima e combina métodos heurísticos para resolver o problema dual da relaxação da formulação exata.

Em outro trabalho, Toth e Vigo (1999) utilizam a informação dada pelo relaxamento lagrangeano para obter limites inferiores e resolver o problema por algoritmos exatos. No trabalho de Osman e Wassan (2002), os autores propõem um algoritmo baseado em Busca

Tabu que alcança soluções melhores que em Toth e Vigo (1999), mas com um custo computacional maior. Brandão (2006) apresenta um novo algoritmo Busca Tabu para o PRVB, capaz de encontrar as melhores soluções através de uma versão em que obtém soluções iniciais de pseudo-limites inferiores.

Gajpal e Abad (2009) utilizam o algoritmo Colônia de Formigas para resolver o PRVB. Nesse trabalho, as formigas artificiais são usadas para construir uma solução com base em informações obtidas a partir de soluções geradas anteriormente. Zachariadis e Kiranoudis (2011) propõem uma metaheurística para o Problema de Roteamento de Veículo com Backhaul baseada em busca local.

Poucos trabalhos, no entanto, tratam do Problema de Roteamento de Veículos com Backhaul e Frota Heterogênea Fixa (PRVBFHF). Assim, Tütüncü (2010) gerou um conjunto de instâncias para o problema, com o intuito de validar seu algoritmo desenvolvido. Essas instâncias combinam dois conjuntos de instâncias da literatura associadas ao Problema de Roteamento de Veículos com Frota Heterôgena Fixa (PRVFHF) e o Problema de Roteamento de Veículos com Backhauls (PRVB). Tütüncü (2010) propõe resolver o PRVFHF e uma extensão deste problema, chamada de Problema de Roteamento de Veículos *Backhaul* com Frota Heterogêna Fixa (PRVBFHF), utilizando o procedimento de busca GRAMPS, que opera dentro de um sistema de apoio à decisão. Ele utiliza um sistema semelhante ao utilizado em Tütüncü et al. (2009), com modificações na interface gráfica do usuário. Os resultados encontrados pelo autor mostram que a abordagem visual por ele proposta pode obter bons resultados em um tempo razoável.

Este artigo apresenta uma proposta de resolução para o PRVBFHF utilizando uma adaptação do algoritmo GRASP. O restante deste artigo está estruturado como segue: a próxima seção descreve o Problema de Roteamento de Veículo Backhauls com Frota Heterogênea apresentando as definições e restrições de interesse deste trabalho; a seção 3 apresenta o algoritmo proposto; a seção 4 apresenta os resultados computacionais alcançados pelo algoritmo desenvolvido aplicado a instâncias da literatura e, por fim, a seção 5 apresenta as conclusões obtidas a respeito do desenvolvimento realizado nesta pesquisa.

2 Problema de Roteamento de Veículos com Backhauls e Frota Heterogênea

O Problema de Roteamento de Veículos com Backhauls (PRVB) consiste em determinar um conjunto de rotas, com o menor custo, de forma a atender, primeiramente, em cada rota, todos os clientes *linehauls*, e posteriormente, os *backhauls*. Um exemplo de rota pode ser visto na Figura 1.

O PRVB pode ser formulado por meio de um grafo completo, em que cada vértice representa um cliente. Neste trabalho, o PRVB é definido da seguinte forma: seja $G = (V, E)$ um grafo não direcionado, em que V é o conjunto de vértices e $E = \{(i, j)i, j \in V\}$ é o conjunto de arestas. Para cada aresta (i, j), é associado um custo (distância) positivo c_{ij} para cada $i, j \in V$, tal que $i \neq j$ e $c_{ii} = +\infty$ para cada $i \in V$. Os clientes *linehaul* e *backhaul* são representados, respectivamente, pelos subconjuntos $L = \{1, 2, \cdots, N\}$ e $B = \{N + 1, N + 2, \cdots, N + M\}$, sendo $N + M$ o número total de clientes. O depósito é representado pelo vértice 0. Desse modo, $V = \{0\} \cup L \cup B$. A frota consiste em K veículos idênticos, de mesma capacidade Q. Cada cliente $i \in L \cup B$ requer uma determinada demanda q_i para ser entregue ($i \in L$) ou para ser coletada ($i \in B$). A quantidade máxima de veículos é dada por $k \geq \max\{K_L, K_B\}$, em que K_L e K_B são, respectiva-

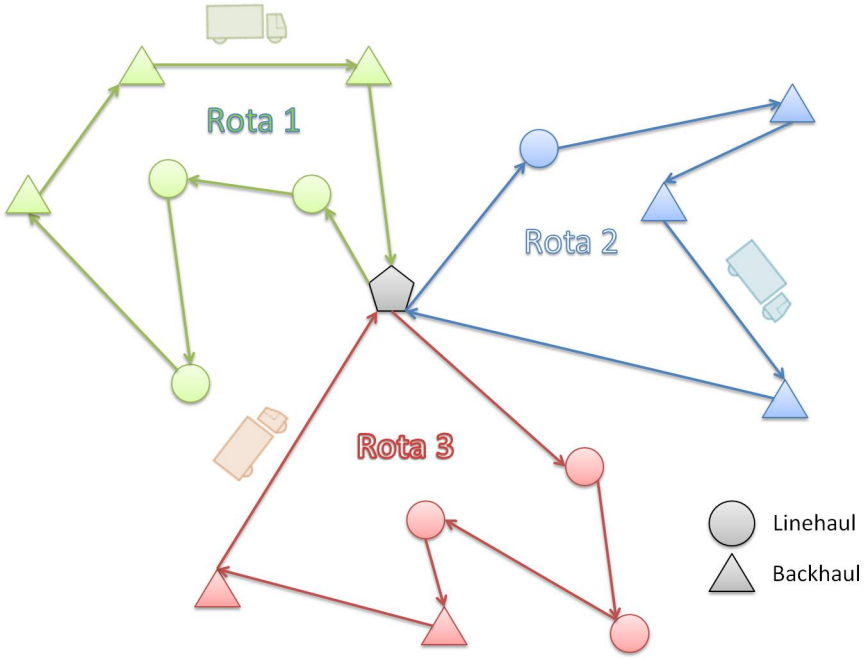

Figura 1. Exemplo de uma rota para o PRVB. O veículo indica o sentido da rota. Observe que clientes *linehaul* são atendidos primeiramente e, em seguida, os *backhaul*.

mente, o número mínimo de veículos necessários para atender a todos os clientes *linehauls* e *backhauls*. Dessa forma, a resolução do PRVB consiste em encontrar um conjunto de rotas, com início e fim no depósito, com um custo total mínimo, que atenda às exigências de demanda e coleta dos clientes, respeitando a capacidade dos veículos. O custo é calculado pela soma dos custos das arestas da solução, respeitando as seguintes restrições:

i) cada veículo realiza apenas um percurso, ou seja, percorre apenas uma rota;
ii) cada rota começa e termina no depósito;
iii) em uma rota, a demanda total dos clientes *linehaul* e *backhaul*, separadamente, não pode exceder a capacidade do veículo;
iv) os clientes *linehaul* devem ser atendidos antes dos clientes *backhaul* (restrição de precedência) e nenhuma rota pode ser formada apenas por clientes *backhaul*. Assim, o custo $c_{ji} = c_{0j} = Y$ para cada $j \in B$ e cada $i \in L$, sendo Y uma constante de valor alto;
v) cada cliente é visitado apenas uma vez por um dos veículos, e, nesta única visita, sua demanda deve ser atendida.

Neste trabalho, no entanto, é tratado o Problema de Roteamento de Veículos com *Backhaul* e Frota Heterogênea Fixa(PRVBFHF), que se diferencia do PRVB descrito anteriormente pelo acréscimo de restrições do Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH). Assim, para a formulação do PRVBFHF, são adicionadas, à descrição do PRVB, as restrições sobre a frota, que é composta por m diferentes tipos de veículos, com $M = \{1, \ldots, m\}$. Para cada tipo $k \in M$, existem m_k veículos disponíveis, cada um com uma capacidade Q_k. Ou seja, duas fortes distinções são presentes, quais sejam, a heterogeneidade da frota de veículos, bem como a limitação do número dos mesmos.

3 Algoritmo Proposto

Nesta seção, são descritas a representação de uma solução; a metodologia de construção da solução inicial para o PRVBFHF; a função de avaliação utilizada; a estrutura de vizinhança para explorar o espaço de soluções do problema; e o algoritmo proposto para solucionar o PRVBFHF.

3.1 Representação de uma solução

Cada solução do PRVBFHF é representada por um vetor $s = [r_1, \ r_2, \ \cdots, r_n]$, que constitui um conjunto de n rotas. Para cada posição deste vetor, ou seja, para cada rota r_i, são atribuídos um veículo, com suas devidas características, e um conjunto de clientes $[C_1, \cdots, C_m]$ a serem atendidos.

3.2 Estruturas de Vizinhança

A vizinhança de uma solução s é o conjunto de soluções $N(s)$, em que cada solução $s' \in N(s)$ é obtida a partir de um movimento feito na solução corrente s. A fim de explorar o espaço de soluções do problema, foram utilizados cinco tipos de movimentos:

- *Shift(1,0)*: consiste na transferência de um cliente de uma rota para outra;
- *Shift(2,0)*: semelhante ao *Shift(1,0)*, porém consiste na realocação de dois clientes de uma mesma rota para outra;
- *Swap(1,1)*: consiste na troca de um cliente i de uma rota r_a com um cliente j de uma outra rota r_b;
- *Swap*: consiste em trocar dois clientes de uma mesma rota;
- *Reinsertion*: consiste em remover um cliente i de uma rota e reinseri-lo em uma outra posição da mesma rota;

3.3 Função de Avaliação

O objetivo do problema abordado é minimizar o custo total do deslocamento. Desse modo, a função de avaliação é dada pela equação (1):

$$f(s) = \sum_{k \in P} \sum_{(i,j) \in E} c_{ijk} x_{ijk} \tag{1}$$

sendo P o conjunto de veículos disponíveis; E o conjunto de arestas (i, j) com $i, j \in V \backslash \{0\}$; c_{ijk} a distância entre um cliente i a um cliente j vezes o custo variável utilizando o veículo k; e x_{ijk} é uma variável de decisão binária, que assume valor 1 se existe uma aresta $(i, j) \in E$ utilizando um veículo k; e 0, caso contrário.

3.4 Algoritmo GRASP Proposto

O algoritmo *Greedy Randomized Adaptive Search Procedure* – GRASP (Feo e Resende, 1995), é um método de duas fases iterativas que, a cada iteração, contrói uma solução, elemento a elemento, e em seguida, realiza uma busca local na vizinhança da solução construída, retornando o ótimo local daquela vizinhança. Após todas as iterações realizadas, o algoritmo retorna a melhor solução encontrada. Seu pseudocódigo está ilustrado pelo Algoritmo 1. As demais subseções detalham cada uma de suas componentes.

Algoritmo 1: *GRASP*

Entrada: $f(.)$, $g(.)$, $N(.)$, *GRASPmax*,s
Saída: s
início
  $f^* \leftarrow \infty$;
  para *Iter*=1 **até** *GRASPmax* **faça**
    Construção($g(.)$, α, s);
    Buscalocal ($f(.)$, $N(.)$, s);
    se $(f(s) < f^*)$ **então**
      $s^* \leftarrow s$;
      $f^* \leftarrow f(s)$;
    fim
  fim
  $s \leftarrow s^*$;
  Retorne s;
fim

3.4.1 Fase de Construção

A cada iteração do GRASP, uma solução é construída, elemento a elemento. A fase de construção está descrita pelo Algoritmo 2.

Algoritmo 2: Construção da Solução

Entrada: $g(.)$, α,s
Saída: s
início
  $s \leftarrow \emptyset$;
  $l \leftarrow \emptyset$;
  Inicialize o conjunto C de candidatos;
  enquanto $C \neq \emptyset$ **faça**
    $g_{\min} = \min\{g(t) \mid t \in C\}$;
    $g_{\max} = \max\{g(t) \mid t \in C\}$;
    $LRC = \{t \in C \mid g(t) \leq g_{\min} + \alpha(g_{\max} - g_{\min})\}$;
    Selecione, aleatoriamente, um elemento $t \in LRC$;
    $l \leftarrow l \cup \{t\}$;
    Atualize o conjunto C de candidatos;
  fim
  Seja *bool* um número binário aleatório;
  se *bool* = 1 **então**
    $s \leftarrow PFIH(l)$
  fim
  senão
    $s \leftarrow Sweep(l)$
  fim
  Retorne s;
fim

Os possíveis candidatos a serem incluídos na solução são colocados na lista C de candidatos, segundo um critério guloso, que avalia o benefício do candidato ser incluído na solução. Este benefício é dado pela função de avaliação, calculada pela expressão de avaliação utilizada no procedimento *Push-Forward Insertion Heuristic* – PFIH (Solomon, 1987). A lista C é formada somente por clientes *Linehauls*, em vista da restrição de atendimento prioritário aos clientes de entrega (*Linehaul*) em relação aos clientes de coleta (*Backhaul*). Em seguida, é criada uma lista restrita de candidatos (LRC), cujo número de

elementos é determinado pelo parâmetro α. A LRC é formada pelos melhores elementos da lista C. Assim, a solução é construída escolhendo-se, aleatoriamente, um elemento da LRC.

Para a construção de rotas a partir da lista de clientes *Linehaul* l, são utilizados os procedimentos PFIH e *Sweep Algorithm* (Renaud e Boctor, 2002), tal como mostrado nas linhas 13 a 19 do Algoritmo 2. A definição de qual desses procedimentos será usado é feita a partir da geração de um valor booleano aleatório na linha 13. Esta forma de construção tem a vantagem de promover uma maior diversidade de soluções a cada chamada GRASP.

3.4.2 Fase de busca local

O procedimento de busca local da metaheurística GRASP (linha 5 do Algoritmo 1) é feita pelo método *Randomized Variable Neighborhood Descent* - RVND, proposto em Subramanian et al. (2010). Esta heurística é uma variação do método *Variable Neighborhood Descent* – VND (Mladenovic e Hansen, 24). A diferença reside na aleatoriedade da ordem das vizinhanças utilizadas para explorar o espaço de soluções. No RVND, a cada chamada ao método, altera-se aleatoriamente a ordem de escolha das estruturas de vizinhanças disponíveis. O pseudocódigo deste procedimento é apresentado no Algoritmo 3.

Algoritmo 3: RVND

```
   Entrada: f(·), N(·), r, s
   Saída: s
 1 início
 2     Seja r o número total de diferentes estruturas de vizinhança;
 3     Seja L_r a lista que contém as diferentes estruturas de vizinhança;
 4     L_r ← embaralha(L_r);
 5     k ← 1;
 6     enquanto k ≤ r faça
 7         p ← L_r(k);
 8         Encontre o melhor vizinho s' ∈ N^(p)(s);
 9         se (f(s') < f(s)) então
10             s ← s';
11             k ← 1;
12         senão
13             k ← k + 1;
14         fim
15     fim
16 fim
17 Retorne s;
18 fim
```

É importante destacar o funcionamento da lista L_r descrita na linha 3 do Algoritmo 3. Essa lista é composta pelas estruturas de vizinhanças disponíveis; por exemplo, seja $L_r = \{1, 2, 3, \cdots, \}$. Na linha 4 do Algoritmo 3, esta lista é embaralhada ($L_r = \{3, 1, 2, \cdots, \}$) e utilizada na linha 7, em que dado o valor de k, atribui-se a p o conteúdo da posição k na lista L_r.

4 Resultados Computacionais

O Algoritmo GRASP foi desenvolvido utilizando-se a plataforma Java 5. Os experimentos foram realizados em um PC Intel Core 2 Quad 2,66 GHz, com 4 GB de memória RAM, em ambiente Windows 7.

Tabela 1. Problemas-teste de (Tütüncü,2010)

Problema	Instância PRVB	n	L	B	Tipos de Veículos														
					A			B			C			D			E		
					Q	c	K	Q	c	K	Q	c	K	Q	c	K	Q	c	K
PRVBFHF1	eil22.62	50	25	25	50	1,0	2	100	1,6	2	160	2,0	1						
PRVBFHF2	eil22.63	50	34	16	50	1,0	2	100	1,6	3	160	2,0	1						
PRVBFHF3	eil22.64	50	40	10	50	1,0	2	100	1,6	3	160	2,0	1						
PRVBFHF4	eil22.62	50	25	25	40	1,0	1	80	1,6	2	140	2,1	2						
PRVBFHF5	eil22.63	50	34	16	40	1,0	1	80	1,6	3	140	2,1	2						
PRVBFHF6	eil22.64	50	40	10	40	1,0	2	80	1,6	3	140	2,1	2						
PRVBFHF7	eil22.65	75	37	38	50	1,0	2	120	1,2	1	200	1,5	1	350	1,8	1			
PRVBFHF8	eil22.66	75	50	25	50	1,0	2	120	1,2	2	200	1,5	1	350	1,8	1			
PRVBFHF9	eil22.67	75	60	15	50	1,0	3	120	1,2	2	200	1,5	2	350	1,8	1			
PRVBFHF10	eil22.65	75	37	38	20	1,0	3	50	1,3	3	100	1,9	2	150	2,4	2	250	2,9	1
PRVBFHF11	eil22.66	75	50	25	20	1,0	4	50	1,3	3	100	1,9	2	150	2,4	2	250	2,9	1
PRVBFHF12	eil22.67	75	60	15	20	1,0	4	50	1,3	4	100	1,9	2	150	2,4	2	250	2,9	1
PRVBFHF13	eil22.77	100	50	50	100	1,0	1	200	1,4	2	300	1,7	1						
PRVBFHF14	eil22.78	100	67	33	100	1,0	3	200	1,4	2	300	1,7	1						
PRVBFHF15	eil22.79	100	80	20	100	1,0	3	200	1,4	3	300	1,7	1						
PRVBFHF16	eil22.77	100	50	50	100	1,0	2	140	1,7	2	200	2,0	2						
PRVBFHF17	eil22.78	100	67	33	100	1,0	4	140	1,7	3	200	2,0	2						
PRVBFHF18	eil22.79	100	80	20	100	1,0	4	140	1,7	4	200	2,0	2						

Para realizar os experimentos computacionais, foi utilizado o conjunto de problemas-teste proposto em Tütüncü (2010). Esse autor gerou o conjunto de problemas-teste do PVRBFH utilizando dois conjuntos de instâncias de versões distintas do PRV. O primeiro corresponde a instâncias do Problema de Roteamento de Veículos com Frota Heterogênea e Fixa (PRVFHF), propostas em Taillard (1996), enquanto o segundo corresponde a instâncias geradas por Toth e Vigo (1999) para o PVRB. A Tabela 1 apresenta as características das instâncias obtidas a partir da combinação desses dois conjuntos de instâncias. As colunas 3, 4 e 5 representam, respectivamente, o número total de clientes n, a quantidade de clientes *linehauls* L e a quantidade de clientes *backhauls* B para cada instância. As colunas 6 a 20 apresentam, para cada tipo m de veículo (A, B, C, D e E), os valores Q e c, que indicam, respectivamente, a capacidade e o custo variável associados ao veículo k e a quantidade de veículos disponíveis desse tipo, como nas instâncias de Taillard (1996).

O algoritmo proposto foi executado 100 vezes para cada instância. O conceito de *Multi-Threading* foi aplicado ao algoritmo; assim, foram utilizados os quatro núcleos disponíveis pelo *hardware*, e, em cada núcleo, o algoritmo GRASP foi executado com valores distintos de α, variando nos valores $10\%, 30\%, 50\%$ e 70%. Como critério de parada, utilizou-se o número de iterações igual a 500.

A Tabela 2 apresenta os resultados obtidos pela aplicação do algoritmo proposto na resolução do Problema de Roteamento de Veículos com *Backhaul* e Frota Heterogênea, utilizando-se as instâncias descritas na Tabela 1. A primeira coluna mostra a instância usada, as colunas 2 e 3 apresentam os resultados encontrados para as 100 execuções do algoritmo na forma de melhor resultado e média dos resultados. Na última coluna é mostrado o tempo médio de cada execução do algoritmo, em segundos.

As instâncias para as quais não foram apresentados resultados computacionais não possuem soluções factíveis, pois a demanda total dos clientes *linehaul* ultrapassa a capacidade total dos veículos, violando-se, assim, a restrição de capacidade dos veículos

Tabela 2. Resultados obtidos pelo GRASP

Problema	Solução		Tempo
	Melhor	Média	Médio (seg.)
PRVBFHF1	991,823	1051,790	24,626
PRVBFHF2	1035,214	1076,326	16,584
PRVBFHF3	-	-	-
PRVBFHF4	1162,496	1210,868	23,787
PRVBFHF5	1191,518	1227,690	18,465
PRVBFHF6	-	-	-
PRVBFHF7	1270,259	1308,198	47,815
PRVBFHF8	-	-	-
PRVBFHF9	1128,584	1180,004	79,524
PRVBFHF10	1745,573	1847,229	109,760
PRVBFHF11	1789,999	1899,065	60,585
PRVBFHF12	-	-	-
PRVBFHF13	1399,311	1437,758	223,319
PRVBFHF14	-	-	-
PRVBFHF15	1495,717	1541,782	114,571
PRVBFHF16	1639,060	1679,886	317,617
PRVBFHF17	1457,279	1508,575	362,323
PRVBFHF18	1612,567	1647,914	412,577

Tabela 3. Comparação dos resultados obtidos para o PRVBFHF

Problema	(Tütüncu, 2010)			%
	GRAMPS	ADVISER	GRASP	
PRVBFHF1	1111,67	1056,44	**991,823**	-6,12
PRVBFHF2	1067,28	982,86	1035.214	5,33
PRVBFHF3	1124,14	998,22	-	-
PRVBFHF4	1094,08	1070,06	1162,496	8,64
PRVBFHF5	1135,21	1127,97	1191,518	5,63
PRVBFHF6	1200,58	1183,36	-	-
PRVBFHF7	1190,63	1190,63	1270,259	6,69
PRVBFHF8	1211,28	1182,66	-	-
PRVBFHF9	1222,66	1203,09	**1128,584**	-6,19
PRVBFHF10	1845,75	1781,50	**1745,573**	-2,02
PRVBFHF11	2035,39	1941,74	**1789,999**	-7,81
PRVBFHF12	1945,35	1917,54	-	-
PRVBFHF13	1228,24	1227,81	1399,311	13,97
PRVBFHF14	1136,87	1109,02	-	-
PRVBFHF15	1228,56	1216,65	1495,717	22,94
PRVBFHF16	1629,47	1555,35	1639,060	5,38
PRVBFHF17	1609,03	1585,30	**1457,279**	-8,08
PRVBFHF18	1618,27	1615,08	**1612,567**	-0,16

disponíveis.

Na Tabela 3, as colunas 2 e 3 apresentam, respectivamente, os melhores resultados obtidos pelos procedimentos GRAMPS e ADVISER, listados em Tütüncü (2010). A coluna 4 apresenta a melhor solução encontrada pelo GRASP e a última coluna mostra o *gap* % entre as soluções dadas pelo algoritmo ADVISER e as obtidas pelo algoritmo proposto, calculado na forma:

$$gap\% = \frac{EE_i - ADV_i}{ADV_i} \times 100\%$$

sendo EE_i o resultado encontrado pela aplicação do algoritmo proposto e ADV_i o resultado obtido pelo algoritmo ADVISER, ambos aplicado à instância i.

De acordo com a Tabela 2, é possível notar que o GRASP fornece boas soluções para o problema em um tempo computacional baixo. Para as instâncias em que a solução está em negrito na coluna 4 da Tabela 3, o GRASP obteve soluções melhores que as obtidas pelo GRAMPS e pelo ADVISER. Assim, foram encontrados, para 6 instâncias, resultados melhores que os disponíveis em Tütüncü (2010).

5 Conclusões

Este trabalho tratou o Problema de Roteamento de Veículos com *Backhauls* e Frota Heterogênea (PRVBFHF) por meio de um algoritmo GRASP. Na fase de construção, uma solução do problema é gerada utilizando os procedimentos *Sweep Method* e *Push-Forward Insertion Heuristic* (PFIH); na fase de busca local, é aplicada a heurística de refinamento *Randomized Variable Neighborhood Descent* (RVND).

Para avaliar o algoritmo proposto, foram utilizados problemas-teste da literatura, e os resultados comparados com o trabalho de Tütüncü (2010).

Os resultados obtidos se mostraram promissores, visto que, das 13 instâncias que possuem solução factível (já que 5 delas violam a restrição de capacidade do PRVBFHF), o algoritmo proposto se mostrou superior ao de Tütüncü (2010) em 6.

Por fim, a qualidade alcançada dos resultados mostram que o algoritmo proposto obteve sucesso para a solução do PRVBFHF.

Em trabalhos futuros, pretende-se testar outras estruturas de vizinhança e outras heurísticas de refinamento dentro do algoritmo GRASP.

Agradecimentos

Os autores agradecem ao CEFET-MG, à CAPES, à FAPEMIG e ao CNPq pelo apoio ao desenvolvimento deste trabalho.

Referências

Brandão, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhaul. *European Journal of Operational Research*, v. 173, p. 540–555.

Casco, D; Golden, B. L. e Wasil, E. A. (1988). *Vehicle Routin: Methods and Studies*, volume 16 of *Studies in Management Science and Systems*, Capítulo Vehicle Routing with backhauls: Models, algorithms, and case studies, p. 127–147. North-Holland, Amsterdam.

Clarke, G. e Wright, J. W. (1963). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, v. 11, p. 568–581.

Deif, I. e Bodin, L. (1984). Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling. Kidder, A., editor, *Proceedings of the Banson Conference on Software Uses in Transportation and Logistic Management*, p. 75–96, (1984).

Feo, Thomas A. e Resende, Maurício G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 9, p. 849–859.

Gajpal, Yuvraj e Abad, P. L. (2009). Multi-ant colony system (macs) for a vehicle routing problem with backhauls. *European Journal of Operational Research*, v. 196, p. 102–117.

Goeschalckx, M e Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls. *Eur*, v. 42, p. 39–51.

Mingozzi, A.; Giorgi, S. e Baldacci,. (1999). An exact method for the vehicle routing problem with backhauls. *Transactions Science*, v. 33, p. 315–329.

Mladenovic, N. e Hansen, P. (24). Variable neighborhood search. *Computers and Operations Research*, v. 1097-1100, p. 1997.

Osman, I. H. e Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with backhauls. *Journal of Scheduling*, v. 5, p. 263–285.

Renaud, J. e Boctor, F. F. (2002). A sweep based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, v. 140, p. 618–628.

Solomon, Marius M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265.

Subramanian, A.; Drummond, L. M. A.; Bentes, C.; Ochi, L. S. e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, p. 1899–1911.

Taillard, E. D. *A Heuristic Column Generation Method for the Heterogeneous Fleet VRP*. Publication CRT-03-96, Universite Montreal., (1996).

Toth, P e Vigo, D. (1996). A heuristic algorithm for the vehicle routing problem with backhauls. Bianco, L. e Toth, P., editors, *Proc. Second TRISTAN Conference*, p. 585–608. Advanced Methods inTransportation Analisys, (1996).

Toth, P e Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. *Transactions Science*, v. 31, p. 372–385.

Toth, Paolo e Vigo, Daniele. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing prolems with backhauls. *European Journal of Operational Research*, v. 113, p. 528–543.

Tütüncü, G. Yazgi. (2010). An interactive gramps algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls. *European Journal of Operational Research*, v. 201, p. 593–600.

Tütüncü, G. Yazgi; Carreto, Carlos A. C. e Baker, Barrie M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega - The International Journal of Management Science*, v. 37, p. 138–154.

Zachariadis, Emmanouil E. e Kiranoudis, Chris T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*, v. 38, n. 3, p. 2717 – 2726. ISSN 0957-4174.