

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

ÁREA DE CONCENTRAÇÃO: AUTOMAÇÃO

BUSCA TABU APLICADA AO PROBLEMA DE ROTEAMENTO PERIÓDICO DE VEÍCULOS

Camila Frederico Mortati

Orientador: Prof. Dr. Vinícius Amaral Armentano

Banca Examinadora:

José Vicente Caixeta Filho – USP/Piracicaba

Franklina M. Bragion de Toledo – USP/São Carlos

Akebo Yamakami – FEEC/UNICAMP

Tese de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Campinas, São Paulo
- Junho 2005 -

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M841b Mortati, Camila Frederico
Busca tabu aplicada ao problema de roteamento periódico de veículos / Camila Frederico Mortati. --Campinas, SP: [s.n.], 2005.

Orientador: Vinícius Amaral Armentano
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Otimização combinatória. 2. Heurística. 3. Logística.
I. Armentano, Vinícius Amaral. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: A tabu search algorithm for the periodic vehicle routing problem.

Palavras-chave em Inglês: Combinatorial optimization, Heuristic e Logistics

Área de concentração: Automação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: José Vicente Caixeta Filho, Franklina Maria Bragion de Toledo e Akebo Yamakami.

Data da defesa: 17/06/2005

*Aos meus pais José e Rosa, à minha irmã Amanda e
ao Fábio,
com todo o meu amor.*

AGRADECIMENTOS

Agradeço a todos que colaboraram para a realização deste trabalho. Em especial:

Ao Professor Vinícius A. Armentano pela orientação guiada com sabedoria, pelo incentivo, dedicação, apoio e paciência;

Aos meus pais e à minha irmã, que tanto os amo, pelo constante apoio e motivação. Mesmo distantes fisicamente, em todos os momentos mostraram-se presentes com muito carinho;

Ao Fábio, meu companheiro de longos anos, por seu amor, compreensão e apoio irrestritos;

A Kika, por toda sua demonstração de saudade e de alegria a cada retorno meu para casa, e por quem sinto imenso carinho;

A Alynne, minha irmã de coração;

À Eliane, à Juliana, ao Luiz e ao Elias pela amizade;

A todos os colegas do Densis, pela ajuda e por tantos momentos de alegria;

À Fundação de Amparo à pesquisa do Estado de São Paulo pelo apoio financeiro.

SUMÁRIO

INTRODUÇÃO	1
1. PROBLEMA DE ROTEAMENTO PERIÓDICO DE VEÍCULOS	3
1.1 <i>Introdução</i>	3
1.2 <i>Descrição do Problema</i>	3
1.3 <i>Revisão Bibliográfica</i>	6
2. HEURÍSTICAS CONSTRUTIVAS	11
2.1 <i>Introdução</i>	11
2.2 <i>Heurística de Cordeau et al. (2001)</i>	13
2.3 <i>Variações das Heurísticas de Beltrami e Bodin (1974) e Cordeau et al. (2001)</i>	13
3. BUSCA LOCAL	16
3.1 <i>Introdução</i>	16
3.2 <i>Busca Local baseada em Cordeau et al. (2001)</i>	17
3.3 <i>Busca Local baseada em Løkketangen e Glover (1998)</i>	19
4. BUSCA TABU	24
4.1 <i>Introdução</i>	24
4.2 <i>Busca Tabu Proposta</i>	24
4.2.1 Critério de Aspiração e Condição de Parada.....	24
4.2.2 Memória de Curto Prazo	24
4.2.3 Movimentos de melhoria.....	26
4.2.4 Memória de Longo Prazo.....	27
4.3 <i>Busca Tabu de Cordeau et al. (2001)</i>	34
4.3.1 Descrição do Algoritmo de Busca Tabu	35
5. TESTES COMPUTACIONAIS	39
5.1 <i>Introdução</i>	39
5.2 <i>Descrição das instâncias testadas</i>	39
5.3 <i>Testes e Considerações sobre as Heurísticas Construtivas</i>	42
5.4 <i>Testes e Considerações sobre os Algoritmos de Busca Local</i>	43
5.5 <i>Testes e Considerações Referentes aos Algoritmos de Busca Tabu</i>	44
5.5.1 Testes referentes ao algoritmo de busca tabu implementado.....	45
5.5.2 Testes referentes ao algoritmo de busca tabu de Cordeau et. al. (2001).....	55
5.5.3 Comparação entre Algoritmos para o PRPV.....	58

6. CONCLUSÕES.....62
REFERÊNCIAS BIBLIOGRÁFICAS.....64

LISTA DE FIGURAS

<i>FIGURA 1 – PROGRAMAÇÃO DAS VISITAS AOS CLIENTES</i>	4
<i>FIGURA 2 – DETERMINAÇÃO DAS ROTAS EM CADA DIA DO HORIZONTE DE TEMPO</i>	5
<i>FIGURA 3 – CLARKE E WRIGHT (1964)</i>	11
<i>FIGURA 4 – HEURÍSTICA CONSTRUTIVA DE CLARKE E WRIGHT ADAPTADA PARA O PRPV</i>	12
<i>FIGURA 5 – HEURÍSTICA CONSTRUTIVA DE CORDEAU ET AL. (2001)</i>	13
<i>FIGURA 6 – HEURÍSTICA CONSTRUTIVA MODIFICADA, BASEADA EM CORDEAU ET AL. (2001)</i>	14
<i>FIGURA 7 – PARÂMETROS E VARIÁVEIS DE DECISÃO</i>	14
<i>FIGURA 8 – PROGRAMA INTEIRO PARA DETERMINAR UMA SOLUÇÃO INICIAL PARA O PRPV</i>	15
<i>FIGURA 9 – BUSCA LOCAL PENALIZADA UTILIZANDO A VIZINHANÇA N_2 E MOVIMENTOS DE MELHORIA INTRA E INTER-ROTAS</i>	19
<i>FIGURA 10 – CLASSIFICAÇÃO DOS TIPOS DE MOVIMENTOS</i>	20
<i>FIGURA 11 – REPRESENTAÇÃO DOS TIPOS DE MOVIMENTOS</i>	21
<i>FIGURA 12 – BUSCA LOCAL BASEADA EM REGRAS DE AVALIAÇÃO DE MOVIMENTOS DE LØKKETANGEN E GLOVER E MOVIMENTOS DE MELHORIA INTRA E INTER ROTAS</i>	23
<i>FIGURA 13 – BUSCA TABU</i>	26
<i>FIGURA 14 – BUSCA TABU COM MOVIMENTOS DE MELHORIA INTRA E INTER ROTAS</i>	26
<i>FIGURA 15 – ESTRATÉGIA DE DIVERSIFICAÇÃO 1</i>	28
<i>FIGURA 16 – ESTRATÉGIA DE DIVERSIFICAÇÃO 2</i>	29
<i>FIGURA 17 – PROCEDIMENTO DE CONSTRUÇÃO DE UMA NOVA SOLUÇÃO BASEADA NA ESTRATÉGIA DE ROCHAT E TAILLARD</i>	31
<i>FIGURA 18 – MECANISMO DE RELIGAÇÃO DE CAMINHO</i>	34
<i>FIGURA 19 – DESCRIÇÃO DO ALGORITMO DE BUSCA TABU DE CORDEAU ET AL. (2001)</i>	38

LISTA DE TABELAS

<i>TABELA 1 – INSTÂNCIAS COM RESTRIÇÕES DE DURAÇÃO E DE CAPACIDADE</i>	40
<i>TABELA 2 - INSTÂNCIAS SOMENTE COM RESTRIÇÃO DE CAPACIDADE</i>	41
<i>TABELA 3 – RESUMO DOS RESULTADOS DOS TESTES REALIZADOS PARA AS HEURÍSTICAS CONSTRUTIVAS IMPLEMENTADAS</i>	42
<i>TABELA 4 – RESUMO DOS TESTES REALIZADOS PARA OS ALGORITMOS DE BUSCA LOCAL BASEADOS EM (CORDEAU ET AL.2001)</i>	44
<i>TABELA 5 – RESUMO DOS TESTES REALIZADOS PARA O ALGORITMO DE BUSCA LOCAL BASEADO EM (LØKKETANGEN E GLOVER) COM MOVIMENTOS DE MELHORIA OR-OPT INTRA E INTER ROTAS</i>	44
<i>TABELA 6 – RESUMO DOS TESTES REALIZADOS COM O ALGORITMO DE BUSCA TABU UTILIZANDO MEMÓRIA DE CURTO PRAZO BASEADA NA ESTRUTURA DA LISTA TABU 1</i>	45
<i>TABELA 7 – RESUMO DOS TESTES REALIZADOS COM O ALGORITMO DE BUSCA TABU UTILIZANDO MEMÓRIA DE CURTO PRAZO BASEADA NA ESTRUTURA DA LISTA TABU 2</i>	46
<i>TABELA 8 - RESULTADOS REFERENTES A BUSCA TABU COM DIVERSIFICAÇÃO 1 PARA INSTÂNCIAS COM RESTRIÇÕES DE CAPACIDADE E DE DURAÇÃO</i>	47
<i>TABELA 9 - RESULTADOS REFERENTES A BUSCA TABU COM DIVERSIFICAÇÃO 1 PARA INSTÂNCIAS COM RESTRIÇÃO DE CAPACIDADE</i>	48
<i>TABELA 10 - RESULTADOS REFERENTES A BUSCA TABU COM DIVERSIFICAÇÃO 2 PARA INSTÂNCIAS COM RESTRIÇÕES DE CAPACIDADE E DE DURAÇÃO</i>	49
<i>TABELA 11 - RESULTADOS REFERENTES A BUSCA TABU COM DIVERSIFICAÇÃO 2 PARA INSTÂNCIAS COM RESTRIÇÃO DE CAPACIDADE</i>	50
<i>TABELA 12 - RESULTADOS REFERENTES A BUSCA TABU COM RELIGAÇÃO DE CAMINHO PARA INSTÂNCIAS COM RESTRIÇÕES DE CAPACIDADE E DE DURAÇÃO</i>	52
<i>TABELA 13 - RESULTADOS REFERENTES A BUSCA TABU COM RELIGAÇÃO DE CAMINHO PARA INSTÂNCIAS COM RESTRIÇÃO DE CAPACIDADE</i>	53
<i>TABELA 14 – RESUMO DOS RESULTADOS OBTIDOS ATRAVÉS DO ALGORITMO BT PARA AS QUATRO REGRAS DE AVALIAÇÃO DE MOVIMENTOS</i>	55
<i>TABELA 15 – RESULTADOS DOS ALGORITMOS DE CORDEAU ET. AL. (1997), BT E DE CORDEAU ET. AL.(2001) PARA AS INSTÂNCIAS COM RESTRIÇÕES DE CAPACIDADE E DE DURAÇÃO</i>	56
<i>TABELA 16 – RESULTADOS DOS ALGORITMOS DE CORDEAU ET. AL. (1997), BT E DE CORDEAU ET. AL.(2001) PARA AS INSTÂNCIAS SOMENTE COM RESTRIÇÃO DE CAPACIDADE</i>	57
<i>TABELA 17 –COMPARAÇÃO ENTRE ALGORITMOS PARA O PRPV PARA AS INSTÂNCIAS COM RESTRIÇÕES DE CAPACIDADE E DE DURAÇÃO</i>	59
<i>TABELA 18 – COMPARAÇÃO ENTRE ALGORITMOS PARA O PRPV PARA AS INSTÂNCIAS COM RESTRIÇÃO DE CAPACIDADE</i>	60

RESUMO

Este trabalho aborda o problema de roteamento periódico de veículos, que consiste em designar uma combinação de dias de visitas a cada cliente, e definir as rotas de veículos em cada dia de um horizonte de planejamento, de forma a minimizar o custo ou a duração total das rotas. Um algoritmo de busca tabu é proposto para a resolução do problema. A história da busca tabu, usada para guiar o processo de busca, é representada através de memórias de curto e longo prazo. A eficiência das estratégias sugeridas para diversificação e intensificação, associadas à memória de longo prazo, são verificadas experimentalmente. O desempenho do algoritmo de busca tabu é testado computacionalmente em problemas da literatura. Um procedimento de busca tabu proposto na literatura é implementado e comparado com o algoritmo aqui proposto.

Palavras-Chave: Roteamento Periódico de Veículos, Busca Tabu, Heurísticas, Otimização Combinatória

ABSTRACT

This work addresses the periodic vehicle routing problem that consists of assigning a combination of visiting days to each client, and defining the routes every day of a planning horizon, in such a way as to minimize the cost or duration of the routes. A tabu search algorithm is proposed for solving this problem. The history of the tabu search, used to guide the search process, is represented by short and long term memories. The efficacy of the suggested strategies for diversification and intensification, associated to the long term memory, is verified experimentally. The performance of the tabu search algorithm is tested computationally in instances from the literature. A tabu search procedure suggested in the literature is implemented and its performance is tested against the tabu search algorithm developed in this work.

Keywords: Periodic Vehicle Routing, Tabu Search, Heuristics, Combinatorial Optimization

INTRODUÇÃO

O problema de roteamento periódico de veículos (PRPV) é uma extensão do problema clássico de roteamento de veículos, em que um conjunto de clientes deve ser visitado em um ou mais dias de um dado horizonte de tempo (planejamento). Combinações de possíveis dias de visitas são associadas a cada cliente. Uma frota de veículos é disponibilizada em cada dia, e cada veículo parte do depósito, visita os clientes pertencentes à rota que deve percorrer e ao final da rota retorna ao depósito. O problema consiste em associar uma combinação de dias de visitas a cada cliente e, para cada dia do horizonte de tempo, definir as rotas dos veículos de tal forma a visitar os clientes alocados para cada dia de forma a minimizar o custo total das rotas percorridas pelos veículos ao longo do horizonte de tempo, sujeito a restrições operacionais.

O PRPV ocorre nas áreas de coleta de lixo (Beltrami e Bodin, 1974; Chang et al., 1997; Hokkanen e Salminen, 1997; Shi e Lin, 1999; Cunha e Caixeta Filho, 2002; Angelelli e Speranza, 2002; Baptista et al., 2002), entrega de roupas em hospital (Brodeur et al., 1998), distribuição (Golden e Wasil, 1987; Carter et al., 1996) e manutenção (Blakeley et al., 2003).

Estes problemas pertencem à área de otimização combinatória e são na grande maioria intratáveis em situações reais, no sentido de que não existem algoritmos exatos que forneçam uma solução ótima em tempo computacional viável. Por este motivo, recorre-se a métodos heurísticos de resolução. A metaheurística busca tabu tem sido usada com sucesso em uma grande variedade de problemas de otimização combinatória (Glover e Laguna, 1997), incluindo problemas de roteamento de veículos (Laporte et al., 2000; Cordeau et al., 2002). No entanto, a literatura sobre implementações de busca tabu e outras metaheurísticas, tais como algoritmos genéticos e *simulated annealing*, para o problema de roteamento periódico de veículos é muito escassa.

A relevância do problema de roteamento periódico de veículos e a potencialidade da busca tabu são a motivação deste trabalho, que tem como objetivo desenvolver e implementar algoritmos da metaheurística busca tabu, e comparar os resultados obtidos com os resultados apresentados na literatura.

Para descrever o problema com mais detalhes, assim como apresentar os métodos de resolução propostos, este trabalho está dividido em seis capítulos apresentados sucintamente a seguir.

O primeiro capítulo apresenta a descrição do problema de roteamento periódico de veículos e a revisão bibliográfica da literatura relacionada. O segundo capítulo contém a descrição das heurísticas construtivas apresentadas na literatura assim com as variações

propostas. O terceiro capítulo contém a descrição da busca local para o PRPV apresentada na literatura e as variações propostas. O quarto capítulo apresenta a descrição e evolução da implementação do algoritmo de busca tabu proposto e o algoritmo de busca tabu descrito por Cordeau et *al.* (2001). O quinto capítulo apresenta todos os resultados obtidos nos testes das heurísticas construtivas, das buscas locais e dos algoritmos de busca tabu. Por fim, o sexto capítulo apresenta as conclusões.

1. PROBLEMA DE ROTEAMENTO PERIÓDICO DE VEÍCULOS

1.1 Introdução

Problemas de roteamento de veículos e métodos exatos e heurísticos de otimização para a resolução desses problemas são descritos em (Toth e Vigo, 2002). Estes problemas são importantes pois, segundo estes autores, diversas aplicações reais nos Estados Unidos e Europa mostram que o uso de ferramentas computacionais leva a uma redução de 5% a 20% do custo de transporte, que por sua vez corresponde de 10% a 20% do custo total de produtos. Um desses problemas é o problema de roteamento periódico de veículos (PRPV), que ocorre nas áreas de coleta de lixo (Beltrami e Bodin, 1974; Chang et al., 1997; Hokkanen e Salminen, 1997; Shi e Lin, 1999; Cunha e Caixeta Filho, 2002; Angelelli e Speranza, 2002; Baptista et al., 2002), entrega de roupas em hospital (Brodeur et al., 1998), distribuição (Golden e Wasil, 1987; Carter et al., 1996) e manutenção (Blakeley et al., 2003).

O problema de roteamento periódico de veículos (PRPV) é uma extensão do problema clássico de roteamento de veículos (PRV), em que um conjunto de clientes deve ser visitado em um ou mais dias de um dado horizonte de T dias. Combinações de possíveis dias de visitas são associadas a cada cliente, isto é, cada cliente i especifica uma freqüência de visitas f_i e um conjunto C_i de combinações de dias de visitas. Por exemplo, se $T=5$, $f_i=2$ e $C_i = \{\{1,3\}, \{2,4\}, \{3,5\}\}$, então o cliente i deve ser visitado duas vezes, e as visitas devem ocorrer nos dias 1 e 3, ou nos dias 2 e 4, ou nos dias 3 e 5. Uma frota de veículos é disponibilizada em cada dia, e cada veículo parte do depósito, visita os clientes pertencentes à rota que deve percorrer e ao final da rota retorna ao depósito. O problema consiste em programar as visitas aos clientes e em estabelecer as rotas dos veículos em cada dia do horizonte de tempo, de forma a minimizar a duração total das rotas, satisfazendo restrições operacionais. Neste capítulo, apresenta-se uma descrição detalhada do problema de roteamento periódico de veículos, objeto de estudo neste trabalho, e uma visão geral da literatura associada.

1.2 Descrição do Problema

O Problema de Roteamento Periódico de Veículos (PRPV) pode ser definido em um multigrafo $G=(V,A)$, onde $V=\{v_0, v_1, \dots, v_n\}$ representa o conjunto de nós e $A=\left\{\left(v_i, v_j\right)^{k,l} : v_i, v_j \in V, i \neq j\right\}$ o conjunto de arcos, em que k e l referem-se ao número do veículo

e ao dia de visita, respectivamente. O nó v_0 representa um depósito no qual situam-se m veículos de capacidade Q_k , e os demais nós de V representam clientes a serem servidos. Para um horizonte de T dias e um cliente i , é dada a demanda q_i de i , o tempo de serviço d_i de i , a frequência f_i de visitas a i e a combinação C_i de dias permitidos para visitas. A cada arco $(v_i, v_j)^{k,l}$ é associado um custo não negativo ou tempo de viagem c_{ijkl} . O PRPV consiste em programar as visitas aos clientes e em determinar as rotas dos veículos em cada dia do horizonte de tempo de forma que:

- i) cada rota inicia e termina no depósito;
- ii) cada cliente no dia t pertence somente a uma rota;
- iii) a demanda total de uma rota não excede a capacidade do veículo;
- iv) o tempo total de uma rota não excede um turno (duração) D ;
- v) o custo total das rotas ao longo de T é minimizado.

As Figuras 1 e 2 exemplificam um problema com $n=10$ clientes, em um horizonte de $T=5$ dias, sendo disponibilizados $k=2$ veículos em cada dia do horizonte. A frequência de visitas de cada cliente é $f_1 = f_2 = 5$, $f_3 = f_4 = f_5 = 1$ e $f_6 = f_7 = f_8 = f_9 = f_{10} = 2$. As combinações de dias de visitas de cada cliente são $C_1 = C_2 = \{\{1,2,3,4,5\}\}$, $C_3 = C_4 = C_5 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$, $C_6 = C_8 = C_{10} = \{\{1,3\}, \{2,4\}\}$ e $C_7 = C_9 = \{\{1,4\}, \{3,5\}\}$. A Figura 1 refere-se à programação das visitas aos clientes, apresentando para cada dia do horizonte os clientes que nele devem ser visitados, e a Figura 2 apresenta as rotas determinadas em cada dia do horizonte.

Programação	
Dia 1:	1; 2; 3; 8; 9
Dia 2:	1; 2; 6; 10
Dia 3:	1; 2; 4; 8; 7
Dia 4:	1; 2; 6; 10; 9
Dia 5:	1; 2; 5; 7

Figura 1 – Programação das visitas aos clientes

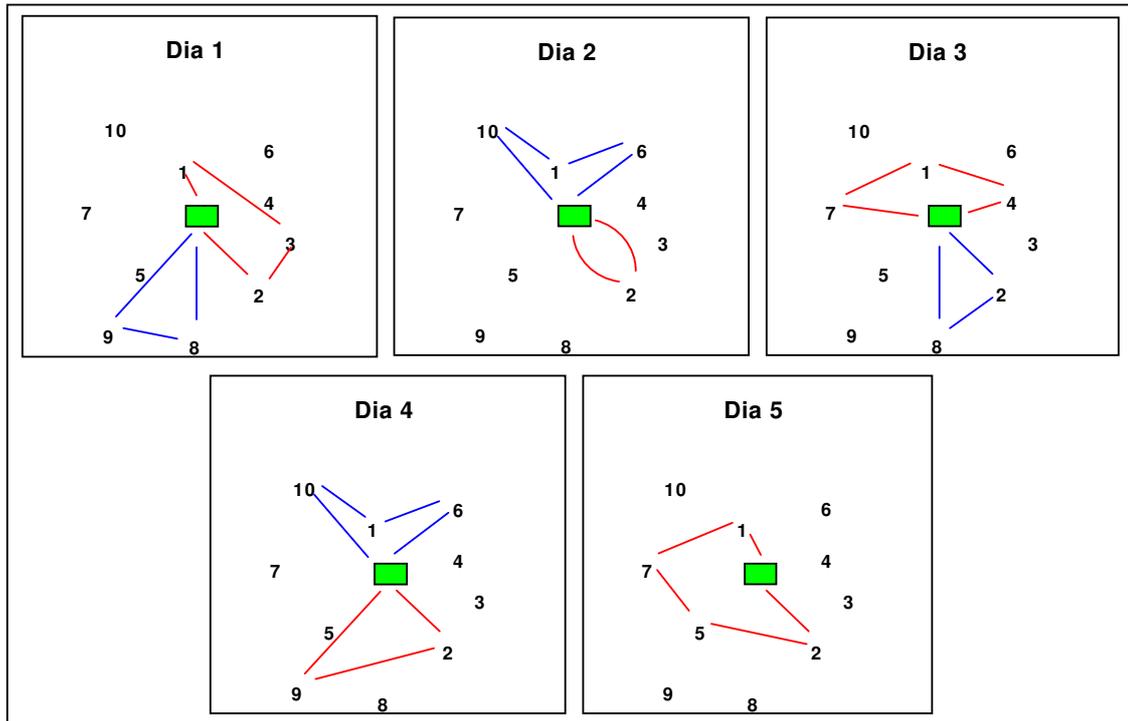


Figura 2 – Determinação das rotas em cada dia do horizonte de tempo

As variáveis do problema são:

$$a_{rl} = \begin{cases} 1, & \text{se e somente se o dia } l \text{ pertence a combinação de visitas } r \\ 0, & \text{caso contrário} \end{cases}$$

$$x_{ijkl} = \begin{cases} 1, & \text{se e somente se o veículo } k \text{ visita o cliente } j \text{ imediatamente após o cliente } i \text{ no dia } l (i \neq j) \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ir} = \begin{cases} 1, & \text{se e somente se a combinação de dias de visitas } r \in C_i \text{ é designada ao cliente } i \\ 0, & \text{caso contrário} \end{cases}$$

A formulação matemática que representa o problema é apresentada a seguir, e assume que $d_0 = 0$ e $q_0 = 0$.

$$\text{Minimizar } \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m \sum_{l=1}^t c_{ijkl} x_{ijkl} \quad (1)$$

sujeito a:

$$\sum_{r \in C_i} y_{ir} = 1 \quad (i = 1, \dots, n) \quad (2)$$

$$\sum_{j=0}^n \sum_{k=1}^m x_{ijkl} - \sum_{r \in C_i} a_{rl} y_{ir} = 0 \quad (i = 1, \dots, n; l = 1, \dots, t) \quad (3)$$

$$\sum_{i=0}^n x_{ihkl} - \sum_{j=0}^n x_{hjkl} = 0 \quad (h = 0, \dots, n; k = 1, \dots, m; l = 1, \dots, t) \quad (4)$$

$$\sum_{j=1}^n x_{0jkl} \leq 1 \quad (k = 1, \dots, m; l = 1, \dots, t) \quad (5)$$

$$\sum_{i=0}^n \sum_{j=0}^n q_i x_{ijkl} \leq Q_k \quad (k = 1, \dots, m; l = 1, \dots, t) \quad (6)$$

$$\sum_{i=0}^n \sum_{j=0}^n (c_{ijkl} + d_i) x_{ijkl} \leq D \quad (k = 1, \dots, m; l = 1, \dots, t) \quad (7)$$

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ijkl} \leq |S| - 1 \quad (k = 1, \dots, m; l = 1, \dots, t; S \subseteq V \setminus \{0\}; |S| \geq 2) \quad (8)$$

$$x_{ijkl} \in \{0, 1\} \quad (i = 0, \dots, n; j = 0, \dots, n; k = 1, \dots, m; l = 1, \dots, t) \quad (9)$$

$$y_{ir} \in \{0, 1\} \quad (i = 1, \dots, n; r \in C_i) \quad (10)$$

A restrição (2) assegura que uma combinação de dias de visitas factível deve ser associada a cada cliente e a (3) que cada cliente é visitado somente nos dias correspondentes à combinação de dias de visitas a ele associada. A restrição (4) garante que um veículo chega e sai de um cliente no mesmo dia e a (5) que cada veículo é usado no máximo uma vez por dia. As restrições (6) e (7) garantem que a capacidade e a duração das rotas não sejam ultrapassadas. A restrição (8) garante a eliminação de sub-rotas.

1.3 Revisão Bibliográfica

De nosso conhecimento não existem algoritmos ótimos para o PRPV propostos na literatura. As heurísticas propostas, de modo geral, partem de uma solução inicial e tentam melhorar esta solução.

Beltrami e Bodin (1974) relatam uma das primeiras aplicações do PRPV na coleta de lixo de Nova Iorque. Russel e Igo (1979) também abordaram problemas de grande porte de coleta de lixo. Em ambos trabalhos, as heurísticas propostas são baseadas no algoritmo de Clarke e Wright (1964) e movimentos k -opt (Lin e Kernighan, 1973).

Chistofides e Beasley (1984) propõem uma heurística composta de duas fases. Na primeira fase, busca-se uma combinação inicial de dias de visitas para cada cliente e a seguir as rotas em cada dia são otimizadas. A segunda fase consiste de trocas de combinação de dias de visitas para um número restrito de clientes. Como cada troca envolve a otimização de novos

PRVs para os dias afetados (computacionalmente caro), os autores substituem o novo problema de roteamento por dois tipos de problemas relaxados. O primeiro consiste em minimizar a soma das distâncias dos clientes a um conjunto de centros, definidos como um certo número de clientes bem espalhados. O segundo consiste em resolver um problema dos m -problemas de caixeiro viajante para as m rotas do dia.

Tan e Beasley (1984) propõem uma heurística composta de três fases. Na primeira fase, geram-se sementes correspondentes a um conjunto de clientes bem espalhados. Na segunda fase, resolve-se um problema de programação linear para designar uma combinação de dias de visitas a cada cliente de forma a minimizar a soma das distâncias das sementes aos clientes. A última fase consiste da resolução de um PRV para cada dia através do algoritmo de Fisher e Jaikumar (1981).

Russel e Gribbin (1991) sugerem uma heurística com quatro fases. A primeira fase é semelhante às fases 1 e 2 dos algoritmos de Tan e Beasley (1984). A terceira fase é idêntica ao procedimento que envolve trocas de combinações de dias de visitas de Christofides e Beasley (1984) com avaliação das trocas através do m -caixeiro viajante. Uma variante adotada pelos autores nesta fase consiste em re-otimizar as m rotas do problema real em cada dia, não utilizando, portanto, o m -caixeiro viajante. A quarta fase envolve um problema de programação binária para alterar a combinação de visitas de cada cliente de forma a maximizar a soma dos ganhos de distância (duração) decorrentes da alteração.

Chao et al. (1995) propõem uma heurística com três fases. Na primeira fase buscam alocar clientes a combinações de dias de visitas através de um problema de programação linear que procura balancear a demanda a ser atendida em cada dia. A segunda fase consiste de movimentos de inserção de clientes em outras rotas. Neste movimento, a combinação de dias de visitas associada ao cliente pode mudar ou não. A terceira fase envolve uma re-otimização de cada rota através da inserção de cada cliente em todas as posições da rota. Para as rotas onde não ocorre melhoria utiliza-se o movimento 2-opt. Quando a razão entre a demanda total ao longo do horizonte dividida pela capacidade total dos veículos é superior a 90%, então a capacidade dos veículos é aumentada artificialmente de 10%. Neste caso, a solução resultante da fase 3 pode ser infactível, e um procedimento de factibilização é sugerido. Os autores sugerem também duas formas de reinicializar a segunda fase, na tentativa de obter soluções melhores. A heurística foi comparada com as heurísticas propostas por Christofides e Beasley, Tan e Beasley, e Russel e Gribbin em 13 instâncias geradas por Christofides e Beasley e Russel e Gribbin, envolvendo tamanhos de 50 a 417 clientes e horizonte de planejamento de 2 a 10 dias. A heurística de Chao

et al.(1995) mostrou-se superior em todas as 13 instâncias. Além disso, foram geradas 19 novas instâncias envolvendo tamanhos de 20 a 184 clientes e horizonte de planejamento de 4 e 6 dias. Os resultados foram comparados com soluções ótimas estimadas.

Em todas as heurísticas acima, assume-se um limite superior de veículos para cada dia. Gaudioso e Palleta (1992) propõem uma heurística para minimizar o número de veículos, baseada em procedimento de inserção de clientes em rotas e um algoritmo de bin-packing para minimizar o número de rotas.

A primeira metaheurística proposta para o PRPV é uma busca tabu (Cordeau et al., 1997). Para a solução inicial, assume-se que os clientes são representados por coordenadas Euclidianas, e que estão ordenados em ordem de ângulo crescente com o depósito e um raio arbitrário. A seguir, escolhe-se aleatoriamente uma combinação de dias de visitas para cada cliente. Para cada dia do horizonte de tempo, cada cliente que deve ser visitado neste dia é inserido em uma rota através da heurística GENI (Gendreau, 1992). Se a inserção de um cliente em quaisquer das rotas de um determinado dia de sua combinação de dias de visitas for inactível, este deve ser incluído na m -ésima rota deste dia. Ao final deste procedimento, somente a rota m pode resultar inactível.

Durante a busca tabu, uma solução s é avaliada pela função de custo

$$f(s) = c(s) + \alpha q(s) + \beta d(s) \quad (1)$$

tal que, α e β são parâmetros positivos, $q(s)$ é a soma das violações da capacidade dos veículos e $d(s)$ é a soma das violações das durações das rotas percorridas pelos veículos. Os parâmetros α e β são ajustados dinamicamente para facilitar a exploração do espaço de busca.

Para cada solução s associa-se um atributo $B(s) = \{(i, k, t): \text{cliente } i \text{ visitado pelo veículo } k \text{ no dia } t\}$. Assim, a vizinhança de uma dada solução s é obtida pela execução dos seguintes movimentos:

1. Remover o cliente i da rota k no dia t e inseri-lo em outra rota k' do mesmo dia
2. Substituir a combinação r do cliente i por outra combinação $r' \in C_i$

A regra de proibição é a seguinte: o cliente i é removido da rota k no dia t e a re-inserção nesta rota é proibida (tabu) por θ iterações. O critério de aspiração é o clássico: execute um movimento tabu se a solução obtida for factível e superior à melhor solução encontrada até o momento.

Para diversificar a busca, qualquer solução \bar{s} tal que $f(\bar{s}) \geq f(s)$ é penalizada pela soma

da frequência de seus atributos, isto é, seja ρ_{ik} o número de vezes que o atributo (i, k) foi adicionado à solução durante a busca. Uma penalidade

$$p(\bar{s}) = \lambda c(\bar{s}) + \sqrt{nm} \sum_{(i,k) \in B(\bar{s})} \rho_{ik}$$

é adicionada a $f(\bar{s})$, e o parâmetro λ controla a intensidade da diversificação.

Para ajustar os parâmetros do algoritmo os autores geraram 10 novas instâncias e através destes foi determinado o critério de parada em 15.000 iterações. O algoritmo de busca tabu é testado nas 32 (13 + 19) instâncias acima mencionadas e seus resultados são comparados com os resultados gerados através da heurística de Chao et al. (1995) A busca tabu foi superior em 24 e igual em 6 instâncias.

A segunda metaheurística para o PRPV é um algoritmo genético paralelo proposto por Drummond et al. (2001). A solução é representada por um vetor, tal que a componente i representa a combinação de dias de visitas associada ao cliente i . As operações de *crossover* e mutação são clássicas. A população do algoritmo genético é particionada em sub-populações que evoluem independentemente. No entanto, a seguinte estratégia interessante de migração de elementos entre sub-populações é usada. Quando a taxa de renovação dos elementos em uma sub-população cai para abaixo de 5%, então as outras sub-populações enviam suas melhores soluções para aquela sub-população. Testes computacionais são apresentados para 20 instâncias da literatura e os resultados mostram que o algoritmo proposto é competitivo com a busca tabu de Cordeau et al. (1997).

Cordeau et al. (2001) propõem um algoritmo de busca tabu, baseado no algoritmo proposto por Cordeau et al. (1997), para os problemas de roteamento de veículos com janelas de tempo, de roteamento periódico de veículos com janelas de tempo e de roteamento de veículos com janelas de tempo com múltiplos depósitos. Neste algoritmo a solução inicial é obtida assim como em Cordeau et al. (1997), porém ao invés de usar a heurística GENI como mecanismo de inserção de clientes em rotas, faz-se uso do mecanismo de inserção de menor custo.

Angelelli e Speranza (2002) estendem o problema de roteamento periódico de veículos, considerando um conjunto de facilidades intermediárias para servir como meio de carga e descarga. Desta forma, um veículo só retorna ao depósito quando os clientes a ele associados tiverem sido visitados, podendo ou não ter passado pelas facilidades intermediárias. Neste estudo foi utilizado o algoritmo de busca tabu de Cordeau et al. (2001) como base de desenvolvimento do algoritmo de busca tabu proposto.

Recentemente, Alegre et al. (2004) propuseram um algoritmo de *scatter search* como

método de resolução de um caso específico do PRPV. O algoritmo consiste de dois laços, sendo o primeiro responsável por controlar o número de vezes que o conjunto de soluções de referência é reconstruído. Primeiramente, são geradas soluções através de um método de diversificação, e subseqüentemente cada uma dessas soluções é submetida a um método de busca em vizinhança. O conjunto de soluções de referencia é (re)construído a partir destas soluções melhoradas. O laço interno controla a geração de novas soluções até que pelo menos uma destas seja admitida no conjunto de soluções de referencia, isto é, uma nova solução é admitida no conjunto de soluções de referencia se ela for melhor, em termos de valor de função objetivo, que a pior solução de referencia corrente. Neste laço, primeiramente um método de combinação é aplicado a todos os pares de soluções de referencia e novas soluções são geradas, e subseqüentemente estas novas soluções são melhoradas pelo método de busca em vizinhança. Por fim, o conjunto de soluções de referencia é atualizado. Os métodos de diversificação, melhoramento e combinação são os mesmos desenvolvidos por Delgado et *al.*(2005). Os resultados mostram que o método proposto é competitivo quando comparado com métodos da literatura.

2. HEURÍSTICAS CONSTRUTIVAS

2.1 Introdução

Heurísticas construtivas geram uma solução através da adição de componentes individuais (ex: nós, arcos, variáveis) um de cada vez até que uma solução seja obtida.

Heurísticas construtivas para a geração de soluções iniciais para o PRPV foram propostas por Beltrami e Bodin (1974), Christofiedes e Beasley (1984), Russel e Gribbin (1991), Chao et al. (1995), Cordeau et al. (1997), e de modo geral, quanto melhor a solução, maior o tempo computacional gasto para obtê-la.

Os seguintes passos são comuns nas heurísticas construtivas acima citadas.

- i) Ordenar os clientes segundo algum critério;
- ii) Associar a cada cliente uma combinação de dias de visitas factível;
- iii) Para cada dia da combinação de visitas associada a cada cliente, inseri-lo em uma rota.

O objetivo foi implementar heurísticas construtivas que gerassem boas soluções em um pequeno tempo computacional. Porém, assim como a maioria das heurísticas apresentadas na literatura, a obtenção de uma solução inicial factível não é garantida. As heurísticas construtivas implementadas seguiram basicamente duas heurísticas sugeridas por Beltrami e Bodin (1974) e Cordeau et al. (2001).

Clarke e Wright (1964) desenvolveram duas heurísticas construtivas para o PRV, uma paralela e outra seqüencial, baseadas no conceito de economias (*savings*). A versão paralela, em geral, gera soluções de melhor qualidade e o algoritmo correspondente é apresentado na Figura 3.

Criar n rotas, onde cada rota inicia e termina no depósito e é composta por um único cliente;

Para cada par de cidades **faça**

 Calcular s_{ij} ;

$s_{ij} = t_{0i} + t_{0j} - t_{ij}$, onde t_{ij} é o tempo de viagem gasto entre as cidades i e j

Se $s_{ij} > 0$ **então** Armazenar s_{ij} em uma lista de *savings*;

Fim para

Ordenar a lista de *savings* decrescentemente;

Percorrendo a lista de *savings*, ligar as cidades i e j , desde que a rota resultante respeite as restrições de capacidade e de duração das rotas.

Figura 3 – Clarke e Wright (1964).

A adaptação da versão paralela da heurística de Clarke e Wright (1964) para o PRPV foi proposta primeiramente por Beltrami e Bodin (1974), e exige que duas cidades só sejam ligadas

se ambas tiverem o dia de visita em comum. O algoritmo correspondente é apresentado na Figura 4.

```
Associar a cada cliente uma combinação de dias de vistas factível, determinada aleatoriamente;
Para cada dia  $l$  do período faça
  Para cada cliente  $i$  faça
    Se o cliente  $i$  deve ser visitado no dia  $l$  então
      Criar uma rota que contenha somente o cliente  $i$  e que inicie e termine no depósito;
    Fim se
  Fim para
Fim para
Para cada par de cidades faça
  Calcular  $s_{ij}$ ;
   $s_{ij} = t_{0i} + t_{0j} - t_{ij}$ , onde  $t_{ij}$  é o tempo de viagem gasto entre as cidades  $i$  e  $j$ 
  Se  $s_{ij} > 0$  então Armazenar  $s_{ij}$  na lista de savings;
Fim para
Ordenar a lista de savings decrescentemente;
Para cada dia  $l$  do período faça
  Para cada elemento  $e$  da lista de savings faça
    Se os clientes  $i$  e  $j$  de  $e$  devem ser visitados no dia  $l$  então
      Ligar os clientes  $i$  e  $j$ , desde que a rota resultante respeite as restrições de capacidade e de duração das rotas;
    Fim se
  Fim para
Fim para
```

Figura 4 – Heurística Construtiva de Clarke e Wright adaptada para o PRPV.

Provavelmente, ao final deste procedimento, o número de rotas em cada dia excede o número máximo de veículos permitido em cada dia do período. Para cada dia do período que possui mais rotas que o número de veículos, as rotas deste dia são ordenadas crescentemente em relação ao número de clientes que cada uma possui. Até que o número de rotas seja factível, e respeitando a ordenação, cada cliente da rota que é desfeita deve ser inserido em uma outra rota deste mesmo dia, considerando-se a inserção de menor custo. Desta forma, o número de rotas torna-se factível, porém as capacidades dos veículos e/ou as durações das rotas percorridas pelos veículos podem estar violadas.

Outra forma de se implementar esta heurística é gerar uma lista de *savings* para cada dia do período, contendo cada uma somente os pares de clientes que devem ser visitados em cada dia. Assim, um mesmo par de clientes pode pertencer à lista de *savings* de vários dias do período. Neste sentido, esta estratégia não é interessante, pois para cada dia do período uma lista de *savings* deve ser ordenada e os métodos de ordenação são computacionalmente caros. Além disso, testes computacionais revelaram não haver ganho na qualidade com esta estratégia.

2.2 Heurística de Cordeau et al. (2001)

A heurística construtiva de Cordeau et al. (2001), como mencionado na revisão bibliográfica, é baseada na heurística construtiva de Cordeau et al. (1997), e o algoritmo associado é apresentado na Figura 5.

```
1 Se os clientes são representados por coordenadas Euclidianas então
2   Ordená-los crescentemente em relação ao ângulo que eles formam com o depósito
   e um raio arbitrário;
3 Senão
4   Ordená-los arbitrariamente;
5 Fim se
6   Associar a cada cliente  $i$  uma combinação de dias de visitas pertencente ao conjunto
    $C_i$ , escolhida aleatoriamente;
7 Para cada dia  $l$  do período faça
8   Considerando a ordenação acima dos cliente, escolha um cliente  $j$  aleatoriamente;
9   Faça  $k = 1$ ;
10  Usando a seqüência de clientes  $j, j + 1, \dots, n, 1, \dots, j - 1$ , executar os seguintes
    passos para cada cliente  $i$  que possua dia de visita  $l$ ;
11  Se a inserção do cliente  $i$  na rota  $k$  do dia  $l$  resultar na violação de capacidade
    ou de duração da rota
12  então
13     $k = \min\{k + 1, m\}$ ;
14  Fim se
15  Inserir o cliente  $i$  na rota  $k$  do dia  $l$  minimizando o aumento da duração desta;
16 Fim para
```

Figura 5 – Heurística Construtiva de Cordeau et al. (2001).

Nas linhas 1 – 5 os clientes são ordenados, e depois, na linha 6, a cada um dos clientes é associada uma combinação de dias de visitas, escolhida aleatoriamente. O laço nas linhas 7 – 16 percorre todos os dias do horizonte de tempo. Nas linhas 10 – 15 cada cliente que deve ser visitado no dia é inserido na rota cuja sua inserção gera o menor custo desde que não viole as restrições de capacidade e de duração desta rota. Caso não haja uma rota em que a inserção do cliente não implique a violação das restrições, este é inserido com o menor custo na m -ésima rota do dia. Ao final deste procedimento somente a m -ésima rota de cada dia do período pode estar inactivável em relação à capacidade e/ou à duração das rotas.

2.3 Variações das Heurísticas de Beltrami e Bodin (1974) e Cordeau et al. (2001)

Duas variações para as heurísticas apresentadas acima são sugeridas neste trabalho. A primeira consiste em se executar a heurística de Cordeau et al. (2001), com o diferencial de que

caso não exista uma rota na qual a inserção do cliente seja factível o mesmo deve ser inserido na rota cujo custo de inserção seja o menor. O algoritmo da Figura 6 apresenta esta modificação.

Se os clientes são representados por coordenadas Euclidianas **então**
 Ordená-los crescentemente em relação ao ângulo que eles formam com o depósito e um raio arbitrário;
Senão
 Ordená-los arbitrariamente;
Fim se
 Associar a cada cliente i uma combinação de dias de visitas escolhida aleatoriamente do conjunto C_i ;
Para cada dia l do período **faça**
 Considerando a ordenação acima dos cliente, escolha um cliente j aleatoriamente, sendo que os clientes mais próximos ao depósito tem maior chance de serem escolhidos;
 Faça $k = 1$;
 Usando a seqüência de clientes $j, j + 1, \dots, n, 1, \dots, j - 1$, executar os seguintes passos para cada cliente i que possua dia de visita l ;
 Encontrar a rota k do dia l para o cliente i de tal forma que as restrições sejam respeitadas e o custo de inserção seja o menor possível;
Se não houver rota no dia l para a qual a inserção do cliente i não viole as restrições
então
 Inserir o cliente i na rota cujo o custo de inserção seja o menor;
Fim se
Fim para

Figura 6 – Heurística construtiva modificada, baseada em Cordeau et al. (2001).

A segunda variação refere-se à determinação da combinação de dias de visitas inicial de cada cliente. Como descrito até o momento, esta combinação é determinada aleatoriamente, respeitando-se o conjunto de combinações de dias de visitas possíveis (factíveis) de cada cliente. Outra forma possível de serem determinadas estas combinações iniciais é a apresentada em (Chao et al., 1995). A idéia é começar com uma solução inicial que tenha o total das demandas dos clientes, que são visitados em cada dia do período, balanceado. Esta solução inicial é obtida através do problema de programação inteira apresentado nas Figuras 7 e 8.

L = quantidade máxima de demanda entregue em um único dia
 S_i = conjunto de possíveis combinações de visitas para o cliente i
 $a_{kl} = \begin{cases} 1, & \text{se o dia } l \text{ pertence a combinação de visitas } k \\ 0, & \text{caso contrário} \end{cases}$
 q_{ikl} = demanda do cliente i no dia l na combinação de visitas k
 $x_{ik} = \begin{cases} 1, & \text{se o cliente } i \text{ está associado à combinação de visitas } k \\ 0, & \text{caso contrário} \end{cases}$

Figura 7 – Parâmetros e Variáveis de decisão.

$\min L$	(1)
sujeito a:	
$\sum_{k \in C_i} x_{ik} = 1 \quad (i = 1, \dots, n)$	(2)
$\sum_{i=1}^n \sum_{k \in S_i} a_{kl} q_{ikl} x_{ik} \leq L \quad (l = 1, \dots, t)$	(3)
$x_{ik} \in \{0,1\} \quad (\forall k \in S_i; i = 1, \dots, n)$	(4)

Figura 8 – Programa Inteiro para determinar uma solução inicial para o PRPV.

As restrições (2) e (4) garantem que exatamente uma única combinação de visitas é associada a cada cliente, e a restrição (3) garante que o total de demanda entregue em um único dia seja menor ou igual a L .

Ao invés de resolver este problema, relaxa-se a restrição (4) para $0 \leq x_{ik} \leq 1$, obtendo-se assim um problema de programação linear. Na solução deste problema relaxado, a maior fração do cliente i , isto é, a maior fração de x_{ik} , é arredondada para 1, e então a combinação de dias de visitas k é associada ao cliente i .

3. BUSCA LOCAL

3.1 Introdução

A busca local determina uma trajetória no espaço de busca e depende da estrutura da vizinhança, bem como da forma que esta é explorada. Em diversos problemas de otimização combinatória, uma trajetória composta somente por soluções factíveis pode ser muito restritiva em termos de se encontrar soluções de alta qualidade. De nosso conhecimento, a implementação eficaz da busca tabu de Gendreau et al. (1994) para o problema de roteamento de veículos (PRV) representa o primeiro trabalho nesta área onde se permitem trajetórias de busca com soluções infactíveis.

A permissão de soluções infactíveis adiciona um alto grau de flexibilidade na busca, no entanto coloca uma dificuldade associada à avaliação de uma solução infactível e de como guiar a busca ao se utilizar este tipo de solução. Em geral, uma solução infactível é avaliada através de um custo penalizado que consiste do custo original desta solução adicionado das restrições violadas, multiplicadas por escalares que variam dinamicamente durante a busca. Por exemplo, Gendreau et al. (1994) estabelecem uma frequência de atualização dos parâmetros que multiplicam as restrições de capacidade e de duração das rotas. Se h soluções consecutivas são factíveis com relação a uma das restrições, então o parâmetro é dividido por 2, e se estas h soluções consecutivas são infactíveis, o parâmetro é multiplicado por 2. Cordeau et al. (1997, 2001) também propõem algoritmos de busca tabu para o PRPV que permitem soluções infactíveis penalizadas.

Løkketangen e Glover (1998) propõem uma implementação sofisticada de busca tabu para problemas de programação inteira-mista com variáveis binárias com busca nos pontos extremos do poliedro definido pela relaxação linear. Soluções infactíveis, que no caso correspondem a variáveis binárias que assumem valores fracionários, são permitidas durante a busca. Neste trabalho são propostas estratégias elaboradas para guiar a busca a partir de soluções infactíveis.

A busca local tem papel crucial no desenvolvimento da busca tabu para o PRPV, e por este motivo foram implementadas buscas locais baseadas nos trabalhos de Cordeau et al. (2001) e Løkketangen e Glover (1998).

3.2 Busca Local baseada em Cordeau et al. (2001)

A busca local de Cordeau et al. (2001) utiliza dois tipos de movimentos descritos a seguir.

- inserção de um cliente no mesmo dia: um cliente é removido de sua rota atual e inserido em uma outra rota do mesmo dia com custo de inserção mínimo, definindo uma vizinhança N_1 .
- troca de combinação de dias de visitas de um cliente: a combinação de dias de visitas de um cliente i é alterada para uma outra combinação de dias de visitas pertencente a C_i . Para os dias comuns a ambas combinações, as rotas não se alteram. Para os dias não comuns às combinações, o cliente é removido de todos os dias que pertencem à combinação de dias de visitas antiga e é inserido nas rotas dos dias da nova combinação com menor custo de inserção. Este movimento define uma vizinhança N_2 .

O melhor movimento da vizinhança $N = N_1 \cup N_2$ de acordo com a função de custo penalizada, descrita abaixo, é executado por Cordeau et al. (2001).

Seja s a solução corrente,

$$c(s) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m \sum_{l=1}^t c_{ijkl} x_{ijkl}$$

o custo de s ,

$$q(s) = \sum_{k=1}^m \sum_{l=1}^t \left[\left(\sum_{i=0}^n \sum_{j=0}^n q_i x_{ijkl} \right) - Q \right]^+$$

a violação total das capacidades dos veículos, e

$$d(s) = \sum_{k=1}^m \sum_{l=1}^t \left[\left(\sum_{i=0}^n \sum_{j=0}^n (c_{ijkl} + d_i) x_{ijkl} \right) - D \right]^+$$

a soma dos excessos das durações das rotas.

Sejam α e β fatores de penalidade associados a $q(s)$ e $d(s)$, respectivamente, e δ o parâmetro utilizado para atualizar α e β . Assim, o custo penalizado $f(s)$ da solução s pode ser expresso como:

$$f(s) = c(s) + \alpha q(s) + \beta d(s).$$

A atualização dos parâmetros α e β é feita da seguinte forma. Seja s' uma solução vizinha a s , obtida por um dos dois movimentos descritos anteriormente. Se $q(s') = 0$, então $\alpha = \frac{\alpha}{(1 + \delta)}$. Senão $\alpha = \alpha(1 + \delta)$. A atualização de β é análoga.

Ainda é possível fazer uso de movimentos intra e inter rotas em cada dia para reduzir o custo total das rotas. Para tal, foram implementados os movimentos Or-OPT intra e inter rotas e Troca-Cross, que geram boas soluções em pouco tempo de processamento, conforme resultados apresentados em (Mitsumoto, 2003).

Movimentos intra-rota são aqueles aplicados a cada rota separadamente, a fim de reduzir seus custos. Os movimentos inter-rotas são aplicados a pares de rotas de um conjunto de rotas de forma a minimizar a soma dos custos das rotas.

O movimento Or-OPT foi proposto por Or (1976), para movimentos intra e inter rotas e usado pela primeira vez para movimentos inter-rotas por Savelsbergh (1992). No caso dos movimentos intra-rota, k nós consecutivos de uma rota são re-inseridos em uma outra posição desta mesma rota. No caso dos movimentos inter-rotas, estes k nós consecutivos são inseridos em uma rota distinta a que pertencem. O parâmetro k é um limitante superior para o número de nós consecutivos que é considerado.

O movimento Troca-Cross (inter-rotas) foi proposto por Taillard et al.(1997) e consiste em trocas entre subconjuntos de nós consecutivos de duas rotas. Sua avaliação envolve o cálculo da diferença entre o custo de adicionar e remover arcos, executando o movimento com o menor custo a cada passo.

A avaliação destes movimentos é assim realizada. Seja s a solução corrente e $f(s) = c(s) + \alpha q(s) + \beta d(s)$ o seu custo penalizado e s' uma nova solução obtida através da execução de algum dos movimentos de melhoria e $f(s') = c(s') + \alpha q(s') + \beta d(s')$ o seu custo penalizado.

Para movimentos intra-rota, o objetivo é obter rotas com custos menores que os atuais, e como os clientes pertencentes às rotas não se alteram, a capacidade das mesmas também não é afetada.

Para um movimento inter-rotas executa-se o movimento que acarreta na maior soma das diminuições dos custos das rotas envolvidas. Após a execução deste movimento a solução s' só é aceita se $f(s') < f(s)$.

O algoritmo da Figura 9 apresenta o algoritmo de busca local penalizada com movimentos de melhoria intra e inter rotas.

```

Gerar uma solução inicial  $s$  através de uma heurística construtiva e iniciar os
parâmetros  $\alpha$ ,  $\beta$  e  $\delta$ ;
Calcular  $f(s)$ ;
Se  $q(s) = 0$  e  $d(s) = 0$  então  $s^* = s$ ;
Enquanto condição de parada não for atingida faça
  Seja  $s'$  a melhor solução pertencente a  $N_2(s)$ ;
  Se  $f(s') < f(s)$  então  $s = s'$ ;
  Se  $q(s') = 0$  e  $d(s') = 0$  e  $f(s') < f(s^*)$  então  $s^* = s'$ ;
  Para cada dia  $l$  do período faça
    Executar movimento inter-rotas, e se possível atualizar  $s^*$ ;
    Executar movimento intra-rota para cada rota deste dia, e se possível atualizar  $s^*$ ;
  Fim para
Fim enquanto

```

Figura 9 – Busca Local Penalizada utilizando a vizinhança N_2 e movimentos de melhoria intra e inter-rotas

3.3 Busca Local baseada em Løkketangen e Glover (1998)

A estratégia de classificar movimentos de acordo com a (in)factibilidade e o valor da função objetivo depende da combinação da variação do valor da função objetivo resultante de um movimento e de sua respectiva variação de inactibilidade.

Seja s uma solução, $c(s)$ o valor de sua função objetivo e $d(s)$ e $q(s)$ a soma total das violações das restrições de duração das rotas e de capacidade, respectivamente. Seja s' uma solução pertencente à $N_2(s)$, $c(s')$ o valor de sua função objetivo e $d(s')$ e $q(s')$ a soma total das violações das restrições de duração das rotas e de capacidade, respectivamente. As variações das violações das restrições de duração das rotas e de capacidade e a variação do valor da função objetivo são dados por:

$$\Delta d(s') = d(s') - d(s)$$

$$\Delta q(s') = q(s') - q(s)$$

$$\Delta c(s') = c(s') - c(s)$$

Como as restrições de capacidade e de duração das rotas podem ocorrer simultaneamente no problema, ao invés de analisar cada variação separadamente estas devem ser analisadas da seguinte forma:

$$\Delta u(s') = w_1 \Delta q(s') + w_2 \Delta d(s')$$

Os parâmetros w_1 e w_2 são atualizados dinamicamente ao longo do procedimento de busca local, conforme descrito na seção 3.2.

Logo, $\Delta q(s') < 0$ indica que a nova solução viola menos a restrição de capacidade, e análoga é a interpretação para $\Delta d(s')$. Assim, se $\Delta u(s') < 0$, então a soma ponderada das violações de ambas as restrições da nova solução é inferior a da solução atual. Se $\Delta c(s') < 0$, então a nova solução apresenta uma melhora no valor da função objetivo.

Classificação dos tipos de movimento

Existem quatro tipos de movimento, que são classificados em relação à mudança no valor da função objetivo e na medida de infactibilidade. Sejam $H1$, $H2$, $H3$ e $H4$ os conjuntos de movimentos pertencentes a cada tipo de movimento, e $h1$, $h2$, $h3$ e $h4$ qualquer movimento pertencente a cada um dos correspondentes grupos. Definem-se $u(s)$ e $u(s')$ como a soma das violações das restrições de uma solução s e de uma solução $s' \in N_1(s)$ e/ou $N_2(s)$, respectivamente. Então, os movimentos podem ser assim classificados:

$$H1 = \{s' \in N_1(s) \text{ e/ou } N_2(s) : c(s') \leq c(s) \text{ e } u(s') \leq u(s)\}$$

$$H2 = \{s' \in N_1(s) \text{ e/ou } N_2(s) : c(s') > c(s) \text{ e } u(s') < u(s)\}$$

$$H3 = \{s' \in N_1(s) \text{ e/ou } N_2(s) : c(s') < c(s) \text{ e } u(s') > u(s)\}$$

$$H4 = \left\{ \begin{array}{l} s' \in N_1(s) \text{ e/ou } N_2(s) : c(s') > c(s) \text{ e } u(s') > u(s) \text{ ou} \\ c(s') = c(s) \text{ e } u(s') > u(s) \text{ ou } c(s') > c(s) \text{ e } u(s') = u(s) \end{array} \right\}$$

A Figura 10 mostra os tipos de movimentos no espaço de busca:

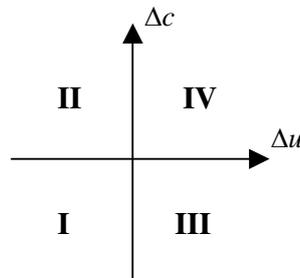


Figura 10 – Classificação dos tipos de movimentos.

Avaliação dos movimentos e regras de escolha

Fazendo-se uso de $\Delta c(s')$ e $\Delta u(s')$ e da classificação dos movimentos, e considerando-se s a solução corrente, quatro regras de avaliação de movimentos são propostas:

1ª Soma ponderada: é estabelecida uma soma ponderada entre $\Delta c(s')$ e $\Delta u(s')$, sendo que o movimento $s' \in N_1(s)$ e/ou $N_2(s)$ escolhido é aquele que possui o menor valor $E(s')$, definido por:

$$E(s') = w_3 \Delta c(s') + \Delta u(s') = w_3 \Delta c(s') + w_1 \Delta q(s') + w_2 \Delta d(s')$$

2ª Teste da razão: Uma vez que cada conjunto de tipo de movimento (H) pode conter mais que um elemento (movimento), para cada conjunto de movimentos é estabelecida uma regra que escolhe um movimento pertencente a este. A Figura 11 auxilia a compreensão destas regras:

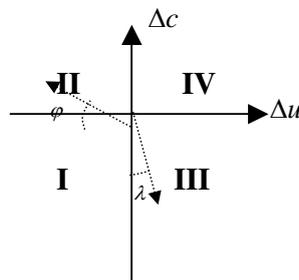


Figura 11 – Representação dos tipos de movimentos.

$h1 = Arg \max(\Delta c(s') \Delta u(s') : s' \in H1)$, se mais de um movimento é determinado, escolha aquele que maximiza $\min(\Delta c(s'), \Delta u(s'))$. Se $\Delta u(s') = 0$, escolha o movimento que maximiza $(-\Delta c(s'))$, e se $\Delta c(s') = 0$ escolha o movimento que maximiza $(-\Delta u(s'))$. Assim, são preferidos movimentos que tenham grande melhora, mas de forma balanceada.

$$h2 = Arg \max \left(\frac{\Delta c(s')}{\Delta u(s')} : s' \in H2 \right),$$

desta forma são preferidos os movimentos que caminhem na direção contrária à infactibilidade. Analisando graficamente através da Figura 11, escolha o movimento que forma o menor ângulo φ . Se mais de um movimento é determinado, escolha aquele que minimize $d(s') + q(s')$.

$$h3 = Arg \max \left(\frac{\Delta u(s')}{\Delta c(s')} : s' \in H3 \right),$$

desta forma são preferidos os movimentos que apresentam melhora na função objetivo. Analisando graficamente através da Figura 11, escolha o movimento que forma o menor ângulo λ . Se mais de um movimento é determinado, escolha aquele que minimize $c(s')$,

$h4 = \text{Arg min}(\Delta c(s')\Delta u(s') : s' \in H4)$. Assim, são preferidos os movimentos que caminham muito pouco na direção indesejada (infactibilidade), mas de forma balanceada. Se mais de um movimento é determinado, escolha aquele que minimiza $\max(-\Delta c(s'), -\Delta u(s'))$.

Então, para cada conjunto de movimentos, no máximo um movimento é considerado. O movimento mais desejado de ser realizado é do tipo 1 ($h1$), e o menos desejado é o do tipo 4 ($h4$). Se existir um movimento do tipo 1, este será o movimento executado, se pelo menos $\Delta c(s') < 0$ ou $\Delta u(s') < 0$. Caso o movimento do tipo 1 não tenha sido executado, execute um movimento do tipo 2 ou um movimento do tipo 3, se os dois tipos de movimentos não ocorrerem simultaneamente. O movimento do tipo 4 só é executado caso não exista movimentos dos outros tipos. Quando um movimento do tipo 1 não for executado e existir tanto um movimento do tipo 2 quanto um movimento do tipo 3, a decisão de qual dos dois movimentos deve ser executado é feita por normalização.

Normalização 1: Defina uma variação de factibilidade $F(w, q)$ como uma função de um multiplicador w e de um expoente q por $F(w, q) = w \sum |\Delta u(s')|^q : s' \in H$ e também defina as razões

$$R = \frac{\sum |\Delta c(s')| : s' \in H}{F(w, q)}, \quad R1(s') = \frac{\Delta c(s')}{R}, \quad R2(s') = \frac{\Delta u(s')}{\Delta c(s')} R.$$

O movimento a ser escolhido é aquele que satisfaz $\text{Arg max}\{R1(h2), R2(h3)\}$. Com referência à Figura 11, significa que os ângulos φ e λ são comparados e o movimento executado é o que forma o menor destes ângulos.

Normalização 2: Seja c^* o valor da melhor solução factível encontrada até o momento na busca. Caso não exista uma solução factível, c^* recebe o valor estimado da melhor solução para a instância em questão. Se $c^* \leq c(s) + \varepsilon$, onde ε é um valor bem pequeno, escolha o movimento do tipo 2 para ser executado. Mas, se $c^* > c(s) + \varepsilon$, faça $R' = \frac{c^* - c(s)}{u(s)}$, e determine o movimento a ser executado assim como na normalização 1, fazendo as devidas substituições de R por R' .

3ª Soma ponderada ordenada dentro de cada tipo de movimento: Calcule $E(s')$ para todos os movimentos. Depois, agrupe os movimentos por tipo, e ordene cada grupo em relação ao valor de E . A ordem de prioridade dos movimentos é: tipo 1, tipo 2, tipo 3 e tipo 4, porém os movimentos dos tipos 2 e 3 são ordenados conjuntamente.

4ª Teste da razão, preferindo o movimento H_2 ao movimento H_3 : A escolha de se optar pelo movimento do tipo 2 ao movimento do tipo 3 faz com que se priorizem os movimentos que estejam caminhando em sentido contrário ao da infactibilidade. Assim, primeiro tenta-se achar uma solução factível para depois tentar melhorá-la em relação à função objetivo.

A Figura 12 apresenta o algoritmo de busca local baseado nestas regras de avaliação de movimentos com busca em vizinhança:

Gerar uma solução inicial s através de uma heurística construtiva e calcular $f(s)$;
Se $q(s) = 0$ **e** $d(s) = 0$ **então** $s^* = s$;
Escolher uma regra de avaliação de movimentos;
Enquanto condição de parada não for atingida **faça**
 Determinar a vizinhança $N_2(s)$;
 Utilizando a regra de avaliação escolhida, determinar $s' \in N_2(s)$;
 $s = s'$;
 Se $f(s') < f(s^*)$ **então** $s^* = s'$;
 Para cada dia l do período **faça**
 Executar movimento inter-rotas, e se possível atualizar s^* ;
 Executar movimento intra-rota para cada rota deste dia, e se possível atualizar s^* ;
 Fim para
Fim enquanto

Figura 12 – Busca Local baseada em regras de avaliação de movimentos de Løkketangen e Glover e movimentos de melhoria intra e inter rotas.

4. BUSCA TABU

4.1 Introdução

Busca tabu (Glover, 1989, 1990, Glover e Laguna, 1997) é uma metaheurística que usa exploração reativa e memória flexível para guiar a busca no espaço de soluções. Através da exploração reativa, determina-se uma direção de busca baseada em propriedades da solução corrente e da história da busca. A memória flexível consiste de estruturas de memória de curto e longo prazo que armazenam a história da busca. A memória de curto prazo armazena atributos de soluções visitadas em passado recente. Estes atributos são armazenados numa lista tabu para impedir o retorno a soluções visitadas. A memória de longo prazo contém uma história seletiva de soluções e seus atributos encontrados durante o processo de busca, e é utilizada em estratégias de diversificação e intensificação da busca.

4.2 Busca Tabu Proposta

A proposta de nosso trabalho é desenvolver um algoritmo de busca tabu para o PRPV que seja robusto, com número reduzido de parâmetros e competitivo com os algoritmos de busca tabu de Cordeau et al. (1997, 2001). A seguir, apresentamos a descrição do algoritmo de busca tabu aqui implementado.

4.2.1 Critério de Aspiração e Condição de Parada

O critério de aspiração é introduzido na busca tabu para determinar quando as regras de proibição devem ser desconsideradas, revogando o status tabu associado a um movimento. Implementa-se, neste trabalho, o critério clássico, que consiste em executar um movimento tabu se a solução resultante deste movimento é melhor que a incumbente (melhor solução encontrada até o momento).

A condição de parada utilizada é a execução de um número máximo de iterações, independente do tamanho da instância, para que desta forma seja possível fazer uma comparação com os resultados apresentados em Cordeau et al.(1997).

4.2.2 Memória de Curto Prazo

A eficiência da memória de curto prazo depende da vizinhança utilizada, da regra de proibição e da duração da proibição. A proibição de movimentos ocorre através do

armazenamento dos atributos das soluções visitadas ao longo da busca em uma lista tabu que é consultada a cada vez que se avalia um novo movimento. Se um atributo está presente na lista tabu, então o movimento associado é proibido de ser executado, exceto se satisfaz o critério de aspiração.

Como já mencionado na seção 3.2, nos trabalhos de Cordeau et al. (1997) e Cordeau et al. (2001) são utilizados dois tipos de movimentos:

- inserção de um cliente no mesmo dia: um cliente é removido de sua rota atual e inserido em uma outra rota do mesmo dia com custo de inserção mínimo, definindo uma vizinhança N_1 .
- troca de combinação de dias de visitas de um cliente: a combinação de dias de visitas de um cliente i é alterada para uma outra combinação de dias de visitas pertencente a C_i . Para os dias comuns a ambas combinações, as rotas não se alteram. Para os dias não comuns às combinações, o cliente é removido de todos os dias que pertencem à combinação de dias de visitas antiga e é inserido nas rotas dos dias da nova combinação com menor custo de inserção. Este movimento define uma vizinhança N_2 .

A regra de proibição definida nestes trabalhos é a seguinte. Seja i o cliente que pertence à rota r do dia l . Este cliente fica proibido (tabu) de retornar à rota r do dia l por θ iterações, exceto se satisfizer o critério de aspiração. Desta forma uma troca de combinação de dias de visitas de um cliente i é possível somente se a nova combinação de dias contenha pelo menos uma rota não tabu associada ao cliente i . A lista tabu é implementada em uma matriz tridimensional, em que a primeira dimensão se refere ao dia do período, a segunda ao cliente e a terceira à rota, e é denotada *lista tabu 1*.

Neste trabalho, utilizamos somente a vizinhança N_2 devido à seguinte observação. Ao se designar uma combinação de dias de visitas para cada cliente, a resolução do problema consiste em se resolver um problema de roteamento de veículos clássico para cada dia do horizonte de tempo. Desta forma, definimos uma regra de proibição que envolve somente troca de combinações de dias de visitas. Seja o cliente i , $c \in C_i$ sua combinação de dias de visitas atual e $c' \in C_i$ sua nova combinação de dias de visitas. Então o cliente i não pode retornar à combinação de dias de visitas c (tabu) por θ iterações. A lista tabu consiste de uma matriz bidimensional, tal que a primeira dimensão se refere ao cliente e a segunda às suas possíveis combinações de dias de visitas, e é denotada *lista tabu 2*.

A Figura 13 apresenta a estrutura do algoritmo de busca tabu implementado com

memória de curto prazo, sendo aplicável tanto a *lista tabu 1* quanto para a *lista tabu 2*.

Gerar uma solução inicial s através de uma heurística construtiva e calcular $f(s)$;
Se $q(s) = 0$ **e** $d(s) = 0$ **então** $s^* = s$;
Escolher uma regra de avaliação de movimentos;
Enquanto condição de parada não for atingida **faça**
 Determinar a vizinhança $N_2(s)$;
 Utilizando a regra de avaliação escolhida, determinar $s' \in N_2(s)$ que ou satisfaça
 $f(s') < f(s^*)$, ou não seja tabu;
 $s = s'$;
 Atualizar a lista tabu;
 Se s' é factível **e** $f(s') < f(s^*)$ **então** $s^* = s'$;
Fim enquanto

Figura 13 – Busca Tabu

4.2.3 Movimentos de melhoria

O objetivo da inclusão de movimentos de melhoria intra e inter-rotas, descritos na seção 3.2, é minimizar a duração das rotas assim como a infactibilidade das soluções. A avaliação destes consiste em se calcular o valor da função objetivo penalizada associada a cada movimento comparando-o com o valor da função objetivo penalizada da solução corrente. Executa-se o movimento que possui o menor valor de função objetivo penalizada, sendo que este valor tem que ser menor que o valor da função objetivo penalizada da solução corrente. A inclusão destes movimentos de melhoria é mostrada na Figura 14.

Gerar uma solução inicial s através de uma heurística construtiva e calcular $f(s)$;
Se $q(s) = 0$ **e** $d(s) = 0$ **então** $s^* = s$;
Escolher uma regra de avaliação de movimentos;
Enquanto condição de parada não for atingida **faça**
 Determinar a vizinhança $N_2(s)$;
 Utilizando a regra de avaliação escolhida, determinar $s' \in N_2(s)$ que ou satisfaça
 $f(s') < f(s^*)$ ou não seja tabu;;
 $s = s'$;
 Atualizar a lista tabu;
 Se $f(s') < f(s^*)$ **então** $s^* = s'$;
 Enquanto houver melhora **faça**
 Para cada dia l do período **faça**
 Executar movimento inter-rotas, e se possível atualizar s^* ;
 Executar movimento intra-rota para cada rota deste dia, e se possível
 atualizar s^* ;
 Fim para
 Fim Enquanto
Fim enquanto

Figura 14 – Busca Tabu com movimentos de melhoria intra e inter rotas.

4.2.4 Memória de Longo Prazo

A memória de longo prazo armazena atributos de soluções e/ou um conjunto de soluções visitadas ao longo da busca, através da utilização de matrizes de frequência e de conjuntos que armazenam soluções. Esta memória é utilizada para gerar estratégias de intensificação, que aprofundam a busca em regiões promissoras, e de diversificação, que exploram regiões não visitadas.

As seguintes estratégias foram testadas isoladamente na busca tabu:

- 1) Diversificação durante a busca tabu, baseada em frequência de residência de um atributo;
- 2) Diversificação baseada em reinício da busca tabu, em que a cada reinício é construída uma nova solução baseada em frequência de residência de um atributo;
- 3) Intensificação baseada em reinício da busca tabu a partir de cada solução de um conjunto de boas soluções encontradas durante a busca.
- 4) Estratégia de reinício da busca tabu que integra diversificação e intensificação, em que se armazenam rotas de boas soluções ao longo da busca para construir uma nova solução de partida.

Além disso, foram testadas duas estratégias combinadas, envolvendo as estratégias de diversificação 1 e 2 com o mecanismo de religação de caminho (*path relinking*). Este mecanismo consiste a construção de uma trajetória entre duas soluções de qualidade, tendo portanto, um caráter intensificador.

As estratégias de diversificação 1 e 2 utilizam uma matriz de frequência de residência, denominada $freq$, que armazena o número de iterações que um cliente i esteve associado a cada uma de suas possíveis combinações de visitas $c \in C_i$ ($freq[i][c]$).

Na estratégia de diversificação 1, após um determinado número de iterações sem melhora da solução incumbente, a troca de uma combinação de um cliente por outra combinação muito frequente é penalizada proporcionalmente a $freq[i][c]$. Seja Δ_c a variação da função objetivo ao se alterar a combinação de dias c do cliente i para a combinação c' . A mudança do cliente para a combinação c' é feita através de movimentos de inserção de menor custo, e a variação penalizada Δ'_c da função objetivo é dada por

$$\Delta'_c = \Delta_c + d\Delta_{\max} \frac{freq[i][c]}{freq_{\max}},$$

em que $\Delta_{\max} = \max\{\Delta_c \in C_i\}$, $freq_{\max}$ representa a frequência máxima da matriz de residência, e d

é um parâmetro que controla o grau de diversificação.

Esta penalização é executada por um determinado número (κ) de iterações, e então desativada, para que a busca tabu de curto prazo possa explorar a nova região. A Figura 15 mostra o procedimento de diversificação 1.

1	Se diversificação durante a busca então
2	Determine $freq_{\max}$;
3	Para cada cliente i faça
4	Para cada combinação de dias de visitas $c \in C_i$ faça
5	Calcule a variação (Δ_c) no valor da função objetivo ao se associar o cliente i a sua combinação de dia de visitas c , considerando-se a inserção de menor custo em cada dia de c ;
6	Fim para
7	Fim para
8	Para cada cliente i faça
9	Determine $\Delta_{\max} = \max\{\Delta_c \in C_i\}$
10	Para cada combinação de dias de visitas $c \in C_i$ faça
11	$\Delta'_c = \Delta_c + d\Delta_{\max} \frac{freq[i][c]}{freq_{\max}};$
12	Fim para
13	Fim para
14	Faça $c = \min\{\Delta'_c : c \in C_i, \text{ para todo cliente } i\}$ e associe a combinação de dias de visitas c ao cliente i , inserindo-o com o menor custo nas rotas dos dias pertencentes a c ;
15	Fim se

Figura 15 – Estratégia de diversificação 1.

Na estratégia de diversificação 2, após um determinado número de iterações sem melhora da solução incumbente o reinício é aplicado através da construção de uma nova solução em um processo iterativo. Os clientes são ordenados decrescentemente em relação ao número de dias de visitas das combinações. Ao primeiro cliente i resultante desta ordenação é associada sua combinação de dias de visitas menos freqüente, isto é, $\min(freq[i][c]: c \in C_i)$. Depois, cliente a cliente, é determinada sua combinação de dias de visitas, de tal forma que as mais freqüentes são penalizadas. Associa-se ao cliente sua combinação de dias de visitas que gera o menor aumento na função objetivo, e então em cada dia referente à combinação de dias de visitas o cliente é inserido com o menor custo. A Figura 16 mostra o procedimento diversificação 2.

Se diversificação baseada em reinício **então**

Determine $freq_{\max}$;

Ordene os clientes decrescentemente em relação ao número de dias de visita das combinações;

Para cada cliente i **faça**

Se i é o primeiro cliente **então**

Associe a i sua combinação de dias de visitas menos freqüente ($\min(freq[i][c] : c \in C_i)$), e insira-o nas rotas dos dias desta combinação;

Senão

Para cada combinação de dias de visitas $c \in C_i$ **faça**

Estime a variação (Δ_c) no valor da função objetivo em se associar o cliente i a sua combinação de dias de visitas c , considerando-se a inserção de menor custo em cada dia de c ;

Fim para

Fim se

Fim para

Para cada cliente i , onde i não é o primeiro cliente **faça**

Determine $\Delta_{\max} = \max\{\Delta_c \in C_i\}$

Para cada combinação de dias de visitas $c \in C_i$ **faça**

$$\Delta'_c = \Delta_c + d\Delta_{\max} \frac{freq[i][c]}{freq_{\max}};$$

Fim para

Associe ao cliente i a combinação de dias de visitas $c : \min\{\Delta'_c : c \in C_i\}$, e insira-o com o menor custo nas rotas dos dias pertencentes a c ;

Fim para

Fim se

Figura 16 – Estratégia de diversificação 2.

A estratégia de intensificação baseada em reinício 3) utiliza como base um conjunto de soluções de referência *Ref*, que armazena soluções de alta qualidade com um grau mínimo de diversidade entre elas.

A diversidade das soluções é calculada a partir de uma medida de distância entre duas soluções, que consiste do número de combinações distintas dos clientes. Por exemplo, se o problema envolve 10 clientes e 6 clientes têm a mesma combinação de dias nas soluções s_1 e s_2 , então a distância (Hamming) entre estas soluções é 4. Assim, uma solução é considerada diversa no conjunto das soluções de referência se ela é *dmin* distante de todas as soluções deste conjunto, onde *dmin* representa a distância a ser respeitada.

Uma solução entra no conjunto *Ref* se tiver valor menor que o da melhor solução do conjunto, ou se tiver valor menor que o da pior solução do conjunto e aumentar a distância entre as soluções do conjunto. Em ambos os casos, a pior solução do conjunto é excluída.

Na implementação desta estratégia, cada reinício é dado após um determinado número de iterações sem melhora da solução incumbente, utilizando uma das soluções do conjunto *Ref*, como solução de partida para a busca tabu, que é posteriormente descartada deste conjunto. A busca tabu é encerrada quando uma das duas situações ocorre: o número máximo de iterações executadas estabelecido é alcançado ou o conjunto *Ref* se torna vazio.

Rochat e Taillard (1995) propuseram uma estratégia de duas fases que integra diversificação e intensificação para o PRV. Na primeira fase, um conjunto de boas soluções distintas entre si é gerado com o auxílio de um procedimento de busca local. Na segunda fase, as rotas destas soluções são extraídas e utilizadas na construção de uma nova solução. Esta idéia parte do princípio de que se uma solução inclui rotas que pertencem a boas soluções, provavelmente o valor de sua função objetivo é melhor que o de uma solução que não contém estas rotas. Assim, esta segunda fase favorece a escolha de rotas que pertencem às melhores soluções geradas na primeira fase, porém sem excluir totalmente a escolha de rotas que pertencem às piores soluções do conjunto.

Para que as informações das soluções encontradas ao longo da busca tabu sejam armazenadas, cria-se um conjunto *T* onde cada elemento corresponde a uma rota destas soluções. Cada rota armazenada é identificada pelo valor da função objetivo da solução correspondente. O conjunto *T* é ordenado crescentemente em relação ao valor de função objetivo de cada elemento e as rotas com um único cliente são removidas do conjunto.

Tendo as informações necessárias armazenadas, o procedimento de reinício, estratégia 4, é iniciado sempre que um número máximo de iterações sem melhora da solução incumbente for atingido. A cada reinício uma nova solução é construída baseada nas informações de $T' = T$, selecionando-se e removendo-se rotas deste conjunto, até que o mesmo fique vazio. A escolha de um elemento de T' é feita probabilisticamente, sendo que o *i*-ésimo elemento tem probabilidade

$\frac{2i}{|T'| + 1}$ de ser escolhido, preferindo-se rotas com valores de função objetivo menores. Cada

vez que uma rota em T' é escolhida para auxiliar a construção da nova solução, as demais rotas de T' que possuem pelo menos um cliente em comum com esta rota são removidas deste conjunto. No caso do PRPV, inserir a rota escolhida em T' como uma rota da nova solução não é de todo consistente, pois os clientes pertencentes a esta rota podem necessitar de mais de um dia de visitas ao longo do horizonte de tempo, ficando assim indefinidas as rotas destes clientes

nos demais dias. Sendo assim, as rotas armazenadas em T' , em nossa abordagem, determinam a combinação de dias de visitas associada a cada cliente de uma rota. Quando $T' = \phi$, então as rotas nele armazenadas foram analisadas e combinações de dias de visitas foram associadas aos clientes destas rotas. Porém, ao final deste procedimento nem todos os clientes necessariamente estão associados a uma combinação de dias de visitas. Neste caso, a estes clientes associa-se uma combinação de dias de visitas aleatoriamente. Terminada a fase de designação de combinação de dias de visitas, a heurística construtiva de Beltrami e Bodin (1974) é aplicada para que as rotas de cada dia do horizonte de planejamento sejam definidas, gerando-se assim uma nova solução completa, que posteriormente é submetida a um procedimento de busca local, baseado em movimentos de melhoria intra e inter rotas. Cada rota desta nova solução é identificada pelo valor da função objetivo da solução e inserida em T .

A Figura 17 apresenta o procedimento de geração de novas soluções da estratégia 4, baseado em Rochat e Taillard (1995).

<p>Se reinício baseado em Rochat e Taillard então Ordenar T crescentemente em relação ao valor da função objetivo associada a cada rota pertencente a T ; Iniciar a solução a ser gerada $S = \phi$, Faça $T' = T$; Enquanto $T' \neq \phi$ faça Escolher uma rota t pertencente a T' , sendo que a $i - \acute{e}$sima pior rota de T' tem probabilidade $\frac{2i}{ T' + 1}$ de ser escolhida; Para cada cliente em S que pertencente a t faça Associar ao cliente a respectiva combinação de dias de visitas; Fim para Remover de T' todas as rotas que possuam pelo menos um cliente em comum com t ; Fim enquanto Se algum cliente de S não possuir uma combinação de dias de visitas determinada então Determinar aleatoriamente; Fim se Aplicar a heurística construtiva de Beltrami e Bodin (1974) para determinar as rotas de cada dia do horizonte de planejamento; Executar sobre S uma busca local, gerando S' ; Associar a cada rota de S' o valor da função objetivo e inserir em T todas as rotas que possuam mais de um cliente; Fim se</p>
--

Figura 17 – Procedimento de construção de uma nova solução baseada na estratégia de Rochat e Taillard.

O tamanho da lista T é grande, para que informações de boas soluções não sejam descartadas, porém possui tamanho limitado. Com isso, depois que a lista fica completa com rotas, a inserção de rotas de soluções que não são muito boas se torna mais difícil, o que faz com que a cada reinício a qualidade das soluções geradas seja melhor. Nos primeiros reinícios as soluções geradas são mais diversas, funcionando então como estratégia de diversificação. Após vários reinícios o processo automaticamente intensifica a busca em regiões promissoras, uma vez que as piores rotas são removidas de T . Como rotas idênticas não são removidas de T e existem rotas que não são modificadas pela busca local, as melhores rotas de T são mais freqüentemente extraídas durante o processo de construção da nova solução, e a busca muda progressivamente de diversificação para intensificação.

Religação de caminho foi originalmente proposta por Glover (1996) como um enfoque para integrar estratégias de intensificação e diversificação no contexto de busca tabu (Glover e Laguna, 1997) e *scatter search* (Glover, 1998; Glover, Laguna e Martí, 2000). Este enfoque gera novas soluções pela exploração de trajetórias que conectam soluções de alta qualidade (intensificação) ou soluções de regiões distintas ou que possuem características contrastantes (diversificação). Partindo de uma destas soluções, chamada solução inicial s_i , um caminho é gerado em direção a outras soluções chamadas soluções guias. Isto é realizado pela seleção de movimentos que introduzem atributos contidos nas soluções guias. O melhor movimento que reduz a distância de s_i às soluções guias é executado. A religação de caminho tem sido usada com sucesso em combinação com GRASP (Resende e Ribeiro, 2005) e como estratégia de combinação de soluções (veja por exemplo, Jain e Meeran, 2002 e Yamashita et al., 2004).

No presente contexto, o processo de religação inicia de uma solução de partida e executa troca de combinações de dias de visitas de forma a diminuir a distância em relação à solução guia. Suponha duas soluções $s_1(0,1,1,2,0)$ e $s_2(0,2,1,0,3)$, em que cada elemento representa a combinação de dias de visita associada ao cliente 1, 2, 3, 4 e 5, respectivamente. A distância entre s_1 e s_2 é de três unidades, pois os clientes 2, 4 e 5 possuem combinações de dias de visitas distintas em s_1 e s_2 . O processo de religação consiste em escolher a melhor solução que diminua de uma unidade a distância atual entre as duas soluções. Uma possível trajetória traçada partindo de s_1 e chegando em s_2 , considerando-se que a cada passo é executado o melhor movimento, é mostrada abaixo.



Ao longo da trajetória, a cada execução de um movimento de troca de combinação de dias de visitas escolhe-se sempre o movimento que gera a melhor solução factível, isto é, com o menor custo possível. Porém, não há garantia de que a cada passo da trajetória exista um movimento de troca de combinação de dias de visitas que gere uma solução factível em relação à capacidade dos veículos e à duração das rotas. Nos passos da trajetória onde não é possível obter uma solução factível, executa-se o movimento de troca de combinação de dias de visitas que gera a solução de menor custo penalizado.

Vários estudos têm mostrado experimentalmente que é conveniente aplicar procedimento de busca local em algumas das soluções geradas ao longo das trajetórias traçadas, como proposto em Glover (1994). Note que duas soluções consecutivas após um passo de religação são muito similares e diferem somente nos atributos que foram introduzidos. Portanto, geralmente não é eficiente aplicar um método de melhoria a cada passo do processo de religação. Foi testado introduzir um parâmetro *NumImp* que controla a aplicação do método de melhoria, mas os melhores resultados foram obtidos aplicando o método de melhoria a cada solução factível encontrada ao longo do processo de religação. O método de melhoria aplicado é uma busca local baseada em movimentos de melhoria intra e inter rotas, como apresentado na Figura 18.

O mecanismo de religação de caminho precisa da definição de quais soluções vão ser ligadas através de uma trajetória. Para isto, mantemos o conjunto de referência *Ref*, como explicado acima, e a primeira forma de religação é a seguinte. Cada vez que uma solução *s* entra neste conjunto, percorre-se uma trajetória iniciando por *s* e cada solução do conjunto *Ref*, bem como uma trajetória iniciando em cada solução do conjunto *Ref* e *s*. A segunda forma de religação é mais econômica e consiste de uma pós-otimização, na qual todos os pares de soluções de *Ref*, após o término da busca tabu, são conectadas bi-direcionalmente. A Figura 18 mostra o procedimento de religação de caminho implementado.

Considere s a solução que atualiza o conjunto Ref ;

Para cada solução s_{ref} do conjunto de soluções de referencia **faça**

Executar os passos abaixo duas vezes: ora considerando $\bar{s} = s_{ref}$ a solução inicial e s a solução guia e ora considerando $\bar{s} = s$ a solução inicial e s_{ref} a solução guia;

Calcular a distância $dist$ entre \bar{s} e a solução guia;

Determinar a vizinhança $N_2(\bar{s})$;

Se existe $s' \in N_2(\bar{s})$ tal que $f(s') < f(s^*)$ **então** $s^* = s'$;

Faça $\bar{N}(\bar{s}) = \{s' \in N_2(\bar{s}) \mid s' \text{ seja } dist - 1 \text{ em relação a solução guia}\}$;

Se existe $\bar{s} \in \bar{N}(\bar{s})$ factível **então**

Escolha \bar{s} factível de menor custo;

Enquanto houver melhora **faça**

Para cada dia l do período **faça**

Executar movimento inter-rotas, e se possível atualizar s^* ;

Executar movimento intra-rota para cada rota deste dia, e se possível atualizar s^* ;

Fim para

Fim enquanto

Senão

Escolha \bar{s} de menor custo penalizado;

Fim se

$\bar{s} = s$ e $dist = dist - 1$;

Fim para

Insira s no conjunto de soluções de referencia;

Figura 18 – Mecanismo de religação de caminho.

4.3 Busca Tabu de Cordeau et al. (2001)

Cordeau et al. (2001) propõem um algoritmo unificado de busca tabu, baseado no algoritmo proposto por Cordeau et al. (1997), para os problemas de roteamento de veículos com janelas de tempo, de roteamento periódico de veículos com janelas de tempo e de roteamento de veículos com janelas de tempo com múltiplos depósitos. Neste algoritmo a solução inicial é obtida assim como em Cordeau et al. (1997), porém ao invés de usar a heurística GENI como mecanismo de inserção de clientes em rotas, faz-se uso do mecanismo de inserção de menor custo.

Neste trabalho implementa-se o algoritmo de busca tabu proposto por Cordeau et al. (2001) com o fim de testar as instâncias apresentadas na literatura para o PRPV e comparar os resultados obtidos com os resultados apresentados em Cordeau et al. (1997) e com os resultados obtidos no algoritmo de busca tabu aqui proposto.

4.3.1 Descrição do Algoritmo de Busca Tabu

O algoritmo de busca tabu desenvolvido é composto essencialmente de duas partes: construção de uma solução inicial, apresentada na Figura 5, e busca tabu propriamente dita.

A solução inicial gerada pelo procedimento descrito na Figura 5 não necessariamente é factível, pois o número de veículos permitidos em cada dia do período é limitado. Neste sentido, soluções infeasíveis são permitidas ao longo da busca tabu. A função de avaliação de uma solução, utilizada neste algoritmo, é a mesma apresentada na seção 3.2, assim como a vizinhança N utilizada.

A cada solução s é associado um conjunto de atributos $B(s) = \{(i, k, l) : \text{cliente } i \text{ é visitado pelo veículo } k \text{ no dia } l\}$, e sendo assim, a transição de uma solução corrente s para uma solução $s' \in N(s)$ pode ser expressa pela adição e remoção de atributos de $B(s)$.

Se um cliente i é removido da rota r no dia l , reinseri-lo nesta rota é proibido por θ iterações, assim como descrito na subseção 4.2.2. O número da última iteração para a qual o atributo (i, k, l) é declarado tabu é denotado por τ_{ikl} . O *status* do atributo pode, no entanto, ser desconsiderado se a inserção deste atributo na solução resultar em uma solução factível de menor custo que a melhor solução identificada com este atributo.

O nível de aspiração σ_{ikl} de um atributo (i, k, l) é primeiramente iniciado com o custo $c(s)$ se (i, k, l) pertence ao conjunto de atributos de uma solução inicial factível e ∞ caso contrário. Toda vez que uma solução factível s é identificada, o nível de aspiração de cada um de seus atributos (i, k, l) é atualizado para $\min\{\sigma_{ikl}, c(s)\}$.

A cada iteração, o subconjunto $M(s) \subseteq N(s)$ consiste de todas as soluções $\bar{s} \in N(s)$, sendo que estas soluções \bar{s} são tais que pelo menos um de seus atributos que devem ser adicionados a s para se obter \bar{s} não é considerado tabu ou $c(\bar{s})$ é menor que o nível de aspiração deste atributo.

Para diversificar a busca, o custo $f(s)$ de uma solução s é penalizado de forma a priorizar atributos pouco adicionados a solução. Seja ρ_{ikl} o número de iterações durante as quais o atributo (i, k, l) foi adicionado a solução corrente. O termo de penalidade é proporcional ao produto de $c(s)$, ρ_{ikl} e \sqrt{nm} , e um fator constante γ é utilizado para ajustar a intensidade da diversificação.

A cada iteração, os valores de α e β são modificados por um fator $1+\delta > 1$. Se a solução corrente é factível em relação a capacidade (duração) das rotas o valor de α (β) é dividido por $1+\delta$ e caso contrário é multiplicado por $1+\delta$.

A seguir é apresentada a notação utilizada na descrição do algoritmo de busca tabu, apresentado na Figura 19.

$B(s) = \{(i, k, l) : \text{cliente } i \text{ é visitado pelo veículo } k \text{ no dia } l\}$: conjunto de atributos da solução s ;

$c(s)$: custo (duração) das rotas da solução s

$d(s)$: excesso de duração das rotas da solução s

$f(s)$: custo da solução s

$g(s)$: custo penalizado da solução s

$M(s)$: um subconjunto de $N(s)$

$N(s)$: Vizinhança da solução s

s, \bar{s} : soluções

s^* : solução incumbente

α : fator de penalização para o excesso de capacidade das rotas

β : fator de penalização para o excesso de duração das rotas

γ : fator utilizado para ajustar a intensidade da diversificação

δ : parâmetro utilizado para atualizar α e β

η : número total de iterações a serem executadas

θ : duração tabu

λ : contador de iterações

ρ_{ikl} : número de vezes que o atributo (i, k, l) é adicionado a solução

σ_{ikl} : nível de aspiração do atributo (i, k, l)

τ_{ikl} : última iteração para qual o atributo (i, k, l) é declarado tabu

A Figura 19 contém a descrição do algoritmo. Primeiramente, se a solução inicial é factível a solução incumbente é atualizada, e os fatores de penalização são iniciados em 1. A seguir, no laço correspondente às linhas 3 – 7, para todos os atributos (i, k, l) , ρ_{ikl} e τ_{ikl} são iniciados em 0, e se o atributo pertence ao conjunto de atributos da solução corrente e esta é factível, o nível de aspiração do atributo é iniciado com o custo das rotas da solução corrente, caso contrário é iniciado com um valor muito alto. E assim o processo iterativo da busca tabu é iniciado.

O laço nas linhas 8 – 36 faz o controle do número de iterações a serem executadas, e em cada iteração são realizados os seguintes passos. Na linha 9 o subconjunto das soluções vizinhas da solução corrente é iniciado vazio. O laço nas linhas 10 – 13 faz o controle para que todas as soluções vizinhas da solução corrente sejam analisadas, considerando-se o status tabu e o nível de aspiração de todos os atributos pertencentes à solução analisada e que não pertencem

a solução corrente, inserido-as ou não no subconjunto de soluções vizinhas da solução corrente.

O laço nas linhas 14 – 17 controla o subconjunto determinado no passo anterior para que todas as soluções pertencentes a ele sejam analisadas, e para cada solução analisada é determinado seu custo penalizado.

Na linha 18 é identificada a solução pertencente a este subconjunto que possua o menor custo penalizado e subsequente é executado o laço nas linhas 19 – 22, que percorre todos os atributos pertencentes à solução corrente e que não pertencem à solução identificada no passo anterior, retirando estes atributos da solução corrente, atualizando-a, e atualizando seus *status* tabu.

O laço nas linhas 23 – 26 percorre todos os atributos que pertencem à solução identificada com o menor custo penalizado e que não pertencem à solução corrente e os insere com o menor custo na solução corrente, atualizando-a, e atualizando o número de vezes que cada um destes atributos foi adicionado à solução.

Nas linhas 27 e 28, se a solução corrente, resultante destas inserções e remoções de atributos, é factível, se seu custo penalizado é menor que o custo penalizado da solução incumbente a solução incumbente é atualizada, e cada um de seus atributos tem seu nível de aspiração redefinido nas linhas 29 – 31.

Por fim, os fatores de penalização de excesso de capacidade dos veículos e de excesso de duração das rotas, respectivamente, são atualizados nas linhas 32 – 35.

1 Faça $\alpha = \beta = 1$;

2 **Se** s é factível **então** $s^* := s$;

3 **Para** todo (i, k, l) **faça**

4 $\rho_{ikl} = 0$ e $\tau_{ikl} = 0$;

5 **Se** $(i, k, l) \in B(s)$ e s é factível **então** $\sigma_{ikl} = c(s)$;

6 **Senão** $\sigma_{ikl} = \infty$;

7 **Fim para**

8 **Para** $\lambda = 1, \dots, \eta$ **faça**

9 $M(s) = \phi$;

10 **Para** cada $\bar{s} \in N(s)$ **faça**

11 **Se** existe $(i, k, l) \in B(\bar{s}) \setminus B(s)$ tal que $\tau_{ikl} < \lambda$ **ou**

12 tal que \bar{s} é factível e $c(\bar{s}) < \sigma_{ikl}$ **então** $M(s) = M(s) \cup \{\bar{s}\}$;

13 **Fim para**

14 **Para** cada $\bar{s} \in M(s)$ **faça**

15 **Se** $f(\bar{s}) \geq f(s)$ **então** $g(\bar{s}) = f(\bar{s}) + \gamma \sqrt{nm} c(s) \sum_{(i,k,l) \in B(\bar{s}) \setminus B(s)} \frac{\rho_{ikl}}{\lambda}$;

16 **Senão** $g(\bar{s}) = f(\bar{s})$;

17 **Fim para**

18 Identifique uma solução $s' \in M(s)$ minimizando $g(\bar{s})$;

19 **Para** cada atributo $(i, k, l) \in B(s) \setminus B(s')$ **faça**

20 Remova o cliente i da rota k do dia l ;

21 $\tau_{ikl} = \lambda + \theta$;

22 **Fim para**

23 **Para** cada atributo $(i, k, l) \in B(s') \setminus B(s)$ **faça**

24 Insira o cliente i na rota k do dia l com o menor custo;

25 $\rho_{ikl} = \rho_{ikl} + 1$;

26 **Fim para**

27 **Se** s' é factível **então**

28 **Se** $c(s') < c(s^*)$ **então** $s^* = s'$;

29 **Para** cada $(i, k, l) \in B(s')$ **faça**

30 $\sigma_{ikl} = \min\{\sigma_{ikl}, c(s')\}$

31 **Fim para**

32 **Se** $q(s') = 0$ **então** $\alpha = \frac{\alpha}{1 + \delta}$;

33 **Senão** $\alpha = \alpha(1 + \delta)$;

34 **Se** $d(s') = 0$ **então** $\beta = \frac{\beta}{1 + \delta}$;

35 **Senão** $\beta = \beta(1 + \delta)$;

36 **Fim para**

Figura 19 – Descrição do algoritmo de busca tabu de Cordeau et al. (2001)

5. TESTES COMPUTACIONAIS

5.1 Introdução

Neste capítulo são descritos os testes realizados para avaliar o desempenho das heurísticas construtivas, dos algoritmos de busca local e dos algoritmos de busca tabu, implementados neste trabalho para o PRPV. Os programas foram implementados em linguagem C++ e os testes realizados em um PC Compaq, com processador AMD, frequência de 1,7GHz e memória RAM de 512MB.

Os resultados do algoritmo de busca tabu implementado são comparados com os resultados apresentados em Cordeau et al. (1997) e Alegre et al. (2004), assim como com os resultados obtidos pelo algoritmo de busca tabu apresentado em Cordeau et al. (2001).

5.2 Descrição das instâncias testadas

As instâncias testadas estão divididas em dois grupos: instâncias que possuem restrições de capacidade e de duração (pr01 – pr10) e instâncias que possuem somente restrição de capacidade (p02 – p32). As Tabelas 1 e 2 apresentam a descrição de cada instância de cada grupo.

Nestas tabelas, a coluna “n” indica o número de clientes, a coluna “m” indica o número máximo de veículos permitido em cada dia do período, a coluna “t” indica o número de dias do horizonte de planejamento e as colunas “D” e “Q” indicam a duração máxima de cada rota percorrida por um veículo e a capacidade máxima de cada veículo, respectivamente.

Em todas as tabelas a seguir são apresentados valores de desvio ou desvio médio em relação ao valor da melhor solução conhecida, mostrada na Tabela 1.

Tabela 1 – Instâncias com restrições de duração e de capacidade

Instância	Melhor solução conhecida (Cordeau et al.(1997))	n	m	t	D	Q
pr01	2209,02	48	2	4	500	200
pr02	3799,28	96	4	4	480	195
pr03	5218,13	144	6	4	460	190
pr04	6012,79	192	8	4	440	185
pr05	6769,8	240	10	4	420	180
pr06	8422,64	288	12	4	400	175
pr07	4997,41	72	3	6	500	200
pr08	7094,52	144	6	6	475	190
pr09	10370,45	216	9	6	450	180
pr10	13370,04	288	12	6	425	170

Tabela 2 - Instâncias somente com restrição de capacidade

Instância	Melhor solução conhecida (Cordeau et al.(1997))	n	m	t	Q
p02	1322,87	50	3	5	160
p03	524,61	50	1	5	160
p04	835,43	75	6	5	140
p05	2027,99	75	1	10	140
p06	836,37	75	1	10	140
p07	826,14	100	4	2	200
p08	2034,15	100	5	5	200
p09	826,14	100	1	8	200
p10	1595,84	100	4	5	200
p11	779,29	126	4	5	235
p12	1195,88	163	3	5	140
p13	3511,62	417	9	7	2000
p14	954,81	20	2	4	20
p15	1862,63	38	2	4	30
p16	2875,24	56	2	4	40
p17	1597,75	40	4	4	20
p18	3147,24	76	4	4	30
p19	4834,34	112	4	4	40
p20	8367,4	184	4	4	60
p21	2184,04	60	6	4	20
p22	4271,11	114	6	4	30
p23	6602,59	168	6	4	40
p24	3687,46	51	3	6	20
p25	3777,15	51	3	6	20
p26	3795,33	51	3	6	20
p27	21956,46	102	6	6	20
p28	22934,71	102	6	6	20
p29	22909,36	102	6	6	20
p30	75016,58	153	9	6	20
p31	78179,89	153	9	6	20
p32	80479,2	153	9	6	20

5.3 Testes e Considerações sobre as Heurísticas Construtivas

Foram executados seis testes referentes às heurísticas construtivas implementadas apresentadas no Capítulo 2.

A Tabela 3 mostra para cada teste realizado a média dos desvios em relação ao valor da melhor solução conhecida e a média dos tempos computacionais. Cada teste é composto por uma heurística construtiva e por um tipo de designação. Por exemplo, o teste 1 utiliza a heurística de Beltrami e Bodin (1974) para a construção inicial das rotas e designação aleatória da combinação de dias de visitas para cada cliente. O teste 2 utiliza a heurística de Beltrami e Bodin (1974) para a construção inicial das rotas e designação da combinação de dias de visitas dos clientes através do PL.

Tabela 3 – Resumo dos resultados dos testes realizados para as heurísticas construtivas implementadas

		Testes					
		1	2	3	4	5	6
Heurística	Beltrami e Bodin (1974)	X	X				
	Cordeau et al. (2001)			X	X		
	Cordeau et al. (2001) Modificado					X	X
Tipo de Designação	Determinação aleatória da combinação de vistas inicial	X		X		X	
	Determinação da combinação de visitas inicial através do PL		X		X		X
	Média das somas dos excessos de capacidade	136,536	3,243	128,024	17,487	172,195	70,902
	Média das somas dos excessos de duração das rotas	63,676	118,890	135,037	136,591	481,021	485,162
	Desvio médio (%)	23,671	24,291	74,922	73,539	116,994	112,286
	CPU médio (seg)	0,009	0,177	0,000	0,169	0,002	0,171

A partir da Tabela 3 pode-se concluir que:

- A heurística construtiva baseada em (Beltrami e Bodin, 1974) foi a que apresentou os melhores resultados tanto no que se refere à qualidade das soluções geradas quanto à violação das restrições;
- Não é possível observar dominância quanto à determinação da combinação de dias de visitas aleatória e através do PL. Como esperado, a solução gerada pelo PL tem um baixo excesso de capacidade às custas de um alto excesso de duração de rota.

- a heurística construtiva de Cordeau et al. (2001) gera soluções melhores que a heurística construtiva modificada, baseada em (Cordeau et al., 2001), tanto em relação à qualidade como em relação aos excessos de capacidade e de duração das rotas;
- os tempos computacionais gastos nos diversos testes são muito semelhantes, notando-se que quando a combinação de dias de visitas é determinada pelo PL o tempo computacional sofre um pequeno aumento. Porém, mesmo com este acréscimo, a média dos tempos não chega a 1 segundo de processamento.

5.4 Testes e Considerações sobre os Algoritmos de Busca Local

Os resultados apresentados nesta seção foram obtidos iniciando os parâmetros utilizados nos algoritmos do capítulo 3 com os valores: $\alpha = \beta = w_1 = w_2 = w_3 = 1$; $q = 1$; $\varepsilon = 0.01$; $\delta = 0.5$; $w = 0,3$. Os valores de α , β e δ foram extraídos de Cordeau et al. (2001) e os valores de w, w_1, w_2, w_3 e q de Løkketangen e Glover (1998). O valor de ε , utilizado no algoritmo de busca local de Løkketangen e Glover (1998), foi determinado experimentalmente. A condição de parada, determinada experimentalmente, é atingida quando o número de iterações realizadas é $10n$ ou quando um ótimo local é atingido antes de $10n$ iterações.

A solução inicial é gerada pela heurística construtiva baseada em (Beltrami e Bodin, 1974) com a determinação da combinação de dias de visitas inicial através do PL.

Os movimentos Or-OPT intra e inter rotas e o movimento Troca-Cross foram testados variando k de 1 (um) até 4 (quatro). A melhor média dos resultados foi obtida utilizando-se os movimentos Or-OPT intra-rota com $k = 4$ e Or-OPT inter-rotas com $k = 1$, e desta forma, os testes que seguem nesse capítulo usam esta configuração de movimentos de melhoria. Note que $k = 1$ no movimento inter-rota corresponde a um movimento de inserção de um cliente de uma rota em outra rota como nos algoritmos de Cordeau et al. (1997, 2001)

A Tabela 4 apresenta o resumo dos testes realizados com os algoritmos de busca local baseados em (Cordeau et al., 2001). Nela, são apresentadas as médias dos desvios em relação ao valor da melhor solução conhecida e a média dos tempos computacionais. Também é apresentado o número de instâncias para as quais não foi possível obter uma solução factível.

Tabela 4 – Resumo dos testes realizados para os algoritmos de busca local baseados em (Cordeau et al.2001)

	Vizinhança N (original de Cordeau et al.)	Vizinhança N_2 com movimentos Or-OPT intra e inter rotas
Desvio médio (%)	9,04	7,71
CPU médio (seg)	25,50	24,53
Instâncias Infactíveis	2	2

Estes resultados mostram que o uso de movimentos de melhoria Or-OPT intra e inter rotas acarreta melhora na qualidade das soluções.

O tempo computacional gasto nas implementações que utilizam movimentos de melhoria é muito semelhante ao tempo gasto nas implementações que não os utilizam, porém o primeiro é um pouco menor pelo fato de algumas instâncias atingirem um ótimo local antes de $10n$ iterações.

A Tabela 5 mostra os resultados da busca local baseada em (Løkketangen e Glover, 1998) para as quatro regras de avaliação de movimentos com movimentos de melhoria Or-OPT intra e inter rotas. Observa-se que nenhuma regra é dominante.

Tabela 5 – Resumo dos testes realizados para o algoritmo de busca local baseado em (Løkketangen e Glover) com movimentos de melhoria Or-OPT intra e inter rotas.

	Primeira Regra	Segunda Regra usando normalização 1	Segunda Regra usando normalização 2	Terceira Regra	Quarta Regra
Desvio médio(%)	7,71	7,86	7,25	7,69	7,63
CPU médio (seg)	29,20	31,33	32,45	31,02	31,51
Instâncias Infactíveis	2	1	1	2	1

5.5 Testes e Considerações Referentes aos Algoritmos de Busca Tabu

Nesta seção são apresentados resultados do algoritmo de busca tabu proposto e os resultados obtidos na implementação do algoritmo de busca tabu de Cordeau et al (2001). Ao final, são comparados os resultados obtidos com resultados apresentados na literatura.

5.5.1 Testes referentes ao algoritmo de busca tabu implementado

Nesta subseção são apresentados testes realizados com o algoritmo de busca tabu implementado, que utilizam os seguintes parâmetros: $\alpha = \beta = w_1 = w_2 = w_3 = 1$; $q = 1$; $\varepsilon = 0.01$ e $w = 0,3$. Os valores de α e β foram extraídos de Cordeau et al. (2001) e os valores de w, w_1, w_2, w_3 e q de Løkketangen e Glover (1998). O valor de ε , utilizado no algoritmo de busca local de Løkketangen e Glover (1998), foi determinado experimentalmente. O valor do parâmetro θ , que determina a duração tabu, e δ , que atualiza os parâmetros de penalização, e a condição de parada são apresentados em cada teste realizado.

Os primeiros testes executados para o algoritmo de busca tabu, apresentado na Figura 13 do capítulo 4, visam analisar qual a melhor estrutura de memória de curto prazo, isto é, qual a regra de proibição de movimentos que apresenta os melhores resultados.

Nas Tabelas 6 e 7 são apresentadas as médias dos desvios dos resultados obtidos em relação a melhor solução conhecida para as quatro regras de avaliação de movimentos e a média dos tempos computacionais gastos. Os resultados da tabela 6 referem-se ao algoritmo de busca tabu com memória de curto prazo baseada na estrutura da *lista tabu 1*, utilizando $\theta = 7,5 \log_{10}^n$, assim como em Cordeau et al.(2001), e $\delta = 0.5$, determinado experimentalmente. Os resultados da tabela 7 referem-se aos resultados da busca tabu com memória de curto prazo baseada na estrutura da *lista tabu 2*, utilizando $\theta = \frac{2n}{5}$. e $\delta = 0.6$, ambos determinados experimentalmente. A condição de parada utilizada é atingida após a execução de 10.000 iterações.

Tabela 6 – Resumo dos testes realizados com o algoritmo de busca tabu utilizando memória de curto prazo baseada na estrutura da lista tabu 1.

	Primeira Regra	Segunda Regra usando normalização 1	Segunda Regra usando normalização 2	Terceira Regra	Quarta Regra
Desvio médio (%) – instâncias pr01 a pr10	3,97	2,41	3,16	3,32	3,53
Desvio médio(%) – instâncias p02 a p32	3,59	5,06	5,55	3,71	5,98
Desvio médio global(%)	3,68	4,42	4,96	3,62	5,38
Média global CPU (seg)	135,15	141,08	140,81	139,39	140,51

Tabela 7 – Resumo dos testes realizados com o algoritmo de busca tabu utilizando memória de curto prazo baseada na estrutura da lista tabu 2.

	Primeira Regra	Segunda Regra usando normalização 1	Segunda Regra usando normalização 2	Terceira Regra	Quarta Regra
Desvio médio (%) – instâncias pr01 a pr10	1,27	1,70	2,15	1,24	1,70
Desvio médio(%) – instâncias p02 a p32	2,18	4,62	4,18	2,15	4,62
Desvio médio global (%)	1,96	3,91	3,68	1,93	3,91
Média global CPU (seg)	205,75	330,80	298,84	440,21	387,40

Comparando-se os resultados das tabelas 6 e 7 verifica-se que a busca tabu com memória de curto prazo baseada na estrutura da *lista tabu 2* apresenta desvios médios globais menores que os desvios médios globais quando a busca tabu de curto prazo é baseada na estrutura da *lista tabu 1*. Uma explicação plausível é que a regra de proibição que define a estrutura *lista tabu 2* consiste em não permitir que clientes sejam associados a combinações de dias de visitas a que recentemente estiveram associados. Com isso, mais regiões do espaço de busca são exploradas, pois aumentam as possibilidades dos clientes se associarem as suas diferentes combinações de visitas. Esta regra de proibição não leva em consideração as rotas de visitas aos clientes pertencem, e estas são determinadas, para cada combinação de dias de visitas de cada cliente, pelo procedimento de busca local baseado em movimentos de melhoria intra e inter rotas em cada dia do horizonte de planejamento.

A regra de proibição que define a estrutura *lista tabu 1* não permite que um cliente retorne a uma rota de um dia do horizonte de tempo a que recentemente pertenceu. Desta forma, o movimento de troca de combinação de dias de visitas permite que um cliente mude para uma nova combinação de dias de visitas, desde que seja inserido em uma rota em que não esteja proibido de pertencer. Portanto, esta regra de proibição restringe a troca de combinação de dias de visita, o que acarreta em pior qualidade. Como a *lista tabu 2* não proíbe inserção em rotas, a busca local trabalha muito mais, como refletido no tempo computacional em relação ao da *lista tabu 1*.

Quanto às regras de avaliação de movimentos, analisando-se os resultados da Tabela 7 é possível perceber que a primeira e a terceira regras são as que apresentam o melhor desempenho, no que se refere à qualidade dos resultados. Porém, a terceira regra exige um tempo computacional muito maior que a primeira regra para obter praticamente o mesmo desvio médio.

Desta forma, os próximos testes realizados que são apresentados utilizam como regra de proibição aquela que define a estrutura da *lista tabu 2*, sendo que estes testes são aplicados somente à primeira regra de avaliação de movimentos. Ao final desta subseção, o algoritmo completo de busca tabu é então testado para todos as regras de avaliação de movimentos.

O próximo passo no desenvolvimento do algoritmo de busca tabu é a inserção de memória de longo prazo. Nesta etapa, cada estratégia descrita no Capítulo 4 foi testada individualmente, e a seguir são apresentados os resultados e considerações somente daquelas que mostraram bom desempenho.

As Tabelas 8 e 9 apresentam os resultados obtidos pelo algoritmo de busca tabu com diversificação 1. Nestes testes $d = 0.5$, parâmetro responsável por controlar o grau de diversificação, é determinado experimentalmente. A cada $70n$ iterações sem melhora da solução incumbente 6 iterações de diversificação são executadas, sendo estes valores também determinados experimentalmente.

Tabela 8 - Resultados referentes a busca tabu com diversificação 1 para instâncias com restrições de capacidade e de duração.

Instância	Condição de parada atingida após a execução de 5000 iterações		Condição de parada atingida após a execução de 10000 iterações		Condição de parada atingida após a execução de 15000 iterações		
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	
pr01	2209,02	0,00	6,75	0,00	13,65	0,00	23,39
pr02	3799,28	1,12	49,76	0,90	86,89	0,72	161,97
pr03	5218,13	1,10	46,86	1,10	103,33	1,10	187,86
pr04	6012,79	0,39	167,76	0,39	331,62	0,37	505,35
pr05	6769,8	3,42	318,30	2,50	559,90	1,13	841,14
pr06	8422,64	3,45	508,54	2,53	900,87	2,41	1.272
pr07	4997,41	0,33	51,74	0,33	92,18	0,33	136,77
pr08	7094,52	2,40	176,81	2,40	326,64	2,03	519,56
pr09	10370,45	1,79	316,77	0,30	640,78	0,16	994,27
pr10	13370,04	2,17	1182,48	0,98	2.176	0,91	2.994
Média (%)		1,61	282,58	1,14	523,27	0,92	763,74

Tabela 9 - Resultados referentes a busca tabu com diversificação 1 para instâncias com restrição de capacidade.

Instância		Condição de parada atingida após a execução de 5000 iterações		Condição de parada atingida após a execução de 10000 iterações		Condição de parada atingida após a execução de 15000 iterações	
		Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)
p02	1322,87	2,11	13,03	1,64	25,15	1,05	36,42
p03	524,61	0,58	4,46	0,00	9,23	0,00	15,41
p04	835,43	13,41	17,77	9,74	37,96	7,62	58,85
p05	2027,99	3,32	70,16	3,32	232,49	1,23	278,32
p06	836,37	2,59	20,62	2,05	41,97	2,05	63,22
p07	826,14	1,06	12,42	1,06	24,88	0,50	36,41
p08	2034,15	2,55	31,96	2,55	68,52	0,71	97,25
p09	826,14	1,90	17,25	1,31	36,35	3,11	54,21
p10	1595,84	3,34	32,25	3,11	65,23	3,11	96,63
p11	779,29	2,41	92,85	2,41	223,32	1,00	323,04
p12	1195,88	3,92	68,04	3,92	150,92	3,54	193,12
p13	3511,62	7,42	1.891	7,86	4.092	4,94	6.022
p14	954,81	0,00	1,84	0,00	3,52	0,00	5,42
p15	1862,63	1,00	5,65	1,00	10,89	1,00	16,01
p16	2875,24	2,30	6,52	0,07	12,13	0,07	17,51
p17	1597,75	2,89	20,61	1,50	44,74	0,00	57,87
p18	3147,24	1,58	55,64	1,57	119,50	1,55	152,07
p19	4834,34	1,47	55,97	1,08	119,29	0,94	164,62
p20	8367,4	2,72	62,18	2,53	117,95	2,53	148,90
p21	2184,04	1,21	34,91	1,20	62,28	1,04	90,36
p22	4271,11	2,87	72,21	2,44	140,75	2,23	185,31
p23	6602,59	3,38	177,36	3,25	337,49	-0,13	435,76
p24	3687,46	1,94	9,66	1,94	18,96	1,94	21,27
p25	3777,15	0,37	11,27	0,26	22,83	0,26	30,57
p26	3795,33	1,71	11,50	1,20	23,25	1,02	35,59
p27	21956,46	0,94	59,83	0,87	124,15	0,71	149,39
p28	22934,71	-2,10	56,78	-2,10	112,10	-2,18	155,68
p29	22909,36	-0,85	79,09	-0,94	173,73	-0,94	234,71
p30	75016,58	1,53	161,65	1,53	330,31	1,51	400,88
p31	78179,89	-0,63	136,41	-0,92	392,86	-0,92	496,26
p32	80479,2	2,26	217,21	1,66	442,23	1,14	610,38
Média (%)		2,23	113,17	1,84	245,73	1,24	344,65

Os resultados das Tabelas 8 e 9 mostram que o aumento do número de iterações executadas pelo algoritmo de busca tabu implica uma melhora significativa nos resultados obtidos. Comparando-se ainda estes resultados com os resultados da primeira regra de avaliação de movimentos apresentados na Tabela 7, verifica-se melhora nos resultados obtidos com o uso da diversificação 1 com aumento no tempo computacional. Nesta estratégia de diversificação, combinações de dias de visitas freqüentemente associadas aos clientes são penalizadas, e cada vez que esta é aplicada muda-se a combinação de dias de visitas do cliente que gera o menor aumento no custo da solução, permitindo à busca tabu explorar espaços de busca antes pouco explorados, garantindo a melhora dos resultados.

Outra estratégia que utiliza memória de longo prazo implementada e que apresentou resultados promissores é a diversificação 2. Os resultados obtidos são apresentados nas Tabelas 10 e 11, e consideram o parâmetro responsável por controlar o grau de diversidade $d = 0.5$ e que cada reinício é executado a cada $70n$ iterações sem melhora da solução incumbente, valores estes determinados experimentalmente.

Tabela 10 - Resultados referentes a busca tabu com diversificação 2 para instâncias com restrições de capacidade e de duração.

Instância		Condição de parada atingida após a execução de 5000 iterações		Condição de parada atingida após a execução de 10000 iterações		Condição de parada atingida após a execução de 15000 iterações	
		Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)
pr01	2209,02	0,00	6,68	0,00	13,66	0,00	20,35
pr02	3799,28	1,12	48,90	0,90	92,76	0,52	130,67
pr03	5218,13	1,10	43,67	1,10	121,53	1,10	191,85
pr04	6012,79	0,39	161,14	0,39	338,92	0,37	504,99
pr05	6769,8	3,42	288,45	2,50	573,91	1,13	839,07
pr06	8422,64	3,45	506,92	2,53	923,16	1,46	1273,80
pr07	4997,41	0,33	43,02	0,33	77,86	0,33	111,65
pr08	7094,52	2,40	164,00	2,40	334,98	0,86	530,07
pr09	10370,45	1,79	299,24	0,30	653,67	0,16	1013,89
pr10	13370,04	2,17	1183,89	0,98	2196,75	-0,04	3046,32
Média (%)		1,62	274,59	1,14	532,72	0,59	766,26

Tabela 11 - Resultados referentes a busca tabu com diversificação 2 para instâncias com restrição de capacidade.

Instância		Condição de parada atingida após a execução de 5000 iterações		Condição de parada atingida após a execução de 10000 iterações		Condição de parada atingida após a execução de 15000 iterações	
		Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)
p02	1322,87	2,11	13,40	1,64	25,64	1,64	34,66
p03	524,61	0,58	4,36	0,00	8,99	0,00	13,26
p04	835,43	13,41	17,89	9,74	38,86	7,62	50,96
p05	2027,99	3,32	72,45	3,32	233,22	2,11	244,67
p06	836,37	2,59	20,55	2,05	41,08	2,05	62,86
p07	826,14	1,06	12,45	1,06	31,85	1,06	52,38
p08	2034,15	2,55	33,10	2,55	65,20	0,50	96,93
p09	826,14	1,90	17,53	1,31	35,58	0,71	53,62
p10	1595,84	3,34	33,04	3,11	64,36	3,11	95,04
p11	779,29	2,41	107,42	2,41	214,41	1,00	305,87
p12	1195,88	3,92	69,27	3,92	143,65	3,92	279,29
p13	3511,62	7,42	1966,37	7,42	3997,72	4,94	5947,19
p14	954,81	0,90	3,04	0,90	5,15	0,00	6,82
p15	1862,63	1,00	4,94	1,00	9,95	1,00	15,49
p16	2875,24	2,70	7,05	2,70	16,74	0,13	27,35
p17	1597,75	2,89	17,52	2,89	23,15	2,89	28,84
p18	3147,24	1,58	55,35	1,58	90,03	1,58	110,58
p19	4834,34	1,47	55,61	1,08	108,04	0,94	144,68
p20	8367,4	2,72	58,43	2,53	118,36	2,53	174,42
p21	2184,04	1,21	32,05	1,21	47,69	1,21	61,06
p22	4271,11	2,87	67,41	2,44	138,81	2,23	198,26
p23	6602,59	3,38	170,21	3,25	344,53	-0,13	485,58
p24	3687,46	1,94	9,80	1,94	18,95	1,94	27,89
p25	3777,15	0,37	11,49	0,37	20,95	0,37	30,23
p26	3795,33	1,71	10,95	1,71	19,93	1,02	28,65
p27	21956,46	0,94	44,33	0,87	123,00	0,71	163,47
p28	22934,71	-2,10	43,09	-2,10	103,18	-2,14	135,27
p29	22909,36	-0,85	80,54	-0,94	172,50	-0,94	214,98
p30	75016,58	1,53	164,12	1,53	313,64	1,53	450,21
p31	78179,89	-0,63	169,97	-0,92	345,49	-0,92	513,94
p32	80479,2	2,26	235,45	1,66	424,74	1,14	618,08
Média (%)		2,27	116,43	2,01	236,95	1,41	344,27

Analogamente à estratégia de diversificação 1, na estratégia de diversificação 2 quanto maior o número de iterações executado melhor a qualidade dos resultados obtidos, sem que haja um grande comprometimento do tempo computacional necessário para a obtenção destes. A cada reinício uma nova solução é construída baseada na freqüência com que cada cliente se mantém associado a cada uma de suas possíveis combinações de dias de visitas. Durante a construção da nova solução as associações cliente-combinação de dias de visitas mais freqüentes são penalizadas e cliente a cliente determina-se a combinação de dias de visitas que gera o menor aumento no valor da função objetivo. Desta forma, espaços de busca pouco ou nunca explorados passam a assim o ser pela busca tabu, o que garante a melhora nos resultados obtidos.

Comparando-se os resultados das Tabelas 8 e 10, que correspondem às instâncias com restrições de capacidade dos veículos e de duração das rotas, verifica-se que a estratégia de diversificação 2 apresenta o melhor desempenho, enquanto que se comparando os resultados das Tabelas 9 e 11, que correspondem às instâncias somente com restrições de capacidade dos veículos, verifica-se que a estratégia de diversificação 1 apresenta o melhor desempenho. Desta forma, não é possível estabelecer uma dominância entre estas estratégias de diversificação.

A última etapa da implementação do algoritmo de busca tabu consiste de testes que verificam o comportamento e o desempenho do algoritmo de busca tabu quando o mesmo utiliza estratégias baseadas em memória de longo prazo combinadas. Dos testes realizados, os que apresentam os melhores resultados são os que combinam estratégias de diversificação com o mecanismo de religação de caminho, e são apresentados nas Tabelas 12 e 13. As colunas identificadas por DIVERS-2 indicam a utilização da estratégia de diversificação 2 e as colunas identificadas por DIVERS-1 indicam a utilização da estratégia de diversificação 1. Nestes testes, considera-se como critério de parada a execução de 15000 iterações, sendo que as estratégias de diversificação são executadas a cada $70n$ iterações sem melhora da solução incumbente e consideram $d = 0.5$. A estratégia de diversificação 1, uma vez iniciada, é executada por 6 vezes consecutivas. O tamanho do conjunto de soluções de referência *Ref* utilizado no mecanismo de religação de caminho é igual a 10.

Tabela 12 - Resultados referentes a busca tabu com religação de caminho para instâncias com restrições de capacidade e de duração.

Instância		DIVERS-2		DIVERS-1	
		Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)
pr01	2209,02	0,00	98	0,00	39,99
pr02	3799,28	0,53	130	1,73	94,20
pr03	5218,13	0,64	346	1,24	219,11
pr04	6012,79	0,17	747,532	0,83	429,95
pr05	6769,8	0,00	1101,73	0,00	873,68
pr06	8422,64	1,17	2925,09	1,61	1.748,42
pr07	4997,41	0,30	107	0,17	98,77
pr08	7094,52	0,80	74	1,14	437,63
pr09	10370,45	0,57	1918,33	0,59	1.254,19
pr10	13370,04	-0,13	3202,66	1,35	3.041,16
Média (%)		0,41	1.065,14	0,87	823,71

Tabela 13 - Resultados referentes a busca tabu com religação de caminho para instâncias com restrição de capacidade.

Instância	DIVERS-2		DIVERS-1		
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	
p02	1322,87	0,86	50	1,43	41,72
p03	524,61	0,00	29	0,00	21,45
p04	835,43	6,03	44	5,66	36,77
p05	2027,99	0,77	163	2,20	110,65
p06	836,37	1,61	232	0,72	118,94
p07	826,14	0,70	71	0,41	55,72
p08	2034,15	0,55	255	0,59	169,95
p09	826,14	0,59	263	0,15	152,11
p10	1595,84	2,35	167,12	3,70	148,27
p11	779,29	1,16	419	2,64	274,93
p12	1195,88	1,17	331	3,18	234,39
p13	3511,62	5,35	7	4,94	5.002,53
p14	954,81	0,00	145	0,45	7,16
p15	1862,63	0,00	4537,88	0,00	95,92
p16	2875,24	0,00	23,75	0,00	596,68
p17	1597,75	0,00	56	0,00	25,54
p18	3147,24	1,49	132,25	1,49	69,52
p19	4834,34	0,85	2367,03	0,88	141,95
p20	8367,4	0,69	62	0,69	2.812,68
p21	2184,04	1,13	224	0,49	56,46
p22	4271,11	0,12	305	-0,01	164,26
p23	6602,59	-0,18	36	-0,05	319,18
p24	3687,46	1,50	41	2,01	34,58
p25	3777,15	0,34	44	0,24	35,88
p26	3795,33	1,02	220,75	1,02	35,63
p27	21956,46	0,47	163	0,65	154,30
p28	22934,71	-2,16	194	-2,23	157,42
p29	22909,36	-0,90	175,25	-0,83	175,25
p30	75016,58	0,93	1221,75	1,51	482,16
p31	78179,89	-0,65	1354,23	-0,92	481,62
p32	80479,2	-1,94	1694,59	1,14	552,41
Média (%)		0,77	484,71	1,04	411,81

A utilização do mecanismo de religação de caminho proporcionou melhora significativa nas soluções geradas pelo algoritmo de busca tabu, tanto utilizando a estratégia de diversificação 1 quanto utilizando a estratégia de diversificação 2. Nestes testes, a atualização do conjunto de soluções de referencia *Ref* é feita somente quando uma solução candidata tem valor de função objetivo menor que a melhor solução do conjunto (incumbente). Isto porque, os testes mostraram que a religação é efetiva somente neste caso, isto é, com trajetórias bi-direcionais entre a solução que entra e as demais do conjunto. A melhoria ao se religar uma boa solução que aumenta a distância entre as soluções do conjunto com as soluções do conjunto é marginal, e por isto foi descartada. Portanto, este tipo de religação caracteriza-se por um procedimento altamente intensificador.

Comparando-se os resultados das colunas DIVERS-1 e DIVERS-2 nas Tabelas 12 e 13, verifica-se que com a utilização do mecanismo de religação de caminho a estratégia de diversificação baseada em reinício apresenta desempenho médio melhor nos dois grupos de instâncias testadas. Para as instâncias com restrições de capacidade dos veículos e de duração das rotas, embora o tempo computacional médio apresentado em DIVERS-1 seja menor que a média do tempo computacional apresentado em DIVERS-2, a média dos desvios dos resultados obtidos em relação aos melhores resultados conhecidos apresentada em DIVERS-2 é significativamente menor que a média apresentada em DIVERS-1. A estratégia DIVERS-2 obteve resultados melhores em 7 das 10 instâncias, empatou em 2 e obteve resultado pior em uma instância. Para as instâncias somente com restrição de capacidade dos veículos a média dos tempos computacionais apresentados em DIVERS-1 e DIVERS-2 são muito semelhantes, porém em relação aos resultados, a estratégia DIVERS-2 apresenta média dos desvios dos resultados obtidos em relação aos melhores resultados conhecidos consideravelmente menor que a média apresentada em DIVERS-1. A estratégia DIVERS-2 obteve resultados melhores em 14 das 31 instâncias, empatou em 7 e obteve resultado pior em 10 instâncias.

Portanto, o algoritmo de busca tabu com mecanismo de religação de caminho e estratégia de diversificação baseada em reinício, que passamos a denominá-lo BT, na média, gera resultados melhores que o algoritmo de busca tabu com mecanismo de religação de caminho e estratégia de diversificação baseada em frequência de residência, e por este motivo os resultados gerados por este serão comparados com os resultados gerados por outros algoritmos apresentados na literatura.

Como mencionado no começo da subseção, o algoritmo BT é testado para as quatro regras de avaliação de movimentos, e o resumo dos resultados é apresentado na Tabela 14.

Tabela 14 – Resumo dos resultados obtidos através do algoritmo BT para as quatro regras de avaliação de movimentos.

	Primeira Regra	Segunda Regra usando normalização 1	Segunda Regra usando normalização 2	Terceira Regra	Quarta Regra
Desvio médio (%) – instâncias pr01 a pr10	0,41	0,79	0,93	0,62	1,23
Desvio médio(%) – instâncias p02 a p32	0,77	1,66	1,75	1,10	1,81
Desvio médio global (%)	0,68	1,45	1,55	0,99	1,67
Média global CPU (seg)	626,28	751,40	783,43	860,75	835,97

Assim como verificado nos testes iniciais, a primeira e a terceira regras de avaliação de movimentos são as que geram os melhores resultados, sendo que o tempo computacional médio global exigido pela terceira regra é consideravelmente maior que o exigido pela primeira regra. Desta forma, nas próximas subseções são considerados os resultados gerados pelo BT quando o mesmo utiliza a primeira regra de avaliação de movimentos.

5.5.2 Testes referentes ao algoritmo de busca tabu de Cordeau et. al. (2001)

O algoritmo de busca tabu apresentado em Cordeau et. al.(2001) foi implementado para que fosse possível verificar a qualidade dos resultados obtidos para as instâncias do PRPV. Assim como em Cordeau et. al.(1997), os parâmetros utilizados são $\alpha = \beta = 1$, $\delta = 0,5$, $\lambda = 0,015$ e $\theta = \lceil 7,5 \log_{10}^n \rceil$, onde $\lfloor x \rfloor$ é o inteiro mais próximo de x . Os resultados apresentados nas Tabelas 15 e 16 consideram como condição de parada a execução de 15000 iterações, sendo que os resultados referentes à Cordeau et. al.(1997) foram obtidos em uma Sun Sparcstation 10.

Tabela 15 – Resultados dos algoritmos de Cordeau et. al. (1997), BT e de Cordeau et. al.(2001) para as instâncias com restrições de capacidade e de duração.

Instância	Cordeau et. al. (1997)		BT		Cordeau et. al. (2001)		
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	
pr01	2209,02	1,14	231,60	0,00	98	4,08	5,98
pr02	3799,28	0,98	587,40	0,53	130	4,04	28,22
pr03	5218,13	1,14	1117,20	0,64	346	3,07	71,88
pr04	6012,79	1,00	1694,40	0,17	747,532	4,99	199,00
pr05	6769,8	0,00	2197,20	0,00	1101,73	3,26	296,78
pr06	8422,64	0,47	3058,20	1,17	2925,09	2,49	532,44
pr07	4997,41	0,07	577,80	0,30	107	3,45	25,33
pr08	7094,52	1,25	1443,00	0,80	74	4,62	165,66
pr09	10370,45	1,32	2739,00	0,57	1918,33	3,39	443,26
pr10	13370,04	1,94	4293,60	-0,13	3202,66	2,36	1.086,30
Média (%)		0,93	1793,94	0,41	1.065,14	3,57	285,48

Tabela 17 – Comparação entre algoritmos para o PRPV para as instâncias com restrições de capacidade e de duração.

Instância	Cordeau et. al. (1997)		BT		BT*	
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	
pr01	2209,02	1,14	231,60	0,00	98	0,00
pr02	3799,28	0,98	587,40	0,53	130	0,53
pr03	5218,13	1,14	1117,20	0,64	346	0,55
pr04	6012,79	1,00	1694,40	0,17	747,532	-0,03
pr05	6769,8	0,00	2197,20	0,00	1101,73	0,00
pr06	8422,64	0,47	3058,20	1,17	2925,09	1,12
pr07	4997,41	0,07	577,80	0,30	107	0,17
pr08	7094,52	1,25	1443,00	0,80	74	0,66
pr09	10370,45	1,32	2739,00	0,57	1918,33	-0,56
pr10	13370,04	1,94	4293,60	-0,13	3202,66	-0,13
Média (%)		0,93	1793,94	0,41	1.065,14	0,23

Comparando-se os resultados das colunas Cordeau et. al. (1997) e BT, fica clara a dominância do algoritmo BT sobre o algoritmo de Cordeau et. al.(1997), sendo que o primeiro obtém resultados melhores em 7 das 10 instâncias, empata em 1 e perde em 2 instâncias. Para este conjunto de instâncias, Alegre et. al. (2004) não apresentam resultados.

Tabela 16 – Resultados dos algoritmos de Cordeau et. al. (1997), BT e de Cordeau et. al.(2001) para as instâncias somente com restrição de capacidade.

Instância	Cordeau et. al. (1997)		BT		Cordeau et. al. (2001)		
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	
p02	1322,87	0,55	243,60	0,86	50	1,92	10,84
p03	524,61	0,00	223,80	0,00	29	0,59	6,38
p04	835,43	0,30	311,40	6,03	44	2,67	11,48
p05	2027,99	1,65	448,80	0,77	163	2,26	46,19
p06	836,37	0,47	470,40	1,61	232	2,45	33,11
p07	826,14	0,39	457,80	0,70	71	0,93	14,69
p08	2034,15	1,02	642,00	0,55	255	1,51	55,99
p09	826,14	0,40	601,80	0,59	263	0,87	29,55
p10	1595,84	2,14	580,80	2,35	167,12	5,63	41,80
p11	779,29	4,91	850,20	1,16	419	4,15	57,20
p12	1195,88	3,65	1102,20	1,17	331	2,85	65,63
p13	3511,62	2,60	3598,80	5,35	7	5,04	789,56
p14	954,81	0,00	69,00	0,00	145	0,40	2,87
p15	1862,63	0,00	154,80	0,00	4537,88	4,02	5,50
p16	2875,24	0,00	256,80	0,00	23,75	7,76	9,25
p17	1597,75	0,00	180,60	0,00	56	0,00	10,44
p18	3147,24	0,38	387,60	1,49	132,25	1,09	23,73
p19	4834,34	1,41	714,00	0,85	2367,03	6,14	37,01
p20	8367,4	0,00	1406,40	0,69	62	13,57	87,36
p21	2184,04	0,00	312,00	1,13	224	0,00	17,89
p22	4271,11	0,84	687,60	0,12	305	1,35	54,40
p23	6602,59	0,27	1174,80	-0,18	36	7,28	94,78
p24	3687,46	0,45	255,60	1,50	41	1,38	16,76
p25	3777,15	0,11	260,40	0,34	44	0,11	16,79
p26	3795,33	0,00	255,60	1,02	220,75	0,00	16,92
p27	21956,46	4,83	678,60	0,47	163	5,16	55,99
p28	22934,71	-1,59	667,80	-2,16	194	-2,15	65,95
p29	22909,36	4,82	673,20	-0,90	175,25	3,47	65,81
p30	75016,58	2,88	1243,20	0,93	1221,75	5,09	158,85
p31	78179,89	1,54	1218,00	-0,65	1354,23	2,05	176,15
p32	80479,2	0,53	1237,20	-1,94	1694,59	-1,35	185,10
Média (%)		1,11	689,19	0,77	484,71	2,78	73,03

Comparando-se os resultados apresentados nesta tabela, verifica-se que o desempenho do algoritmo de busca tabu de Cordeau et. *al.*(2001) é significativamente inferior ao desempenho dos algoritmos BT e de Cordeau et. *al.*(1997).

O mau desempenho do algoritmo de Cordeau et. *al.*(2001) é explicado pela regra de proibição de movimentos utilizada e por este não fazer uso do algoritmo GENI, um procedimento sofisticado para otimização de rotas, que usa movimentos 4-Opt (Gendreau et al. 1992), para inserção e remoção de clientes nas soluções.

Quanto aos tempos computacionais, comparando-se as colunas BT e Cordeau et. *al.*(2001), verifica-se que a média da segunda é significativamente menor que a média da primeira. Isto ocorre porque o algoritmo de busca tabu de Cordeau et. *al.*(2001) não utiliza busca em vizinhança como procedimento de melhoria, tornando-o mais rápido.

Em relação aos tempos computacionais de Cordeau et. *al.*(1997) nenhuma comparação é feita pelo fato dos testes terem sido realizados em máquinas com capacidade de processamento diferentes.

5.5.3 Comparação entre Algoritmos para o PRPV

Nesta subseção são apresentados os resultados obtidos com o algoritmo desenvolvido e implementado BT, com o algoritmo de busca tabu de Cordeau et. *al.*(1997) e com o algoritmo de *scatter search* de Alegre et. *al.*(2004) para instâncias do PRPV, presentes nas Tabelas 17 e 18. Nestas tabelas, também são apresentados os melhores resultados obtidos ao longo de todos os testes realizados com o algoritmo de busca tabu implementado, identificados nas colunas denotadas por BT*.

Os resultados apresentados nas colunas referentes à Alegre et. *al.*(2004) foram obtidos através de testes executados em um PC Pentium III, 600Mhz.

Tabela 18 – Comparação entre algoritmos para o PRPV para as instâncias com restrição de capacidade.

Instância	Cordeau et. al. (1997)		BT		BT*	Alegre et.al. (2004)		
	Desvio (%)	CPU (seg)	Desvio (%)	CPU (seg)	Desvio (%)	Desvio (seg)	CPU (seg)	
p02	1322,87	0,55	243,60	0,86	50	0,55	0,86	494,00
p03	524,61	0,00	223,80	0,00	29	0,00	2,43	45,00
p04	835,43	0,30	311,40	6,03	44	3,95	1,26	1426,00
p05	2027,99	1,65	448,80	0,77	163	0,77	0,78	1280,00
p06	836,37	0,47	470,40	1,61	232	0,42	0,45	1797,00
p07	826,14	0,39	457,80	0,70	71	0,47	0,42	199,00
p08	2034,15	1,02	642,00	0,55	255	0,42	0,89	3584,00
p09	826,14	0,40	601,80	0,59	263	0,15	0,42	970,00
p10	1595,84	2,14	580,80	2,35	167,12	1,13	1,59	9467,00
p11	779,29	4,91	850,20	1,16	419	1,00	0,37	6492,00
p12	1195,88	3,65	1102,20	1,17	331	1,17	2,93	515,00
p13	3511,62	2,60	3598,80	5,35	7	4,94	-	-
p14	954,81	0,00	69,00	0,00	145	0,00	0,00	5,00
p15	1862,63	0,00	154,80	0,00	4537,88	0,00	0,00	1,00
p16	2875,24	0,00	256,80	0,00	23,75	0,00	0,00	2,00
p17	1597,75	0,00	180,60	0,00	56	-1,41	0,00	96,00
p18	3147,24	0,38	387,60	1,49	132,25	1,49	0,31	401,00
p19	4834,34	1,41	714,00	0,85	2367,03	0,85	0,25	20,00
p20	8367,4	0,00	1406,40	0,69	62	0,69	0,53	60,00
p21	2184,04	0,00	312,00	1,13	224	0,37	-0,48	373,00
p22	4271,11	0,84	687,60	0,12	305	-0,05	1,39	528,00
p23	6602,59	0,27	1174,80	-0,18	36	-0,35	3,19	42,09
p24	3687,46	0,45	255,60	1,50	41	1,50	0,39	114,31
p25	3777,15	0,11	260,40	0,34	44	0,20	0,11	69,00
p26	3795,33	0,00	255,60	1,02	220,75	1,02	0,00	7,53
p27	21956,46	4,83	678,60	0,47	163	0,47	2,75	219,00
p28	22934,71	-1,59	667,80	-2,16	194	-2,26	-1,62	435,00
p29	22909,36	4,82	673,20	-0,90	175,25	-0,94	3,68	19,00
p30	75016,58	2,88	1243,20	0,93	1221,75	-0,05	2,37	19712,00
p31	78179,89	1,54	1218,00	-0,65	1354,23	-1,21	-0,30	7650,00
p32	80479,2	0,53	1237,20	-1,94	1694,59	-1,98	0,72	8316,00
Média (%)		1,11	689,19	0,77	484,71	0,38	0,85	2144,63

Os resultados apresentados nas colunas Cordeau et. *al.*(1997) e BT mostram que na média o segundo obtém resultados de melhor qualidade que o primeiro. Comparando-se instância a instância, o algoritmo BT obteve resultados melhores em 13 das 31 instâncias, empatou em 5 e perdeu em 13.

Para todas as instâncias testadas, é conhecida a representação da melhor solução obtida por Cordeau et. *al.* (1997). Desta forma foi possível compará-las com as representações das soluções obtidas pelo algoritmo BT, e verificou-se que nas instâncias onde ocorreu empate as representações são idênticas, o que significa que a busca local com movimentos de melhoria intra e inter rotas apresentam o comportamento esperado. Contudo, nas instâncias para as quais não ocorreram empates, as representações das soluções são distintas no que se refere às combinações de dias de clientes, e, conseqüentemente, as rotas de cada dia do horizonte de tempo assim o são também. Para um grupo de 5 instâncias que não empataram, escolhidas aleatoriamente, observou-se que pelo menos 20% dos clientes possuem combinações de dias de visitas distintas.

Em relação aos resultados obtidos pelo algoritmo de *scatter search* apresentado em Alegre et. *al.* (2004), sendo que o resultado da instância 13 não é informado, o algoritmo BT mostra-se competitivo, sendo que este obteve resultados melhores em 12 das 30 instâncias comparadas, empata em 6 e perde em 12 instâncias.

Quanto aos tempos computacionais, nenhuma conclusão pode ser realizada, pois os testes referentes a cada um dos métodos de solução do PRPV apresentados nas Tabelas 17 e 18 foram executados em máquinas distintas, que possuem capacidade de processamento também distintas.

6. CONCLUSÕES

Neste trabalho foi estudado o problema de roteamento periódico de veículos (PRPV) com restrições de capacidade dos veículos e de duração das rotas por estes percorridas.

O método de solução desenvolvido e implementado é uma busca tabu, que faz uso de memórias de curto e longo prazo. A motivação para a utilização deste método vem do fato deste ser usado com sucesso em uma grande variedade de problemas de otimização e do fato da literatura sobre implementações de busca tabu para o PRPV ser muito escassa.

A implementação do algoritmo de busca tabu foi realizada em etapas, sendo que a solução de partida é obtida através de um método de heurística construtiva. Em cada etapa sempre se avaliou a qualidade das soluções geradas em função do tempo computacional gasto.

A etapa inicial consistiu da implementação de várias heurísticas construtivas, para que fosse escolhida aquela que gerasse a melhor solução inicial, isto é, a que violasse menos as restrições de duração das rotas e de capacidade dos veículos. Nesta etapa conclui-se que a melhor heurística construtiva a ser utilizada é a baseada em Beltrami e Bodin (1974).

A segunda etapa consistiu em se implementar algoritmos de busca local, utilizados como base do algoritmo de busca tabu. A vizinhança utilizada é determinada por movimentos de troca de combinação de dias de visitas dos clientes e de movimentos de troca de clientes entre rotas de um mesmo dia de visita. Foram testadas 4 regras de avaliação dos movimentos da vizinhança, também testadas no algoritmo de busca tabu, além de se testar a introdução de movimentos de melhoria intra e inter rotas, cujos resultados mostram serem fundamentais.

A terceira etapa foi a definição efetiva do algoritmo de busca tabu, utilizando inicialmente somente memória de curto prazo, com o fim de se determinar a melhor regra proibitiva. Os testes mostraram que a melhor regra proibitiva é a que define a estrutura de matriz bidimensional, onde clientes ficam impossibilitados de voltar a determinadas combinações de dias de visitas por um número de iterações da busca tabu, sendo que este número foi determinado experimentalmente. A regra proibitiva só é violada no caso do critério de aspiração ser satisfeito, isto é, a nova solução ser melhor que a solução incumbente. Nestes testes, a regra 1 de avaliação de movimentos apresentou os melhores resultados.

A quarta e última etapa foi a implementação de estratégias de intensificação e de diversificação: diversificação baseada em frequência de residência, ao longo da busca, e como reinício, penalizando movimentos realizados com muita frequência ao longo da busca;

intensificação baseada em reinício, armazenando boas soluções em um conjunto de elite e reiniciando a busca com as soluções pertencentes a este conjunto; estratégia de diversificação e intensificação baseadas em Rochat e Taillard(1995), onde combinações de dias de visitas de boas soluções são armazenadas para que novas soluções de partida sejam construídas a partir destas boas combinações armazenadas, e estratégia de religação de caminho, onde trajetórias são traçadas entre soluções armazenadas em um conjunto de soluções de referência. Vários testes foram realizados combinando as diversas estratégias implementadas e executando-as separadamente, além de testes variando os parâmetros envolvidos, e o teste que apresentou os melhores resultados combina as estratégias de diversificação baseada em reinício e de religação de caminho.

Também foi implementado o algoritmo de busca tabu de Cordeau et al. (2001), observando através dos resultados gerados que este algoritmo perdeu sua maior eficiência devido a regra de proibição de movimentos utilizada juntamente com o fato de ter deixado de fazer uso do algoritmo GENI para inserção e remoção de clientes nas rotas.

Os resultados gerados pela melhor configuração do algoritmo de busca tabu implementado foram comparados com os resultados de Cordeau et al. (1997, 2001) mostrando-se mais eficiente que ambos, principalmente nas instâncias com restrições de duração das rotas e de capacidade dos veículos. Quando estes resultados são comparados com os resultados apresentados em Alegre et al.(2004) verifica-se que ambos os algoritmos são competitivos.

É possível afirmar que o uso da busca tabu como ferramenta de resolução do PRPV com restrições de duração das rotas e de capacidade dos veículos é bastante eficiente e proporciona muitos bons resultados.

Como extensões deste trabalho podemos citar a geração de novas instâncias teste e possíveis melhorias das memórias de curto e de longo prazos utilizadas, a implementação de outras metaheurísticas, por exemplo *scatter search*, como métodos de solução do PRPV, além de estender o PRPV para o problema de roteamento periódico de veículos com múltiplos depósitos (PRPVMD).

REFERÊNCIAS BIBLIOGRÁFICAS

- Alegre J., Laguna M. e Pacheco J. (2004). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts, submetido para publicação em março de 2004, <http://leeds-faculty.colorado.edu/laguna/publications.htm>.
- Angelelli E. e Speranza M.G. (2002). The application of a vehicle routing model to a waste-collection problem: two case studies, *Journal of the Operational Research Society*, 53, 944-952.
- Baptista S. Oliveira R.C. e Zúquete E. (2002). A period vehicle routing case study, *European Journal of Operational Research*, 139, 220-229.
- Beltrami E.L. e Bodin L.D. (1974). Networks and vehicle routing for municipal waste collection, *Networks*, 4, 65-94.
- Blakeley F., Bozkaya B., Cao B. Hall W. e Knolmayer J. (2003). Optimizing periodic maintenance operations for Schindler elevator corporation, *Interfaces*, 33, 76-79.
- Brodeur M.B. Cordeau J-F, Laporte G. e Lasary A. (1998). Scheduling Linen Deliveries in a Large Hospital, *Journal of the Operational Research Society*, 49, 777-780.
- Carter M.W., Farvolden J.M., Laporte G. E Xu J. (1996). Solving an integrated logistics problem arising in grocery distribution, *INFOR*, 34, 290-306.
- Chao I.M., Golden B.L. e Wasil E.A. (1995). An improved heuristic for the period vehicle routing problem, *Networks*, 26, 25-44.
- Chang N-B, Liu H. Y. e Wei Y.L. (1997). GIS technology for vehicle routing and scheduling in solid waste collection systems, *Journal of Environmental Engineering*, 123, 901- 910.
- Christofides N. e Beasley (1984). The period vehicle routing problem, *Networks*, 14, 237-256.
- Clarke G. e Wright J.V. (1964). Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581.

Cordeau J-F, Gendreau M. e Laporte G. (1997). A tabu search for periodic and multi-depot vehicle routing problems, *Networks*, 30, 105-119.

Cordeau J-F, Gendreau M. e Mercier A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52, 928-936.

Cordeau J-F, Gendreau M., Laporte G., Potvin J-Y e Semet F. (2002). A guide to vehicle routing problems, *Journal of the Operational Research Society*, 53, 512-522.

Cunha V. e Caixeta Filho (2002). Gerenciamento da coleta de resíduos sólidos urbanos: Estruturação e aplicação de modelo não-linear de programação por metas, *Gestão & Produção*, 9, 143-161.

Delgado C., Laguna M. e Pacheco J. (2005). Minimizing labor requirements in a periodic vehicle loading problem, a ser publicado em *Computational Optimization and Applications*.

Drummond L.M.A., Ochi L.S. e Vianna D.S. (2001). An asynchronous parallel metaheuristic for the period vehicle routing problem, *Future Generation Computer Systems*, 17, 379-386.

Fisher M.L. e Jaikumar R. (1981). A generalized assignment heuristic for general routing problems, *Networks*, 11, 109-124.

Gaudioso M. e Paletta (1992). A heuristic for the periodic vehicle routing problem, *Transportation Science*, 26, 86-92.

Gendreau M., Hertz A. e Laporte G. (1992). New insertion and postoptimization procedures for the traveling salesman problem, *Operations Research*, 40, 1086-1094.

Gendreau M., Hertz A. e Laporte G. (1994). A Tabu Search Heuristic for the Vehicle routing Problem, *Management Science*, 40, 1276-1290.

Glover F. (1989). Tabu Search Part I, *ORSA Journal on Computing*, vol.1, 3, 190-206.

Glover F. (1990). Tabu Search Part II, *ORSA Journal on Computing*, vol.2, 1, 4-32.

Glover F. (1994). Tabu search for nonlinear and parametric optimization (with links to genetic algorithms), *Discrete Applied Mathematics*, 49, 231-255.

Glover, F. (1996). Tabu search and adaptive memory programming – Advances, applications and challenges, in Barr R.S., Helgason R.V. and Kennington J.L., (editors), *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, 1-75, Kluwer.

Glover F. (1998). A template for scatter search and path relinking in artificial evolution, *Lecture Notes in Computer Science*, 1363, 13-54. J-K. Hao, E. Lutton, E. Ronald, M. Schoenauer e D. Snyers, Eds, Springer.

Glover F. e Laguna M. (1997). *Tabu Search*, Kluwer.

Glover F. Laguna M. e Martí R. (2000). Fundamentals of scatter search and path relinking, *Control and Cybernetics*, 29(3), 653-684.

Golden B.L. e Wasil E.A. (1987). Computerized vehicle routing in the soft drink industry, *Operations Research*, 35, 6-17.

Jain, A.S., Meeran, S., 2002. A multi-level hybrid framework applied to the general flow-shop scheduling problem, *Computers & Operations Research*, 29, 1873-1901.

Hokkanen J. e Salminen (1997). Choosing a solid waste management system using multicriteria decision analysis, *European Journal of Operational Research*, 98, 19-36.

Laporte G., Geandreau M., Potvin J.Y. e Sement F. (2000). Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operational Research*, 7, 285-300.

Lin S. e Kernighan B.W. (1973). An effective heuristic algorithm for the traveling salesman problem, *Operations Research* 21, 498-516.

Løkketangen A. e Glover F. (1998). Solving zero-one mixed integer programming problems using tabu search, *European Journal of Operational Research*, 106, 624-658.

Mitsumoto M. (2003). Busca em vizinhanças para o Problema de Roteamento de Veículos, relatório Fapesp, processo 03/04237-3.

Or I. (1976). Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking, Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University.

Resende M.G.C. e Ribeiro C.C. (2005). "GRASP with path-relinking: Recent advances and applications", in Ibaraki T., Nonobe K. and Yagiura M., *Metaheuristics: Progress as Real Problem Solvers*, Kluwer.

Rochat Y. e Taillard E.D. (1995). Probabilistic diversification and intensification in local search for vehicle routing, *Journal of heuristics* 1, 147-167.

Russel R.A. e Igo W. (1979). An assignment routing problem, *Networks*, 9, 1-17.

Russel R.A. e Gribbin D. (1991). A multiphase approach to the vehicle routing problem, *Networks*, 21, 747-765.

Savelsbergh M.W.P. (1992). The vehicle routing problem with time windows: minimizing route duration, *ORSA Journal on Computing*, 4, 146-154.

Shi L-H e Lin Y-T (1999). Optimal routing for infectious waste collection, *Journal of Environmental Engineering*, 125, 479-484.

Taillard E., Badeau P., Gendreau M., Guertain F. e Potvin J.Y. (1997). A Tabu Search for the vehicle routing problem with time windows, *Transportation Science*, 31, 170-186.

Tan C.C.R. e Beasley J. (1984). A heuristic algorithm for the period vehicle routing problem, *OMEGA*, 12, 497-504.

Yamashita D.S., Armentano V. A. e Laguna M. (2004). "Scatter search for project scheduling with resource availability cost", *European Journal of Operational Research*, to appear.