



ISSN: 1984-3151

# RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM BACKHAULS COM HEURÍSTICA BASEADA EM BUSCA LOCAL

## RESOLUTION OF THE VEHICLE ROUTING PROBLEM WITH BACKHAULS WITH HEURISTICS BASED ON LOCAL SEARCH

Douglas Souza Lima<sup>1</sup>; Luciana Assis<sup>2</sup>; Alessandro Vivas<sup>3</sup>

1 Graduando em Sistemas de Informação UFVJM. Diamantina, MG. lima.sistemas38@gmail.com.

2 Mestre em Ciência da Computação. UFMG, 2007. Professora do Curso de Sistemas de Informação, UFVJM. Diamantina, MG. lpassis@ufvjm.edu.br.

3 Doutor em Engenharia Elétrica. UFMG, 2008. Professor do Curso de Sistemas de Informação, UFVJM. Diamantina, MG. alessandrovivas@ufvjm.edu.br

Recebido em: 15/06/2012 - Aprovado em: 07/07/2012 - Disponibilizado em: 30/07/2012

*RESUMO: Este trabalho apresenta uma heurística para resolução do Problema de Roteamento de Veículos com Backhauls. Este é um problema de coleta e entrega, no qual as demandas de entrega devem ser atendidas antes das demandas de coleta. O algoritmo proposto é baseado na metaheurística Iterated Local Search (ILS). Para encontrar uma solução inicial para o problema, é apresentado um novo método que transforma o problema abordado em dois Problemas de Roteamento de Veículos Capacitado. Um deles contendo apenas os pontos de demanda de entrega e, o outro, os pontos de coleta. Dada as rotas encontradas após resolução dos dois problemas, busca-se uma melhor forma de unir as duas soluções utilizando um algoritmo de atribuição. Em seguida, diversos mecanismos de busca local e perturbação são aplicados à solução inicial. Os testes realizados em instâncias encontradas na literatura revelam a efetividade do algoritmo proposto.*

*PALAVRAS-CHAVE: Logística e Transportes, Metaheurísticas, Otimização Combinatória.*

*ABSTRACT: This paper presents a heuristic algorithm to solving the Vehicle Routing Problem with Backhauls. This is a Pickup and Delivery Problem where on each route all delivery must be satisfied before pickups. The proposed algorithm is based on Iterated Local Search (ILS) metaheuristic. To set an initial solution to the problem, we present a new method that transforms the problem addressed in two Capacitated Vehicle Routing Problems. One containing only the linehauls demands and other backhauls demands. Given the routes found after solving the two problems, we seek a better way to combine the two solutions using an assignment algorithm. Then, various search local and perturbation movements are applied to the initial solution. Tests on instances taken from literature reveal the effectiveness of the proposed algorithm. KEYWORDS: Logistics & Transport, Metaheuristics, Combinatorial Optimization. Main area. Logistics & Transport.*

## 1 INTRODUÇÃO

O transporte de cargas no Brasil desempenha um papel fundamental nas grandes empresas, pois viabiliza a movimentação de materiais a serem industrializados e de produtos para o consumidor final.

Segundo Fleury (2010) estima-se que no ano de 2000 os gastos com transporte de cargas no Brasil corresponderam a cerca de 10% do PIB nacional, aproximadamente R\$90 bilhões, números estes que podem ser considerados elevados se comparados com os Estados Unidos, onde as despesas com

transporte de cargas correspondem a 6% do PIB. O uso de bons sistemas de roteirização pode reduzir estes custos e, ainda, tornar a empresa mais competitiva, uma vez que o custo repassado aos consumidores finais também será reduzido.

O Problema de Roteamento de Veículos (VRP do inglês, *Vehicle Routing Problem*), proposto por Dantzig e Ramser (1959), é um problema básico de logística. Este problema tem como objetivo definir rotas entre o depósito e um determinado conjunto de clientes, minimizando o custo de transporte, e respeitando algumas restrições básicas: cada cliente só pode ser visitado uma única vez e por apenas um veículo; cada rota é iniciada e finalizada no depósito; em nenhum momento a capacidade do veículo pode ser ultrapassada e a demanda de todos os clientes tem de ser atendidas.

Em diversas situações reais é necessário atender às demandas dos consumidores e também visitar fornecedores em uma mesma rota, a fim de reduzir os custos de transporte. Durante o transporte destas mercadorias, algumas delas podem sofrer algum dano devido à reorganização das cargas no veículo. Este prejuízo pode ser evitado se a empresa adotar alguma estratégia que restrinja a ordem de atendimento dessas demandas como, por exemplo, atender primeiramente as demandas de entrega para, então, atender as demandas de coleta.

Um exemplo que pode ser usado é o de uma empresa de beneficiamento da laranja, os veículos partem de um depósito para entregar o suco e, ao retornarem, podem passar nos produtores para recolher mais frutos para serem processados. Assim sendo, o recolhimento das laranjas após o atendimento das demandas de entrega evita que as cargas sejam rearranjadas e danificadas.

Esta situação caracteriza o Problema de Roteamento de Veículos com Backhauls (VRPB do inglês, *Vehicle Routing Problem with Backhauls*), abordado nesse trabalho. Assim como no VRP, este problema tem como objetivo definir rotas entre o depósito e um determinado conjunto de pontos de demanda, minimizando o custo de transporte. A diferença entre eles é que os pontos de demanda podem ser de entrega ou coleta de mercadorias. Outra característica do VRPB é que as demandas de coleta só podem ser atendidas após todas as demandas de entrega serem satisfeitas.

O Problema de Roteamento de Veículos com Backhauls pode ser reduzido ao Problema de Roteamento de Veículos, sendo este pertencente à classe NP-Difícil. Dessa forma, devido a sua complexidade computacional, existem, na literatura, poucos trabalhos que o solucionam, utilizando métodos exatos. Toth e Vigo (1997) e Mingozzi, Giorgi e Baldacci (1999) apresentam algoritmos exatos para solucionar problemas com até 100 pontos de demanda.

Assim sendo, os esforços têm sido direcionados no desenvolvimento de métodos heurísticos para lidar com problemas com um maior número de clientes, que retratam, de forma mais adequada, os problemas reais. Dentre as abordagens heurísticas vale destacar as seguintes. Brandão (2005) reporta bons resultados utilizando um algoritmo baseado na Busca Tabu. A solução inicial é dada utilizando duas abordagens: uma delas através da resolução do Problema de Roteamento de Veículos (OVRP do inglês *Open Vehicle Routing Problem*) e, a outra, por meio da geração de K-árvores. Tavakkoli-Moghaddam Saremi e Ziaee (2006) propuseram um algoritmo memético que utiliza diferentes buscas locais para melhoria dos resultados. Outros dois algoritmos baseados nas metaheurísticas *Reactive Tabu Search* e *Variable*

*Neighbourhood Descent* foram utilizados por Crispim e Brandão (2001). Goetschalckx e Jacobs-Blecha (1989) utilizam uma metodologia de solução composta por duas fases. A primeira uma solução inicial é gerada, utilizando uma heurística baseada no Preenchimento de Curvas Espaciais (SFC, *Space filling Curve*). Na segunda fase, esta solução é refinada através da resolução de subproblemas identificados no modelo matemático.

Este trabalho apresenta um algoritmo heurístico baseado na Busca Local Iterativa (ILS, *Iterated Local Search*) para solucionar o VRPB. Propõe-se um novo método de geração da solução inicial. Este método constrói uma solução a partir da resolução de dois problemas de roteamento de veículos: um deles considerando as demandas de entrega e outro considerando demandas de coleta. Em seguida, busca-se a melhor forma de unir as rotas contendo os pontos de coleta e os pontos de entrega, resultantes das duas soluções encontradas. Dada a solução inicial, diversos métodos de busca local e perturbação são aplicados a ela, a fim de explorar eficientemente o espaço de busca.

O artigo está organizado da seguinte forma: na Seção 2 é apresentada a definição formal do VRPB e seu modelo matemático; a Seção 3 descreve o algoritmo baseado no ILS, proposto para solucionar o problema abordado; na Seção 4 os resultados obtidos são comparados com os melhores resultados encontrados na literatura. Por fim, as conclusões são apresentadas na Seção 5.

## 2 DESENVOLVIMENTO

### 2.1 DEFINIÇÃO DO PROBLEMA

O Problema de Roteamento de Veículos com *Backhauls*, abordado neste trabalho, pode ser definido

como: dado um grafo não-direcionado  $G = (V, A)$ , onde  $V = 0 \cup L \cup B$  é um conjunto de vértices e  $A = \{i, j\} \mid i, j \in V$  um conjunto de arestas. Cada aresta possui um custo  $c_{i,j}$  não negativo associado, que representa a distância entre os vértices  $i$  e  $j$ . Os subconjuntos  $L = \{1, 2, \dots, n\}$  e  $B = \{n+1, n+2, \dots, n+m\}$  representam respectivamente os pontos de demanda de entrega (*linehauls*) e de coleta (*backhauls*), e  $\{0\}$  representa o depósito.

Cada consumidor  $i \in L$  requer uma demanda  $d_i$  de entrega e cada consumidor  $i \in B$  requer uma demanda  $p_i$  de coleta. O número total de consumidores é dado por  $m+n$ . As demandas devem ser atendidas por uma frota de  $K$  veículos homogêneos de capacidade  $Q$ . O número máximo de veículos utilizados é dado por:  $K \geq \max\{K_L, K_B\}$ , onde  $K_L$  e  $K_B$  representam o número mínimo de veículos necessários para servir todas as demandas de entrega e coleta, respectivamente.

O objetivo do Problema de Roteamento de Veículos com *Backhauls* consiste em encontrar um conjunto de rotas de custo mínimo que iniciam e finalizam no depósito, sendo este custo definido como a soma dos custos  $c_{i,j}$  dos arcos inseridos na solução. As restrições do problema são: cada veículo deve ser utilizado uma única vez; as demandas de entrega devem ser satisfeitas antes do atendimento das demandas de coleta; nenhuma rota deve atender apenas demandas de coleta; o total de demanda de coleta e entrega de todos os consumidores de uma rota não deve exceder a capacidade máxima do veículo; e, por fim, cada consumidor deve ser visitado uma única vez por um único veículo para atendimento da sua demanda.

### 2.1.1 NOTAÇÃO

$c_{i,j}$	custo de percorrer a aresta $(i, j)$
$n$	número de pontos de demanda de entrega ( <i>linehauls</i> )
$m$	número de pontos de demanda de coleta ( <i>backhauls</i> )
$K$	número de veículos
$Q$	capacidade dos veículos
$d_i$	demanda de entrega do cliente $i$
$p_i$	demanda de coleta do cliente $i$

### 2.1.2 VARIÁVEL DE DECISÃO

$$x_{ijk} = \begin{cases} 1, & \text{se o veículo } k \text{ percorrer a aresta } (i,j) \\ i,j & 0,1,\dots,n,n+1,n+m, \text{ e } i \neq j \\ 0, & \text{caso contrário} \end{cases}$$

### 2.1.3 FORMULAÇÃO MATEMÁTICA

O modelo matemático do VRPB, definido por Tavakkoli-Moghaddam Saremi e Ziaee (2006), é apresentado na Eq.1.

$$\text{Minimizar } Z = \sum_{i=0}^{n+m} \sum_{j=0}^{n+m} \sum_{k=1}^K c_{i,j} x_{ijk} \quad (1)$$

O modelo possui as restrições apresentadas nas Equações 2 a 11.

$$\text{Minimizar } Z = \sum_{j=0}^{n+m} \sum_{k=1}^K x_{ijk} = 1, j = 1, \dots, n+m, i \neq j \quad (2)$$

$$\text{Minimizar } Z = \sum_{i=0}^{n+m} \sum_{k=1}^K x_{ijk} = 1, i = 1, \dots, n+m, j \neq i \quad (3)$$

$$\sum_{j=0}^{n+m} x_{0,jk} \leq 1, k = 1, \dots, K \quad (4)$$

$$\sum_{i=1}^{n+m} x_{i0k} \leq 1, k = 1, \dots, K \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^{n+m} d_i x_{ijk} \leq Q, k = 1, \dots, K \quad (6)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^{n+m} p_i x_{ijk} \leq Q, k = 1, \dots, K \quad (7)$$

$$\sum_{i=n+1}^{n+m} x_{ijk} - \sum_{i=n+1}^{n+m} x_{jik} = 0, j = 1, \dots, n+m, k = 1, \dots, K \quad (8)$$

$$\sum x_{ijk} \leq |S| - 1, S \subset 2, 3, \dots, n+m \quad (9)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^n \sum_{k=1}^K x_{ijk} = 0 \quad (10)$$

$$\sum x_{ijk} = 0, i, j = 0, \dots, n+m \text{ e } k = 1, \dots, K \quad (11)$$

O objetivo do problema apresentado no modelo (Eq. 1) é minimizar os custos de transporte. As equações (Eq. 2) e (Eq. 3) apresentam restrições que garantem que os pontos de demandas sejam atendidos uma única vez. As equações (Eq. 4) e (Eq. 5) indicam restrições do número máximo de veículos  $K$ , que poderá ser utilizado para satisfazer as demandas e que todas as rotas iniciem e finalizem no depósito. As equações (Eq. 6) e (Eq. 7) garantem que a carga máxima do veículo  $Q$  não seja extrapolada durante o atendimento das demandas de coleta e entrega. A equação (Eq. 8) representa a conservação de fluxo. A equação (Eq. 9) evita sub-rotas. A precedência do atendimento das demandas de entrega sobre as demandas de coleta é indicada na equação (Eq. 10).

## 2.2 ESTRATÉGIA DE SOLUÇÃO

Neste trabalho, foi utilizado um algoritmo baseado na metaheurística busca local iterativa (ILS) elaborado por Lourenço, Martin e Stützle, (2003) para resolver o VRPB. Este algoritmo requer a geração de uma solução inicial que é refinada após o uso de uma determinada busca local. Quando um mínimo local é alcançado, a esta solução aplica-se uma perturbação a fim de explorar outras regiões do espaço de soluções. A nova solução é então refinada e o processo é repetido até que determinado critério de parada seja atendido.

O pseudocódigo do método proposto para solucionar o VRPB é apresentado no Algoritmo 1.

Algoritmo 1 – método proposto para solucionar o VRPB.

Algoritmo ILS

**Require:** MaxIter, maxIterILS;

1:  $S = \text{gerarSolucaoInicial}();$

2:  $S = \text{BuscaLocal}(S);$

3:  $S^* = S;$

4:  $\text{iteracao} = 0;$

5: **while**  $\text{iteracao} < \text{MaxIter}$  **do**

6:      $S' = S;$

7:      $\text{iteracaoILS} = 0;$

8:     **while**  $\text{iteracaoILS} < \text{maxIterILS}$  **do**

9:          $S'' = \text{Perturbacao}(S');$

10:          $S'' = \text{BuscaLocal}(S'');$

11:         **if**  $f(S'') < f(S')$  **then**

12:              $S' = S'';$

13:              $\text{iteracaoILS} = 0;$

14:         **else**

15:              $\text{iteracaoILS} = \text{iteracaoILS} + 1;$

16:         **end if**

17:     **end while**

18:     **if**  $f(S') < f(S^*)$  **then**

19:          $S^* = S';$

20:     **end if**

21:      $\text{iteracao} = \text{iteracao} + 1;$

22: **end while**

23: **return**  $S^*;$

Fim Algoritmo ILS

A solução  $S$  representa a solução inicial,  $S^*$  a melhor solução encontrada até o momento,  $S'$  é a

solução corrente e,  $S''$ , a solução encontrada após procedimentos de perturbação e busca local.

O primeiro passo do algoritmo é encontrar uma solução inicial viável  $S$  para o problema (linha 1). Em seguida, esta solução é refinada aplicando-se um procedimento de busca local (linha 2). A variável  $S^*$  é inicializada com a solução  $S$ , sendo esta a melhor solução encontrada até o momento. A variável  $S'$  representa a solução corrente. Esta é inicialmente instanciada com a solução inicial  $S$  (linha 6). Em seguida, a solução  $S'$  passa por procedimentos de perturbação (linha 9), permitindo explorar outras soluções do espaço de busca, e busca local (linha 10), que refinam as soluções encontradas durante a busca. A solução modificada após estes procedimentos é armazenada em  $S''$ . Se  $S''$  for melhor que a solução corrente  $S'$  (linhas 11-16), então os procedimentos de perturbação e busca local serão novamente aplicados a  $S''$ . Este procedimento (linhas 8-17) é repetido até que determinado critério de interrupção seja satisfeito. Quando isso ocorre, verifica-se se uma melhor solução foi encontrada, atualizando  $S^*$ , se necessário (linhas 18-20), a solução corrente  $S'$  é novamente instanciada com a solução inicial  $S$  e os métodos de perturbação e busca local serão novamente aplicados a  $S'$ . Este procedimento (linhas 5-22) é repetido até que determinado critério de interrupção seja satisfeito.

Os detalhes dos métodos de geração da solução inicial, buscas locais e perturbações são descritos nas Seções 2.2.1, 2.2.2 e 2.2.3, respectivamente.

### 2.2.1 SOLUÇÃO INICIAL

Para gerar a solução inicial, o VRPB foi tratado inicialmente como dois Problemas de Roteamento de Veículos Capacitado (CVRP, *Capacitated Vehicle Routing Problem*), um contendo apenas os pontos de

demanda de coleta (*backhauls*), e outro contendo os pontos de demanda de entrega (*linehauls*). Para solucionar os dois problemas gerados, foi aplicado o algoritmo baseado no ILS, proposto por Penna, Subramanian e Ochi (2011).

Dadas as duas soluções encontradas, o próximo passo é encontrar uma melhor forma de unir as rotas contendo apenas os pontos de demanda de entrega às rotas que atendem aos pontos de demanda de coleta. Assim, foi utilizado um algoritmo de atribuição (RIGDON, 2008) em um grafo bipartido  $G'=(V',E')$  em que  $V'=V'_L \cup V'_B$ .  $V'_L$  corresponde às rotas obtidas na resolução do CVRP, considerando apenas as demandas de entrega e,  $V'_B$ , as rotas obtidas após resolução do CVRP, considerando as demandas de coleta. Para cada aresta  $(l,b) \in E'$  é associado um valor  $C_{l,b}$  que indica o custo da rota resultante da união entre as rotas  $l$  (*linehauls*) e  $b$  (*backhauls*). Este custo é definido após avaliar 4 maneiras distintas de conectar estas duas rotas, apresentadas nas figuras a seguir.

Dada uma rota  $l$  que atenda a determinados pontos de entrega e outra rota  $b$  que atenda a pontos de coleta (Figura 1), a primeira possibilidade de junção de ambas as rotas consiste na remoção das arestas de retorno ao depósito da rota  $l$ , remoção da aresta de partida do depósito da rota  $b$  e inclusão da aresta que conecta o último cliente da rota  $l$  ao primeiro cliente da rota  $b$  (Figura 2). A segunda situação considera a conexão entre o último cliente da rota  $l$  e último cliente da rota  $b$ . Assim, é necessário inverter o sentido da rota  $b$  (Figura 3). A terceira situação, ilustrada na (Figura 4), considera a conexão entre o primeiro cliente da rota  $l$  e o primeiro cliente da rota  $b$ . Neste caso, é necessário inverter o sentido da rota  $l$ . Por fim, na quarta situação, as duas rotas devem

ser invertidas e é feita a conexão entre o primeiro cliente da rota  $l$  com o último cliente da rota  $b$  (Figura 5). Assim, o custo da rota gerada após união das rotas  $l$  e  $b$  é dado por:

$$C_{lb} = \min(C_{lb}^1, C_{lb}^2, C_{lb}^3, C_{lb}^4)$$

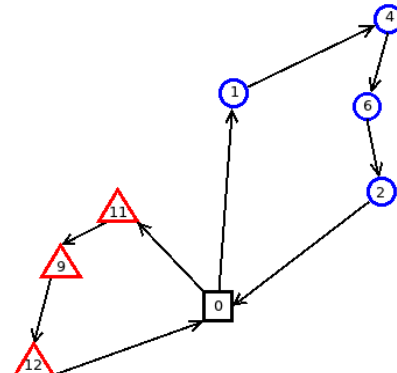


Figura 1 - Soluções Originais

1

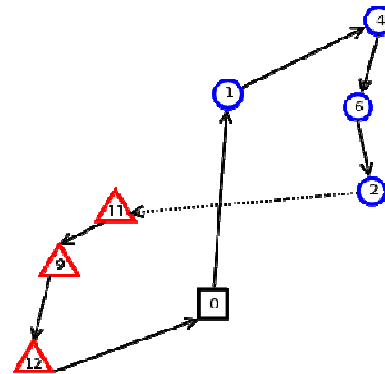


Figura 2 - Situação 1 custo  $C_{lb}^1 = 110$

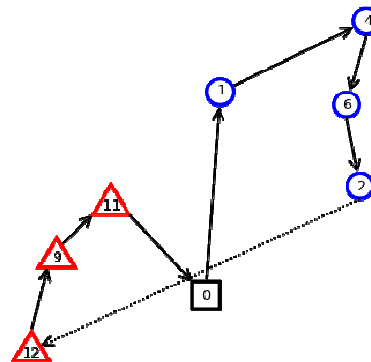
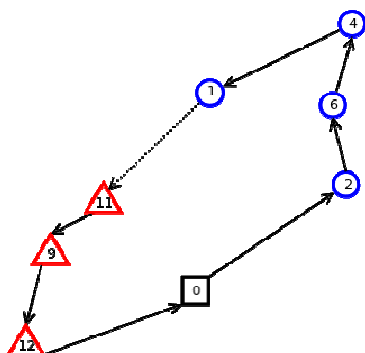
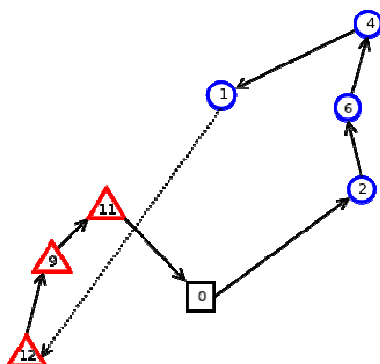
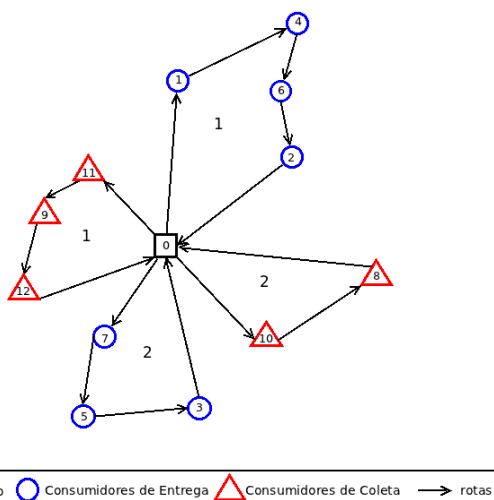


Figura 3 - Situação 2 custo  $C_{lb}^2 = 140$

Figura 4 - Situação 3 custo  $C_{lb}^3 = 100$ Figura 5 - Situação 4 custo  $C_{lb}^4 = 130$ 

Essa avaliação é feita para todas as combinações possíveis entre as rotas contidas na solução do CVRP, a Figura 6 contendo apenas os pontos de demanda de entrega e as rotas contidas na solução do CVRP com os pontos de demanda de coleta. Na Figura 7 é apresentado o problema de atribuição gerado, dadas estas duas soluções do CVRP.



□ Depósito ○ Consumidores de Entrega ▲ Consumidores de Coleta → rotas

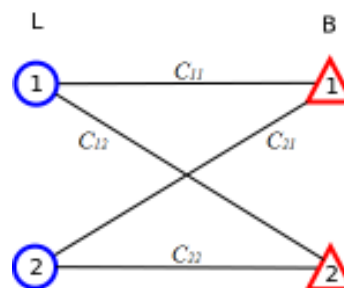
Figura 6 - Soluções *linehaults* e soluções *backhaults*

Figura 7 - Atribuição 1

## 2.2.2 BUSCAS LOCAIS

Para realizar o procedimento de busca local foi implementado um algoritmo baseado no RVND (*Random Variable Neighborhood Descent*), proposto por Penna, Subramanian e Ochi (2011). A busca local procura encontrar a melhor solução dentro de um espaço de soluções através de movimentos específicos. Neste trabalho foi utilizado estruturas de vizinhança inter-rotas e intra-rota.

Nas estruturas de vizinhança inter-rotas procura-se melhorar as soluções trocando consumidores entre as rotas presentes nas soluções. As buscas locais utilizadas são: *Shift(1,0)*, *Shift(2,0)*, *Swap(1,1)*, *Swap(2,1)*, *Swap(2,2)* e *Crossover*. A busca local *Shift(1,0)* reinsere um consumidor  $k$  de uma rota  $R_1$  para uma rota  $R_2$ . Na *Shift(2,0)* dois consumidores adjacentes são removidos de uma rota  $R_1$  e inseridos em uma rota  $R_2$ , mas não necessariamente na mesma ordem. A busca local *Swap(1,1)* permuta dois consumidores de rotas distintas. A busca local *Swap(2,1)* são selecionados e removidos dois consumidores adjacentes  $C_1$  e  $C_2$  de uma rota  $R_1$  e um cliente  $C_3$  de uma rota  $R_2$ , os consumidores  $C_1$  e  $C_2$  são inseridos na rota  $R_2$  e  $C_3$  é inserido na rota  $R_1$ , sendo que  $C_1$  e  $C_2$  não necessitam permanecer na mesma ordem. A busca local *Swap(2,2)* permuta dois consumidores adjacentes  $C_1$

e  $C_2$  da rota  $R_1$  com os consumidores  $C_3$  e  $C_4$  também adjacentes de uma rota  $R_2$ , sendo que, após a troca estes consumidores não precisam permanecer adjacentes. Na busca local *Crossover* são selecionados e removidos dois consumidores adjacentes  $C_1$  e  $C_2$  pertencentes a uma rota  $R_1$ , e dois consumidores adjacentes  $C_3$  e  $C_4$  de uma rota  $R_2$ . Em seguida, uma aresta é inserida ligando  $C_1$  a  $C_4$  e outra ligando  $C_3$  a  $C_2$ .

Todas as buscas locais são melhor-aprimorantes e antes de sua execução as restrições do problema são verificadas. Dessa forma, assegura-se que apenas soluções viáveis sejam geradas.

Nas estruturas de vizinhança intra-rota, troca-se a posição dos consumidores dentro de uma mesma rota. As buscas utilizadas são: *2-Opt*, *Exchange*, *OrOpt*, *OrOpt2* e *OrOpt3*. Na busca local *2-Opt* são removidas duas arestas não adjacentes  $(i_1, i_2)$  e  $(i_3, i_4)$ , e são inseridas duas novas arestas  $(i_1, i_3)$  e  $(i_2, i_4)$ . A busca local *Exchange* troca dois consumidores de posição na rota. A busca local *OrOpt* muda a posição de um consumidor na rota e as buscas locais *OrOpt2* e *OrOpt3* trocam 2 e 3 consumidores adjacentes, respectivamente.

### 2.2.3 PERTURBAÇÕES

As perturbações selecionam clientes aleatórios e os realocam em uma posição viável qualquer. O objetivo é explorar novas soluções sem preocupar com a qualidade das mesmas. Neste trabalho foram utilizadas as perturbações: *DoubleSwap*, *EjectionChain*, *MultipleSwap* e *MultipleShift*.

A perturbação *DoubleSwap* realiza dois movimentos *Swap(1,1)*. Na perturbação *EjectionChain* é escolhido

um consumidor da rota  $R_1$  de forma aleatória e o insere na rota  $R_2$ . Logo após é escolhido um consumidor aleatório de  $R_2$  e insere em  $R_3$ . Este processo continua até que um consumidor da última rota seja inserido na primeira rota. Os mecanismos de perturbação *MultipleSwap* e *MultipleShift* realizam múltiplos movimentos *Swap(1,1)* e *Shift(1,0)*, respectivamente. O número de movimentos realizados varia entre  $[0, \text{número de rotas}/2]$ . Em todos os movimentos de permutação o consumidor é inserido na primeira posição viável encontrada.

## 2.3 RESULTADOS COMPUTACIONAIS

O algoritmo proposto neste trabalho foi implementado na linguagem C++ e executado em uma máquina com processador Intel Core I7, 2.93GHz, com 8 GB de memória RAM e Sistema Operacional Linux - Distribuição Ubuntu 11.04.

Para a realização dos testes, os parâmetros *MaxIter* e *maxIterILS* do ILS foram fixados com os valores 15 e 80, respectivamente. O algoritmo foi avaliado utilizando um grupo de 62 instâncias, propostas por Goetschalckx e Jacobs-Blecha (1989), que são comumente encontradas na literatura. Estas instâncias podem conter entre 25 a 100 pontos de demanda de entrega ( $n$ ) e entre 5 a 50 pontos de demanda de coleta ( $m$ ). O número total de pontos de demanda ( $n + m$ ) varia entre 25 a 150.

As Tabelas 1, 2 e 3 apresentam uma comparação dos resultados obtidos pelo algoritmo proposto e os melhores resultados encontrados na literatura, Brandão (2005). A primeira coluna indica o nome da instância. A segunda coluna apresenta o número total de pontos de demanda ( $n + m$ ). A terceira e quarta colunas mostram o número de pontos de demanda de



entrega ( $n$ ) e coleta ( $m$ ), respectivamente, a quinta coluna a capacidade máxima dos veículos e a sexta o número máximo de veículos. A sétima coluna mostra os melhores resultados da literatura e a oitava e nona colunas mostram os resultados obtidos com o algoritmo proposto baseado no ILS. A última coluna indica a diferença percentual entre os resultados obtidos e os melhores resultados da literatura.

Observa-se nas tabelas que o método proposto foi

capaz de reduzir o custo de transporte em 7 instâncias avaliadas (A3, D1, E2, F2, G1, G4 e G5). Em outras 10 instâncias as soluções apresentam custo igual aos resultados da literatura. Nas demais instâncias estes valores são muito próximos. A diferença percentual média entre os resultados da literatura e do algoritmo proposto foi de 3,45%. Portanto, o método demonstrou uma capacidade em gerar resultados competitivos, próximos aos encontrados na literatura.

Tabela 1

Tabela de Resultados

Instâncias						Melhores da Literatura	ILS		GAP
Nome	$n+m$	$n$	$m$	Q	Veículos	Custo	Veículos	Custo	-
A1	25	20	5	1550	8	<b>229.886</b>	8	<b>229.886</b>	0
A2	25	20	5	2550	5	<b>180.119</b>	5	<b>180.119</b>	0
A3	25	20	5	4050	4	163.405	3	<b>155.796</b>	-4,65
A4	25	20	5	4050	3	<b>155.796</b>	3	<b>155.796</b>	0
B1	30	20	10	1600	7	<b>239.080</b>	7	<b>239.080</b>	0
B2	30	20	10	2600	5	<b>198.048</b>	4	198.150	0,05
B3	30	20	10	4000	3	<b>169.372</b>	3	<b>169.372</b>	0
C1	40	20	20	1800	7	<b>249.448</b>	6	260.776	4,54
C2	40	20	20	2600	5	<b>215.020</b>	5	<b>215.020</b>	0
C3	40	20	20	4150	5	<b>199.346</b>	3	201.550	1,10
C4	40	20	20	4150	4	<b>195.366</b>	3	201.550	3,16
D1	38	30	8	1700	12	322.530	11	<b>318.301</b>	-1,31
D2	38	30	8	1700	11	<b>316.709</b>	11	318.301	0,50
D3	38	30	8	2750	7	<b>239.479</b>	7	<b>239.479</b>	0
D4	38	30	8	4075	5	<b>205.832</b>	5	<b>205.832</b>	0
E1	45	30	15	2650	7	<b>238.880</b>	7	<b>238.880</b>	0
E2	45	30	15	4300	4	212.263	4	<b>212.123</b>	-0,06
E3	45	30	15	5225	4	<b>206.659</b>	4	206.788	0,06
F1	60	30	30	3000	6	<b>263.173</b>	6	265.276	0,79
F2	60	30	30	3000	7	265.213	6	<b>264.878</b>	-0,12
F3	60	30	30	4400	5	<b>241.120</b>	4	245.277	1,72
F4	60	30	30	5500	4	<b>233.861</b>	4	235.709	0,79

Tabela 2

Continuação da Tabela de Resultados

Nome	Instâncias				Veículos	Melhores da Literatura	ILS		GAP
	$n+m$	$n$	$m$	$Q$		Custo	Veículos	Custo	-
G1	57	45	12	2700	10	306.305	9	<b>305.317</b>	-0,32
G2	57	45	12	4300	6	<b>245.441</b>	6	<b>245.441</b>	0
G3	57	45	12	5300	5	<b>229.507</b>	5	231.045	0,67
G4	57	45	12	5300	6	232.521	5	<b>231.045</b>	-0,63
G5	57	45	12	6400	5	221.730	4	<b>220.927</b>	-0,36
G6	57	45	12	8000	4	<b>213.457</b>	3	226.367	6,04
H1	68	45	23	4000	6	<b>268.933</b>	6	271.214	0,84
H2	68	45	23	5100	5	<b>253.365</b>	5	267.237	5,47
H3	68	45	23	6100	4	<b>247.449</b>	4	261.020	5,48
H4	68	45	23	6100	5	<b>250.221</b>	4	260.748	4,20
H5	68	45	23	7100	4	<b>246.121</b>	4	276.358	12,28
H6	68	45	23	7100	5	<b>249.135</b>	3	269.412	8,13
I1	90	45	45	3000	10	<b>350.246</b>	9	359.355	2,60
I2	90	45	45	4000	7	<b>309.943</b>	7	334.703	7,98
I3	90	45	45	5700	5	<b>294.507</b>	5	312.512	6,11
I4	90	45	45	5700	6	<b>295.988</b>	5	312.655	5,63
I5	90	45	45	5700	7	<b>301.226</b>	5	314.670	4,46
J1	94	75	19	4400	10	<b>335.006</b>	10	338.912	1,16
J2	94	75	19	5600	8	<b>310.417</b>	8	316.892	2,08
J3	94	75	19	8200	6	<b>279.219</b>	6	313.705	12,35
J4	94	75	19	6600	7	<b>296.533</b>	7	311.673	5,10
K1	113	75	38	4200	10	<b>394.376</b>	10	415.859	5,44
K2	113	75	38	5200	8	<b>362.130</b>	8	388.749	7,35
K3	113	75	38	5200	9	<b>365.693</b>	8	374.540	2,41

Tabela 3

Continuação da Tabela de Resultados

Instâncias						Melhores da Literatura	ILS		GAP
Nome	$n+m$	$n$	$m$	$Q$	Veículos	Custo	Veículos	Custo	-
K4	113	75	38	6200	7	<b>348.950</b>	7	364.345	4,41
L1	150	75	25	4400	10	<b>425.772</b>	10	441.910	3,79
L2	150	75	25	5000	8	<b>401.228</b>	8	435.819	8,62
L3	150	75	25	5000	8	<b>402.720</b>	8	432.867	7,48
L4	150	75	25	6000	6	<b>384.637</b>	6	421.760	9,65
L5	150	75	25	6000	7	<b>387.928</b>	7	422.527	8,91
M1	125	100	25	5200	11	<b>398.593</b>	10	408.756	2,54
M2	125	100	25	5200	10	<b>396.917</b>	10	415.496	4,68
M3	125	100	25	6200	9	<b>376.309</b>	9	387.328	2,92
M4	125	100	25	8000	7	<b>348.418</b>	7	372.066	6,78
N1	150	100	50	5700	11	<b>408.101</b>	10	444.135	8,82
N2	150	100	50	5700	10	<b>408.066</b>	10	441.206	8,12
N3	150	100	50	6600	9	<b>394.338</b>	9	415.374	5,33
N4	150	100	50	6600	10	<b>396.055</b>	9	423.783	7,00
N5	150	100	50	8500	7	<b>373.477</b>	7	393.026	5,23
N6	150	100	50	8500	8	<b>374.691</b>	7	407.131	8,65

**Negrito – Melhores Resultados.**

### 3 CONCLUSÃO

Para resolver o Problema de Roteamento de Veículos com Backhauls (VRPB), este trabalho propôs um algoritmo heurístico baseado na Busca Local Iterativa (ILS). A geração da solução inicial deste algoritmo foi obtida através da junção de duas outras soluções do Problema de Roteamento de Veículos Capacitado já otimizadas.

Este método se mostrou bastante eficiente ao ser comparado com os melhores algoritmos encontrados

na literatura, retornando soluções de boa qualidade. A diferença percentual entre eles não foi muito inferior, na média 3,45%. Em algumas instâncias o método foi capaz de superar os resultados da literatura. A aplicação do ILS demonstrou boa capacidade em aprimorar as soluções obtidas.

Trabalhos futuros podem ser direcionados na aplicação de técnicas paralelas para redução no tempo de processamento do algoritmo, bem como a aplicação de novas estruturas de vizinhança com o intuito de aprimorar os resultados obtidos.

## REFERÊNCIAS

- BRANDÃO, José. A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, Braga, 01, abr. 2005. Disponível em <<http://www.inf.u-szeged.hu/~cimreh/backhauls.pdf>>. Acesso em: 02 dez. 2011.
- CRISPIM, José; BRANDÃO, José, Reactive tabu search and variable neighbourhood descent applied to the vehicle routing problem with backhauls. In: *Metaheuristics International Conference*, 4, 2001, Braga. Disponível em <<http://140.113.119.114/Thesis/thesis/097/1/Literature/VRPB/2001-Reactive%20Tabu%20Search%20and%20Variable%20Neighbourhood%20Descent%20Applied%20to%20the%20vehicle%20routing%20probl.pdf>>. Acesso em: 23 abr. 2012.
- DANTZIG, G. B.; RAMSER, J. H.. The truck dispatching problem. *Management Science*. v.6, n.1, 80-91, out. 1959. ISSN 0025-1909.
- FLEURY, P. F., *Nota Sobre O Setor De Transporte De Cargas No Brasil*, COPPEAD/UFRJ, 2010. Disponível em : <<http://www.logisticadescomplicada.com/o-setor-de-transporte-de-cargas-no-brasil/>>. Acesso em: 20 mar. 2012.
- GOETSCHALCKX, M.; JACOBS-BLECHA, C.. The vehicle routing problem with backhauls. *European Journal of Operational Research*, North-Holland, Out 1989. Theory and Methodology, 39–51.
- LOURENÇO, H. R.; MARTIN, O. C., STÜTZLE, T., Iterated Local Search, *International Series in Operations Research & Management Science*, v.57, capítulo 11, 2003.
- MINGOZZI, A.; GIORGI, S.; BALDACCI, R., An exact method for the vehicle routing problem with backhauls. *Transportation Science*, 3 Ago 1999. Disponível em <<http://140.113.119.114/students/thesis/098/1/Literature/4292826.pdf>>. Acesso em : 30 ago. 2011.
- PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *J Heuristics*, 08 Set 2011, Disponível em : <<http://www.ic.uff.br/~satoru/conteudo/artigos/PAPER%20PUCA-JHeuristics-2011.pdf>>. Acesso em: 21 mar. 2012.
- RIGDON, R., *An implementation of the Kuhn-Munkres Assignment Algorithm*, Disponível em : <<http://code.google.com/p/hungarianassignment/>>. Acesso em : 14 jan. 2012.
- TAVAKKOLI-MOGHADDAM, R.; SAREMI, A.; ZIAEE, M., A memetic algorithm for a vehicle routing problem with backhauls, *Applied Mathematics and Computation*, Tehran, 2006. Disponível em <<http://140.113.119.114/Thesis/thesis/097/1/Literature/VRPB/2006-A%20memetic%20algorithm%20for%20a%20vehicle%20routing%20problem%20with%20backhauls.pdf>>. Acesso em: 29 mar. 2012.
- TOTH, P.; VIGO, D.. An exact algorithm for the vehicle routing problem with backhauls. *Transportations Science*, Bologna, Nov 1997. Disponível em <<http://140.113.119.114/students/thesis/098/1/Literature/4452817.pdf>>. Acesso em: 30 ago. 2011.