

UMA ANÁLISE DA APLICAÇÃO DE ESTRATÉGIAS EVOLUTIVAS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS *BACKHAULS* E FROTA HETEROGÊNEA FIXA

MARCELUS XAVIER OLIVEIRA*, MARCONE JAMILSON FREITAS SOUZA†, SÉRGIO RICARDO DE SOUZA*, DAYANNE GOUVEIA COELHO‡, PUCA HUACHI VAZ PENNA§

**Programa de Pós-Graduação em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil*

*†Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais, Brasil*

*‡Departamento de Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil*

*§Instituto do Noroeste Fluminense de Educação Superior
Universidade Federal Fluminense
Santo Antônio de Pádua, Rio de Janeiro, Brasil*

Emails: marcelusxavier@gmail.com, marcone.freitas@yahoo.com.br,
sergio@dppg.cefetmg.br, dayagc@gmail.com, pucahuachi@gmail.com

Abstract— This paper presents an algorithm based on Evolutionary Strategy for solving the Heterogeneous Fixed Fleet Vehicle Routing Problem with backhauls (HFFVRPB). This problem is an extension of Vehicle Routing Problem. Customers are divided into two groups: customers backhauls and customers linehauls. Customers linehauls are those who receive an amount of product coming from a deposit, while clients backhauls are those who are collecting a quantity of product to be transported to the deposit. The problem is to serve all customers while respecting the constraints that linehaul customers must be met first to backhaul customers, respecting the capacities of vehicles that serve them. To solve the HFFVRPB, it is combined the Evolution Strategy with the metaheuristic Variable Neighborhood Descent and the Sweep Method of constructing routes. The algorithm was evaluated through computational experiments on a set of instances of the literature. The results validate the proposed algorithm.

Keywords— Vehicle Routing Problem with Heterogeneous Fleet and backhauls, Evolutionary Strategy.

Resumo— Este artigo apresenta um algoritmo baseado em Estratégias Evolutivas para a solução do Problema de Roteamento de Veículos com Backhauls e Frota Heterogênea Fixa (PRVBFHF). Neste problema, que é uma extensão do Problema de Roteamento de Veículos, os clientes são divididos em dois conjuntos: clientes *linehauls* e clientes *backhauls*. Os clientes *linehauls* são aqueles que recebem uma quantidade de produto oriunda de um depósito, enquanto os clientes *backhauls* são os em que se coleta uma quantidade de produto para ser transportado até o depósito. O problema consiste em atender todos os clientes respeitando a restrição de que clientes *linehauls* devem ser atendidos obrigatoriamente antes que os clientes *backhauls*, respeitando as capacidades dos veículos que os atendem. Para resolver o PRVBFHF, combina-se a Estratégia Evolutiva com a metaheurística *Variable Neighborhood Descent* e o procedimento *Sweep Method* de construção de rotas. O algoritmo desenvolvido foi avaliado a partir de experimentos computacionais em um conjunto de instâncias presentes na literatura. Os resultados obtidos validam o algoritmo proposto.

Keywords— Problema de Roteamento de Veículos com Backhauls, Estratégias Evolutivas.

1 Introdução

O presente trabalho estuda uma extensão do Problema de Roteamento de Veículos (PRV) clássico, conhecida como Problema de Roteamento de Veículos com Backhauls (PRVB), acrescido de Frota Heterogênea Fixa (PRVBFHF). Esta extensão inclui um conjunto de clientes a quem os produtos serão entregues, designados *linehauls*, e um conjunto de fornecedores cujos produtos precisam ser transportados de volta para o centro de distribuição denominados *backhauls*. Em adição ao problema, todas as entregas de cada rota devem ser feitas antes que qualquer mercadoria possa ser

coletada, a fim de evitar a reorganização das cargas no veículo. Este tipo de situação pode ser encontrada em diversos casos práticos em que ocorra planejamento da logística de distribuição.

Na literatura pode-se encontrar vários trabalhos que estudam e desenvolvem técnicas para resolver o Problema de Roteamento de Veículos Backhauls. 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 resolver o PRVB. Casco et al. (1988) propuseram resolver o PRVB, primeiramente criando rotas *linehauls*, através do

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 serão 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. Em Goeschalckx e Jacobs-Blecha (1989), os clientes *linehauls* são visitados antes dos clientes *backhauls*. A metodologia utilizada para resolver o PRVB é dividida em duas fases: na primeira fase, é gerada uma solução inicial viável e na segunda fase, é feita uma melhora na solução inicial gerada através de heurísticas de busca local.

Toth e Vigo (1996) propõem uma heurística *Cluster-first-route-second*, baseado em uma relaxação lagrangeana de uma formulação do PRVB, em que é construída um conjunto de rotas viáveis através de movimentos de vértices e trocas de arcos. Esta metodologia 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 através 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. Neste artigo, 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 através de algoritmos exatos. No trabalho de Osman e Wassan (2002), os autores propõem uma metaheurística 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. Neste 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 (PRVBFH). Assim, Tütüncü (2010) gerou um conjunto de instâncias para o problema, com o intuito de validar a metodologia proposta. Esse novo conjunto de instâncias combina dois conjuntos de instâncias da literatura, associadas ao Problema de Roteamento de Veículos com Frota Heterogênea 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 Veículo *Backhaul* com Frota Heterogênea Fixa (PRVBFHF), utilizando um procedimento de programação de pesquisa 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 Estratégias Evolutivas. O restante do 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 a metodologia de solução proposta; a seção 4 apresenta os resultados computacionais alcançados pela metodologia proposta aplicada 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 Fixa

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, a 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 modelo de grafo completo, em que cada vértice representa um cliente. Neste trabalho, o PRVB é definido da seguinte forma: seja $G = (V, A)$ um grafo não direcionado, em que V é o conjunto de vértices e $A = \{(i, j) \mid i, j \in V\}$ é o conjunto de arcos. Para cada arco (i, j) , é associado um custo (distância) positivo c_{ij} , tal que $i \neq j$ e $c_{ii} = +\infty$ para cada $i \in V$. Os clientes *linehaul* e *backhaul* são representados, respectivamente, pelos subconjuntos $L = \{1, 2, \dots, N\}$ e $B = \{N + 1, N + 2, \dots, N + M\}$, sendo $N + M$

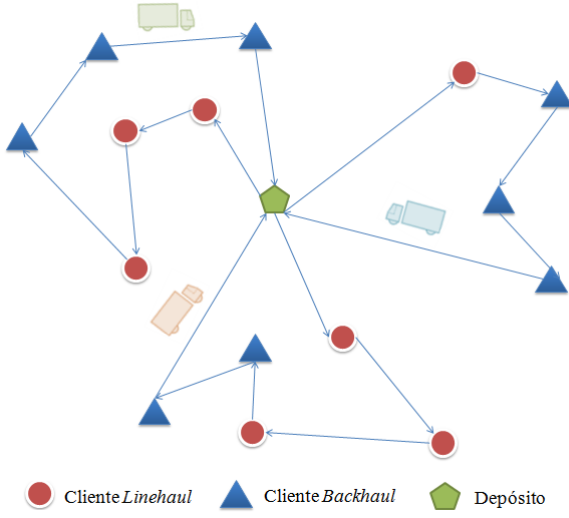


Figura 1: Exemplo de uma rota para o PRVB. O veículo indica o sentido da rota. Observe que clientes *linehaul* são atendidos primeiramente e, em seguida, os *backhaul*.

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, respectivamente, o número mínimo de veículos necessários para atender 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 dos arcos, 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 cliente *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.

Para o PRVBFHF utilizado neste trabalho, é usada a formulação do PRVB descrita anteriormente, com o acréscimo de restrições do PRVHFH. Assim, para a formulação do PRVBFHF, são adicionadas à composição do PRVB as restrições sobre a frota, em que é composta por m diferentes tipos de veículos, com $M = \{1, \dots, m\}$. Para cada tipo $k \in M$, existem m_k veículos disponíveis, cada um com uma capacidade Q_k .

3 Metodologia Proposta

Nesta seção, são descritas a representação e como é feita a construção de uma solução para o PRVBFHF; a função de avaliação utilizada; a estrutura de vizinhança utilizada 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 $s = [R_1, R_2, \dots, R_n]$, que constitui um conjunto de n rotas. Para cada posição deste vetor, ou seja, para cada rota R_i , são atribuídos um tipo de veículo e um conjunto de clientes $[C_1, \dots, C_m]$ que serão 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*: este movimento consiste na transferência de um cliente de uma rota para outra.
- *Shift(2,0)*: semelhante ao *Shift*, porém consiste na realocação de dois clientes de uma rota para outra.
- *Swap*: este movimento consiste na troca de um cliente i de uma rota r_1 com um cliente j de uma rota r_2 .
- *Swap-intra*: este movimento consiste na troca entre dois clientes na mesma rota.
- *Or-Opt*: este movimento 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 expressão 1:

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

sendo P o conjunto de veículos; A o conjunto de arcos (i,j) com $i, j \in V \setminus \{0\}$; c_{ijk} é a distancia entre um cliente i a um cliente j vezes o custo variável utilizando o veículo k ; e $x_{i,j,k}$ é uma variável binária, que assume valor 1 se existe um arco $(i,j) \in A$ utilizando um veículo k , 0 caso contrário.

3.4 Algoritmo Proposto: Estratégia Evolutiva

Neste trabalho, o algoritmo proposto é uma adaptação do Algoritmo Estratégias Evolutivas – EE (Rechenberg, 1973), sendo inspirado na abstração do algoritmo de Coelho et al. (2011). Para gerar cada solução da população inicial do problema, é incorporado ao EE o método da varredura (*Sweep Method*) (Renaud e Boctor, 2002), na construção de cada indivíduo da população inicial. O pseudocódigo do método utilizado pode ser visto no Algoritmo 1.

A população inicial do algoritmo (linhas 2 à 5) é composta por μ indivíduos. Durante a construção desta população (linha 3), parte dos indivíduos são gerados de forma aleatória e parte é construída pelo método *Sweep*, aplicando, em cada indivíduo gerado, a heurística de refinamento *Variable Neighborhood Descent* – VND (Mladenovic e Hansen, 1997). Construída a população inicial, enquanto não atender ao critério de parada, calcula-se a probabilidade de mutação de cada indivíduo (linha 9); aplica-se a mutação neste indivíduo (linha 10); e seleciona-se (linha 13), dentre todos os indivíduos ($Pop + PopFilhos$), os μ indivíduos que irão para a próxima geração. O critério de parada adotado é o número de gerações.

Algoritmo 1: Estratégia Evolutiva

Entrada: *IterMax*, Função $f(\cdot)$;
Saída: *Pop*;

```

1 início
2   para  $i=1$  até  $\mu$  faça
3      $ind_i \leftarrow ConstróiSolução()$ ;
4      $Pop[i] = ind_i$ ;
5   fim
6   enquanto Critério de parada não satisfeito
7     faça
8       para  $i=1$  até  $\mu$  faça
9          $ind_i \leftarrow Pop[i]$ ;
10         $ind_i \leftarrow ProbabilidadeMutacao(ind_i)$ ;
11         $ind_i \leftarrow Mutacao(ind_i)$ ;
12         $PopFilhos[i] = ind_i$ ;
13      fim
14       $Pop = Selecao(Pop, PopFilhos)$ ;
15 fim
```

O cálculo da Probabilidade de Mutação (linha 9 do Algoritmo 1) é realizado conforme o Algoritmo 2. Cada posição no vetor p está ligada a uma estrutura de vizinhança e representa a probabilidade do indivíduo sofrer um tipo de mutação (linha 5 do Algoritmo 2), calculada pela distribuição normal. Cada posição do vetor a , que também está relacionada a uma estrutura de vizinhança, representa a quantidade de movimentos que o indivíduo sofrerá (linha 6 Algoritmo 2), calculada por distribuição binomial.

A Mutação (linha 10 do Algoritmo 1) é dada conforme o Algoritmo 3. Para cada estrutura de vizinhança $\mathcal{N}(\cdot)$, é calculado um número aleatório $x \in [0, 1]$ (linha 5 Algoritmo 3) e, a seguir, verificado se é satisfeita a probabilidade p_i de sofrer

Algoritmo 2: ProbabilidadeMutacao

Entrada: Indivíduo Ind_i ;
Saída: Indivíduo Ind_i ;

```

1 início
2    $p \leftarrow$  Vetor de Parâmetros  $P$  de probabilidade
   do indivíduo  $Ind_i$  sofrer mutação;
3    $a \leftarrow$  Vetor de Parâmetros  $A$  de quantidades
   de movimentos no indivíduo  $Ind_i$ ;
4   para  $i=1$  até número de vizinhança faça
5      $p[i] \leftarrow Normal(0, \sigma_{real})$ ;
6      $a[i] \leftarrow a_i + Binomial(0, \sigma_{binomial})$ ;
7   fim
8 fim
```

mutação. Satisfeita essa condição, aplica-se a_i número de movimentos (linha 8 Algoritmo 3), calculado pela distribuição binomial.

Algoritmo 3: Mutacao

Entrada: Indivíduo Ind_i , vizinhanças $\mathcal{N}(\cdot)$;
Saída: Indivíduo Ind_i ;

```

1 início
2    $p \leftarrow$  Vetor de Parâmetros  $P$  de Probabilidade
   do indivíduo  $Ind_i$  sofrer mutação
3    $a \leftarrow$  Vetor de Parâmetros  $A$  de Aplicação de
   quantidades de movimentos no indivíduo  $Ind_i$ 
4   para  $i=1$  até número de vizinhança faça
5     Gere um número aleatório  $x \in [0, 1]$ 
6     se  $x < p_i$  então
7       para  $j=1$  até  $a_i$  faça
8          $ind_i \leftarrow Movimento \mathcal{N}(j)$ 
9       fim
10    fim
11  fim
12 fim
```

Na linha 13 do Algoritmo 1 são escolhidos μ indivíduos para a próxima geração. Desses μ indivíduos, são selecionados, por elitismo, os cinco melhores, ou seja, aqueles que possuem menor valor na função de avaliação. Os demais indivíduos, são escolhidos pelo operador *Seleção*($\cdot, \cdot$). Neste trabalho, o operador utilizado é o Método por Torneio Binário.

4 Resultados Computacionais

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

Para realizar os experimentos computacionais, foi utilizado o conjunto de problemas-teste de Tütüncü (2010). Esse autor gerou os problemas-teste do PVRBFH a partir de dois conjuntos de problemas distintos de roteamento de veículos da literatura. O primeiro conjunto corresponde a instâncias para o PRVFHF propostas por Taillard (1996), que contêm a frota heterogênea, e o segundo corresponde a instâncias geradas para o PVRB, propostas por Toth e

Tabela 2: Resultados obtidos pela Estratégia Evolutiva

Problema	Solução		Tempo Médio (seg.)
	Melhor	Média	
PRVBFHF1	945,476	1058,569	9,706
PRVBFHF2	938,151	1053,049	12,953
PRVBFHF3	-	-	-
PRVBFHF4	1054,014	1207,810	11,507
PRVBFHF5	1089,300	1207,239	13,076
PRVBFHF6	-	-	-
PRVBFHF7	1250,250	1372,836	36,369
PRVBFHF8	-	-	-
PRVBFHF9	1137,536	1250,105	62,625
PRVBFHF10	1470,324	1706,327	53,452
PRVBFHF11	1494,426	1720,344	57,889
PRVBFHF12	-	-	-
PRVBFHF13	1496,544	1602,406	156,729
PRVBFHF14	-	-	-
PRVBFHF15	1405,279	1567,458	147,521
PRVBFHF16	1672,066	1817,687	183,899
PRVBFHF17	1429,363	1602,153	237,426
PRVBFHF18	1574,866	1724,927	271,584

Vigo (1999). A Tabela 1 apresenta as características dos problemas-teste obtidos a partir da combinação dos dois conjuntos. 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 . Das colunas 6 a 20, para cada tipo m de veículo (A, B, C, D e E), os valores Q e c indicam, respectivamente, a capacidade e o custo variável associados ao veículo k e a quantidade de veículos disponíveis desse tipo, conforme as instâncias de Taillard (1996).

O algoritmo proposto foi executado 100 vezes para cada problema-teste. Os parâmetros utilizados foram: número de gerações igual a 10.000 e população inicial de tamanho 100, ou seja $\mu = 100$.

A Tabela 2 apresenta os resultados obtidos na aplicação da EE na resolução do Problema de Roteamento de Veículos *Backhaul* com Frota Heterogênea Fixa, utilizando os problemas-teste apresentados na Tabela 1. A primeira coluna mostra o problema-teste usado, 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. Os problemas-teste para os quais não foram apresentados resultados computacionais não possuem soluções factíveis, pois a demanda total dos clientes *linehaul* ultrapassa a capacidade total dos veículos, violando, assim, a restrição de capacidade dos veículos 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 pela Estratégia Evolutiva e a última coluna mostra o *gap* % entre as soluções

Tabela 3: Comparação dos resultados obtidos para o PRVBFHF

Problema	(Tütüncü, 2010)		Estratégia Evolutiva	<i>gap</i> %
	GRAMPS	ADVISER		
PRVBFHF1	1111,67	1056,44	945,476	-10,50
PRVBFHF2	1067,28	982,86	938,151	-4,55
PRVBFHF3	1124,14	998,22	-	-
PRVBFHF4	1094,08	1070,06	1054,014	-1,50
PRVBFHF5	1135,21	1127,97	1089,300	-3,43
PRVBFHF6	1200,58	1183,36	-	-
PRVBFHF7	1190,63	1190,63	1250,250	5,01
PRVBFHF8	1211,28	1182,66	-	-
PRVBFHF9	1222,66	1203,09	1137,536	-5,45
PRVBFHF10	1845,75	1781,50	1470,324	-17,47
PRVBFHF11	2035,39	1941,74	1494,426	-23,04
PRVBFHF12	1945,35	1917,54	-	-
PRVBFHF13	1228,24	1227,81	1496,544	21,89
PRVBFHF14	1136,87	1109,02	-	-
PRVBFHF15	1228,56	1216,65	1405,279	15,50
PRVBFHF16	1629,47	1555,35	1672,066	7,50
PRVBFHF17	1609,03	1585,30	1429,363	-9,84
PRVBFHF18	1618,27	1615,08	1574,866	-2,49

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 a Estratégia Evolutiva fornece boas soluções para o problema em um tempo computacional baixo. Para os problemas-teste em que a solução está em negrito na coluna 4, a EE obteve soluções melhores que as obtidas pelo GRAMPS e pelo ADVISER. Assim, foram encontrados para 9 instâncias resultados melhores que os disponíveis em (Tütüncü, 2010).

5 Conclusões

Este artigo apresentou uma solução para o Problema de Roteamento de Veículos *Backhauls* com Frota Heterogênea Fixa, utilizando um algoritmo baseado em uma adaptação da Estratégia Evolutiva. Para a construção da população inicial no método proposto, foram utilizadas a construção aleatória, o método de varredura (*Sweep Method*) e a aplicação da heurística de refinamento VND nos indivíduos gerados.

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

Os resultados se mostraram promissores, visto que das 18 instâncias, 5 violam a restrição de capacidade do PRVBFHF e nas 13 restantes, foram obtidos soluções factíveis. Destas, o algoritmo proposto se mostrou superior em 9 instâncias aos resultados apresentados em (Tütüncü,

Tabela 1: Características dos 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						

2010). Deve-se ressaltar que para as instâncias PRVBFHF10 e PRVBFHF11 foram encontrados valores 17,47% e 23,04%, respectivamente, menores que os da referência citada.

Em trabalhos futuros, pretende-se testar outras estruturas de vizinhança e outras heurísticas de refinamento para o algoritmo desenvolvido, e verificar se podem elas contribuir com a melhoria da solução final.

Agradecimentos

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

Referências

- Brandão, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhaul, *European Journal of Operational Research* **173**: 540–555.
- Casco, D., Golden, B. L. e Wasil, E. A. (1988). *Vehicle Routin: Methods and Studies*, Vol. 16 of *Studies in Management Science and Systems*, North-Holland, Amsterdam, chapter Vehicle Routing with backhauls: Models, algorithms, and case studies, pp. 127–147.
- Clarke, G. e Wright, J. W. (1963). Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research* **11**: 568–581.
- Coelho, V. N., Souza, M. J. F., Coelho, I. M., Guimarães, F. G. e Coelho, B. N. (2011). Estratégias evolutivas aplicadas a um problema de programação inteira mista, *Anais do Congresso Brasileiro de Inteligência Computacional – CBIC 2011*.
- Deif, I. e Bodin, L. (1984). Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling, in A. Kidder (ed.), *Proceedings of the Banson Conference on Software Uses in Transportation and Logistic Management*, pp. 75–96.
- Gajpal, Y. e Abad, P. L. (2009). Multi-ant colony system (macs) for a vehicle routing problem with backhauls, *European Journal of Operational Research* **196**: 102–117.
- Goeschalckx, M. e Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls, *Eur* **42**: 39–51.
- Mingozzi, A., Giorgi, S. e Baldacci (1999). An exact method for the vehicle routing problem with backhauls, *Transactions Science* **33**: 315–329.
- Mladenovic, N. e Hansen, P. (1997). Variable neighborhood search, *Computers and Operations Research* **24**: 1097–1100.
- Osman, I. H. e Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with backhauls, *Journal of Scheduling* **5**: 263–285.
- Rechenberg, I. (1973). Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution, *Frommann-Holzboog*.
- 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* **140**: 618–628.
- Taillard, E. D. (1996). *A Heuristic Column Generation Method for the Heterogeneous*

- Toth, P. e Vigo, D. (1996). A heuristic algorithm for the vehicle routing problem with backhauls, in L. Bianco e P. Toth (eds), *Proc. Second TRISTAN Conference*, Advanced Methods in Transportation Analysis, pp. 585–608.
- Toth, P. e Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls, *Transactions Science* **31**: 372–385.
- Toth, P. e Vigo, D. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls, *European Journal of Operational Research* **113**: 528–543.
- Tütüncü, G. Y. (2010). An interactive gramps algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls, *European Journal of Operational Research* **201**: 593–600.
- Tütüncü, G. Y., Carreto, C. A. C. e Baker, B. M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls, *Omega - The International Journal of Management Science* **37**: 138–154.
- Zachariadis, E. E. e Kiranoudis, C. T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries, *Expert Systems with Applications* **38**(3): 2717 – 2726.