

**VARIABLE NEIGHBORHOOD SEARCH E ITERATED LOCAL SEARCH APLICADOS
AO PROBLEMA DE REDE DE TRANSPORTE RODOVIÁRIO COM CARGA
FRACIONADA**

Carlos Eduardo Dutra

Universidade Candido Mendes – UCAM-Campos, NPD
Rua Anita Pessanha 100, Campos dos Goytacazes, RJ, 28040-320, Brasil.
carlos2004.dutra@terra.com.br

Fermín Alfredo Tang Montané

Universidade Candido Mendes – UCAM-Campos, NPD
Rua Anita Pessanha 100, Campos dos Goytacazes, RJ, 28040-320, Brasil.
tang@ucam-campos.br

RESUMO

O problema de Transporte Rodoviário com Cargas Fracionadas (PTRCF) consiste em atender pedidos de transporte de carga entre pares de depósitos geograficamente dispersos. Os depósitos são interligados entre si mediante um conjunto de rotas, que são percorridas por veículos idênticos. O objetivo é determinar o conjunto de rotas que minimize o custo total de transporte e atenda às restrições de demanda, prazo de entrega e capacidade dos veículos. O transbordo de carga é permitido entre as rotas. O presente artigo propõe o uso das metaheurísticas *Variable Neighborhood Search* (VNS) e *Iterated Local Search* (ILS) para resolver o problema. Os algoritmos propostos foram testados em um conjunto de 48 instancias teste, onde varia-se o número de depósitos e as demandas dos pedidos. Os conjuntos de demandas correspondem aos dados reais de uma transportadora com sede no estado de Espírito Santo.

PALAVRAS CHAVE. Variable Neighborhood Search. Iterated Local Search. Transporte de carga. Metaheurísticas.

ABSTRACT

The Less-than-Truck Trucking Problem (LTTP) is to meet requests for cargo transport between pairs of depots geographically dispersed. The depots are connected to each other through a set of routes that are traveled by identical vehicles. The objective is to determine the set of routes that minimizes total transportation costs and satisfies all the demands, not exceeding the maximum delivery time and vehicle capacity constraints. Cargo transshipments are allowed among the routes. This paper proposes the use of two metaheuristics: Variable Neighborhood Search (VNS) and Iterated Local Search (ILS) to solve the problem. The algorithms were tested on a set of 48 test instances, in which the number of depots and the amount demanded by clients varies. The sets of data correspond to the demands of an actual carrier based in Espírito Santo state.

KEYWORDS. Variable Neighborhood Search. Iterated Local Search. Freight transportation. Metaheuristics.

1. Introdução

No Problema de Transporte Rodoviário (PTR) considera-se um conjunto de depósitos que servem como ponto de concentração de cargas para os clientes de uma região próxima ao depósito. O problema consiste em definir uma rede de transporte através de um conjunto de rotas interligadas entre si, que permita atender pedidos de transporte de carga entre pares de depósitos, de maneira a minimizar o custo total de transporte. No caso mais geral, o problema considera frota de veículos heterogênea. O PTR compreende 4 tipos de operações de cargas: coleta, transferência, transbordo e entrega.

Aborda-se a variante do PTR no caso específico com carga fracionada (PTRCF). Entende-se por carga fracionada aquela onde um mesmo veículo atende pedidos de transporte de diversos clientes de maneira simultânea. Neste problema, consideram-se restrições no prazo de entrega dos pedidos, assim como uma frota de veículos homogênea e ilimitada. Além disso, vale observar que as rotas são dependentes umas das outras, devido às operações de transbordo.

A carga a ser transportada entre um depósito origem e um depósito destino, pode utilizar várias rotas interligadas entre si (ver Figura 1). No exemplo da Figura 1, procura-se atender ao pedido de transporte entre o depósito origem 1 ao depósito destino 8. O pedido é atendido seguindo o percurso (1-2-4-5-7-8) que é coberto pelas rotas 1, 2 e 3. Observa-se que, as três rotas estão interligadas entre si mediante 2 pontos de transbordo, nos depósitos 4 e 5. Enquanto as rotas 1 e 3 são seguidas por inteiro, apenas o trecho 4-5 da rota 2 é utilizado. No exemplo, a rota 1 concentra carga no trecho 4-5 da rota 2, mediante uma operação de transbordo. As operações de transbordo visam o aproveitamento máximo da capacidade do veículo, mediante a concentração de carga em trechos específicos. Uma hipótese adicional deste problema é que a rota de ida entre dois depósitos não é necessariamente igual à rota de retorno.

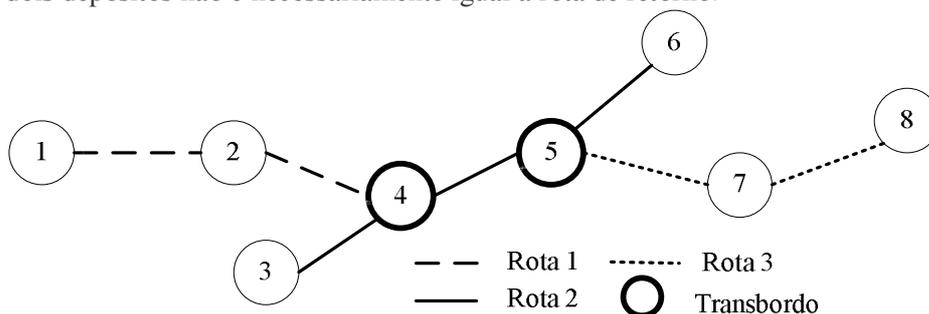


Figura 1. – Operações de transportes.

O PTRCF pode ser definido matematicamente da seguinte maneira: Seja $G = (V, A)$ um grafo direcionado onde todos os vértices $V = \{1, \dots, n\}$ representa o conjunto de depósitos e todos os arcos $A = \{(i, j) : i, j \in V, i \neq j\}$ representa o conjunto de trechos possíveis. A cada trecho (i, j) , associa-se o custo de percurso, c_{ij} , assim como o tempo de percurso, t_{ij} , ambos em função da distância. Para cada depósito k , considera-se, sempre que necessário, o tempo de transbordo, b_k , que é uma constante independente do volume de carga. Existem pedidos de transporte de carga entre os pares de depósitos i, j . Cada pedido deverá ser atendido dentro de um prazo comercial (tempo máximo disponível). As rotas adotadas pelos veículos são denotadas como uma seqüência de depósitos $(i \dots k \dots j)$. O tempo gasto no percurso da rota é denotado como $T(i \dots k \dots j)$. Este tempo é calculado como a soma dos tempos de percurso de cada trecho mais os tempos de transbordo caso existam. Considera-se que a frota de veículos é homogênea, onde Q representa a capacidade do veículo. O PTRCF consiste em determinar um conjunto de rotas de maneira que: (1) todos os pedidos sejam atendidos; (2) a carga máxima na rota não exceda a capacidade do veículo; (3) o tempo gasto no atendimento de cada pedido não exceda seu prazo comercial; e (4) o custo total seja mínimo.

Ao projetar a rede de transporte, por um lado é importante considerar percursos com vários transbordos entre rotas para concentrar carga e reduzir os custos de transferência. No

entanto, em termos de prazo de entrega é melhor manter um número baixo de transbordos. Sempre que o prazo de entrega for violado o algoritmo procurará caminhos alternativos.

No presente trabalho são propostos dois algoritmos com base na metaheurística *Variable Neighborhood Search* (VNS) (MLADENOVIC e HANSEN, 1997) e *Iterated Local Search* (ILS) (LOURENÇO *et. al.*, 2002), ambas utilizam como busca local o método *Variable Neighborhood Descent* (VND) (MLADENOVIC e HANSEN, 1997) para resolver o PTRCF. Os algoritmos implementados utilizam como sub-rotinas os algoritmos de caminho mínimo (DIJKSTRA, 1959) e k-caminho mínimo restrito (*Constrained k-Shortest Paths*) (VAN DER ZIJPP *et. al.*, 2004). O trabalho está organizado da seguinte forma: Seção 2, revisão bibliográfica referente ao problema PTRCF; Seção 3 a metodologia proposta para a resolução do PTRCF; Seção 4 resultados computacionais e Seção 5 conclusões do trabalho.

2. Revisão Bibliográfica

Problemas co-relacionados foram estudados em (COUTO, 2004). O autor desenvolveu métodos de solução com base em programação matemática e algoritmos heurísticos. Dois problemas específicos foram estudados: o problema de transferência de carga (PTC) e o problema de alocação de carga (PAC). O problema de transferência de carga (PTC) consiste em determinar a quantidade e os tipos de veículos, assim como os modos de contratação entre frota própria e terceirizada de maneira a maximizar o lucro resultante da transferência de carga entre cidades. Foi proposto um modelo matemático para este problema que considera que todas as demandas devem ser atendidas e incluem 7 restrições operacionais. Por outro lado, o problema de alocação de cargas (PAC) consiste em definir a alocação de pedidos de transferência de carga dos clientes às rotas existentes de maneira a minimizar o custo de transporte. Este problema considera restrições de capacidade mínima e máxima para cada veículo. Além disso, considera frota de veículos heterogênea. O PAC foi resolvido desenvolvendo-se três abordagens: um modelo matemático, uma heurística com base na técnica de Geração de Colunas e um algoritmo com base na técnica *Branch-and-Price*.

Em ambos os problemas abordados pelo autor (PTC e PAC), adotou-se o modelo de operação de carga fechada ou lotação. Essa hipótese garante o uso de rotas independentes para cada cliente sem necessidade de transbordos, diferentemente do caso do PTRCF tratado no presente trabalho que considera rotas dependentes com transbordos. Vale observar que o autor não utilizou janelas de tempos para o atendimento dos pedidos.

A literatura é escassa para o problema de transporte rodoviário com carga fracionada (PTRCF). Recente estudo sobre o problema específico foi realizado por Dutra e Montané (2009), que propôs quatro algoritmos com base na heurística VND, denominado de VND iterado. A mesma definição matemática proposta por Dutra e Montané (2009) foi descrita na seção 1. Os algoritmos propostos possuem duas etapas. Uma etapa de busca local utilizando o *movimento de redução e rotas (MOV-RR)* e o *movimento de realocação de pedidos (MOV-RP)* e uma etapa de diversificação com base em um mecanismo de perturbação e *movimento de inserção de depósitos intermediários (MOV-INS)*. O melhor resultado encontrado foi executando o mecanismo de perturbação após a busca local, considerando-se os movimentos a partir do melhor aprimorante.

O trabalho proposto surgiu da necessidade de se encontrar melhores soluções ao problema de roteirização e programação de equipamentos de transportes. Este problema aplica-se a empresas de transportes rodoviários, marítimos e aéreos.

3. Metodologia Proposta

Na presente Seção, descreve-se a metodologia utilizada na construção de diferentes implementações da metaheurística VNS e ILS aplicada ao PTRCF. Na seção 3.1 descreve-se o método construtivo, utilizado para obter a solução inicial. Na seção 3.2 descreve-se a busca local com dois movimentos, MOV-RR e MOV-RP que são descritos nas seções 3.2.1 e 3.2.2 respectivamente. Na seção 3.3 descreve-se a heurística VND. Na seção 3.4 descrevem-se dois movimentos de perturbação, MOV-INS e MOV-RPA que são descritos nas seções 3.4.1 e 3.4.2

respectivamente. E finalmente nas seções 3.5 e 3.6 descrevem-se as metaheurísticas VNS e ILS utilizadas para a resolução do problema PTRCF.

3.1. Heurística construtiva

A heurística construtiva, *criação de rotas sob demanda e alocação de pedidos* (CRSDA) utilizada no presente trabalho, foi proposta por Dutra e Montané (2009). A heurística tem como objetivo definir rotas para atender um conjunto de pedidos. Cada pedido P_i é identificado pelo par de depósitos (o_i, d_i) , origem e destino, respectivamente. Associado a cada pedido P_i existe uma quantidade de carga K_i . A heurística realiza duas operações básicas: a criação de rotas que permitam o transporte da carga de um pedido e a alocação parcial ou total da carga de diferentes pedidos as rotas existentes. A heurística compreende três etapas. As duas primeiras etapas são de criação de rotas, enquanto que a terceira etapa é de alocação de carga. As etapas da heurística são as seguintes:

Etapa 1. Criação de rotas primárias - Dado um pedido $P_i = (o_i, d_i)$, cria-se uma rota para atender o pedido aplicando o algoritmo de caminho mínimo (DIJKSTRA, 1959).

Etapa 2. Criação de rotas alternativas - Caso não seja possível atender ao pedido $P_i = (o_i, d_i)$, utilizando a rota calculada na etapa 1, calcula-se uma rota alternativa aplicando o algoritmo de k-caminho mínimo restrito, *Constrained k-Shortest Path*, proposto por Van der Zijpp et.al, (2004).

Etapa 3. Alocação de pedidos as rotas existentes - Sempre que uma nova rota criada nas etapas 1 ou 2 consegue atender de forma completa o pedido corrente $P_i = (o_i, d_i)$, o algoritmo tenta alocar outros pedidos pendentes na nova rota.

As três etapas são repetidas para cada pedido pendente, até que todos os pedidos se encontrem alocados a uma rota e sejam atendidos de forma completa. O processamento dos pedidos é realizado priorizando aqueles pedidos de maior peso. Para isso, os algoritmos foram classificados em ordem decrescente do peso utilizando o algoritmo *QuickSort* (HOARE, 1961; HOARE, 1962). Cada uma das etapas da heurística construtiva será detalhada nas subseções 3.1.1, 3.1.2 e 3.1.3.

3.1.1. Criação de rotas primárias

Dado um pedido $P_i = (o_i, d_i)$, cria-se uma rota para atendê-lo aplicando o algoritmo de caminho mínimo (DIJKSTRA, 1959). Em seguida, verifica-se se as restrições de prazo de entrega e de capacidade do veículo são satisfeitas. Se ambas as restrições forem satisfeitas, o pedido P_i é alocado à rota criada. Se apenas a restrição de capacidade não for satisfeita, criam-se réplicas da rota original para atender o excedente de carga. No caso da restrição de prazo de entrega ser violada, executa-se a etapa de criação de rota alternativa, detalhada na próxima seção.

3.1.2. Criação de rotas alternativas

Uma rota alternativa é criada utilizando o algoritmo de k-caminho mínimo restrito *Constrained k-Shortest Paths (CKSP)*, proposto por Van der Zijpp et.al, (2004). O algoritmo realiza a enumeração implícita de caminhos mínimos com restrições adicionais. Esse algoritmo é uma variante do algoritmo *K-Shortest Paths* proposto por Lawler, (1976). No caso específico do PTRCF, considera-se a restrição de prazo de entrega como restrição adicional.

Na Figura 2, mostra-se uma rede de transporte que apresenta dados referentes às distâncias a serem percorridas, o tempo de transferência em cada trecho e o tempo de transbordo em cada depósito. A Figura 3 ilustra a execução do algoritmo de k-caminho mínimo restrito para o exemplo da Figura 2. O algoritmo inicia a partir do cálculo do caminho mínimo global (4-5-3-2) (DIJKSTRA, 1959). Considera-se que existe um tempo de transbordo de duas horas ($b_k=2$) em cada depósito intermediário, devido a operações de carregamento e descarregamento dos veículos. Como este caminho não atende a restrição de prazo de entrega (24 horas), diferentes subproblemas são gerados explorando de maneira sistemática todas as combinações possíveis de remoção e inserção dos arcos presentes no caminho mínimo global (4-5-3-2). Cada subproblema

corresponde a um problema de caminho mínimo modificado que é resolvido pelo algoritmo de Dijkstra (1959).

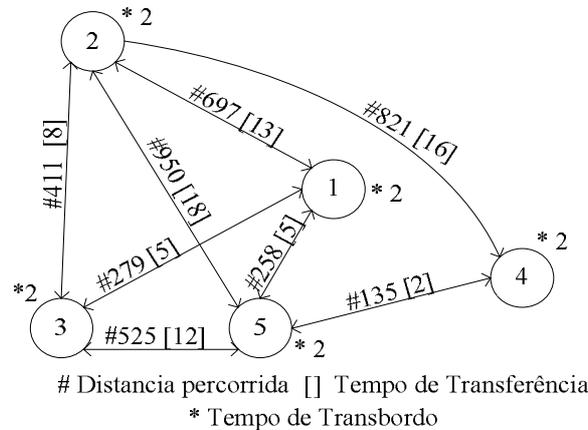


Figura 2. – Exemplo de rede de transporte.

Na Figura 3, o caminho mínimo resultante do terceiro subproblema (4-5-2) atende a restrição de prazo de entrega. O algoritmo termina sem conseguir encontrar outros caminhos mínimos alternativos.

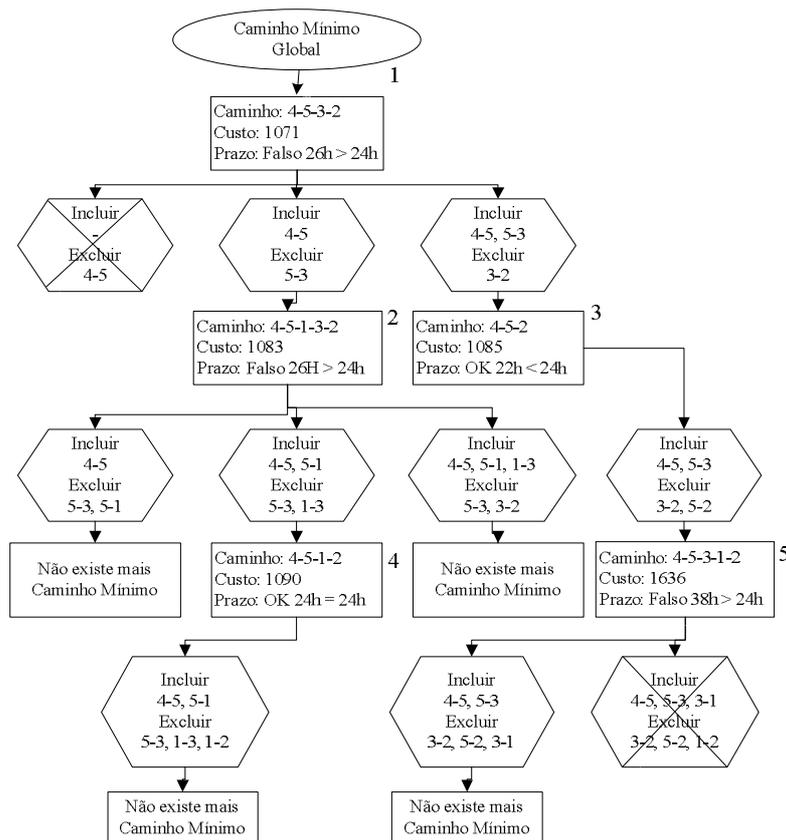


Figura 3. –Execução do algoritmo k-caminho mínimo restrito.

3.1.3. Alocação de pedidos em rotas existentes

Nesta etapa, realiza-se a alocação dos pedidos pendentes às rotas criadas nas etapas 1 e 2 da heurística CRSDA (DUTRA e MONTANÉ, 2009). O algoritmo testa a possível alocação de cada pedido pendente P_k em alguma das rotas existentes até encontrar uma rota destino viável r_i

ou testar todas as rotas. A alocação de um pedido pendente P_k a uma rota destino r_i é validada fazendo-se as seguintes verificações: 1) os depósitos origem e destino do pedido devem fazer parte da rota destino r_i ; 2) a capacidade de carga disponível na rota destino r_i deve ser superior a demanda do pedido P_k ; 3) a restrição de prazo de entrega não deve ser excedida. O algoritmo retorna a etapa 1 da heurística CRSDA quando não é possível fazer a alocação de um pedido P_k . O algoritmo termina quando todos os pedidos estiverem alocados a uma rota.

3.2. Busca Local

Nesta seção descrevem-se dois movimentos de busca local propostos por Dutra e Montané (2009) para resolver o PTRCF: o *movimento de redução de rotas (MOV-RR)* e o *movimento de realocação de pedidos (MOV-RP)*.

3.2.1 Movimento de redução de rota (MOV-RR)

O movimento de redução de rota (MOV-RR) consiste em reduzir o comprimento de uma rota e inclusive eliminar a rota por inteiro, mediante a realocação da sua carga a outras rotas existentes. A rota que é candidata a redução, ou eliminação, deve ser uma rota redundante com taxa de ocupação baixa. O objetivo deste movimento é aumentar a taxa de ocupação das rotas e reduzir os custos de transporte. A execução do movimento compreende as seguintes etapas: i) escolher o segmento da rota origem, cuja carga será realocada; ii) escolher a rota destino, que possui um segmento idêntico ao escolhido em i) e iii) verificar que a operação de realocação não exceda a capacidade do veículo na rota destino. O movimento com maior redução no custo de transporte é escolhido em cada iteração da busca.

3.2.2 Movimento de realocação de pedidos (MOV-RP)

O Movimento de Realocação de Pedidos (MOV-RP) consiste em realocar vários pedidos de uma rota origem, a uma ou mais rotas de destino. A realocação dos pedidos é realizada de maneira conjunta. Cada pedido P_i é identificado pelo par de depósitos (o_i, d_i) , origem e destino, respectivamente. A realocação de cada pedido compreende as seguintes etapas: i) escolher o pedido P_i que será realocado; ii) escolher a rota destino que deverá conter um segmento $(o_i, \dots, i_k, \dots, d_i)$ com início em o_i , fim em d_i e possíveis intermediários i_k ; e iii) verificar que a operação de realocação não exceda as restrições de capacidade do veículo nem de prazo de entrega. O movimento viável com maior redução de custo é escolhido para execução.

3.3 Heurística VND

A heurística VND (MLADENOVIC e HANSEN, 1997) é um método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. O método utiliza uma vizinhança principal visando melhorar a solução corrente, e quando não for possível melhorá-la o método troca a vizinhança corrente por outra. O procedimento retorna a vizinhança principal quando uma melhor solução é encontrada. O algoritmo termina após utilizar todas as estruturas de vizinhança sem conseguir melhorar a solução corrente.

O algoritmo proposto utiliza dois movimentos para gerar as vizinhanças: *Movimento de Redução de Rota (MOV-RR)* e *Movimento de Realocação de Pedidos (MOV-RP)*, que foram descritos nas seções 3.2.1 e 3.2.2, respectivamente. Os movimentos geram duas estruturas de vizinhança diferentes denotadas como N_1 e N_2 . A probabilidade de uso da vizinhança N_1 é maior do que a vizinhança N_2 , devido à ordem de execução dessas vizinhanças no algoritmo.

3.4. Perturbação

Nesta seção descrevem-se dois movimentos de perturbação: *Movimento de Inserção de Depósitos (MOV-INS)* e *Movimento de Realocação de Pedidos Aleatórios (MOV-RPA)* que visam modificar a solução corrente com o intuito de introduzir certo grau de diversificação no processo de busca. Devido a isso, a execução dos movimentos de perturbação pode piorar a

qualidade da solução corrente no curto prazo. Os movimentos são descritos nas seções 3.4.1 e 3.4.2 respectivamente.

3.4.1 Movimento de perturbação (MOV-INS)

O movimento de inserção de depósitos intermediários (MOV-INS) consiste em inserir um depósito intermediário em uma rota candidata. A escolha da rota candidata é definida de maneira aleatória. Vale observar que o depósito candidato para inserção não é removido de nenhuma outra rota, sendo assim, o custo da solução deve aumentar após a execução do movimento. Esse movimento tem uma função de diversificação da busca. Seu objetivo é aumentar a probabilidade de alocação de pedidos na rota escolhida e aumentar a taxa de ocupação na mesma. A inserção de intermediários compreende as seguintes etapas: i) escolher de maneira aleatória a rota r_i ; ii) escolher a posição de inserção (i_k, i_{k+1}) na rota r_i na qual será realizada a inserção, avalia-se em primeiro lugar o trecho mais próximo de depósito inicial; iii) escolher de maneira aleatória o depósito j a ser inserido na rota r_i , a partir de uma lista de candidatos LC, classificada em ordem crescente de distância ao depósito origem i_1 da rota r_i ; iv) verificar que todos os pedidos que percorrem o segmento (i_k, i_{k+1}) não tenham seu prazo de entrega violado após a inserção. A restrição de capacidade não precisa ser verificada após a inserção.

O exemplo da Figura 4 ilustra o movimento de inserção (MOV-INS). Na Figura 4(a), considere a rota corrente (3-5-6). O movimento testa a inserção de um depósito intermediário no primeiro trecho (3-5), a partir da lista de candidatos (LC) ilustrada na Figura 4(b). O movimento viável com menor incremento no custo de transporte é escolhido para execução.

3.4.2 Movimento de perturbação (MOV-RPA)

O Movimento de Realocação de Pedidos Aleatórios (MOV-RPA) consiste em uma variante do MOV-RP proposto por Dutra e Montané (2009), porém, de maneira aleatória o movimento consiste em escolher uma rota origem e realocar seus pedidos a uma ou mais rotas destino. A realocação dos pedidos é realizada de maneira conjunta. A realocação de cada pedido compreende as seguintes etapas: i) escolher a rota início de maneira aleatória e selecionar seus pedidos P_i para que seja efetuada a realocação; ii) escolher a rota destino que deverá conter um segmento $(o_i - \dots - i_k - \dots - d_i)$ com início em o_i , fim em d_i e possíveis intermediários i_k ; iii) verificar que a operação de realocação não exceda as restrições de capacidade do veículo nem de prazo de entrega. Diferentemente do MOV-RP, este movimento é executado mesmo que não seja possível realocar todos os pedidos, realocando-se nesse caso apenas parte dos pedidos. Nesse caso, ao executar o movimento de maneira aleatória, não é possível garantir que a solução corrente terá melhoria. O objetivo desse movimento é liberar capacidade em algumas rotas, aumentando a probabilidade de melhoria na etapa de busca local.

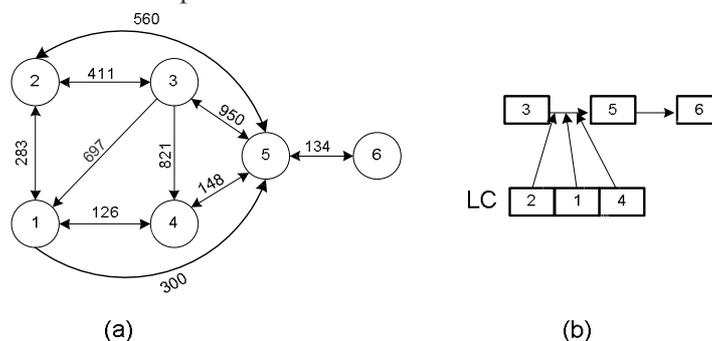


Figura 4. – Movimento de inserção de intermediários (MOV-INS).

3.5. Metaheurística VNS

A Metaheurística de Busca de Vizinhança Variável conhecida como VNS (*Variable Neighborhood Search*) foi proposta por MLADENOVIC e HANSEN (1997). A idéia é uma mudança sistemática de vizinhança dentro de uma busca local. A Metaheurística VNS possui um mecanismo de perturbação na solução corrente, utilizando diferentes estruturas de vizinhança. Esse mecanismo de perturbação pode ser aplicado durante a fase de descida e na fase de exploração permitindo uma piora na qualidade da solução corrente. Após o processo de perturbação da solução é executado um movimento de busca local que busca através do mecanismo de intensificação se aproximar de ótimos locais. Sempre que o mecanismo de intensificação não conseguir melhorar a solução corrente, altera-se o mecanismo de perturbação utilizando outra estrutura de vizinhança.

Procedimento VNS	
início	
1. $s \leftarrow CRSDA ();$	// solução inicial - Construtiva
2. $I_{max} \leftarrow 0; r \leftarrow 2$	// I - número iterações e r - número de busca local
3. enquanto $I_{max} < 50$ faça	// critério de parada
4. $k \leftarrow 1;$	// tipo de estrutura de vizinhança
5. enquanto $(k \leq r)$ faça	
6. $s' \leftarrow$ Gera um vizinho qualquer de $N_k(s);$	// Etapa de diversificação
7. $s'' \leftarrow VND(s');$	// Etapa de busca local
8. se $f(s'') < f(s)$ então	
9. $s \leftarrow s'';$	
10. $k \leftarrow 1;$	
11. senão $k \leftarrow k + 1;$	
12. fim-se;	
13. fim-enquanto;	
14. $I_{max} \leftarrow I_{max} + 1;$	
15. fim-enquanto;	
16. retorne $s;$	
fim VNS-VND;	

Figura 5. – Algoritmo baseado na metaheurística VNS aplicada ao PTRCF.

A Figura 5 mostra o algoritmo VNS proposto para resolver o problema PTRCF. Essa heurística inicia com uma solução s obtida por uma heurística construtiva para o PTRCF (*vide* Figura 5, Passo 1). A solução s é melhorada por um algoritmo VND que realiza um processo de busca local (*vide* Figura 5, Passo 7). É importante observar que a solução s' gerada (*vide* Figura 5, Passo 6) é um vizinho qualquer da vizinhança $N_k(s)$, ou seja, a solução s' não necessariamente é melhor que a solução inicial s . A essa etapa do VNS dar-se o nome de diversificação, com ela é possível evitar redundância de solução, com isso tende-se a aumentar a probabilidade de se escapar de ótimos locais. Faz-se necessário, no entanto, escolher quais as estruturas de vizinhanças que serão utilizadas na etapa de diversificação. Neste trabalho, foram utilizadas duas estruturas de vizinhança N_k , para $k = 1, 2$, que são: MOV-INS, MOV-RPA. O critério de parada do algoritmo VNS é um número máximo de iterações ($I_{max} = 50$).

3.6. Metaheurística ILS

A metaheurística Busca Local Iterativa conhecida como ILS (*Iterated Local Search*), foi proposta por LOURENÇO *et. al.* (2002). O ILS é uma Metaheurística de simples implementação que se baseia em um método de busca local que não procura focar a busca em um espaço completo de soluções, mas em um pequeno subespaço, definido por soluções que são ótimos locais de determinados procedimentos de otimização (LOURENÇO *et. al.*, 2002). A principal idéia do ILS é poder melhorar um procedimento de busca local gerando novas soluções de partida, as quais são obtidas por meio de perturbações na solução ótima local. A perturbação precisa ser suficientemente forte para permitir que a busca local explore diferentes soluções, mas

também, fraca o suficiente para evitar um reinício aleatório (LORENÇO *et al.*, 2002). Segundo LOURENÇO *et al.* (2002) o ILS possui três componentes básicos: Perturbação, Busca Local e Critério de Aceitação.

A Figura 6 apresenta o algoritmo da heurística ILS implementado ao problema PTRCF. A heurística inicia com uma solução s_0 obtida por uma heurística construtiva (*vide* Figura 6, Passo 1). A solução s é melhorada por uma busca local a qual é baseada na heurística VND (*vide* Figura 6, Passo 3). Após a etapa da busca local a solução corrente é perturbada (*vide* Figura 6, Passo 5) e em seguida melhorada pela busca local VND (*vide* Figura 6, Passo 10). É importante observar que a solução s' (*vide* Figura 6, Passo 5) é gerada a partir de uma etapa de diversificação de vizinhança $N_k(s, \text{histórico})$, escolhida de maneira aleatória. A solução s' não necessariamente é melhor que a solução inicial s . Nessa etapa as duas estruturas de vizinhanças utilizadas são: MOV-INS e MOV-RPA e é importante afirmar que na escolha aleatória somente um movimento é executado. Uma diferença nessa etapa de diversificação é o *histórico*, que representa o nível (m) de perturbação que se deve aplicar na solução corrente, onde $m = \{1,2,3\}$, o número de menor valor implica em executar somente 1 movimento de perturbação aleatório, e o número de maior valor implica em executar n vezes o movimento de perturbação aleatório. Cada iteração em que não se melhora a solução, o algoritmo aumenta o nível de perturbação. O critério de parada do algoritmo ILS é um número máximo de iterações ($I = 50$).

```

Procedimento ILS( )
início
1.  $s_0 = \text{CRSDA}()$ ; // Solução inicial – Construtiva
2.  $I \leftarrow 0$ ; // número de iteração
3.  $s \leftarrow \text{VND}(s_0)$ ; // Etapa busca local
4. enquanto ( $I \leq 50$ ) faça
5.  $s' = \text{Gera um vizinho de perturbação } N_k(s, \text{histórico})$ ; //Etapa de diversificação
6.  $s'' \leftarrow \text{VND}(s')$ ; // Etapa busca local
7. se  $f(s'') < f(s)$  então
8.  $s \leftarrow s''$ ;  $\text{histórico} \leftarrow 0$ ;
9. senão  $\text{histórico} \leftarrow \text{histórico} + 1$ ;
15.  $I \leftarrow I + 1$ ;
19. fim-enquanto;
20. retorne  $s$ ;
fim ILS;

```

Figura 6. - Algoritmo baseado na metaheurística ILS aplicada ao PTRCF.

4. Resultados Computacionais

Nesta seção é analisado o comportamento e o desempenho das metaheurísticas VNS e ILS propostas para resolver o problema de transporte rodoviário com carga fracionada PTRCF. Esse algoritmos foram codificados em linguagem Java, usando o compilador NetBeans IDE 6.7.1 e executado em um computador Intel Core 2 Duo, 2.66 GHz, com 4 GB de RAM, 2 GB de Cache e Sistema Operacional Microsoft Windows Vista. Foi gerado um conjunto de dados que compreendem 48 instâncias de teste, considerando-se 4 redes de transporte possíveis, onde varia-se o número de depósitos considerados e 12 conjuntos de demandas para cada rede. As redes consideradas compreendem: 5, 7, 10 e 13 depósitos. A qualidade dos resultados obtidos pelos algoritmos VNS e ILS foi avaliada pela comparação da melhoria alcançada com relação a heurística construtiva CRSDA.

4.1 Comparações dos resultados para o PTRCF

A Tabela 1 apresenta os resultados médios obtidos pela heurística construtiva CRSDA, para 4 grupos de instâncias de dados totalizando 48 instâncias. A descrição dos grupos de instâncias é realizada nas duas primeiras colunas da tabela: Nome: identifica os grupos de instâncias pelo número de depósitos; Tot. Peso: mostra o peso em kg. a ser transportado pelas

rotas. Os resultados obtidos pela heurística construtiva CRSDA são apresentados nas quatro colunas seguintes: N. Veic., número de veículos utilizados; % Ocup. Média, percentual de ocupação médio por veículo; FO. Sol, o valor da função objetivo; T.S., tempo computacional em segundos. Vale observar que, o percentual de ocupação médio é calculado como a média do percentual de ocupação em cada trecho da rota.

Dois algoritmos com base na metaheurística VNS e ILS foram desenvolvidos. Os algoritmos utilizam o algoritmo VND como mecanismo de busca local. Duas vizinhanças foram incorporadas no algoritmo VND. Adotou-se como primeira vizinhança (N_1), aquela gerada pelo movimento MOV-RR. A segunda vizinhança (N_2) é gerada pelo movimento MOV-RP. Ambos os movimentos visam à troca de vizinhança a partir do cálculo do melhor aprimorante. Os algoritmos com base na metaheurística utilizam dois mecanismos de perturbação com base no movimento MOV-INS e MOV-RPA.

Tabela 1. - Resultados do algoritmo CRSDA

Grupo Prob.		CRSDA		
Nome	Tot. Peso	N. Veic.	% Ocup. Média	F.O. Sol
G-5	100536	6	58%	6347
G-7	159761	15	35%	16891
G-10	232086	29	27%	40980
G-13	299181	39	26%	57642
Med. Geral		22	37%	30465

A Tabela 2 apresenta os resultados médios obtidos pelos algoritmos VNS e ILS, para cada grupo de instâncias teste. Para cada algoritmo apresentam-se três colunas de resultados: F.O. sol., o valor da função objetivo; T.M.S., tempo computacional em segundos; e % Mel., percentual de melhoria em relação aos resultados obtidos pela heurística construtiva CRSDA.

Na análise da Tabela 2, pode-se perceber que as metaheurísticas, VNS e ILS, produziram uma melhoria significativa sobre os resultados obtidos pela heurística construtiva CRSDA. Os resultados obtidos pela heurística VNS produziram uma melhoria média de 82%, e os resultados obtidos pela heurística ILS produziram uma melhoria média de 84%. Por outro lado, observa-se também que quanto maior for à instância, maior é a percentagem de melhoria obtida com relação à solução construtiva. Por exemplo, no grupo de problemas G-5, a percentagem de melhoria foi de aproximadamente 7,34%, enquanto que no grupo de problemas G-13, a percentagem de melhoria foi de aproximadamente 132%. Esse fato revela a dificuldade crescente enfrentada pela heurística construtiva no cálculo de soluções iniciais, à medida que o tamanho do problema aumenta.

Tabela 2. Comparação por grupos de instâncias entre os diferentes algoritmos.

Prob.	CRSDA	VNS			ILS		
	F.O. Sol	F.O. Sol	T.M.S.	% Mel.	F.O. Sol	T.M.S.	% Mel.
G-5	6347	5927	29	7%	5927	115	7%
G-7	16891	10854	85	56%	10850	283	56%
G-10	40980	17429	141	136%	17055	668	141%
G-13	57642	25357	228	127%	24756	1101	133%
Med. Geral	30465	14892	121	82%	14647	542	84%

Com relação aos tempos computacionais, observa-se que o tempo requerido pelos diferentes algoritmos foi bastante baixo. Na comparação entre as heurísticas VNS e ILS, observa-se que o tempo médio requerido pela heurística ILS foi bastante superior ao tempo requerido pela heurística VNS. A heurística ILS gastou 542 milissegundos enquanto que a heurística VNS gastou 121 milissegundos.

A Tabela 3 apresenta resultados médios referentes à utilização dos veículos nas soluções obtidas pelas heurísticas CRSDA, VNS e ILS para cada grupo de instância teste. Descreve-se a estrutura da tabela. Para cada algoritmo apresentam-se duas colunas de resultados: a coluna Nº Veículos, mostra o número de veículos utilizados na solução e a coluna % Ocup. Média, mostra o percentual médio de ocupação dos veículos na solução.

Tabela 3. - Utilização dos veículos pelos diferentes algoritmos.

Prob.	CRSDA		VNS		ILS	
	N. Veic.	% Ocup. Média	N. Veic.	% Ocup. Média	N. Veic.	% Ocup. Média
G-5	6	58%	5	61%	5	61%
G-7	15	35%	9	58%	9	60%
G-10	29	27%	13	60%	13	63%
G-13	39	26%	19	56%	18	60%
Med. Geral	22	37%	12	59%	11	61%

Analisando a Tabela 3, pode-se observar que as soluções produzidas pela heurística construtiva CRSDA, em geral, fazem uso de um número elevado de veículos, em média 22 veículos, com taxa de ocupação bastante baixa, 36,63% em média. A taxa de ocupação média aumenta à medida que a qualidade das soluções é melhorada aplicando-se algumas das metaheurísticas propostas, VNS e ILS. Observa-se que, com o aumento da taxa de ocupação média ocorre à redução do número médio de veículos utilizados. A heurística ILS proporciona um aumento da taxa de ocupação, que em média sobe para 61,09% e uma redução do número de veículos utilizados, que em média cai para 11 veículos. Fica evidenciado que quanto menor o espaço ocioso nos veículos, menor será a quantidade de veículos a serem utilizados.

Na Tabela 4 comparam-se entre si os resultados médios obtidos pelas duas metaheurísticas propostas no presente trabalho, VNS e ILS. Analisando a Tabela 4, pode-se observar que a heurística ILS, em média, proporciona resultados que são 1,25% melhores que aqueles produzidos pela heurística VNS. Nos resultados apresentados, observa-se que a heurística ILS conseguiu uma melhoria de até 10,52% em uma determinada instância do grupo G-10 sobre os resultados da heurística VNS. No entanto, deve-se ressaltar que a heurística ILS não conseguiu os melhores resultados em todas as instâncias testes, obtendo um pior resultado de 4,61% em uma determinada instância do grupo G-10 sobre os resultados obtidos pelo algoritmo VNS. Devido a isso, não é possível garantir que a heurística ILS produzirá sempre o melhor resultado.

Tabela 4. - Comparação entre os algoritmos VNS e ILS

Prob.	VNS		ILS		
	F.O. Sol	T.M.S.	F.O. Sol	T.M.S.	% Mel.
G-5	5927	29	5927	115	0,00%
G-7	10854	85	10850	283	0,16%
G-10	17429	141	17055	668	2,31%
G-13	25357	228	24756	1101	2,53%
Med. Geral	14892	121	14647	542	1,25%
Máximo					10,52%
Mínimo					-4,61%

5. Conclusões

O problema abordado no presente artigo é o Problema de Transporte Rodoviário com Carga Fracionada (PTRCF). Metaheurísticas com base nas técnicas VNS e ILS foram desenvolvidas e implementadas gerando soluções de boa qualidade para o Problema PTRCF. As metaheurísticas foram testadas utilizando um total de 48 instâncias teste, considerando-se 4 redes de transporte possíveis, onde varia-se o número de depósitos e 12 conjuntos de demandas

distintos para cada rede. As redes consideradas compreendem: 5, 7, 10 e 13 depósitos. A qualidade dos resultados obtidos pelas metaheurísticas VNS e ILS foi avaliada pela comparação da melhoria alcançada com relação à heurística construtiva CRSDA proposta em Dutra e Montané (2009). Além disso, as duas metaheurísticas foram comparadas entre si.

Com a realização de vários testes, conclui-se que a heurística construtiva CRSDA não produziu resultados de boa qualidade. Observou-se que as heurísticas VNS-VND e ILS foram capazes de introduzir melhorias muito significativas, em média 82% e 84%, respectivamente. As soluções construtivas apresentaram um percentual de ocupação dos veículos muito baixo, em média de 36,63% e um número de veículos utilizados bastante alto, em média 22 veículos.

Com a implementação de ambas as heurísticas, VNS e ILS e com a realização de vários testes, conclui-se que a heurística ILS obteve resultados que foram em média 1,25% melhores do que os resultados encontrados pela heurística VNS. Acredita-se que essa melhoria se dá em função da eficiência da heurística ILS que utiliza um parâmetro histórico, que aumenta o nível de perturbação na etapa de diversificação caso a solução corrente não melhore. Na comparação com os resultados obtidos pela heurística VNS, observou-se que a heurística ILS, conseguiu aumentar o percentual médio de ocupação dos veículos em 2,58% e eliminar em média 1 veículo da frota. Acredita-se que a implementação de uma metaheurística híbrida possa melhorar os resultados produzidos pela heurística ILS. Em geral, o tempo computacional requerido pelas duas metaheurísticas propostas foi bastante baixo.

Referências

- COUTO, P. T. B.**, *Resolução do problema de transporte rodoviário de cargas utilizando programação inteira*. 80f. Dissertação (Mestrado em Engenharia Elétrica) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.
- DIJKSTRA, E. W.**, A note on two problems in connexion with graphs. In *Numerische Mathematik*. Vol. 1. Mathematisch Centrum, Amsterdam, The Netherlands, p.269–271, 1959.
- DUTRA, C. E., MONTANÉ, F. A. T.**, Heurística para o problema de rede de transporte rodoviário com carga fracionada, XVI Simpósio de engenharia de produção, bauru-SP, 2009.
- FREITAS, L. M. B.** *Metaheurísticas para Problemas de Roteamento de Veículo com Coleta e Entrega Simultânea*. 72f. Dissertação (Mestrado em Pesquisa Operacional) - Universidade Candido Mendes – UCAM-Campos, Campos dos Goytacazes, 2008.
- HOARE, C.A.R.**, Algorithm 64: QuickSort, *Communications of the Association for Computing Machinery* 4, p.321, 1961.
- HOARE, C.A.R.**, QuickSort, *Computer Journal* 5, p.10-15, 1962.
- LAWLER, E.L.**, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart & Winston, 1976.
- LOURENCO, H.R.; MARTIN, O. STÜTZLE, T.** Iterated local search, in: GLOVER, F.; KOCHENBERGER G. (Eds.), *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, vol. 57, Kluwer Academic Publishers, Norwell, MA, p. 321–353, 2002.
- MLADENOVIC, N.; HANSEN, N.** Variable neighborhood Search. *Computers and Operations Research*, v. 24, p.1097–1100, 1997.
- VAN DER ZIJPP, N.J.; CATALANO, S. FIORENZO.**, Path enumeration by finding the constrained K-shortest paths, *Transportation Research Part B*, v. 39, p. 545–563, 2004.