



**COPPE/UFRJ**

**BUSCA DISPERSA APLICADA AO PROBLEMA DE ROTEAMENTO DE  
VEICULOS COM MÚLTIPLOS DEPÓSITOS**

Dan Abensur Gandelman

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Produção.

Orientadores: Laura Bahiense e  
Virgílio J. M. Ferreira Filho

Rio de Janeiro  
Março de 2010

BUSCA DISPERSA APLICADA AO PROBLEMA DE ROTEAMENTO DE  
VEICULOS COM MÚLTIPLOS DEPÓSITOS

Dan Abensur Gandelman

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

---

Profa. Laura Silvia Bahiense da Silva Leite, D.Sc.

---

Prof. Virgilio Jose Martins Ferreira Filho, D.Sc.

---

Prof. Luiz Satoru Ochi, D.Sc.

---

Profa. Simone de Lima Martins, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2010

Abensur Gandelman, Dan

Busca Dispersa Aplicada ao Problema de Roteamento de Veículos com Múltiplos Depósitos/ Dan Abensur Gandelman. – Rio de Janeiro: UFRJ/COPPE, 2010.

XI, 69 p.: il.; 29,7 cm.

Orientadores: Laura Silvia Bahiense da Silva Leite e Virgilio Jose Martins Ferreira Filho.

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Produção, 2010.

Referencias Bibliográficas: p. 51-59.

1. Mataheurísticas. 2. Busca Dispersa. 3. Roteamento de Veículos. I. Leite, Laura Silvia Bahiense da Silva et al. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

Dedicado com muito amor:  
Ao meu pai Rubem,  
a minha mãe Sonia (em memória),  
e ao meu irmão Roni.

## AGRADECIMENTOS

Ao povo Brasileiro, por financiar meus estudos através das instituições UFRJ, COPPE/UFRJ, CNPq e CAPES.

Ao Prof. Roberto Diéguez Galvão (em memória), pela oportunidade concedida. Pelo ensino, orientação e pelo apoio dado para a conclusão do meu mestrado, até seu falecimento em 2008. Aos meus orientadores Virgílio J. M. Ferreira Filho e Laura Bahiense, que assumiram minha orientação em 2009, pelo apoio dado em todos aspectos para a conclusão desta dissertação.

Aos meus pais Rubem Gandelman e Sonia Abensur (em memória) e ao meu irmão Roni, família que sempre me apoiou e me suportou em todos os momentos.

Às minhas Avós Symi Abensur e Liba Gandelman (em memória). Aos meus Avôs Jacob Abensur (em memória) e Ovsia Gandelman (em memória). Às minhas Tias Irene e Helena. Aos meus tios Moysés, Áurea, Suelí e Rachel (em memória).

Ao meu grande amigo Luis Menasche Scherter. Ao meu grande amigo Sergio Warszawski. Ambas antigas amigas de inestimável valor.

A Catherine pelo carinho, amizade e apoio. Aos amigos da “Unesco” pelo apoio e amizade durante os últimos 10 anos.

A Gladys Maquera Sosa por toda sua grande amizade e apoio que foram sempre fundamentais. Seus conselhos nesta e em outras pesquisas sempre foram de grande valor.

A minha amiga Anna Lerner uma das minhas mais antigas amigas, por toda torcida e apoio. A minha grande amiga Claudia De Maria Santos, pela grande amizade e seus sempre sábios conselhos.

A meus amigos em geral, pelo apoio e amizade.

Aos professores e funcionários da PO-PEP, a Andreia e todas as pessoas que apoiaram diretamente ou indiretamente a conclusão deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

BUSCA DISPERSA APLICADA AO PROBLEMA DE ROTEAMENTO DE  
VEICULOS COM MÚLTIPLOS DEPÓSITOS

Dan Abensur Gandelman

Março/2010

Orientadores: Laura Silvia Bahiense da Silva Leite  
Virgilio Jose Martins Ferreira Filho

Programa: Engenharia de Produção

Esta dissertação aborda o Problema de Roteamento de Veículos com Múltiplos Depósitos e a metodologia de Busca Dispersa para resolver tal problema. O Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) é uma generalização do Problema de Roteamento de Veículos Clássico (PRV), onde rotas podem ser formadas a partir de múltiplos depósitos. Por se tratar de uma generalização do PRV, o PRVMD é classificado como NP-difícil, o que torna difícil sua solução ótima para instâncias que não sejam pequenas. Este trabalho um algoritmo baseado na metaheurística de Busca Dispersa (*Scatter Search* em inglês) para resolver o PRVMD. O algoritmo é testado para instâncias encontradas na literatura. Os resultados são comparados com os obtidos por outras heurísticas, mostrando que a BD é eficiente para resolver o PRVMD.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SCATTER SEARCH APPLIED TO THE MULTI-DEPOT VEHICLE ROUTING  
PROBLEM

Dan Abensur Gandelman

March/2010

Advisors: Laura Silvia Bahiense da Silva Leite  
Virgilio Jose Martins Ferreira Filho

Department: Industrial Engineering

The present dissertation deals with the Multi-Depot Vehicle Routing Problem (MDVRP) using the Scatter Search methodology. This problem is a generalization of the Classical Vehicle Routing Problem, where routes can belong to more than one depot. The MDVRP is an NP-hard problem, which means that optimal solutions are difficult to find for big instances. This thesis presents a Scatter Search applied to the MDVRP. The algorithm is tested with benchmark instances and compared with other heuristics found in the literature. The result of this comparison is that BD is efficient to solve the problem.

## ÍNDICE

|  |    |
|--|----|
| CAPITULO 1.- INTRODUÇÃO.....   | 1  |
| CAPITULO 2.- O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS..... | 3  |
| 2.1 Descrição do Problema.....   | 3  |
| 2.2 Formulação Matemática para o PRVMD.....                                    | 5  |
| CAPITULO 3.- BUSCA DISPERSA.....   | 7  |
| 3.1 Desenho Básico da Busca Dispersa.....                                      | 8  |
| 3.2 Método Gerador de Soluções Iniciais.....                                   | 10 |
| 3.3 Método da Melhoria.....  | 10 |
| 3.4 Construção do Conjunto de Referência.....                                  | 10 |
| 3.5 Atualização do Conjunto de Referência.....                                 | 11 |
| 3.6 Reconstrução do Conjunto de Referência.....                                | 12 |
| 3.7 Método de Geração de Subconjuntos.....                                     | 12 |
| 3.8 Busca Dispersa aplicada a Problemas de Roteamnto.....                      | 12 |
| CAPITULO 4.- BUSCA DISPERSA APLICADA AO PRVMD.....                             | 14 |
| 5.1 Método Gerador de Soluções Diversas.....                                   | 14 |
| 5.1.1 GRASP aplicado ao PRVMD.....   | 14 |
| 5.2 Método de Melhoria das Soluções.....                                       | 15 |
| 5.2.1 Re-alocação.....   | 17 |
| 5.2. 2 intercâmbio.....  | 17 |
| 5.2.3 Cruzamento.....  | 18 |
| 5.2.4 2-opt.....   | 18 |
| 5.2.5 Métodos de Melhoria Implementados.....                                   | 19 |
| 5.3 Construção do Conjunto de Referência.....                                  | 20 |
| 5.4 Geração de Subconjuntos.....   | 22 |
| 5.5 Combinação de Soluções.....  | 22 |
| 5.6 Atualização do Conjunto de Referência.....                                 | 25 |
| 5.7 Critério de Parada da DB.....  | 26 |



|  |    |
|--|----|
| CAPITULO 6.- RESULTADOS COMPUTACIONAIS.....          | 26 |
| 6.1 Instâncias.....                                  | 26 |
| 6.2 Ajuste dos parâmetros para a busca dispersa..... | 29 |
| 6.3 Resultados computacionais.....                   | 32 |
| 6.4 Testes de Estabilidade da BD implementada.....   | 44 |
| <br>   |    |
| CAPITULO 7.- CONCLUSÕES E PEQUISA FUTURA.....        | 48 |
| <br>   |    |
| REFERÊNCIAS BIBLIOGRAFICAS.....                      | 51 |
| <br>   |    |
| ANEXO 1 – PROBLEMAS DE ROTEAMENTO.....               | 60 |
| <br>   |    |
| ANEXO 2 – PESQUISA BIBLIOGRAFICA PARA O PRV.....     | 65 |

## LISTA DE FIGURAS :

|  |    |
|--|----|
| Figura 2.1.Representação do PRVMD. ....  | 3  |
| Figura 3.1.Pseudocódigo da Busca Dispersa. ....                                  | 9  |
| Figura 4.1.Algoritmo de construção de soluções.....                              | 16 |
| Figura 4.2.Exemplo de Movimento de Inserção Inter-Depósito.....                  | 17 |
| Figura 4.3.Exemplo de Movimento de Intercâmbio Inter-Depósito.....               | 18 |
| Figura 4.4.Exemplo de Movimento de Crossover. ....                               | 19 |
| Figura 4.5.Exemplo de Movimento 2-opt. ....                                      | 20 |
| Figura 4.6.Exemplo de Método de Combinação.....                                  | 24 |
| Figura 5.1. Testes para o método de combinação.....                              | 32 |
| Figura 5.2.Tempo de Execução versus Tamanho da Instância para a BD.....          | 40 |
| Figura 5.3.Tempo de Execução versus Tamanho da Instância para a Busca Tabu...41  |    |
| Figura 5.4. Tempo de Execução versus Tamanho da Instância para método exato...41 |    |
| Figura 5.5. Evolução do GAP durante a execução da instância p01.....             | 42 |
| Figura 5.6. Evolução do GAP durante a execução da instância p22.....             | 42 |
| Figura 5.7. Evolução do GAP para a instância p01.....                            | 43 |
| Figura 5.8. Evolução do GAP para a instância p22.....                            | 44 |
| Figura 5.9. Resultados variando a Matriz Aleatória.....                          | 45 |
| Figura 5.10. Distribuição de Probabilidade da instância p01.....                 | 46 |
| Figura 5.11. Distribuição de Probabilidade da instância p22.....                 | 47 |

## LISTA DE TABELAS :

|  |    |
|--|----|
| Tabela 5.1. Uso de Instâncias pela Literatura.....                         | 27 |
| Tabela 5.2. Características das instâncias utilizadas.....                 | 28 |
| Tabela 5.3. Testes variando o parâmetro alfa.....                          | 30 |
| Tabela 5.4. Testes de estratégias para o método de melhoria.....           | 31 |
| Tabela 5.5. Testes para o tamanho de Refset.....                           | 32 |
| Tabela 5.6 Testes para o tamanho do conjunto P.....                        | 33 |
| Tabela 5.7 Parâmetros adotados para a DB.....                              | 33 |
| Tabela 5.8. Resultados da BD para as instâncias benchmark.....             | 35 |
| Tabela 5.9. Comparação da BD com os resultados obtidos por CHAO.....       | 36 |
| Tabela 5.10. Comparação da BD com os resultados obtidos por PISSINGER..... | 37 |
| Tabela 5.11. Comparação da BD com os resultados obtidos por BALDACCI.....  | 38 |
| Tabela 5.12. Comparação baseado em MFlops.....                             | 39 |

## CAPÍTULO 1

### INTRODUÇÃO

O crescimento econômico mundial e o aumento do consumo na sociedade aumentaram a demanda pela movimentação de cargas. A importância da eficiência dos sistemas de distribuição torna-se evidente quando se considera o impacto dos respectivos custos na operação das empresas e no meio ambiente. Neste contexto os métodos de otimização em problemas de roteamento de veículos recebem grande destaque dentro da Pesquisa Operacional.

O Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) é uma generalização do Problema de Roteamento de Veículos Clássico (PRV), onde rotas podem ser formadas a partir de múltiplos depósitos. Por se tratar de uma generalização do PRV, o PRVMD é NP-difícil, o que torna difícil sua solução ótima para instâncias que não sejam pequenas. Este trabalho apresenta uma versão de um algoritmo baseado na metaheurística de Busca Dispersa para resolver o PRVMD. O algoritmo é testado para instâncias encontradas na literatura. Os resultados também são comparados com os obtidos por outras heurísticas.

A Busca Dispersa (BD), *Scatter Search* em inglês – ver LAGUNA e MARTÍ (2003), é uma meta-heurística que combina soluções pertencentes a um conjunto de referência, com o objetivo de gerar novas soluções com características não contidas nas soluções originais. O conjunto de referência guarda soluções de qualidade e diversidade encontrados durante o processo de busca. A BD apresenta algumas similaridades com os algoritmos genéticos, porém difere deles em princípios fundamentais, tais como o uso de estratégias sistemáticas ao invés de aleatórias; proporciona também um conjunto de parâmetros, gerando um marco flexível que permite o desenvolvimento de diferentes implementações com distintos graus de complexidade. A BD tem sido aplicada na solução de um grande número de problemas de otimização combinatória. Para o PRV clássico ver SOSA et al. (2007); para o problema de roteamento de veículos com janelas de tempo, ver RUSSEL e CHIANG (2006).

A organização da dissertação se dá da seguinte forma: no Capítulo 2 apresentamos formalmente o PRVMD. Já no Capítulo 3 a Busca Dispersa, que será a metodologia utilizada para resolver o problema. No Capítulo 4 apresentamos a BD aplicada ao PRVMD e como se dá a implementação. No Capítulo 5 apresentamos os resultados computacionais para no Capítulo 6 concluir o trabalho, além de apresentar perspectivas futuras. Nos Anexos 1 e 2 apresentamos uma visão geral de problemas de rotamnto e uma pesquisa bibliográfica para o PRV, respectivamente.

A principal contribuição deste trabalho é aplicar a Busca Dispersa ao PRVMD, cuja implementação, de nosso conhecimento, é inédita para tal problema. Podemos assim testar a eficácia de tal método para esta variante do PRV. Outra contribuição é o processo de calibração de parâmetros realizado para a BD, que pode servir de referência para outras implementações de BD.



Para solucionar o PRVMD temos abordagens exatas, que garantem que a melhor solução é encontrada (solução ótima) e abordagens heurísticas, que não têm tal garantia, porém chegam a soluções viáveis em pouco tempo computacional. São poucos os trabalhos que propõem abordagens exatas para o PRVMD. LAPORTE et al. (1984) propõe um algoritmo de branch-and-bound para o problema simétrico, o qual inicialmente relaxa as restrições de integralidade, eliminações de subtours, restrições que impedem a formação de rotas entre dois depósitos. Já LAPORTE et al. (1988) propõem a solução do PRVMD assimétrico, transformando-o inicialmente no problema de alocação capacitada, que é então solucionado utilizando branch-and-bound. Soluções ótimas para instâncias de até 80 clientes e 3 depósitos são encontradas. Mais recentemente, BALDACCI e MINGOZZI (2009) propõe um método exato para o problema que utiliza métodos de cortes baseados nas relaxações Lagrangeanas e LP.

As abordagens heurísticas para o PRVMD são mais comuns. TILLMAN (1969) utiliza o método proposto por CLARKE e WRIGHT (1964). A heurística inicialmente aloca os clientes aos depósitos mais próximos, para então formar rotas. TILLMAN e HERING (1971) utilizam a mesma idéia, porém permitem que um cliente migre para uma rota de outro depósito durante o processo de busca, apesar de sua alocação inicial. WREN e HOLLIDAY (1972) iniciam sua heurística alocando cada cliente ao depósito mais próximo e calculam seu ângulo polar em relação ao depósito respectivo. Em ordem crescente do ângulo polar, os clientes são alocados a rotas existentes ou novas rotas, caso o impacto na função objetivo seja compensador. A partir daí sete movimentos específicos são repetitivamente aplicados, até que melhorias não sejam mais obtidas.

CHAO et al. (1993) propõem uma heurística de múltiplas fases na qual inicialmente os clientes são associados ao depósito mais próximo. Em seguida o PRV Clássico é resolvido para cada depósito. Na fase seguinte tenta-se realocar os clientes para outros depósitos visando melhorar a solução. Dois procedimentos de re-inicialização são implementados buscando diversificar o espaço de solução. Resultados são apresentados para 11 instâncias de GILLET e JOHNSON (1976) com até 360 clientes e 9 depósitos.

SALHI e SARI (1997) propõe uma heurística de três níveis. No primeiro nível é construída uma solução inicial viável. No segundo os autores tentam melhorar a solução do primeiro nível através de movimentos inter-depósitos; no terceiro nível as tentativas são via movimentos intra-depósitos. NAGY e SALHI (2005) apresentam uma

heurística para o PRVMD com coletas e entregas simultâneas. A heurística é composta de um método gerador de soluções iniciais seguido por vários movimentos em uma determinada seqüência. PISINGER e ROPKE (2007) propõe uma heurística que é uma extensão do *adaptive large neighborhood search* que também resolve o PRVMD.

RENAUD et al. (1996) apresentam uma busca tabu para o PRVMD, na qual são testadas 23 instâncias, incluindo as instâncias de GILLET e JOHNSON (1976). BAE et al. (2007) propõem um Algoritmo Genético (AG) para o problema, utilizando operadores genéticos clássicos do PRV modificados para o problema multi-depósito. HO et al. (2008) propõem um AG híbrido onde dois métodos geradores de solução inicial são utilizados, um aleatório e outro baseado na heurística de mínimas distâncias de Clark e Wright. Em seguida é aplicado um método de melhoria a essas soluções, para então iniciar-se um AG Clássico. LI-XIA (2009) propõe uma heurística baseada em simulação utilizando lógica fuzzy para resolver o problema, estes não utilizam instâncias utilizadas por outros autores e se limitam a ajustar a heurística proposta em função de seus próprios parâmetros.

## 2.2 Formulação Matemática para o PRVMD.

Apresentamos aqui uma formulação para o PRVMD.

$$\text{Minimize } \sum_{k \in K} \sum_{i \in H \cup L} \sum_{j \in H \cup L} d_{ij} \cdot x_{ijk}$$

Sujeito a

$$\sum_{i \in H \cup L} \sum_{j \in H \cup L} d_{ij} \cdot x_{ijk} \leq D \quad (k \in K), \quad (\text{R.1})$$

$$t_{ijk} \leq C \quad (i, j \in H \cup L, k \in K), \quad (\text{R.2})$$

$$\sum_{k \in K} \sum_{i \in H \cup L} x_{ijk} = 1 \quad (j \in H), \quad (\text{R.3})$$

$$\sum_{k \in K} \sum_{i \in H \cup L} x_{ijk} \geq 1 \quad (j \in L), \quad (\text{R.4})$$



$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in (H \cup L) \setminus S} x_{ijk} \geq 1 \quad (2 \leq |S|), \quad (\text{R.5})$$

$$\sum_{i \in H \cup L} x_{ijk} = \sum_{i \in H \cup L} x_{jik} \quad (j \in H \cup L, k \in K), \quad (\text{R.6})$$

$$\sum_{i \in H \cup L} \sum_{j \in H \cup L} x_{ijk} \leq 1 \quad (k \in K), \quad (\text{R.7})$$

$$t_{ijk} \leq x_{ijk} \cdot \sum_n (q_n) \quad (i, j \in H \cup L, k \in K), \quad (\text{R.8})$$

$$x_{ijk} = 0,1 \quad (i, j \in H \cup L, k \in K), \quad (\text{R.9})$$

$$t_{ijk} \geq 0 \quad (i, j \in H \cup L, k \in K). \quad (\text{R.10})$$

Onde:

$L = \{1, 2, \dots, t\}$  é o conjunto de depósitos.

$H = \{1, 2, \dots, n\}$  é o conjunto de clientes.

$K = \{1, 2, \dots, m\}$  é o conjunto de veículos.

$d_{ij}$  é a distância entre o cliente  $i$  e  $j$

$q_i$  é a demanda do cliente  $i$

$D$  é o limite de distância que os veículos podem percorrer

$C$  é a capacidade máxima de cada veículo

$x_{ijk} = 1$ , se o arco  $ij$  faz parte da rota  $k$  e  $x_{ijk} = 0$  caso contrário

$t_{ijk}$  é a carga no arco  $ij$  da rota  $k$

Nesta formulação as restrições R.1 e R.2 são de limite de distância e limite de capacidade respectivamente. R.3 e R.4 restringem que cada cliente só pode pertencer a uma única rota, porém os depósitos podem pertencer a mais de uma rota. R.5 garante que cada rota é conectada ao depósito. R.6 garante que em cada cliente o mesmo veículo que chega é o que sai. R.7 garante que um veículo pode partir somente uma vez de um cliente. R.8 garante que somente exista fluxo entre clientes e depósitos conectados. R.9 e R.10 são restrições de integralidade e não negatividade.

## CAPÍTULO 3

### BUSCA DISPERSA

Neste capítulo descreveremos a metodologia de Busca Dispersa. Trata-se de um algoritmo evolutivo que combina soluções através de estratégias determinísticas e tem sido aplicado com sucesso a diversos problemas de otimização combinatória. Seus conceitos básicos foram propostos na década de 70. Em contraste com outros métodos evolutivos como algoritmos genéticos, a Busca Dispersa usa métodos determinísticos para combinar soluções, sem usar métodos aleatórios. Suas estratégias de intensificação e diversificação tem se mostrado eficazes para a resolução de diversos problemas de otimização combinatória.

A Busca Dispersa foi introduzida por GLOVER (1977) como uma heurística para programação inteira. Em sua proposta original soluções iniciais eram geradas de maneira determinística para que então a BD orientasse a exploração de um conjunto de referência destas soluções, levando em conta soluções diversas e com bom valor da Função Objetivo (FO). Combinações lineares de soluções geram novos pontos de exploração no espaço de soluções, realizando a “busca dispersa” pelo espaço de soluções.

Já no final da década de 90, GLOVER (1998) apresentou um template mais elaborado, que atualmente é considerada a referência para implementações de BD. LAGUNA e MARTÍ (2003) publicaram um livro sobre BD, realizando um detalhado estudo da metodologia e abordando estratégias avançadas como uso de memória e de *re-encadeamento de trajetórias*. Usaremos nesta dissertação a terminologia definida por GLOVER (1998), aqui repetida:

|                        |   |
|------------------------|---|
| <i>P</i>               | Conjunto de soluções geradas com o método gerador de soluções diversas;                         |
| <i>PSize</i>           | Tamanho da população <i>P</i> de soluções diversas (i.e., $ P =Psize$ );                        |
| <i>Refset</i>          | Conjunto de Referência;   |
| <i>RefsetQualidade</i> | Conjunto de soluções com qualidade em <i>Refset</i> (em relação à função objetivo do problema); |

|                          |   |
|--------------------------|---|
| <i>RefsetDiversidade</i> | Conjunto de soluções com diversidade em <i>Refset</i> (em relação às soluções que estão em <i>Refset</i> );   |
| <i>Pool</i>              | Conjunto de soluções geradas pelo método da combinação;   |
| <i>b</i>                 | Tamanho do conjunto de referência (i.e., $ RefSet =b$ );  |
| $b_1$                    | Tamanho do subconjunto do <i>Refset</i> correspondente a soluções de qualidade;                               |
| $b_2$                    | Tamanho do subconjunto do <i>Refset</i> correspondente a soluções com diversidade;                            |
| $x^i$                    | A $i^{ésima}$ solução do conjunto de referência . $x^1$ é a melhor solução em <i>Refset</i> e $x^b$ é a pior; |
| $d(x,y)$                 | Distância entre a solução $x$ e a solução $y$ .   |
| $s$                      | Subconjunto de <i>Refset</i> .  |

### 3.1 Desenho Básico da Busca Dispersa

O método de busca dispersa é bastante flexível, já que seus elementos podem ser implementados de diversas maneiras e com diversos níveis de complexidade. Tal flexibilidade é explorada em cada aplicação da BD, onde estratégias de exploração do espaço de soluções se mostram eficientes para uma implementação em particular. Os principais elementos são seus 5 métodos:

- (i) Um Método Gerador de Soluções Iniciais
- (ii) Um Método de Melhoria para atingir ótimos locais
- (iii) Um Método de Atualização de Sub-Conjuntos para manter o conjunto de referência.
- (iv) Um Método de Combinação de Soluções para gerar novas soluções
- (v) Um Método Gerador de Sub-Conjuntos para gerar um conjunto de referência.

A Figura 3.1 mostra como os métodos se combinam para formar a Busca Dispersa.

---

**Passo 1.** Utilizando o método gerador de soluções iniciais, gerar  $PSize$  soluções para armazenar no conjunto  $P$ .

**Passo 2.** Utilizando o Método de Melhoria, levar cada solução do conjunto  $P$  ao seu ótimo local.

**Passo 3.** Construa  $Refset$  utilizando o método Gerador de Sub-conjuntos, movendo do conjunto  $P$ ,  $b_1$  soluções com melhores valor de FO e  $b_2$  soluções com alto grau de diversidade. Faça  $novasoluções = verdadeiro$ ;

**Enquanto (  $|P| > 0$  )**

**Enquanto (  $novasoluções = verdadeiro$  )**

**Passo 4.** Gere novas soluções utilizando o conjunto de referência e o método de combinação de soluções armazenando as novas soluções no conjunto  $Pool$ ;

**Passo 5.** Atualize o conjunto de referência utilizando o método de atualização de Sub-Conjuntos considerando as soluções em  $Refset \cup Pool$ ;

**Passo 6.** Caso o conjunto de referência não tenha sido modificado na atualização do Passo 5, faça  $novasoluções = falso$ ;

**Fim Enquanto**

**Passo 7.** Re-construa  $Refset$  utilizando o método Gerador de Sub-conjuntos, movendo  $b_1$  soluções de  $P$  com melhores valor de FO e  $b_2$  soluções de  $P$  com alto grau de diversidade. Faça  $novasoluções = verdadeiro$ ; Caso não haja mais soluções no conjunto  $P$ , termine a Busca Dispersa.

**Fim Enquanto**

---

**Figura 3.1.** Pseudocódigo da Busca Dispersa.

Na Figura 3.1 apresentamos o pseudocódigo básico da BD. No Passo 1 geramos soluções iniciais através de seu método gerador. No Passo 2 aplica-se o método de melhoria que irá melhorar cada uma das soluções, deixando o conjunto  $P$ , de soluções iniciais, com soluções em seus ótimos locais. No Passo 3 construímos o conjunto de Referência  $Refset$  considerando dois subconjuntos: i)  $RefsetQualidade$ , contendo as  $b_1$  melhores soluções em relação a FO e ii)  $RefsetDiversidade$  contendo as  $b_2$  soluções mais diversas do conjunto  $P$  em relação a  $RefsetQualidade$ . As soluções selecionadas para construir o conjunto de referência são retiradas do conjunto  $P$  de soluções iniciais.

No Passo 4 o método de combinação combina as soluções do conjunto  $Refset$  gerando novas soluções. Essas novas soluções são armazenadas no conjunto  $Pool$ . O método de atualização de sub-conjunto então, no Passo 5, verifica as melhores soluções do conjunto  $Pool$  e atualiza  $RefsetQualidade$  com tais soluções. Também atualiza  $RefsetDiversidade$  com as soluções mais diversas de  $Pool$  em relação a  $RefsetDiversidade$ . O Passo 6 verifica se houve realmente atualização do conjunto de

referência ou se ele permaneceu igual a última atualização. Caso tenha permanecido igual o looping se encerra e novas soluções oriundas do conjunto  $P$  irão entrar no conjunto de referência. No Passo 7, novas soluções do conjunto  $P$  são inseridas no conjunto de referência. Quando não houverem mais soluções naquele conjunto a busca dispersa se encerra.

### 3.2 Método Gerador de Soluções Iniciais

Tal método dita o início do procedimento de busca dispersa, ao gerar soluções iniciais para o processo de busca. O número de soluções é geralmente cinco vezes o tamanho do conjunto de referência, mas esse é um número empírico recomendado por LAGUNA e MARTÍ (2003). É importante que se tenha diversidade entre as soluções geradas por este método, para que a busca possa ocorrer por um espaço amplo do espaço de soluções. A implementação deste método tem grande liberdade e depende do problema sendo tratado.

### 3.3 Método da Melhoria

É um método clássico também encontrado em outras heurísticas como algoritmos genéticos e busca tabu. Trata-se de movimentos a serem aplicados repetidamente buscando levar a solução a um ótimo local. Caso soluções sejam inviáveis, tenta-se aplicar movimentos que levem a solução a viabilidade e uma vez estando viável, aplica-se movimentos buscando melhorá-la.

### 3.4 Construção do Conjunto de Referência

O conjunto de referencia é o que guia a busca dispersa. Ela é composta de dois subconjuntos: *RefsetQualidade* e *RefsetDiversidade*. Mantendo durante o procedimento de busca soluções referência de qualidade e diversidade, procura-se combinar características de ambas tentando sair de ótimos locais e buscar novos ótimos, com a esperança de se encontrar o ótimo global.

O *RefsetQualidade* sempre é composto de soluções boas em relação a FO. Já em *RefsetDiversidade* consideramos soluções diversas em relação ao conjunto *RefsetQualidade*. Para calcular o quanto uma solução é diversa de outra é preciso ter

um procedimento de cálculo estabelecido, que vai depender da estrutura da solução do problema a ser tratado.

O procedimento se inicia selecionando do conjunto  $P$   $b_1$  melhores soluções em relação a FO. Retira-se tais soluções do conjunto  $P$ . Então calcula-se a diversidade entre todos os elementos do conjunto  $P$  e todos os elementos de *RefsetQualidade*. As  $b_2$  soluções mais diversas são retiradas de  $P$  e adicionadas a *RefsetDiversidade*.

### 3.5 Atualização do Conjunto de Referência

LAGUNA e MARTÍ (2003) descrevem dois critérios a serem consideradas para atualização do conjunto de referência. O primeiro trata do momento em que ela deve ocorrer gerando duas opções: a atualização estática ou dinâmica. O outro critério para renovar o conjunto de referência que pode se basear em qualidade e por qualidade/diversidade.

A atualização estática somente atualiza o conjunto de referência após todas as soluções do mesmo conjunto terem sido combinadas. Somente após gerar todo o conjunto *Pool* é que se examina o mesmo para atualizar *Refset*. Já na atualização dinâmica a cada solução que é gerada via combinação de soluções podemos potencialmente atualizar o conjunto de referência e por isso trata-se de um procedimento de maior complexidade.

Em relação ao critério para renovar o conjunto de referência na atualização baseada em qualidade, todo o conjunto de referência (incluindo *RefsetDiversidade*) é atualizado utilizando-se soluções com melhor FO. Já no método de atualização por qualidade e diversidade, temos o cálculo da distância entre soluções e o *RefsetDiversidade* é preenchido com soluções diversas. Ou seja no caso de atualização por qualidade, não consideramos a diversidade na busca, obtendo assim uma convergência mais rápida porém com menor exploração do espaço de soluções.

### 3.6 Reconstrução do Conjunto de Referência: Método de Geração de Subconjuntos

Quando se realiza o procedimento de atualização do conjunto de referência e não se dá nenhuma alteração nele, chegamos a uma estagnação da busca. Para continuar reconstruímos o conjunto de referência trazendo soluções do conjunto P, sempre as removendo deste conjunto. Quando ao buscar-se novas soluções em P, verificar-se que o mesmo se encontra vazio, a busca dispersa termina.

### 3.7 Método de Combinação de Soluções

O método gerador de soluções iniciais cria soluções que em seguida são otimizadas pelo método de melhoria. O *Refset* é o conjunto de referência que o serve justamente para a combinação de soluções. Assim como nos algoritmos genéticos tal método combina características de duas soluções para criar uma terceira. Para cada problema de otimização tenta-se detectar qualidades em duas soluções e preservá-las na solução a ser gerada. A combinação pode ser sistemática ou probabilística, podendo gerar uma ou mais soluções resultantes.

### 3.8 Busca Dispersa aplicada a Problemas de Roteamento

No próximo Capítulo veremos como ocorre a implementação da DB para o PRVMD. Para isso vale verificar outras implementações de BD a problemas de roteamento em geral. Em relação à aplicação da BD a diferentes problemas de roteamento de veículos, existem trabalhos prévios, tais como o de ROCHAT e TAILLARD (1995), no qual apresentam uma técnica probabilística para diversificar e intensificar a busca local. Esta técnica é aplicada a dois tipos de problemas de roteamento: o Problema de Roteamento de Veículo clássico e o PRV com Janelas de Tempo. Ela é composta de duas fases: i) Geração de soluções iniciais; ii) Identificação de tours (bons e ruins) comuns nas soluções iniciais para gerar novas soluções. Nas novas soluções encontradas inicia-se uma busca local, para depois realizar a pós-otimização.

REGO (2000) aplica a BD ao PRV clássico e combina soluções desde o conjunto de referência (*Refset*) por meio do mecanismo de arcos comuns.

CORBERAN et al. (2002) desenvolveu um procedimento eficiente da BD para resolver um problema multiobjetivo do roteamento de ônibus da escola em uma área rural. RUSSELL e CHIANG (2006) aplicam a BD ao Problema de Roteamento de Veículos com Janelas de Tempo no qual estudam os efeitos dos parâmetros relacionados ao conjunto de referência (tamanho, à qualidade e à diversidade). O mecanismo de arcos comuns e um modelo ajustado baseado em recobrimento são utilizados para combinar as soluções. Para o método de melhoria utilizam Busca Tabu Reativa. Resultados computacionais mostraram que a BD é competente e obtém, ao igual que outras metaheurísticas, soluções de boa qualidade.

SOSA et al. (2007) aplicam um algoritmo da BD desenvolvido para o PRV Clássico. Para a etapa geradora de soluções foram testadas duas heurísticas construtivas. Os autores apresentam três métodos de melhoria inter-rotas e um método de melhoria intra-rota para a etapa de melhoria da BD. A construção de Refset é por qualidade e diversidade das soluções. Por meio da geração de sub-conjuntos escolhem pares de soluções que serão combinadas por meio do mecanismo de clientes comuns às rotas. Experimentos computacionais foram realizados em quatro conjuntos de dados disponíveis na literatura e são mostrados resultados para os diferentes parâmetros relacionados ao Refset, assim como para diferentes critérios de atualização de Refset.



## CAPÍTULO 4

### BUSCA DISPERSA APLICADA AO PRVMD

Neste capítulo são apresentadas as diferentes etapas de uma aplicação da BD ao PRVMD. Para o método gerador de soluções iniciais foi construída uma heurística construtiva com base na metodologia GRASP. Para a etapa de melhoria são apresentados diferentes tipos de movimento inter-rotas (entre rotas) e intra-rota (dentro de uma mesma rota). Finalmente, as etapas de construção (atualização) do Refset, geração de sub-conjuntos e combinação de soluções são descritos.

#### 5.1 Método Gerador de Soluções Iniciais

Utilizamos dois métodos para gerar soluções iniciais. Ambos se baseiam na metodologia do GRASP, utilizando como base duas diferentes funções míopes: baseado em heurística construtiva gulosa de menor distância e a Heurística Sweep proposta por GILLETT e MILLER (1974).

Os dois métodos geram ao todo  $Psize$  soluções. Vamos iniciar descrevendo a heurística de Gillett e Miller adaptada para o problema de roteamento com múltiplos depósitos para em seguida mostrar como é feita a adaptação para a metodologia GRASP.

##### 5.1.1 Algoritmo Sweep com base na heurística de Gillett e Miller

No algoritmo Sweep proposto por GILLETT e MILLER (1974), são formados grupamentos de clientes os ordenando segundo seus ângulos polares em relação aos depósitos até que as restrições do PRV não sejam satisfeitas. Cada cliente  $i$  é referenciado por suas coordenadas polares  $(\rho_i, \theta_i)$  em relação ao depósito mais próximo. A heurística adaptada para o problema de roteamento com múltiplos depósitos é definida pelos seguintes passos:

**Passo 1** Agrupar clientes a depósitos. Um cliente será agrupado ao depósito que estiver mais próximo, considerando distâncias euclidianas aos depósitos.

**Passo 2** Para cada agrupamento de clientes/depósitos, criar uma lista ordenada de clientes segundo  $\theta$  em ordem crescente. Se dois clientes têm o mesmo valor de  $\theta$  (ângulo polar relativo ao depósito do agrupamento), o cliente com menor valor de  $\rho$  (distância ao depósito) é selecionado.

**Passo 3** Para cada uma das listas ordenadas do Passo 2, formar rotas que sigam a ordem da lista, respeitando a viabilidade das soluções. Iniciar uma nova rota quando um cliente não conseguir ser inserido na rota anterior sem causar inviabilidade da solução.

O processo pode ser repetido diversas vezes. Podemos fazer uma rotação a esquerda nas listas geradas no Passo 2. Analogamente o que estamos fazendo é iniciar o procedimento de um outro ângulo polar que não zero graus. Podemos iniciar o método nos  $n$  ângulos polares de cada cliente em relação depósito.

A diferença entre a abordagem clássica desta heurística e a adaptação de múltiplos depósitos se dá apenas no Passo 1, onde agrupamos clientes a depósitos. Quando o procedimento é executado para um único depósito, temos como resultado o procedimento como descrito por GILLET e MILLER (1974).

### 5.1.2 GRASP aplicado ao PRVMD

A metodologia GRASP é utilizada para a geração de soluções iniciais no processo de busca da BD. A meta-heurística GRASP - “*Greedy Randomized Adaptive Search Procedure*”, foi desenvolvida ao final da década de 80, ver FEO e RESENDE (1995). GRASP é um procedimento de múltiplos inícios, no qual cada iteração consiste em uma fase de construção e outra de melhoria. Na fase de construção aplica-se um procedimento heurístico construtivo para obter uma boa solução inicial. Esta solução é melhorada na segunda fase mediante um algoritmo de busca local. Esta metodologia foi escolhida por permitir a geração de soluções diversas, ou seja, dispersas no espaço de soluções. Tal característica é fundamental para o sucesso de uma implementação de um algoritmo de BD.

Descreveremos os procedimentos de uma rotina GRASP já adaptada para o PRVMD. Seja  $c(e)$  o valor da função míope para o cliente  $e$ . Seja  $c^*$  o maior valor da função míope e  $c_*$  o menor valor para os elementos candidatos. Seja  $D_{A_k}$  a distância

acumulada no grupo  $k$  e  $d_{ij}$  a distância entre o cliente  $i$  e o cliente  $j$ , onde  $i$  representa o último cliente que esta no agrupamento  $A_k$ , isto é, o cliente que precede o depósito da rota;  $j$  é o cliente candidato a ingressar no agrupamento  $k$  e  $p_j$  é o valor da demanda do cliente  $j$ . Seja  $\phi_{j_0}$  o ângulo polar entre o último cliente roteado  $j$  e o depósito a que a rota pertence.

---

**Passo 1.** Escolha um dos critérios da construção da função míope:

Baseado na Heurística de GILLET e MILLER (1974): a função míope é dada por  $c(j) = \phi_{j_0}$ . Baseado na heurística construtiva gulosa de menor distância:

$$c(j) = D_{A_k} + d_{ij} + d_{j_0}.$$

**Passo 2.** Seja  $k = 1$ .

**Passo 3.** Selecione aleatoriamente um cliente para iniciar uma rota a partir do depósito mais próximo a ele.

**Passo 4.** Se todos os clientes já estiverem roteados ir para o Passo 7. Caso contrário, para cada elemento não roteado calcular o custo da função míope  $c(e)$  encontrando o custo de inserir o cliente na última posição da rota  $k$ .

**Passo 5.** Determine uma Lista Restrita de Candidatos (LRC), que consiste em todos os elementos candidatos cujo valor da função míope (com base na heurística proposta)  $c(e)$  seja tal que (i)  $c(e) \leq c_* + \alpha(c^* - c_*)$ , onde  $c(e)$  é o valor da função míope para o cliente e  $c^*$  e  $c_*$  os maiores e menores valores da função míope para os elementos candidatos. (ii) Só poderão entrar na LRC clientes que, caso inseridos na posição indicada, respeitem todas as condições de viabilidade da rota.

**Passo 6.** Caso a LRC não contenha nenhum candidato, faça  $k=k+1$ , iniciando uma nova rota e selecione aleatoriamente um cliente para iniciar uma rota a partir do depósito mais próximo, vá ao Passo 3. Caso contrário, selecione aleatoriamente um candidato da LRC para acrescentar à solução parcial do PRVMD. Atualize os dados do PRVMD e vá ao Passo 3.

**Passo 7.** Foi construída uma solução para o PRVMD. Aplique o Método de melhoria.

**Passo 8.** Repita os passos anteriores  $Psize$  vezes para encontrar  $Psize$  soluções iniciais.

---

**Figura 4.1:** Algoritmo de construção de soluções com base na metodologia GRASP

## 5.2 Método de Melhoria das Soluções

Para o método de melhoria consideramos quatro movimentos básicos: realocação, intercâmbio, cruzamento e 2-opt. Tais movimentos definem as vizinhanças do processo de busca. Temos duas estratégias para aceitar os movimentos calculados: melhor entre todas e primeira melhoria encontrada.

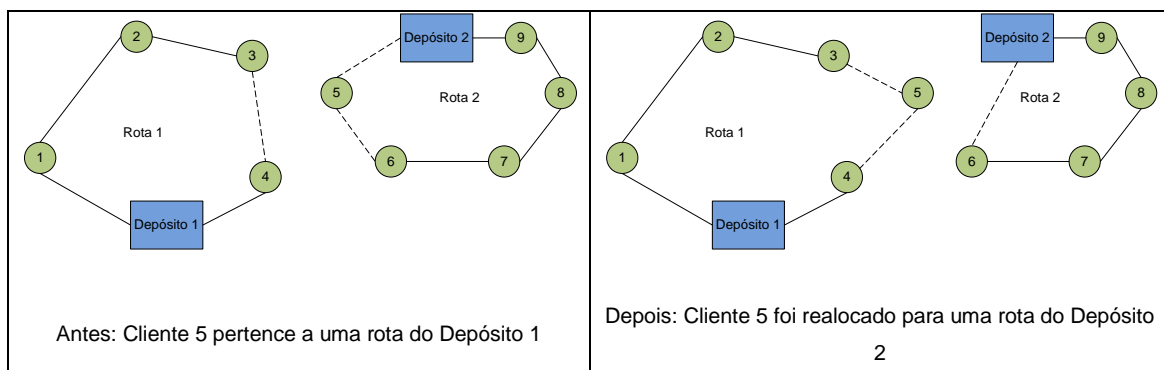
### 5.2.1 Re-alocação

O movimento de realocação pode ser realizado dentro de uma mesma rota (inter-rota), e entre rotas distintas (intra-rota) sendo que as rotas não precisam pertencer ao mesmo depósito (intra-depósito). O movimento só pode ser aceito se gerar uma solução viável melhor do que a solução original. Tal movimento remove um cliente de uma rota para inseri-lo em outra. Neste e nos outros movimentos temos duas estratégias: i) Melhor Movimento e ii) Primeiro Movimento Encontrado.

Melhor Movimento: Testam-se todas as possibilidades de inserção na presente solução, executando-se aquela inserção que tenha maior impacto ao minimizar a FO, respeitando a viabilidade. Tal opção consome maior tempo computacional para ser executada.

Primeiro Movimento Encontrado: A primeira inserção calculada que minimize a FO e respeite a viabilidade é executada.

Na Figura 4.2 temos um exemplo do movimento para o caso inter-rotas. O Cliente 5 pertencia a Rota 1 e foi realocado para uma determinada posição na Rota 2.

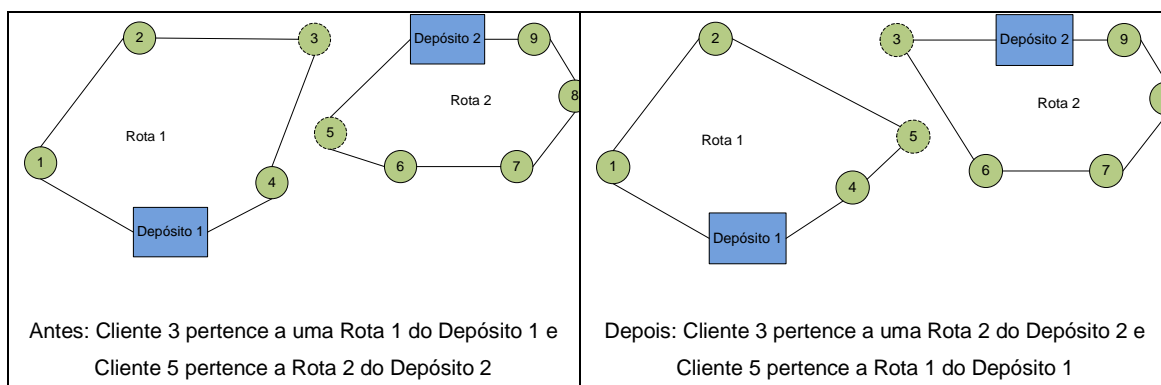


**Figura 4.2:** Exemplo de Movimento de Inserção Inter-Depósito

### 5.2.2 Intercâmbio

O movimento de intercâmbio também pode ser realizado dentro de uma mesma rota (intra-rota), e entre rotas distintas (inter-rota) sendo que as rotas não precisam pertencer ao mesmo depósito (intra-depósito). O movimento só pode ser aceito se gerar uma solução viável melhor do que a solução original. Este movimento seleciona dois clientes da solução e os troca de rotas, respeitando a antiga ordem de clientes das rotas originais.

Na Figura 4.3 podemos observar um exemplo de um movimento de intercâmbio inter-depósitos. Os clientes 3 e 5, trocam de rotas, mantendo a mesma posição que o cliente trocado possuía na rota. Veja que o Cliente 3, originariamente da Rota 1, precedia do Cliente 2 e procedia o Cliente 4. O Cliente 5 que substituiu o Cliente 3 em sua rota, possui os mesmos vizinhos na ordem da Rota. Neste movimento também temos as duas estratégias, já descritas: i) Melhor Movimento e ii) Primeiro Movimento Encontrado.

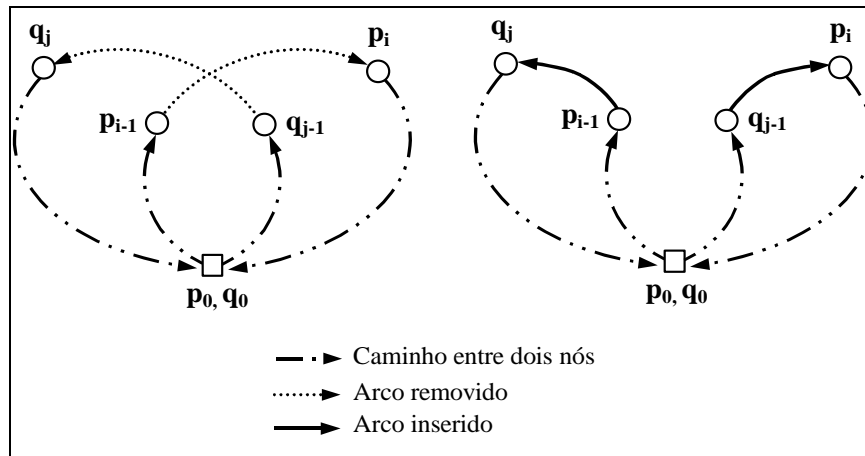


**Figura 4.3:** Exemplo de Movimento de Intercâmbio Inter-Depósito

### 5.2.3 Cruzamento

Tal movimento somente pode ser realizado entre rotas diferentes de um mesmo depósito, caso contrário geraria a inviabilidade da solução. Neste movimento seleciona-se um par de rotas. Cada rota é dividida em duas seções que são re-conectadas. Consegue-se isso removendo um arco de cada rota e substituindo-os por

dois novos arcos que conectam respectivamente a seção inicial da primeira rota com a seção final da segunda rota e a seção inicial da segunda com a seção final da primeira. Na Figura 5.4 ilustramos este movimento para um par de rotas; a rota  $p_0-p_{i-1}-p_i-p_0$  é substituída pela rota  $p_0-p_{i-1}-q_j-p_0$  e a rota  $q_0-q_{j-1}-q_j-q_0$  é substituída pela rota  $q_0-q_{j-1}-p_i-q_0$ .

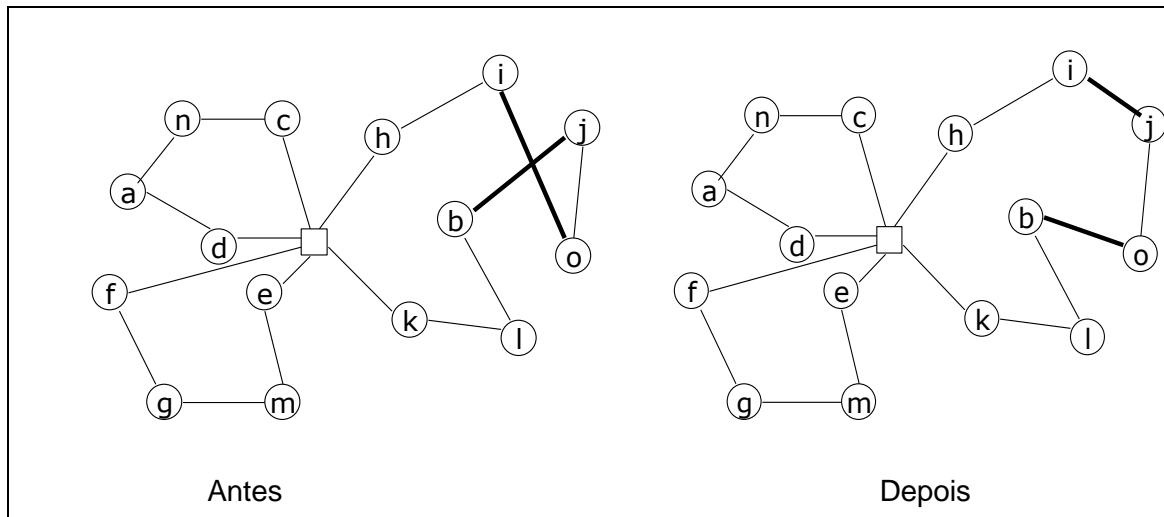


**Figura 4.4.** Exemplo de Movimento de Crossover

O movimento só pode ser aceito se gerar uma solução viável melhor do que a solução original. Neste movimento também temos as duas estratégias, já descritas: i) Melhor Movimento e ii) Primeiro Movimento Encontrado. A Figura 4.4 foi retirado de Sosa et al (2007), cujo o autor desta dissertação é co-autor.

#### 5.2.4 2-opt

Este movimento intra-rotas utiliza a heurística *2-opt*, proposta por LIN (1965); para melhor visualização deste movimento na Figura 4.6 os arcos  $(i,o)$  e  $(b,j)$  são trocados pelos arcos  $(i,j)$  e  $(b,o)$  respectivamente. A Figura 4.5 foi retirado de Sosa et al (2007), cujo o autor desta dissertação é co-autor.



**Figura 4.5.** Exemplo de movimento 2-opt.

### 5.2.5 Métodos de Melhoria implementados

Dois métodos de melhoria foram construídos utilizando os movimentos descritos. O método Seqüencial e o método Paralelo. No Método Seqüencial aplicamos cada um dos movimentos, até que nenhuma melhoria seja possível, seqüencialmente. Já no Método Paralelo, calculamos o impacto de melhoria na FO de cada um dos movimentos para então aplicar o movimento que apresenta maior melhoria na FO.

#### Método Paralelo

**Passo 1:** Seja  $contador = 0$

Enquanto (  $contador < 2$  )

**Passo 2.** Aplicam-se os movimentos seguidos de Realocação até que nenhum outro movimento de inserção seja possível.

**Passo 3.** Aplicam-se os movimentos seguidos de Intercâmbio até que nenhum outro movimento de intercâmbio seja possível.

**Passo 4.** Caso o movimento de Cruzamento esteja Habilitado, aplicam-se movimentos seguidos de Cruzamento até que nenhum outro movimento de cruzamento seja possível.

**Passo 4.** Aplicam-se movimentos seguidos de 2-opt até que nenhum outro movimento 2-opt seja possível.

**Passo 5.**  $contador = contador + 1$

## Método Seqüencial

**Passo 1.** *existenciaMovimentos = verdadeiro*

Enquanto ( *existenciaMovimentos = verdadeiro* )

**Passo 2.** Calcule contribuição na FO do movimento de Realocação.

**Passo 3.** Calcule contribuição na FO do movimento de Intercâmbio.

**Passo 4.** Calcule contribuição na FO do movimento de Cruzamento, se o movimento de cruzamento estiver habilitado.

**Passo 5.** Executar o movimento que apresentar melhor contribuição na FO (calculado nos Passos 2 a 4). Se todos os movimentos apresentarem contribuição nula, faça *existenciaMovimentos = falso*

**Passo 4.** Aplicam-se movimentos seguidos de 2-opt até que nenhum outro movimento 2-opt seja possível.

### 5.3 Construção do Conjunto de Referência

Esta etapa é encontrada em todas as aplicações de Busca Dispersa, sem muitas variações. Aplicamos sem alterações como descrito por MARTÍ et. al. (2006). Destacaremos nesta seção a construção de *Refset* e sua atualização.

O conjunto de referência *Refset* é contruído por soluções com diversidade e com soluções com qualidade. Sejam as soluções de qualidade formadas pelo conjunto  $\{x^1, \dots, x^{b_1}\}$  e as soluções com diversidade formadas pelo conjunto  $\{x^{b_1+1}, \dots, x^b\}$ . Seja o tamanho de *Refset*  $b=b_1+b_2$ . Seja  $f(x)$  o valor da função objetivo. Os valores de  $b_1$  e  $b_2$  são parâmetros para execução.

Como descrevo em SOSA et al (2007), para problemas de roteamento de veículos destacam-se o método de atualização de *Refset* estático por qualidade, por isso aplicamos este método nesta implementação. Definimos a distância entre duas soluções  $x$  e  $y$ ,  $d(x, y) = \sum_{k=1, \dots, n} h_k$  ( $n$  é o número de clientes) onde  $h_k$  é 0 se o cliente  $k$  pertence à mesma rota nas soluções  $x$  e  $y$ ; e 1 caso contrário.



### Procedimento para criação do Conjunto de Referência

**Passo 1.** Dentre as soluções no conjunto  $P$  (gerado pelo método gerador de soluções iniciais), retire as  $b_1$  melhores soluções (com menor FO) e as adicione ao conjunto de referência *Refset*.

**Passo 2.** Calcule a distância  $d_{\min}(x)$  de cada solução  $x \in P$  em relação a cada um dos elementos de *Refset*. Esta distância é dada por  $d_{\min}(x) = \min\{d(x, y)\}, y \in \text{Refset}$ .

**Passo 3.** Selecione as  $b_2$  soluções que maximizam a distância mínima calculada, inserindo assim em *Refset*  $b_2$  soluções diversas em relação as soluções de qualidade.

**Passo 4.** Ordene *Refset* em ordem decrescente de  $f(x)$ .

### 5.4 Geração de Subconjuntos

O objetivo deste método é selecionar pares de soluções do conjunto de referência para serem combinadas. Este método somente seleciona pares de soluções que não tenham sido ainda combinadas em etapas anteriores, evitando o cálculo da combinação de soluções cujo resultado já fora atingido em iterações anteriores da BD. Para isso mantemos o rastreamento dos pares que já foram combinados em iterações anteriores.

Após ter sido contruído o *Refset*, realiza-se a combinação das soluções pertencentes ao *Refset*, que combinará todas as soluções, uma vez que após a criação de *Refset* nenhuma solução fora combinada. Esta estratégia é descrita em CAMPOS *et al.* (1999, 2001) e no trabalho de MARTÍ e LAGUNA (2006).

### 5.5 Combinação de Soluções

O método de geração de subconjuntos seleciona dois pares de soluções para serem combinados pelo Método de Combinação de Soluções. Este método funciona da seguinte maneira:

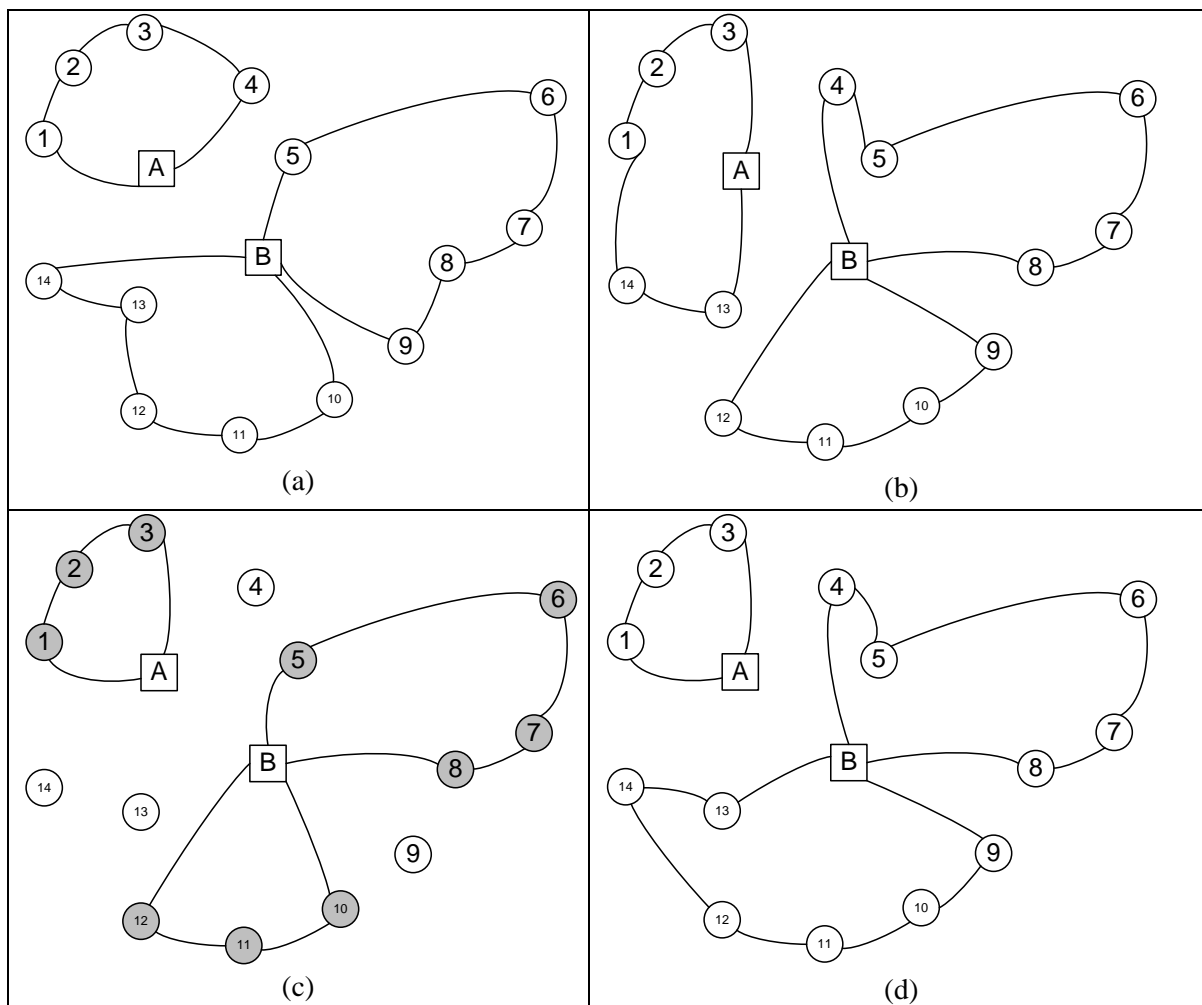
**Passo 1.** Identificam-se os clientes que pertencem às mesmas rotas (e depósitos) em soluções a serem combinadas. Esses clientes são fixados às rotas e respectivos depósitos; os clientes não fixados são desconectados das mesmas;

**Passo 2.** Forme rotas parciais aos clientes fixados, respeitando a ordem que se encontravam nas soluções iniciais.

**Passo 3.** Para cada cliente não fixado, tente fixar em todas as possíveis posições de todas as rotas existentes. Fixe na rota e posição que tenha menor impacto na função objetivo, respeitando a viabilidade. Caso não seja viável inserir em nenhuma rota existente, abra uma nova rota no depósito mais próximo.

**Passo 4.** Aplica-se o Método de Melhoria a nova solução. Esta solução é armazenada em uma lista denominada *Pool*.

Na Figura 4.6 ilustra-se o método de combinação para um problema com 14 clientes em que as duas soluções a serem combinadas têm três veículos cada. As Figuras 4.6(a) e 4.6(b) mostram as duas soluções que serão combinadas. Na Figura 4.6(c) clientes comuns das soluções 1 e 2 são identificados e fixados nas rotas respectivas; a obtenção da nova solução é feita de modo que sejam formadas rotas segundo o algoritmo descrito acima, conforme mostramos na Figura 4.6(d).



**Figura 4.6** –Exemplo de Método de Combinação (a) Solução 1; (b) Solução 2; (c) Clientes comuns; (d) Solução resultante.

Analogamente também foi implementado um método de combinação d soluções baseado em clientes comuns. Ao invés de se fixar clientes fixamos arcos comuns. Os passos para a combinação são:

**Passo 1.** Identificam-se os arcos que pertencem às mesmas rotas (e depósitos) em soluções a serem combinadas. Tais arcos são fixados às rotas e respectivos depósitos; os clientes que não possuem arcos fixados são temporariamente retirados da solução;

**Passo 2.** Forme rotas parciais aos arcos fixados;

**Passo 3.** Para cada cliente não fixado, tente fixar em todas as possíveis posições de todas as rotas existentes. Fixe na rota e posição que tenha menor impacto na função objetivo, respeitando a viabilidade. Caso não seja viável inserir em nenhuma rota existente, abra uma nova rota no depósito mais próximo.

**Passo 4.** Aplica-se o Método de Melhoria a nova solução. Esta solução é armazenada em uma lista denominada *Pool*.

#### 5.6 Atualização do Conjunto de Referência

A solução encontrada pelo método de combinação é armazenada em uma lista denominada *Pool*. *Refset* consistirá das  $b_1$  melhores soluções, baseado na FO, armazenadas em  $Refset \cup Pool$ . Em relação às  $b_1$  melhores soluções calcula-se a diversidade das outras soluções em *Pool*, selecionando-se  $b_2$  soluções mais diversas. Uma solução é considerada diversa em relação as soluções de qualidade quando o número de clientes comuns a rotas é pequeno, como já descrito.

#### 5.7 Critério de Parada da BD

Se na atualização de *Refset* novas soluções são introduzidas, ativa-se novamente o método da combinação de soluções. O critério de parada para a BD acontece quando o conjunto de soluções em *Refset* não foi modificado, isto é, quando este conjunto não foi atualizado na última iteração da BD.

## CAPÍTULO 6

### RESULTADOS COMPUTACIONAIS

Neste capítulo mostraremos os resultados computacionais referentes a implementação da Busca Dispersa descrita no Capítulo 5. Analisaremos o comportamento da metaheurística em relação aos seus parâmetros e as variações disponíveis nos métodos da BD. Iniciaremos analisando a escolha das instâncias *benchmark*. Depois analisaremos os testes realizados para a definição dos parâmetros da BD. Por fim analisaremos os resultados comparando com os obtidos por outras heurísticas. O código foi implementado usando linguagem C++ e compilado com Visual Studio 2008. O código gerado foi otimizado para o processador 64 bits AMD Athlon 64 X2 2GHz. O código gerado não utiliza recursos de paralelismo.

Os resultados serão comparados em relação a FO e ao tempo computacional requerido para encontrar as soluções. Para comparar a FO encontrada pela BD com as soluções existentes na literatura usaremos o GAP, que nesta dissertação definimos como o percentual de diferença entre a solução encontrada e a melhor solução disponível na literatura. Constantemente utilizaremos a média de GAPs entre as soluções das 23 instâncias que utilizaremos para comparação com outras heurísticas.

#### 6.1 Instâncias

Para selecionar as instâncias a serem utilizadas buscou-se na literatura instâncias que tivessem sido utilizadas em mais de um artigo. NAGY e SALHI (2005) utilizaram as 11 instâncias propostas por CHRISTOFIDES e EILON (1969) e as de GILLET e JOHNSON (1976). Tais instâncias contêm entre 50 e 249 clientes e de 2 a 5 depósitos. Já BAE et al (2007) propõe instancias próprias para resolver o problema entre 10 e 99 clientes. RENAUD et al (1996) utiliza as mesmas 11 instancias de CHRISTOFIDES e EILON (1969) e GILLET e JOHNSON (1976) e também testam com 12 instancias propostas por CHAO et al. (1993). HO et al (2007) utilizam instâncias próprias geradas aleatoriamente com 50 e 100 clientes ambas com dois depósitos. Por fim BALDACCI e MINGOZZI (2009) utilizam as instâncias propostas por CHRISTOFIDES e EILON (1969) e GILLET e JOHNSON (1976). Já PISSINGER e ROPKE (2007) utilizam as instâncias propostas por CHRISTOFIDES e EILON (1969),

GILLETT e JOHNSON (1976) e CHAO et al (1993). Não conseguimos obter as instâncias criadas por HO et al (2007), com tamanho máximo de 100 clientes e de BAE et al (2007), com tamanho máximo de 99 clientes.

Na Tabela 5.1 podemos observar as instâncias utilizadas por cada autor. Pela disponibilidade das instâncias para execução dos testes verificamos que utilizamos todas as instâncias disponíveis utilizadas por mais de um autor:

- 1) CHRISTOFIDES e EILON (1969) propuseram 7 instâncias, sem limites de distância e utilizando limites de capacidade de veículo, variando entre 2 e 5 depósitos e entre 50 e 100 clientes.
- 2) GILLETT e JOHNSON (1976) propuseram 4 instâncias variando de 2 a 5 depósitos com 249 clientes, utilizando restrições de capacidade e máxima distância para os veículos.
- 3) CHAO et al (1993) propuseram 12 instancias variando de 2 a 9 depósitos e de 80 a 360 clientes, utilizando restrições de capacidade e máxima distância para os veículos.

|                            | Instância de CHRISTOFIDES e EILON (1969) | Instância de GILLETT e JOHNSON (1976) | Instância de CHAO et al. (1993) | Instância de HO et al (2007) | Instância de BAE et al (2007) |
|----------------------------|--|---------------------------------------|---------------------------------|------------------------------|-------------------------------|
| GILLETT e JOHNSON (1976)   | •  | •                                     |                                 |                              |                               |
| CHAO et al. (1993)         | •  | •                                     | •                               |                              |                               |
| RENAUD et al (1996)        | •  | •                                     | •                               |                              |                               |
| NAGY e SALHI (2005)        | •  | •                                     |                                 |                              |                               |
| PISSINGER e ROPKE (2007)   | •  | •                                     | •                               |                              |                               |
| BAE et al (2007)           |  |                                       |                                 |                              | •                             |
| HO et al (2007)            |  |                                       |                                 | •                            |                               |
| BALDACCI e MINGOZZI (2009) | •  | •                                     |                                 |                              |                               |

**Tabela 5.1:** Uso de Instâncias pela Literatura.

| Instância | Autores da Instância        | Número de Depósitos | Número de Clientes | Capacidade do Veículo | Limite de Distância de Cada Rota |
|-----------|-----------------------------|---------------------|--------------------|-----------------------|----------------------------------|
| p01       | Christofides e Eilon (1969) | 4                   | 50                 | 80                    | $\infty$                         |
| p02       | Christofides e Eilon (1969) | 4                   | 50                 | 160                   | $\infty$                         |
| p03       | Christofides e Eilon (1969) | 5                   | 75                 | 140                   | $\infty$                         |
| p04       | Christofides e Eilon (1969) | 2                   | 100                | 100                   | $\infty$                         |
| p05       | Christofides e Eilon (1969) | 2                   | 100                | 200                   | $\infty$                         |
| p06       | Christofides e Eilon (1969) | 3                   | 100                | 100                   | $\infty$                         |
| p07       | Christofides e Eilon (1969) | 4                   | 100                | 100                   | $\infty$                         |
| p08       | Gillett e Johnson (1976)    | 2                   | 249                | 500                   | 310                              |
| p09       | Gillett e Johnson (1976)    | 3                   | 249                | 500                   | 310                              |
| p10       | Gillett e Johnson (1976)    | 4                   | 249                | 500                   | 310                              |
| p11       | Gillett e Johnson (1976)    | 5                   | 249                | 500                   | 310                              |
| p12       | Chao et al. (1993)          | 2                   | 80                 | 60                    | $\infty$                         |
| p13       | Chao et al. (1993)          | 2                   | 80                 | 60                    | 200                              |
| p14       | Chao et al. (1993)          | 2                   | 80                 | 60                    | 180                              |
| p15       | Chao et al. (1993)          | 4                   | 160                | 60                    | $\infty$                         |
| p16       | Chao et al. (1993)          | 4                   | 160                | 60                    | 200                              |
| p17       | Chao et al. (1993)          | 4                   | 160                | 60                    | 180                              |
| p18       | Chao et al. (1993)          | 6                   | 240                | 60                    | $\infty$                         |
| p19       | Chao et al. (1993)          | 6                   | 240                | 60                    | 200                              |
| p20       | Chao et al. (1993)          | 6                   | 240                | 60                    | 180                              |
| p21       | Chao et al. (1993)          | 9                   | 360                | 60                    | $\infty$                         |
| p22       | Chao et al. (1993)          | 9                   | 360                | 60                    | 200                              |
| p23       | Chao et al. (1993)          | 9                   | 360                | 60                    | 180                              |

**Tabela 5.2:** Características das instâncias utilizadas para os testes computacionais.

A Tabela 5.2 apresenta as características de cada uma das instâncias utilizadas, 23 instâncias ao todo. Os principais fatores indicados como características das instâncias são o número de depósitos; número de clientes; capacidade dos veículos e o limite de distância de cada rota a ser formada. Indicamos também a autoria de cada instância. Todas as instâncias tem limite de capacidade do veículo e somente algumas tem limite de distância. As instâncias mais difíceis de serem resolvidas são aquelas com menor limite de distância (quando existe o limite) e também com menor limite de capacidade do veículo. As instâncias com maior número de clientes são as mais demoradas para serem resolvidas. Quanto ao tamanho das instâncias (relativamente ao número de clientes), utilizamos as maiores instâncias disponíveis na literatura. As instâncias que não obtivemos acesso - HO et al (2007) e BAE et al (2007) - utilizam instâncias de tamanho máximo de 100 clientes. As

instâncias testadas apresentam variação de 50 a 360 clientes e com número de depósitos variando entre 2 e 9.

## 6.2 Ajuste dos parâmetros para a busca dispersa

Diversos são os parâmetros a serem definidos na implementação, descrita no Capítulo 5, da Busca Dispersa para o PRVMD. Precisaremos ajustar tais parâmetros para chegar na versão final da busca dispersa. Os principais grupos de parâmetros são:

- 1) Definição do melhor método de geração de soluções iniciais
- 2) Definição do parâmetro alfa do método de geração de soluções iniciais
- 3) Definição de estratégias do método de melhoria
- 4) Definição da estratégia para o método de combinação de soluções
- 5) Definição do tamanho do conjunto de referência (*Refset*)
- 6) Tamanho do conjunto P de soluções iniciais

Para chegar a um conjunto destes parâmetros para compor a versão final da Busca Dispersa para o PRVMD adotamos a seguinte estratégia: para cada parâmetro a ser definido consideramos fixos (por um valor previamente ou arbitrariamente definido) todos os outros parâmetros e variamos apenas o parâmetro em teste, para então avaliarmos o resultado. Assim, repetindo tal procedimento, parâmetro a parâmetro, chegaremos a combinação final de valores.

Iniciamos com a definição do parâmetro *alfa* da metodologia GRASP e dos três métodos de geração de soluções iniciais. Para isso realizamos os seguintes testes: para cada parâmetro *alfa* variando de 0 a 1 (com incrementos de 0,1) executamos o método gerador de soluções iniciais para todas as instâncias e calculamos a média de GAPs em relação aos melhores resultados disponíveis na literatura. Tais resultados podem ser encontrados na Tabela 5.3, nas colunas “Sem Melhoria”. Já a coluna “Com Melhoria” apresenta o mesmo resultado, porém aplicando-se em seguida o método de melhoria seqüencial para todas as soluções. Podemos verificar que quanto maior o valor de *alfa* mais diversas são as soluções. Quando *alfa* é igual a zero, caso das heurísticas clássicas, melhor o GAP. Para este teste consideramos o método de melhoria seqüencial, com cruzamento e estratégia primeiro movimento encontrado.



| Alfa         | Menor Distância |               | Gillett e Miller |               | Médias         |               |
|--------------|-----------------|---------------|------------------|---------------|----------------|---------------|
|              | Sem Melhoria    | Com Melhoria  | Sem Melhoria     | Com Melhoria  | Sem Melhoria   | Com Melhoria  |
| 0,0          | 219,32%         | 19,36%        | 234,15%          | 19,74%        | <b>189,66%</b> | <b>19,21%</b> |
| 0,1          | 251,00%         | 22,25%        | 312,78%          | 21,21%        | <b>204,83%</b> | <b>20,97%</b> |
| 0,2          | 333,36%         | 20,71%        | 330,62%          | 20,02%        | <b>255,32%</b> | <b>19,96%</b> |
| 0,3          | 367,52%         | 22,81%        | 363,73%          | 20,65%        | <b>296,45%</b> | <b>21,82%</b> |
| 0,4          | 369,09%         | 21,80%        | 389,54%          | 20,92%        | <b>323,23%</b> | <b>21,20%</b> |
| 0,5          | 359,52%         | 22,57%        | 353,90%          | 21,24%        | <b>355,87%</b> | <b>22,09%</b> |
| 0,6          | 405,47%         | 24,24%        | 404,27%          | 22,07%        | <b>366,70%</b> | <b>23,87%</b> |
| 0,7          | 370,98%         | 25,52%        | 422,68%          | 23,02%        | <b>370,64%</b> | <b>24,30%</b> |
| 0,8          | 421,80%         | 21,76%        | 428,15%          | 24,62%        | <b>422,58%</b> | <b>23,89%</b> |
| 0,9          | 431,91%         | 22,98%        | 425,13%          | 26,05%        | <b>421,46%</b> | <b>24,97%</b> |
| 1,0          | 412,78%         | 23,99%        | 430,46%          | 21,84%        | <b>427,56%</b> | <b>23,92%</b> |
| <b>Média</b> | <b>358,43%</b>  | <b>22,54%</b> | <b>372,31%</b>   | <b>21,94%</b> | <b>330,39%</b> | <b>22,38%</b> |

**Tabela 5.3:** Testes variando o parâmetro *alfa*.

Escolhemos o método gerador de soluções iniciais baseado na heurística de GILLETT e MILLER (1974), por apresentar melhores médias gerais e ainda apresentar diversidade de soluções dado pelo GRASP. Em relação ao parâmetro *alfa* da metodologia GRASP, ignoramos os resultados apresentados e utilizamos o valor de 0,5, por ser recomendado pela literatura (por introduzir diversidade, fator importante para a busca dispersa) e pois apresentou bom resultado no teste final, como veremos adiante. Para os próximos testes consideramos fixados os parâmetros *alfa* igual a 0,5 e o método gerador de soluções iniciais baseado na heurística de GILLETT e MILLER (1974),

Para o método de melhoria precisamos escolher:

- i) Estratégia relativa aos movimentos selecionados: “Primeiro Movimento Encontrado” ou “Melhor de Todos”.
- ii) Se ativaremos ou não o movimento de cruzamento.
- iii) Se utilizaremos o método paralelo ou o método seqüencial.

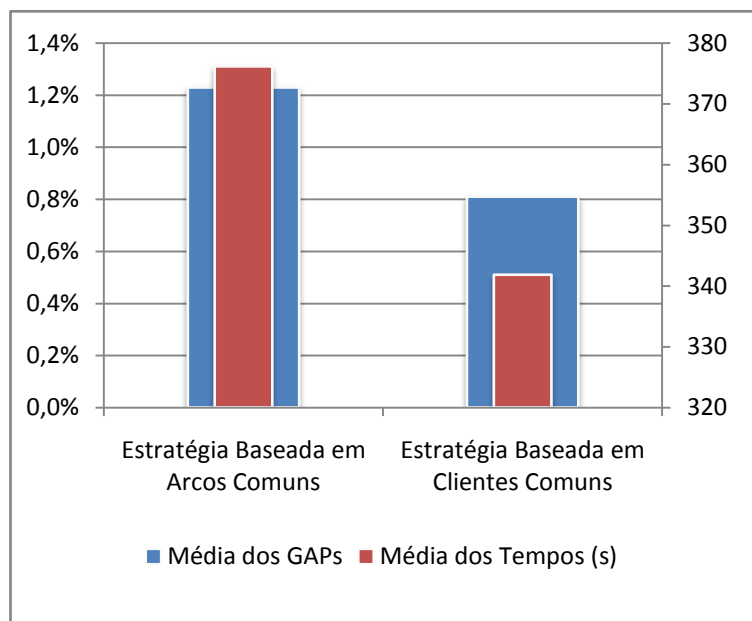
Executamos para todas as 23 instâncias disponíveis todas as combinações possíveis com estas três decisões e observamos o tempo computacional e o a média dos GAPs para cada combinação. Estes resultados estão na Tabela 5.4. Verificamos

que o melhor resultado em relação ao GAP se dá quando utilizamos o método Seqüencial, com Cruzamento e o Método Paralelo, ainda que não seja a opção que apresente a menor média de tempo computacional. Para este teste consideramos o tamanho de Refset (2,8) para os tamanhos de qualidade e diversidade.

| <b>Estratégia: Primeiro Movimento Encontrado</b> |                |                      |                |                      |
|--|----------------|----------------------|----------------|----------------------|
| Com Cruzamento                                   |                | Sem Cruzamento       |                |                      |
|  | Média dos GAPs | Média dos Tempos (s) | Média dos GAPs | Média dos Tempos (s) |
| Método Paralelo                                  | 0,78%          | 289,25               | 1,02%          | 208,27               |
| Método Seqüencial                                | 0,72%          | 203,82               | 0,86%          | 187,47               |
| <b>Estratégia: Melhor de Todos</b>               |                |                      |                |                      |
| Com Cruzamento                                   |                | Sem Cruzamento       |                |                      |
|  | Média dos GAPs | Média dos Tempos (s) | Média dos GAPs | Média dos Tempos (s) |
| Método Paralelo                                  | 0,82%          | 502,29               | 1,18%          | 269,64               |
| Método Seqüencial                                | 0,75%          | 345,69               | 0,89%          | 217,81               |

**Tabela 5.4:** Testes de estratégias para o método de melhoria.

Para escolher a estratégia do método de combinação de soluções testamos as duas combinações implementadas (baseada em arcos comuns e em clientes comuns, como veremos adiante) utilizando os parâmetros já fixados e o tamanho de *Refset* (2,8) para os tamanhos do *Refset* de diversidade e qualidade; e executando para as 23 instâncias disponíveis. Os resultados são apresentados na Figura 5.1. Notamos que a estratégia baseada em clientes comuns apresenta melhor resultado, tempo na média dos GAPs quanto na média de tempo computacional, logo optamos por esta estratégia, que deverá ser considerada fixa para os próximos testes.



**Figura 5.1:** Testes para o método de combinação

Para a escolha do tamanho de *Refset* testamos quatro possibilidades, frequentemente utilizados na literatura, variando entre o tamanho do *Refset* qualidade e do *Refset* diversidade. Utilizamos o tamanho de soluções iniciais igual a 50. O resultado está na Tabela 5.5. O resultado (2,8) para os tamanhos de qualidade e diversidade, respectivamente apresentam a melhor relação entre a média dos GAPs e a média dos tempos computacionais e serão fixadas nestes valores para a versão final da BD.

| Tamanho <i>Refset</i> Qualidade | Tamanho <i>Refset</i> Diversidade | Média dos GAPs | Média dos Tempos (s) |
|---------------------------------|-----------------------------------|----------------|----------------------|
| 2                               | 8                                 | 0,72%          | 203,82               |
| 5                               | 5                                 | 0,81%          | 341,87               |
| 8                               | 2                                 | 1,98%          | 233,12               |
| 10                              | 10                                | 0,69%          | 692,82               |

**Tabela 5.5:** Testes para o tamanho de *Refset*.

O último parâmetro a ser definido é o tamanho do conjunto P, que contém as soluções iniciais. Quanto maior o tamanho de P, maior será a busca e também o tempo computacional. Para este teste consideramos todos os parâmetros já fixados anteriormente e testamos algumas variações de parâmetros do tamanho deste conjunto, observando o tempo computacional e a média de GAPs em relação as melhores soluções encontradas na literatura. Os resultados podem ser observados na

Tabela 5.6. O parâmetro escolhido foi o do tamanho igual a 50, já que apresenta melhor relação entre média de GAPs e Média dos tempos computacionais.

| Tamanho de P | Média de GAPs | Média de Tempos (s) |
|--------------|---------------|---------------------|
| 25           | 2,02%         | 58,37               |
| 50           | 0,72%         | 203,82              |
| 75           | 0,71%         | 602,53              |
| 100          | 0,69%         | 1763,74             |

**Tabela 5.6:** Testes para o tamanho do conjunto P

O resumo das opções testadas e as escolhas estão resumidas na Tabela 5.7. Consideramos estes parâmetros para os testes computacionais finais apresentados adiante.

| Parâmetro/Estratégia                          | Opção Definida                                   |
|---|--|
| Estratégia do Geração de Soluções Iniciais    | baseado na heurística de GILLETT e MILLER (1974) |
| Valor <i>alfa</i> do GRASP                    | 0,5  |
| Estratégia de Movimento                       | Primeiro Movimento Encontrado                    |
| Utilização do Movimento de Cruzamento         | Com Cruzamento                                   |
| Estratégia do Método de Melhoria              | Método Sequencial                                |
| Estratégia do Método de Combinação            | Estratégia Baseada em Clientes Comuns            |
| Tamanho do conjunto <i>Refset</i> diversidade | 8  |
| Tamanho do conjunto <i>Refset</i> qualidade   | 2  |
| Tamanho do conjunto P de soluções iniciais    | 50   |

**Tabela 5.7:** Parâmetros adotados para a DB.

### 6.3 Resultados Computacionais

Selecionadas as instâncias benchmark, e tendo os parâmetros fixos desta implementação de BD, vamos comparar os resultados obtidos com os resultados de outras implementações encontradas na literatura. Iremos também analisar a convergência da BD implementada e sua estabilidade.

Na Tabela 5.8, apresentamos, para cada instância, os resultados da nossa implementação da BD. Em negrito, nesta e em todas as demais tabelas, estão destacadas as melhores soluções conhecidas até então.

Notamos que para instâncias com menor número de clientes, a BD chega a um resultado com GAP baixo. Os maiores GAPS encontrados pela busca dispersa referem-se a instâncias com maior número de clientes, sugerindo que quanto maior o tamanho da instância, mais difícil é para a DB obter um resultado com baixo GAP.

Notamos, ainda, que o tempo computacional para resolver cada instância também é diretamente proporcional ao tamanho das mesmas, ou seja, quanto maior a instância, maior o tempo computacional para resolvê-la.

| Instância     | Instância Proposta Por      | Número de Depósitos | Número de Clientes | Melhor Solução Encontrada na Literatura | Solução Encontrada pela Busca Dispersa | Tempo (s) | GAP          |
|---------------|-----------------------------|---------------------|--------------------|---|--|-----------|--------------|
| p01           | Christofides e Eilon (1969) | 4                   | 50                 | <b>576,87</b>                           | <b>576,87</b>                          | 5,6       | 0%           |
| p02           | Christofides e Eilon (1969) | 4                   | 50                 | <b>473,53</b>                           | <b>473,53</b>                          | 4,0       | 0%           |
| p03           | Christofides e Eilon (1969) | 5                   | 75                 | <b>640,65</b>                           | 642,92                                 | 28,4      | 0,35%        |
| p04           | Christofides e Eilon (1969) | 2                   | 100                | <b>999,21</b>                           | 1007,10                                | 100,0     | 0,79%        |
| p05           | Christofides e Eilon (1969) | 2                   | 100                | <b>750,26</b>                           | 751,55                                 | 20,0      | 0,17%        |
| p06           | Christofides e Eilon (1969) | 3                   | 100                | <b>876,5</b>                            | 881,90                                 | 99,9      | 0,62%        |
| p07           | Christofides e Eilon (1969) | 4                   | 100                | <b>881,97</b>                           | 900,38                                 | 80,6      | 2,09%        |
| p08           | Gillett and Johnson (1976)  | 2                   | 249                | <b>4421,03</b>                          | 4485,57                                | 288,8     | 1,46%        |
| p09           | Gillett and Johnson (1976)  | 3                   | 249                | <b>3669,31</b>                          | 3722,10                                | 275,4     | 1,44%        |
| p10           | Gillett and Johnson (1976)  | 4                   | 249                | <b>3666,85</b>                          | <b>3666,85</b>                         | 348,7     | 0%           |
| p11           | Gillett and Johnson (1976)  | 5                   | 249                | <b>3573,23</b>                          | 3598,47                                | 257,0     | 0,71%        |
| p12           | Chao et al. (1993)          | 2                   | 80                 | <b>1318,95</b>                          | 1319,13                                | 76,7      | 0,01%        |
| p13           | Chao et al. (1993)          | 2                   | 80                 | <b>1318,95</b>                          | <b>1318,95</b>                         | 55,6      | 0%           |
| p14           | Chao et al. (1993)          | 2                   | 80                 | <b>1360,12</b>                          | 1362,98                                | 43,4      | 0,21%        |
| p15           | Chao et al. (1993)          | 4                   | 160                | <b>2505,42</b>                          | 2544,03                                | 172,3     | 1,54%        |
| p16           | Chao et al. (1993)          | 4                   | 160                | <b>2572,23</b>                          | <b>2572,23</b>                         | 139,9     | 0%           |
| p17           | Chao et al. (1993)          | 4                   | 160                | <b>2709,09</b>                          | 2779,67                                | 116,5     | 2,61%        |
| p18           | Chao et al. (1993)          | 6                   | 240                | <b>3736,53</b>                          | <b>3736,53</b>                         | 297,2     | 0%           |
| p19           | Chao et al. (1993)          | 6                   | 240                | <b>3827,06</b>                          | 3867,21                                | 205,6     | 1,05%        |
| p20           | Chao et al. (1993)          | 6                   | 240                | <b>4064,76</b>                          | <b>4064,76</b>                         | 254,7     | 0%           |
| p21           | Chao et al. (1993)          | 9                   | 360                | <b>5501,58</b>                          | 5568,77                                | 738,4     | 1,22%        |
| p22           | Chao et al. (1993)          | 9                   | 360                | <b>5718</b>                             | 5762,19                                | 528,7     | 0,77%        |
| p23           | Chao et al. (1993)          | 9                   | 360                | <b>6092,66</b>                          | 6189,48                                | 550,5     | 1,59%        |
| <b>Média:</b> |                             |                     |                    |   |  |           | <b>0,72%</b> |

**Tabela 5.8:** Resultados da BD para as instâncias *benchmark*.

Segue agora a comparação com as principais implementações disponíveis na literatura. Iniciaremos por CHAO et al. (1993), que propõem uma heurística de múltiplas fases na qual inicialmente os clientes são associados ao depósito mais próximo. Em seguida o PRV Clássico é resolvido para cada depósito. Na fase seguinte tenta-se realocar os clientes para outros depósitos visando melhorar a solução. A Tabela 5.9 apresenta os tempos computacionais e a F.O. de ambas as implementações: de CHAO et al. (1993) e da Busca Dispersa. Os tempos computacionais de CHAO et al. (1993) foram obtidos em um Sun 4/370. Notamos que a BD tem grande vantagem, considerando a FO, tendo melhores resultados para a maioria das instâncias. Das 23 instâncias comparadas, apenas em quatro delas a BD não encontrou melhores resultados do que CHAO et al. (1993) A comparação computacional será feita adiante, considerando que os processadores utilizados são diferentes.

| Instância | CHAO et al. (1993) |           | Busca Dispersa |           |
|-----------|--------------------|-----------|----------------|-----------|
|           | F.O.               | Tempo (s) | F.O.           | Tempo (s) |
| p01       | 582,4              | 66        | <b>576,9</b>   | 5,6       |
| p02       | 476,6              | 72        | <b>473,5</b>   | 4,0       |
| p03       | <b>641,2</b>       | 108       | 642,9          | 28,4      |
| p04       | 1026,9             | 132       | <b>1007,1</b>  | 100,0     |
| p05       | 756,6              | 144       | <b>751,6</b>   | 20,0      |
| p06       | 883,6              | 126       | <b>881,9</b>   | 99,9      |
| p07       | <b>898,5</b>       | 288       | 900,4          | 80,6      |
| p08       | <b>4511,6</b>      | 1266      | 4485,6         | 288,8     |
| p09       | 3950,9             | 1254      | <b>3722,1</b>  | 275,4     |
| p10       | 3815,6             | 432       | <b>3666,9</b>  | 348,7     |
| p11       | 3733,0             | 1002      | <b>3598,5</b>  | 257,0     |
| p12       | 1327,3             | 168       | <b>1319,1</b>  | 76,7      |
| p13       | 1345,9             | 42        | <b>1319,0</b>  | 55,6      |
| p14       | 1372,5             | 78        | <b>1363,0</b>  | 43,4      |
| p15       | 2610,3             | 138       | <b>2544,0</b>  | 172,3     |
| p16       | 2605,3             | 366       | <b>2572,2</b>  | 139,9     |
| p17       | 2816,6             | 390       | <b>2779,7</b>  | 116,5     |
| p18       | 3877,4             | 516       | <b>3736,5</b>  | 297,2     |
| p19       | <b>3863,9</b>      | 1338      | 3867,2         | 205,6     |
| p20       | 4272,0             | 876       | <b>4064,8</b>  | 254,7     |
| p21       | 5791,5             | 4710      | <b>5568,8</b>  | 738,4     |
| p22       | 5857,4             | 7944      | <b>5762,2</b>  | 528,7     |
| p23       | 6494,6             | 1464      | <b>6189,5</b>  | 550,5     |

**Tabela 5.9:** Comparação da BD com os resultados obtidos por CHAO et al (1993).

Já PISINGER e ROPKE (2007) propõe uma heurística que é uma extensão do *adaptive large neighborhood search*, enquanto em RENAUD et al. (1996) foi implementada uma Busca Tabu para o PRVMD. A Tabela 5.10 apresenta os tempos computacionais e a F.O. das implementações. Os tempos computacionais de RENAUD et al. (1996) foram obtidos em um Sun Sparcstation 10. Já PISINGER e ROPKE (2007) utilizam um processador AMD Athlon, mesma classe de processador que utilizamos para os testes da BD. A comparação com os tempos computacionais será feita adiante. Em relação a FO, a Busca dispersa supera a Busca Tabu com um GAP médio de todas as instâncias de 0,71% contra 0,86%. Já o *adaptive large neighborhood search* supera a DB com 0,48% de GAP médio contra 0,71% da BD.

| Instância | RENAUD et al. (1996) |           | PISSINGER e ROPKE (2007) |           | Busca Dispersa |           |
|-----------|----------------------|-----------|--------------------------|-----------|----------------|-----------|
|           | F.O.                 | Tempo (s) | F.O.                     | Tempo (s) | F.O.           | Tempo (s) |
| p01       | <b>576,9</b>         | 192       | <b>576,9</b>             | 29,0      | <b>576,9</b>   | 5,6       |
| p02       | <b>473,5</b>         | 288       | <b>473,5</b>             | 28,0      | <b>473,5</b>   | 4,0       |
| p03       | 645,1                | 348       | <b>641,2</b>             | 64,0      | 642,9          | 28,4      |
| p04       | 1016,1               | 684       | <b>1006,09</b>           | 88,0      | 1007,1         | 100,0     |
| p05       | <b>750,3</b>         | 768       | 752,34                   | 120,0     | 751,6          | 20,0      |
| p06       | <b>876,5</b>         | 504       | 883,0                    | 93,0      | 881,9          | 99,9      |
| p07       | 892,6                | 408       | <b>889,4</b>             | 88,0      | 900,4          | 80,6      |
| p08       | 4500,5               | 4164      | <b>4421,0</b>            | 333,0     | 4485,6         | 288,8     |
| p09       | <b>3669,3</b>        | 2472      | 3892,5                   | 361,0     | 3722,1         | 275,4     |
| p10       | 3720,9               | 2580      | <b>3666,9</b>            | 363,0     | <b>3666,9</b>  | 348,7     |
| p11       | 3696,4               | 624       | <b>3573,2</b>            | 357,0     | 3598,5         | 257,0     |
| p12       | <b>1319,0</b>        | 324       | 1319,1                   | 95,0      | 1319,1         | 76,7      |
| p13       | <b>1319,0</b>        | 288       | <b>1319,0</b>            | 60,0      | <b>1319,0</b>  | 55,6      |
| p14       | 1365,7               | 156       | <b>1360,1</b>            | 58,0      | 1363,0         | 43,4      |
| p15       | 2551,5               | 930       | <b>2519,6</b>            | 253,0     | 2544,0         | 172,3     |
| p16       | <b>2572,2</b>        | 666       | 2574,0                   | 188,0     | <b>2572,2</b>  | 139,9     |
| p17       | 2731,4               | 348       | <b>2709,1</b>            | 179,0     | 2779,7         | 116,5     |
| p18       | 3790,0               | 1392      | <b>3736,5</b>            | 419,0     | <b>3736,5</b>  | 297,2     |
| p19       | <b>3827,1</b>        | 1320      | 3838,8                   | 315,0     | 3867,2         | 205,6     |
| p20       | 4097,1               | 600       | <b>4064,8</b>            | 300,0     | <b>4064,8</b>  | 254,7     |
| p21       | 5678,5               | 2922      | <b>5501,6</b>            | 582,0     | 5568,8         | 738,4     |
| p22       | <b>5718,0</b>        | 2010      | 5722,2                   | 462,0     | 5762,2         | 528,7     |
| p23       | 6145,6               | 1038      | <b>6092,7</b>            | 443,0     | 6189,5         | 550,5     |

**Tabela 5.10:** Comparação da BD com os resultados obtidos por PISSINGER e ROPKE (2007) e RENAUD et al. (1996).



A última comparação é feita com um método exato, baseado em geração de colunas, para o PRVMD. Proposto por BALDACCI e MINGOZZI (2009) o problema é testado para 8 instâncias, sendo que para duas, marcadas com “\*”, não é garantida a optimalidade. Seus testes foram realizados em AMD Athlon 64 X2 Dual Core 2.6GHz utilizando paralelismo. Tendo a mesma classe de processadores podemos comparar os tempos computacionais de maneira justa, comparando uma abordagem exata e a DB, heurística, para as instâncias testadas. Notamos que para instâncias de maior número de clientes o tempo da Busca Dispersa se mostra competitivo, tendo resultados próximos aos obtidos pelo método exato, porém com tempo computacional muito mais competitivo, por se tratar de uma heurística. Tais resultados são apresentados na Tabela 5.11. BALDECCI e MINGOZZI (2009) somente utilizam as instâncias de GILLETT e JOHNSON (1976) e testa apenas a instância p15 de CHAO et al (1993), logo as instâncias marcadas com “\*\*\*” não foram testadas pelos autores.

| Instância | Baldacci e MingoZZi (2009) |           | Busca Dispersa |           |
|-----------|----------------------------|-----------|----------------|-----------|
|           | F.O.                       | Tempo (s) | F.O.           | Tempo (s) |
| p01       | <b>576,9</b>               | 10,9      | <b>576,9</b>   | 5,6       |
| p02       | <b>473,5</b>               | 54,4      | <b>473,5</b>   | 4,0       |
| p03       | <b>640,7</b>               | 92,4      | 642,9          | 28,4      |
| p04       | <b>999,2</b>               | 5106,9    | 1007,1         | 100,0     |
| p05       | <b>751,26*</b>             | 1647,6    | 751,6          | 20,0      |
| p06       | <b>876,5</b>               | 104,3     | 881,9          | 99,9      |
| p07       | <b>882,0</b>               | 294,4     | 900,4          | 80,6      |
| p08       | **                         |           | 4485,6         | 288,8     |
| p09       | **                         |           | 3722,1         | 275,4     |
| p10       | **                         |           | 3666,9         | 348,7     |
| p11       | **                         |           | 3598,5         | 257,0     |
| p12       | **                         |           | 1319,1         | 76,7      |
| p13       | **                         |           | 1319,0         | 55,6      |
| p14       | **                         |           | 1363,0         | 43,4      |
| p15       | <b>2,505.42*</b>           | 2147,9    | 2544,0         | 172,3     |
| p16       | **                         |           | 2572,2         | 139,9     |
| p17       | **                         |           | 2779,7         | 116,5     |
| p18       | **                         |           | 3736,5         | 297,2     |
| p19       | **                         |           | 3867,2         | 205,6     |
| p20       | **                         |           | 4064,8         | 254,7     |
| p21       | **                         |           | 5568,8         | 738,4     |
| p22       | **                         |           | 5762,2         | 528,7     |
| p23       | **                         |           | 6189,5         | 550,5     |

**Tabela 5.11:** Comparação da BD com os resultados obtidos por BALDACCI e MINGOZZI (2009).

Testou-se ainda a DB, variando cada um dos parâmetros fixados no processo de calibração, após obter tais resultados finais e não observou-se melhora dos resultados em nenhuma variação, garantindo que tal combinação de parâmetros é uma boa combinação. Encontrar o valor ótimo de parâmetros é praticamente inviável, exigindo um número demasiadamente grande de testes.

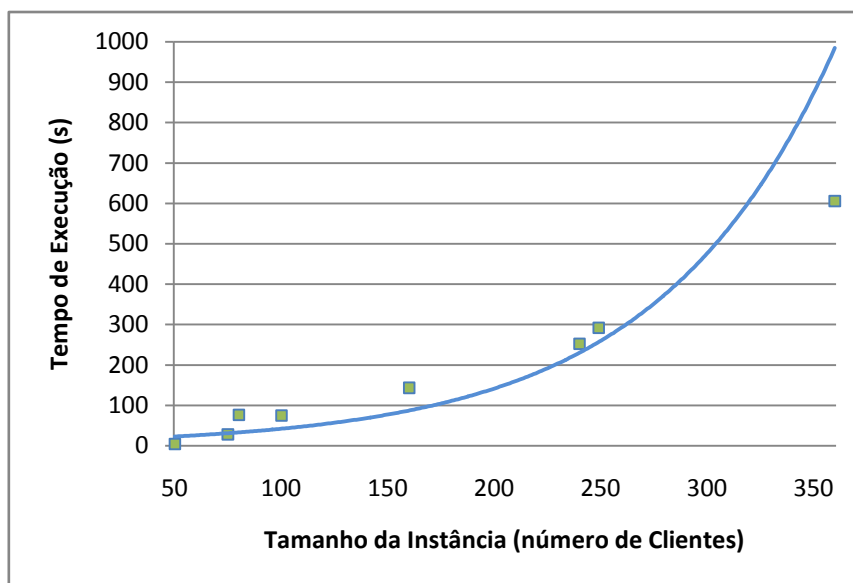
Para consolidar as comparações com as outras abordagens heurísticas utilizamos uma comparação utilizando MFlops (*Floating point Operations Per Second*), que fornece o número de operações de ponto flutuante por segundo, para cada processador utilizado. Ao multiplicar o tempo computacional pelo número de MFlops do computador utilizado, estamos normalizando o tempo computacional, obtendo um custo normalizado e comparável. Como tal número é muito grande dividimos por  $10^{-6}$ . Os resultados são apresentados na Tabela 5.12. PISSINGER e ROPKE (2007) apenas informam que utilizam um processador Athlon, sem citar seu modelo, logo atribuímos a ele o mesmo valor de MFlop do que o processador utilizado na BD.

|  | <b>Chao et al.</b> | <b>Renaud et al.</b> | <b>Pissinger e Ropke</b> | <b>Busca Dispersa</b> |
|--|--------------------|----------------------|--------------------------|-----------------------|
| Média de GAPs                              | 2,76%              | 0,86%                | 0,48%                    | 0,71%                 |
| Soma dos Tempos de todas as instâncias (s) | 22920,0            | 25026,0              | 5278,0                   | 4687,8                |
| Processador                                | SUN 4/370          | Sun Sparcstation 10  | Athlon                   | Athlon 64 2GHz        |
| Mflops do Processador                      | 2,7                | 18,42                | 6320                     | 6320                  |
| Tempo Normalizado                          | 0,06               | 0,46                 | 33,36                    | 29,63                 |

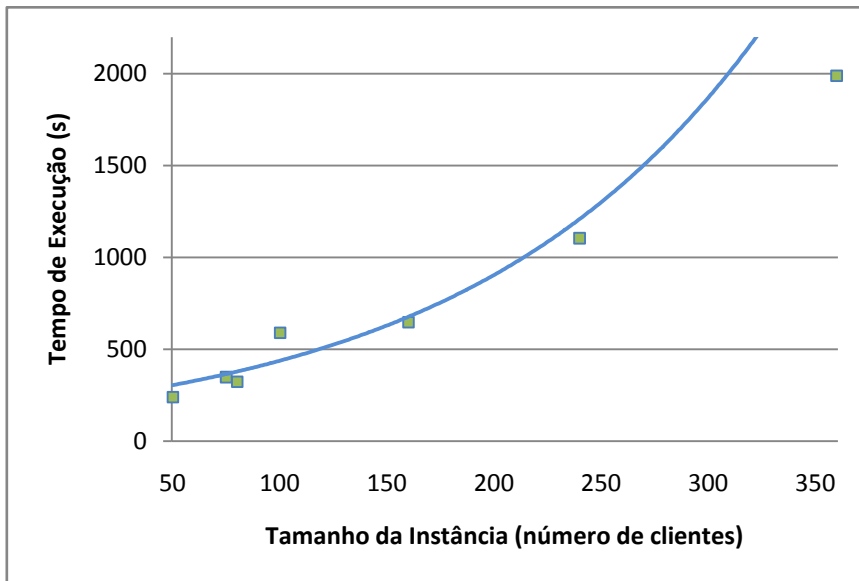
**Tabela 5.12:** Comparação baseado em MFlops.

Podemos observar que embora não obtenha a melhor média de GAPs da literatura, a Busca Dispersa pode ser considerada competitiva para resolver o PRVMD, analisando a média de GAP. Em relação aos tempos normalizados, notamos uma grande desproporção entre as heurísticas implementadas recentemente, com recursos computacionais recentes dos que tiveram suas implementações realizadas antes do ano 2000. Ainda comparando com implementações recentes, os tempos mostram-se competitivos.

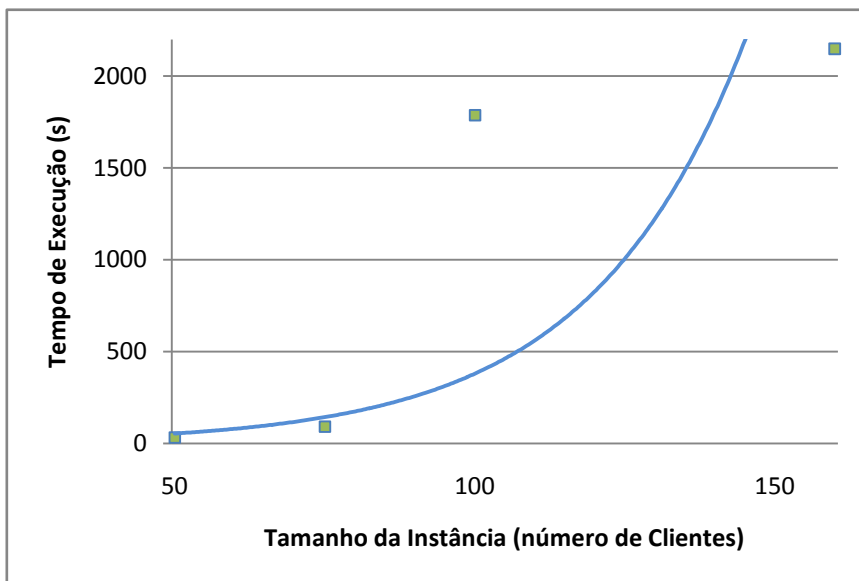
Outra análise importante é como se comporta o tempo computacional com o aumento do número de clientes. Na Figura 5.2 podemos observar um gráfico que mostra no eixo vertical o tempo de execução em segundos para cada tamanho de instância, no eixo horizontal. Notamos um aumento exponencial do tempo computacional com o aumento do tamanho da instância. Na Busca Tabu implementada por RENAUD et al (1996) observamos também um aumento exponencial, porém menos acentuado, como pode-se observar na Figura 5.3. Já para métodos exatos, como o implementado por BALDACCI e MINGOZZI (2009) observamos um aumento mais acentuado do que a BD e Busca Tabu, como mostrado na Figura 5.4.



**Figura 5.2:** Tempo de Execução versus Tamanho da Instância para a BD



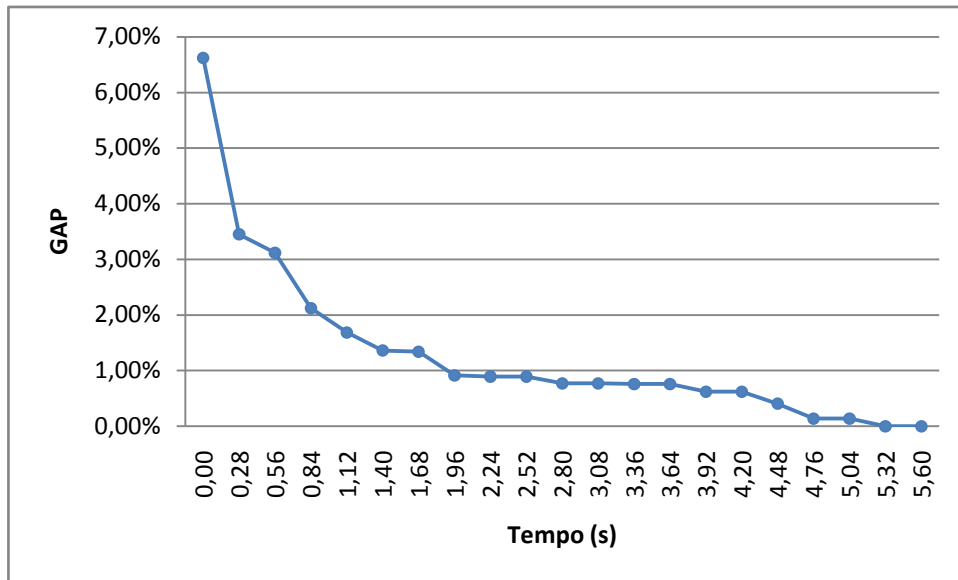
**Figura 5.3:** Tempo de Execução versus Tamanho da Instância para a Busca Tabu



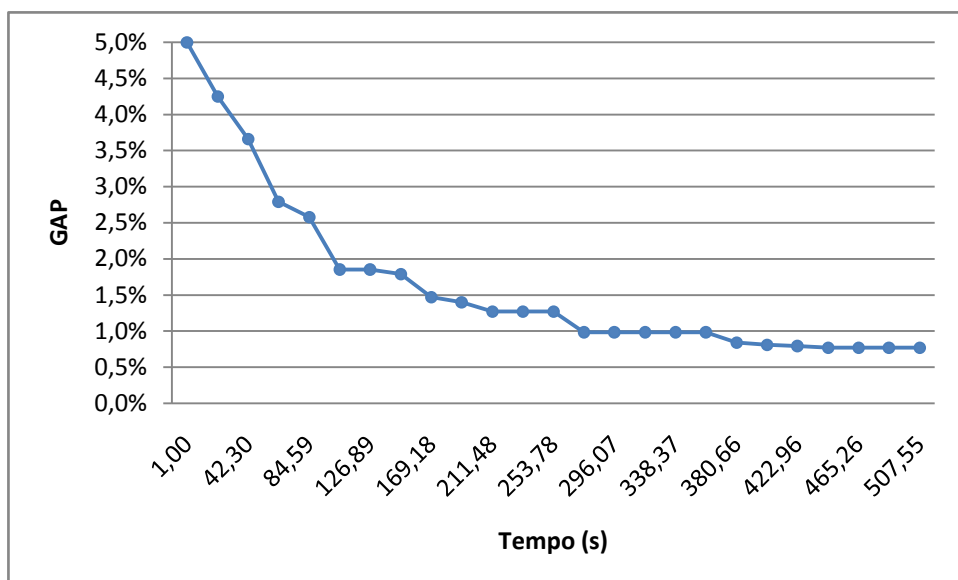
**Figura 5.4:** Tempo de Execução versus Tamanho da Instância para método exato

Uma análise importante é observar como ocorre a evolução do GAP, da melhor solução encontrada, durante uma execução da BD. Para tal avaliação, após cada reconstrução do conjunto de referência guardamos o par GAP e tempo computacional corrido, para plotar os gráficos das Figuras 5.5 e 5.6, que retratam, respectivamente, a evolução das instâncias p01 (considerada uma instância fácil) e p22 (considerada uma instância difícil). Tal análise foi proposta por Chambers (1983) visando observar a velocidade da convergência do processo de busca. Para ambas as instâncias em

análise observamos que nos primeiros momentos temos uma convergência rápida para em seguida ter uma melhora mais lenta da solução, até que o critério de parada interrompa as execuções.

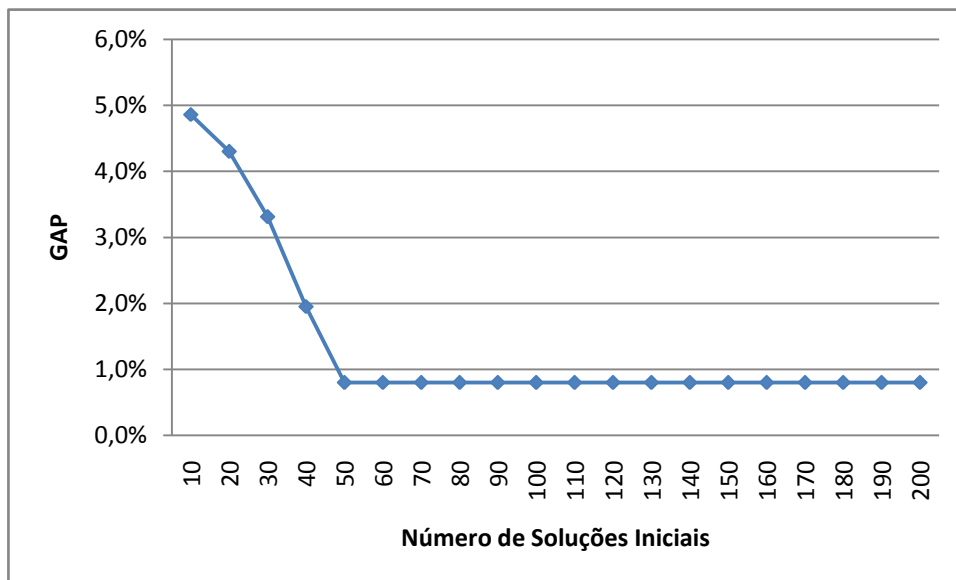


**Figura 5.5:** Evolução do GAP durante a execução da instância p01

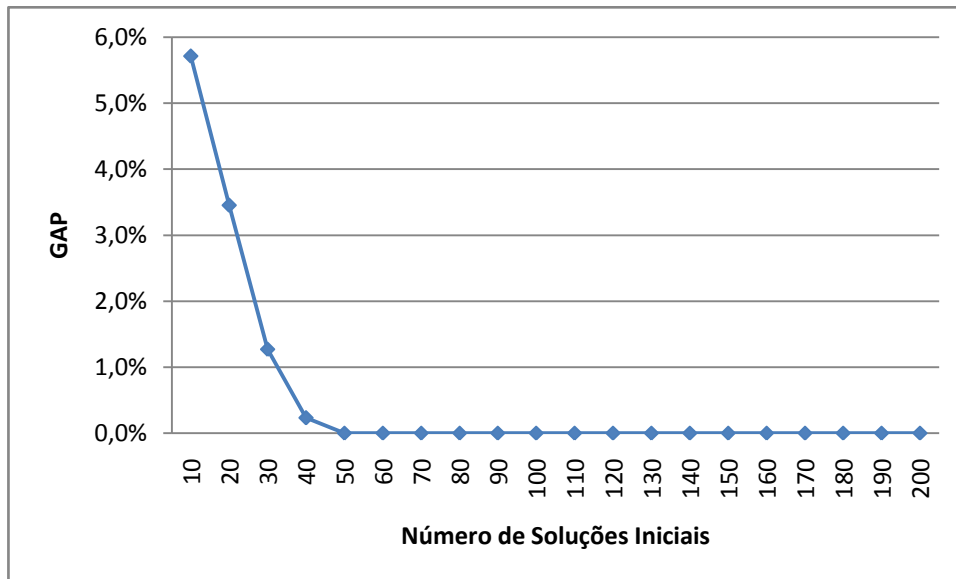


**Figura 5.6:** Evolução do GAP durante a execução da instância p22

O principal fator do critério de parada da BD implementada é o número de soluções iniciais, já que este irá definir o número de reconstruções do conjunto de referência. Quanto maior o número de soluções iniciais, maior a busca. Para análise do número de soluções iniciais no GAP, executamos para duas instâncias (p01 e p22) a BD, variando de 10 a 200 o número de soluções iniciais e observamos o GAP resultante. O resultado pode ser observado na Figura 5.7 e Figura 5.8. Notamos que em ambas as instâncias, para tamanhos maiores que 50 soluções iniciais, a DB estagna. Tal comportamento é observado em todas as instâncias e mostra uma limitação da BD implementada. O método gerador de soluções iniciais, para mais de 50 soluções, não é capaz de trazer maior diversidade que guie a BD para outras regiões promissoras.



**Figura 5.7:** Evolução do GAP para diferentes tamanhos de soluções iniciais para a instância p01



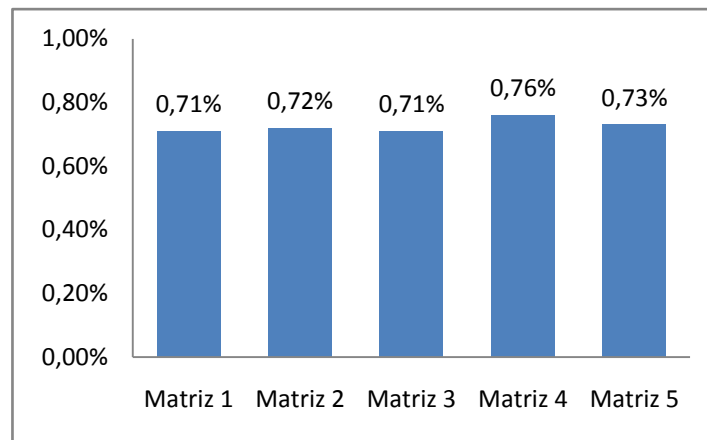
**Figura 5.8:** Evolução do GAP para diferentes tamanhos de soluções iniciais para a instância p22

### 6.3 Testes de Estabilidade da BD implementada

A grande quebra de paradigma que a Busca Dispersa introduziu, quando comparada aos algoritmos genéticos, foi o uso de procedimentos determinísticos e não aleatórios e a utilização de memória no processo. É natural, por isso, questionar o uso de aleatoriedade no método gerador de soluções iniciais. Para testar a robustez e estabilidade da BD implementada, realizamos testes suplementares. O grau de aleatoriedade a ser inserido no método gerador de soluções iniciais já foi testado. Para garantir a robustez e estabilidade precisamos variar a matriz de aleatoriedade utilizada (números aleatórios a serem utilizados) pelo código escrito e verificar como se comporta a média dos GAPS em relação aos melhores resultados observados na literatura .

Realizamos o seguinte teste: para os parâmetros já definidos, variamos apenas a matriz de aleatoriedade e verificamos como se comportou a Busca Dispersa, de forma a testar o impacto da aleatoriedade no método gerador de soluções iniciais. Tais resultados são apresentados na Figura 5.9. Observamos que não há grande variação na média dos GAPS das execuções em relação aos melhores resultados obtidos pela

literatura, garantindo a estabilidade da implementação, ainda que usando aleatoriedade.



**Figura 5.9:** Resultados variando a Matriz Aleatória.

Em ALEX et al (2002) foi proposta uma metodologia para calcular a distribuição de probabilidade, no tempo, para uma heurística encontrar uma solução alvo, com um determinado valor de FO. Tal distribuição, aplicada a heurísticas que utilizam aleatoriedade, mostra a probabilidade acumulada de o método encontrar uma solução considerada boa. Podemos aplicar tal metodologia ao método gerador de soluções iniciais. Para obter tal distribuição, executamos os seguintes passos (considere valor de *alfa* igual a 0,5):

**Passo 1:** Executar 50 iterações do método gerador de soluções iniciais, para uma dada instância, e guardar o valor de FO da melhor solução encontrada, que será a FO alvo.

**Passo 2:** Executar 500 iterações, sendo que, em cada iteração executar seguidas vezes o método gerador de soluções iniciais, até que seja encontrada uma solução com valor de FO igual ou melhor que a FO alvo encontrada no Passo 1. Guardar, para cada iteração o tempo até encontrar a solução.

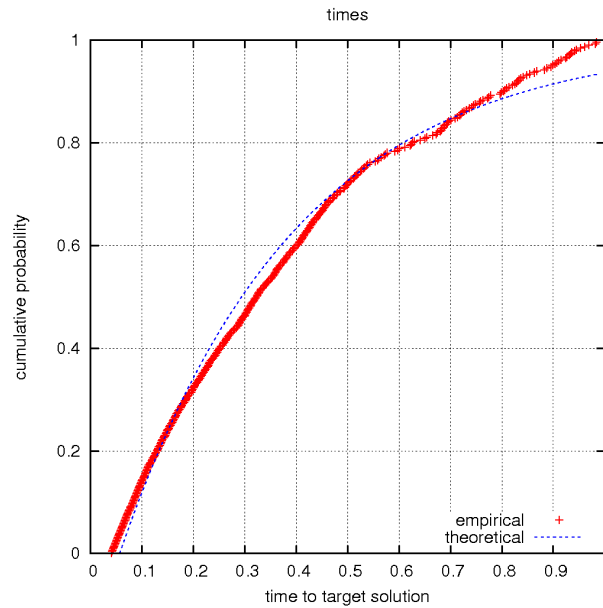
**Passo 3:** Ordenar de maneira crescente os 500 tempos encontrados no Passo 2.

**Passo 4:** Calcular a probabilidade associada ao n-ésimo tempo de execução utilizando a fórmula  $p_i = (i - \frac{1}{2})/n$ , sendo *n* o número da execução.

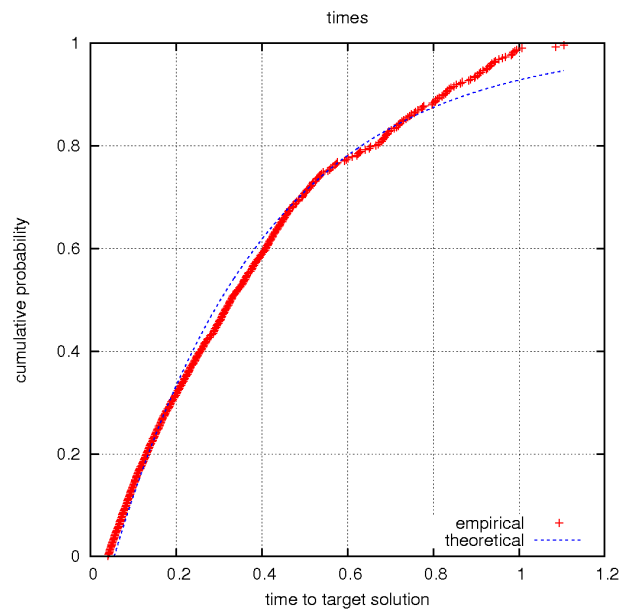
**Passo 5:** Plotar o gráfico das probabilidades calculadas e os respectivos tempos associados.



Aplicamos tal procedimento as instâncias p01 e p22. Os resultados podem ser observados nas Figuras 5.10 e 5.11. Para a geração dos gráficos foi utilizado o aplicativo tttplots que pode ser encontrado em <http://www2.research.att.com/~mgcr/tttplots/>. Podemos observar que a curva de distribuição (semelhante em ambas as instâncias) tem uma curvatura suave, demorando a convergir.



**Figura 5.10:** Distribuição de Probabilidade de se encontrar uma solução alvo para a instância p01



**Figura 5.11:** Distribuição de Probabilidade de se encontrar uma solução alvo para a instância p22

## CAPITULO 7

### CONCLUSÕES E PESQUISA FUTURA

Nesta dissertação descrevemos um algoritmo de Busca Dispersa desenvolvido para o PRVMD. A BD é um método evolutivo que combina soluções a partir de um conjunto de referência, com a finalidade de criar novas soluções melhores que as que as originaram. Foi apresentada uma revisão bibliográfica para o problema em estudo. A metodologia de BD também foi apresentada assim como as adequações deste procedimento para resolver o PRVMD. Foram realizados experimentos computacionais para conjuntos de dados disponíveis na literatura.

Uma descrição detalhada do processo de calibração dos parâmetros de Busca Dispersa foi apresentada, tais processos não são comuns na literatura, onde os autores se limitam a listar os parâmetros utilizados nos artigos, muito pela limitação de páginas dos artigos. Esta foi uma contribuição importante desta dissertação, já que dá um exemplo de processo estruturado da escolha de parâmetros e também apresenta todos os testes e resultados de tais escolhas para o PRVMD.

Observamos que no processo de calibração de parâmetros que uma implementação da Busca Dispersa, principalmente aplicada a problemas de roteamento, requer um ajuste cuidadoso dos mesmos, já que os resultados são bastante sensíveis a variações, o que significa que o sucesso de uma implementação de BD depende de cuidado nesta etapa.

Em relação a implementação de Busca Dispersa, testamos diferentes métodos de melhoria: i) Métodos Seqüencial e Paralelo; ii) Estratégia de Movimentos (Melhor de Todos e Primeiro Melhor) e iii) Utilização do movimento de Cruzamento. Tais abordagens podem ser adotadas para qualquer procedimento de Busca Dispersa, independente do problema a ser resolvido. Verificamos que o método Paralelo apresentou melhores resultados para esta implementação. Mesmo que onerando o tempo computacional o movimento de cruzamento se mostrou eficaz. Em relação a estratégia do método de melhoria verificou-se que o método baseado no primeiro movimento encontrada leva a uma convergência que apresenta melhores resultados do que a estratégia de aplicar o melhor de todos os movimentos calculados, para esta implementação da BD.

Já para o método de combinação de soluções foram testadas duas abordagens: i) Baseada em Clientes Comuns e ii) Baseada em Arcos Comuns. Verificamos que o método baseado em clientes comuns apresentou melhores resultados. Para o método gerador de soluções iniciais implementamos duas abordagens baseadas na heurística GRASP. Cada um dos métodos implementados foi detalhado. O baseado na heurística de GILLETT e MILLER (1974) apresentou melhores resultados. Diversos parâmetros de tamanho para o conjunto de referência também foram testados, apresentando melhores resultados quando o conjunto de referência qualidade têm tamanho 2 e o diversidade tamanho 8. Em relação ao tamanho do conjunto de soluções iniciais, o conjunto P, verificamos que o tamanho igual a 50 apresentou melhor relação entre o GAP e o tempo computacional.

Podemos verificar que uma implementação de Busca Dispersa pode ter diversas implementações para o mesmo problema, tendo diversas opções de implementação em cada método da BD e diversos parâmetros a serem ajustados. Verificamos a sensibilidade da busca dispersa para cada parâmetro testado. A busca dispersa pode apresentar a diversidade que aquele que a implementa desejar. Verificamos uma implementação com boa complexidade, ainda que a versão otimizada dos parâmetros ignore parte do todo implementado para testes.

Utilizamos instâncias disponíveis na literatura para comparar nossos resultados com os obtidos por outras implementações. Através da comparação concluímos que a Busca Dispersa não foi a melhor heurística para resolver o PRVMD em relação a FO, já que PISINGER e ROPKE (2007) e sua heurística baseada no *adaptive large neighborhood search*, apresentaram, em média, melhores resultados, embora em duas instâncias termos encontrado melhores resultados. O tempo computacional da BD é ligeiramente melhor do que o de PISINGER e ROPKE (2007). Uma análise de tempo normalizado considerando o *MFlop* de cada processador utilizado em cada implementação foi apresentada.

A Busca Dispersa se mostrou mais competitiva em relação a implementação de Busca Tabu de RENAUD et al. (1996), em relação a médias dos GAPs, assim como se mostrou, no mesmo aspecto, mais competitiva em relação a heurística proposta por CHAO et al. (1993). Em relação ao resultado computacional normalizado (devido a diferença de computadores) os tempos computacionais são maiores, já de que se trata

de uma meta-heurística que demanda maior processamento por ter diversos métodos atuando em conjunto.

Testamos a robustez e estabilidade da implementação proposta variando a matriz de aleatoriedade (utilizada somente no método gerador de soluções iniciais), para verificar o quanto o resultado depende da matriz de aleatoriedade utilizada. Verificamos que tal dependência é pequena, concluindo que a implementação proposta é robusta.

Entre as perspectivas futuras para a continuidade deste trabalho, destacam-se novas implementações de combinação de soluções, por se tratar de um método essencial para a BD. Como sugestão citamos métodos híbridos de combinação (ver LAGUNA e MARTI, 2003). Também se pode inserir o *Re-encadeamento de Trajetórias* no método de combinação de soluções, visando melhorar os resultados apresentados. Outra idéia seria testar a inserção de outros movimentos de melhoria.

A implementação de problemas de roteamento com múltiplos depósitos, incorporando outras restrições, tais quais janelas de tempo e coleta e entrega simultâneas podem ser feitas, com objetivo de testar a eficácia da BD aplicada e tais problemas, embora sejam raras tais aplicações tanto para métodos exatos ou heurísticos.

Outra perspectiva é a implementação de métodos exatos para a obtenção de limites inferiores que possam avaliar (balizar) a qualidade da heurística para qualquer instância testada. Isto é bastante importante, visto que em apenas algumas instâncias testadas haviam soluções ótimas disponíveis na literatura e, em nenhum outro trabalho, encontramos disponíveis limites inferiores para as instâncias testadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALEX, R. M., RESENDE, M. G. C., RIBEIRO, C. C. Probability distribution of solution time in grasp: An experimental investigation. *Journal of Heuristics*, 8, pp. 200–222, 2000.
- BAE, S. HWANG, H. S., CHO, G. E GOAN, M., “Integrated GA-VRP solver for multi-depot system”, *Computers & Industrial Engineering* 53, pp. 233-240, 2007.
- BALDACCI, R., HADJICONSTANTINO, E. A., e MINGOZZI, A., “An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation, *Operations Research* v. 52, pp. 723-738, 2004.
- BALDACCI, R., MINGOZZI, A., “A unified exact method for solving different classes of vehicle routing problems”, *Mathematical Programming, Ser. A*, 120:347–380, 2009
- BAKER, B.M., AYECHHEW, M.A. “A genetic algorithm for the vehicle routing problem”, *Computers & Operations Research* v. 30, pp. 787-800, 2003.
- BARBAROSOGLU, G., ÖGÜR, D. “A tabu search algorithm for the vehicle routing problem”, *Computers & Operations Research* v. 26, pp. 255-270, 1999.
- BAPTISTA, S., CARVALHO, R.O., ZÚQUETE, E. “A period vehicle routing case study”, *European Journal of Operational Research* v. 139, n.2, pp. 220-229, 2002.
- BELENGUER, J. M., MARTINEZ, M.C., MOTA, E. “A lower bound for the split delivery vehicle routing problem”, *Operations Research* v. 48, No. 5, pp. 801-810, 2000.
- BELTRAMI, E., BODIN, L. “Networks and vehicle routing for municipal waste collection”, *Networks* v. 4, pp. 65-94, 1974.
- BERGER, J., BARKAOUI, M., BRÄYSY, O. “A route-directed hybrid genetic approach for the vehicle routing problem with time windows”, *Information Systems and Operations Research* v. 41, pp. 179-194, 2003.
- BERTSIMAS, D.J., VAN RYZIN, G. “Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane”, *Operations Research* v. 39, pp. 601-615, 1991.
- BIANCHESSI N., RIGHINI G., “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery.”, *Computers & Operations Research* 34(2), p578–594, 2007.
- BRAMEL, J., SIMCHI-LEVI, D. “Probabilistic analysis and practical algorithms for the vehicle routing problem with time windows”, *Operations Research* v. 44, pp. 501-509, 1996.

- BRÄYSI, O., HASLE, G., DULLAERT, W. "A multi-star local search algorithm for the vehicle routing problem with time windows", *European Journal of Operational Research* v. 159, pp. 586-605, 2004.
- BULLNHEIMER, B., HARTL, R., STRAUSS, C. "An improved ant system algorithm for the vehicle routing problem", *Annals Operations Research* v. 89, pp. 319-328, 1999.
- CAMPOS V., LAGUNA M., MARTÍ R. "Scatter Search for the Linear Ordering Problem". In: *New Ideas in optimisation*, D. Corne, M. Dorigo e F. Glover (Eds.), McGraw-Hill, pp. 331-341, 1999.
- CAMPOS V., LAGUNA M., MARTÍ R. *Context-Independent Scatter and Tabu Search for Permutation Problems*. In: Technical report TR03-2001, Departamento de Estadística e I.O., Universidad de Valencia, 2001.
- CASCO, D. O., GOLDEN, B.L., WASIL, E. A. "Vehicle routing with backhauls". In: Golden BL, Assad A.(ed). *Vehicle Routing: Methods and Studies*, North Holland, Amsterdam, pp. 127-147, 1988.
- CLARKE, G., WRIGHT, J.W. "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research* v. 12, pp. 568-581, 1964.
- CHAMBERS, J. M. *Graphical Methods for Data Analysis (Statistics)*. Chapman & Hall/CRC, 1983.
- CHAO, I. M., GOLDEN, B. L., WASIL, E. "A new heuristic for the multi-depot vehicle routing problem that improves upon best-know solutions", *American Journal of Mathematical Management Sciences*. v. 13, n. 3-4, pp. 371-406, 1993.
- CHAO, I. M., GOLDEN, B. L., WASIL, E. "An improved heuristic for the period vehicle routing problem", *Networks* v. 26, pp. 25-44, 1995.
- CHRISTOFIDES, N., BEASLEY, J.E. "The period vehicle routing", *Networks* v. 14, pp. 237-256, 1984.
- CHRISTOFIDES, N., EILON, S., "An algorithm for the vehicle-dispatching problem", *Operational Research Quarterly* 20, pp. 309-318, 1969.
- CHRISTOFIDES, N., MINGOZZI, A., TOTH, P., "The vehicle routing problem". In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, Wiley, Chichester, UK, pp. 315-338, 1979.
- CORBERÁN, A., FERNÁNDEZ, E., LAGUNA, M., MARTÍ, R. "Heuristic Solutions to the Problem of Routing School Buses eith Multiple Objectives", *Journal of the Operational Research Society* v. 53, n. 4, pp. 427-435. 2002.
- CORDEAU, J., GENDREAU, M., LAPORTE, G., *et al.* "A guide to vehicle routing heuristics", *Journal of the Operational Research Society* v. 53, pp. 512-522, 2002.

- DEIF, I., BODIN, L.D., "Extension of the Clarke and Wright algorithm for solving the vehicle routing with backhauling". In: *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, Mass, 1984.
- G. DANTZIG, R. FULKERSON, S. JOHNSON, "Solution of a Large-Scale Traveling-Salesman Problem", *Operations Research*, v. 2, No. 4, pp. 393-410, 1954.
- DETHLOFF, J. "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up", *OR Spektrum* v. 23, pp. 79-96, 2001..
- DROR, M. "Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution", *European Journal of Operational Research* v. 64, Issue 3, pp. 432-441, 1993.
- DROR M., TRUDEAU, P. "Stochastic vehicle routing with modified savings algorithm", *European Journal of Operational Research* v. 23, n. 2, pp. 228-235, 1986.
- DROR, M., POWELL, W. "Special Issue on Stochastic and Dynamic Models in Transportation", *Operations Research* v. 41, 1993.
- DROR, M., TRUDEAU, P. "Savings by split delivery routing", *Transportations Science* v. 23, n. 2, pp. 141-145, 1989.
- DROR, M., TRUDEAU, P. "Split delivery routing", *Naval Research Logistics* v. 37. n. 3, pp. 383-402, 1990.
- DROR, M., LAPORTE, G., TRUDEAU, P. "Vehicle routing with split deliveries", *Discrete Applied Mathematics* v. 50, n. 3, pp. 239-254, 1994.
- DUHAMEL, C., POTVIN, J.Y., ROUSSEAU, J.M. "A tabu search heuristic for the vehicle routing problem with backhauls and time windows", *Transportation Science* v. 31, pp. 49-49, 1997..
- FEO, T., RESENDE, M.G.C. "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization* v. 2, pp. 1-27,1995.
- FISHER, M. L., "Vehicle routing problem". In : T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), *Networks Routing, Handbooks in Operations Research and Management Science*, v. 8, pp. 1-33, Elsevier Publisher BV, Amsterdam, 1995.
- FRIZZELL, P. W., GIFFIN, J.W. "The bounded split delivery vehicle routing problem with grid network distances", *Asia-Pacific Journal of Operational Research* v. 9, No. 1, pp. 101-116, 1992.



- FRIZZELL, P. W., GIFFIN, J.W. "The split delivery vehicle scheduling problem with time windows and grid network distances", *Computers & Operational Research* v. 22, n. 6, pp. 655-667, 1995.
- GALVÃO, R.D., GUIMARÃES, J. "The control of helicopter operations in the Brazilian oil industry: Issues in the design and implementation of a computerized system", *European Journal of Operational Research* v. 49, pp. 266-270, 1990.
- GAMBARDELLA, L., TAILLARD, E., AGAZZI, G., "Macs VRPTW : A multiple ant colony system for vehicle routing problems with time windows", In: Corne, D., Dorigo, M., Glover, F., (Eds.), *New Ideas in Optimization*, McGraw-Hill, London, pp.63-76, 1999.
- GÉLINAS, S., DESROCHERS, M., DESROSIERS, J., SOLOMON, M.M. "A new branching strategy for time constrained routing problems with application to backhauling", *Annals of Operations Research* v. 61, pp. 91-110, 1995.
- GENDREAU, M., HERTZ, A., LAPORTE, G. "A tabu search heuristic for the vehicle routing problem", *Management Science* v. 40, pp. 1276-1290, 1994.
- GENDREAU, M., LAPORTE, G., SÉGUIN, S. "Stochastic vehicle routing", *European Journal of Operational Research*, v. 88, n. 1, pp. 3-12, 1996.
- GILLETT, B., MILLER, L. R. "A heuristic algorithm for the vehicle dispatch problem", *Operations Research* v. 22, pp. 340-349, 1974.
- GILLETT, B.E. E JOHNSON, J.G., "Multi-terminal vehicle dispatch algorithm", *Omega* 4, 711–718. 1976
- GLOVER, F. "Heuristic for integer programming using surrogate constraints", *Decision Sciences* 8, pp. 156-166, 1977.
- GLOVER, F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research* v. 13, pp. 533-549, 1986.
- GLOVER, F., "A Template for Scatter Search and Path Relinking, Artificial Evolution". In: *Lecture Notes in Computer Science* 1363, J. K. Hao, E. Lutton, E. Ronald, M. Schoenauer e D. Snyers (Eds.), Springer-Verlag, pp. 13-54, 1998.
- GLOVER, F., LAGUNA, M. "Tabu search". Kluwer Academic Publishers, 1997.
- GLOVER, F., LAGUNA, M., MARTÍ, R. "Fundamentals of Scatter Search and Path Relinking", *Control and Cybernetics* v. 29 n. 3, pp. 653-684, 2000.
- GOETSCHALCKX, M., JACOBS-BLECHA, C. "The vehicle routing problem with backhauls", *European Journal of Operational Research*, v. 42, pp. 39-51, 1989.
- GOLDEN, B. L., MAGNANTI, T. L., NGUYEN, H. Q. "Implementing vehicle routing algorithms", *Networks* v. 7, pp. 113-148, 1977.

- HAGHANI, A. e JUNG, S. "A dynamic vehicle routing problem with time-dependent travel times", *Computers & Operations Research* v. 32, n. 11, pp. 2959-2986, 2005.
- HO, W., HO. G., JI, PI. E LAU, H., "A hybrid genetic algorithm for the multi-depot vehicle routing problem". *Engineering Applications of Artificial Intelligence* 21, 548-557, 2008.
- HOMBERGER, J., GEHRING, H., "Two evolutionary metaheuristics for the vehicle routing problem with time windows", *Information Systems and Operational Research-Special issue: Metaheuristics for Location and Routing Problems*, v. 37, pp. 297-318, 1999.
- KIRKPATRICK, S., GELATT, C.D.,VECCHI, P.M. "Optimization by simulated annealing", *Science* v. 220, pp.671-680, 1983.
- LAGUNA, M. MARTI, R. "Scatter Search Methodology and Implementations in C", Kluwer Academic Publishers, 2003.
- LAPORTE, G., NORBERT, Y., "Exact algorithms for the vehicle routing problem". In: S. Martello, G. Laporte, M. Minoux, C. Ribeiro (Eds.), *Surveys in Combinatorial Optimization, Annals of Discrete Mathematics*, v. 31, pp. 147-184, 1987.
- LAPORTE, G., NOBERT, Y., ARPIN, D., "Optimal solutions to capacitated multidepot vehicle routing problems", *Congressus Numerantium* v. 44, pp. 283-292, 1984.
- LAPORTE, G., NOBERT Y. E ARPIN D., "Optimal solutions to capacitated multidepot vehicle routing problems". *Congressus Numerantium*, 44, 283-292, 1984.
- LAPORTE, G., OSMAN, I.H., "Routing problems: A bibliography", *Annals of operations research* v. 61, n. 1, pp. 227-262, 1995.
- LARSEN, A., The Dynamic Vehicle Routing Problem, Ph.D. Thesis, Department of Mathematical Modelling , Technical University of Denmark, 2000.
- LAU, H.CH., SIM,M., MENG T.,K. "Vehicle routing problem with time windows and a limited number of vehicles", *European Journal of Operational Research*, 2003.
- LENSTRA, J., RINNOOY K. A. "Complexity of vehicle routing and scheduling problems", *Networks* v. 11, pp. 221-227, 1981.
- LIA-XI, RONG. "Algorithm Study of Multiple-depot Vehicle Routing Problem Based on Fuzzy Simulation", *Annals of the International Conference on Industrial Mechatronics and Automation*, 2009.
- MARTÍ, R., LAGUNA, M. e GLOVER, F. "Principles of scatter search", *European Journal of Operational Research* v.169, n. 2, pp. 359-372, 2006.

- MARTINHON, C., LUCENA, A., MACULAN, N. "Stronger K-tree relaxations for the vehicle routing problem", *European Journal of Operational Research* v. 158, n. 1, pp. 56-71, 2004.
- MIN, H., "The multiple vehicle routing problem with simultaneous delivery and pickup points", *Transportation Research*, v. 23A, pp. 377-386, 1989.
- MESTER, D., BRÄYSY, O. "Active guided evolution strategies for the large scale vehicle routing problem with time windows", *Computers & Operations*, v. 32, n. 6, pp. 1593-1614, 2005.
- MINGOZZI, A., GIORGI, S. "An exact method for the vehicle routing problem with backhauls", *Transportation Science* v. 33, n. 3, pp. 315-329, 1999.
- MONTANÉ, F.A.T. *Problemas de roteamento de veículos com pedidos de coleta e entrega*, Tese de doutorado, Engenharia de Produção, Univ. Federal de Rio de Janeiro, 2004.
- MONTANÉ, F.A.T., GALVÃO, R.D. "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers & Operations Research* v. 33, n. 3, pp. 595-619, 2006.
- MOSHEIOV, G. "The traveling salesman problem with pick-up and delivery", *European Journal of Operational Research*, v. 79, pp. 299-310, 1994.
- MULLASERIL, P.A., DROR, M., LEUNG, J. "Split-delivery routing heuristics in livestock feed distribution", *Journal of the Operational Research Society*, v. 48. n. 2, pp. 107-116, 1997.
- NAGY, G., SALHI, S., "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, v. 162, n. 1, pp. 126-141, 2005.
- OSMAN, I.H. "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem", *Annals of Operations Research*, v. 41, n. 4, pp. 421-451, 1993.
- PISINGER, D., ROPKE, S., "A general heuristic for vehicle routing problems." *Computers & Operations Research* 34, 2403-2435, 2007.
- POTVIN, J., ROUSSEAU, J. "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows", *European Journal of Operational Research*, v. 66, pp. 331-340, 1993.
- POWELL, W.B., JAILLET, P., ODoni, A. "Stochastic and Dynamic Networks and Routing". In: *Network Routing*, v. 8 of *Handbooks in Operations Research and Management Science*, chapter 3, pp. 141--295. North--Holland, Amsterdam, the Netherlands, 1995.

- PRINS, C. "A simple and effective evolutionary algorithm for the vehicle routing problem", *Computers & Operations Research*, v. 31, n. 12, pp. 1985-2002, 2004.
- PSARAFTIS, H. N. "Dynamic Vehicle routing: status and prospects", *Annals of Operations Research* v. 61, 143-164, 1995.
- PISINGER, D., ROPKE, S., "A general heuristic for vehicle routing problems." *Computers & Operations Research* 34, 2403-2435, 2007.
- REGO, C., SPRING, "Scatter Search for vehicle routing problem". National INFORMS Meeting, Salt Lake City, UT, 2000.
- REGO, C., "A subpath ejection method for the vehicle routing problem", *Management Science* v.44, pp. 1447-1459, 1998.
- REGO, C., ROUCAIROL, C., "A parallel tabu search algorithm using ejection chains for the vehicle routing problem", In: *Meta-Heuristics: Theory and Applications*, Kluwer, Boston, pp. 661-675, 1996.
- RENAUD, J., LAPORTE, G., BOCTOR, F.F. "A tabu search heuristic for the multi-depot vehicle routing problem", *Computers & Operations Research* v. 23, n. 3, pp. 229-235, 1996.
- ROCHAT, Y., TAILLARD, E.D. "Probabilistic Diversification and intensification in local search for vehicle routing", *Journal of Heuristics*, v. 1, pp. 147-167, 1995.
- RUSSELL, R., IGO, W. "An assignment routing problem", *Networks* v. 9, pp. 1-17, 1979.
- RUSSELL, R., GRIBBIN, D. "A multiphase approach to the period routing problem", *Networks* v. 21, pp.747-765, 1991.
- RUSSELL, R. "Hybrid heuristics for the vehicle routing problem with time windows", *Transportation Science*, v. 29, pp. 156-166, 1995.
- RUSSELL, R. A., CHIANG, W. Ch. "Scatter search for the vehicle routing problem with time windows", *European Journal of Operational Research* v. 169, n. 2, pp. 606-622, 2006.
- SALHI, S., NAGY, G. "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling", *Journal of the Operational Research Society* v. 50, pp. 1034-1042, 1999.
- SALHI, S. E SARI, M., Models for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research* 103, 95–112, 1997.
- SECOMANDI, N. "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands", *Computers & Operations Research* v. 27, n. 5, pp. 1171-1200 , 2000.

- SÉGUIN, R., POTVIN, J.Y., GENDREAU, M., et al., “Real-Time Decision Problems: An Operational Perspective”, *Journal of the Operational Research Society* v. 48, 162-174, 1997.
- SOLOMON, M. “Algorithms for the vehicle routing and scheduling problem with time windows constraints”, *Operations Research* v. 35, n. 2, pp. 254-265, 1987.
- SOSA, N. G. M., GALVÃO R. D., GANDELMAN, D.A., “Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos.” *Pesquisa Operacional* 27, pp. 293-310, 2007
- STEWART, W. R., GOLDEN, J.B. “Stochastic vehicle routing: A comprehensive approach”, *European Journal of Operational Research*, v. 14 n. 4, pp. 371-385 , 1983.
- ROBERT T. SUMICHRAST, INA S. MARKHAM, “A heuristic and lower bound for a multi-depot routing problems”, *Computers and Operations Research*, 10, 1047-1056, 1995.
- TAILLARD, E. D. “Parallel iterative search methods for vehicle routing problem”, *Networks* v. 23, pp. 661-673, 1993.
- TAILLARD, E.D., BADEAU, P., GENDREAU, P., GUERTIN, F., POTVIN, J.Y. “A tabu search heuristic for the vehicle routing problem with soft time windows”, *Transportation Science* v. 31, pp. 170 - 186, 1997.
- TAN, C.C.R., BEASLEY J. E. “A heuristic algorithm for the period vehicle routing problem”, *Omega* v. 12, n. 5, pp. 497-504, 1984.
- TARANTILIS, C. D., IOANNOU, G. e PRASTACOS, G. “Advanced vehicle routing algorithms for complex operations management problems”, *Journal of food engineering* v. 70, pp. 455-471, 2005.
- TARANTILIS, C. D. “Solving the vehicle routing problem with adaptive memory programming methodology”, *Computers & Operations Research* v. 32, pp. 2309 – 2327, 2005.
- THOMPSON, P., PSARAFTIS, H. “Cyclic transfer algorithms for multivehicle routing and scheduling problem”, *Operations Research* v. 41, pp. 935-946, 1993.
- TILLMAN, F. A. “The multiple terminal delivery problem with probabilistic demands”, *Transportation Science* v. 3, pp. 192-204, 1969.
- TILLMAN, F. A., HERING, R. W. “A study of a look-ahead procedure for solving the multiterminal delivery problem”, *Transportation Research* v. 5, pp. 225-229, 1971.
- TIMON C. D., ELDON Y. L. , DEFROSE CH. “Dynamic vehicle routing for online B2C delivery”, *Omega* v. 33, n.1, pp. 33-45, 2005.

- TOTH, P., VIGO, D. "An exact algorithm for the vehicle routing problem with backhauls", *Transportation Science* v. 31, pp. 372-385, 1997.
- TOTH, P., VIGO, D. "A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls", *European Journal of Operational Research*, v. 113, pp. 528-543, 1999.
- TOTH, P., VIGO, D. "The Vehicle Routing Problem". SIAM Monographs on Discrete . Philadelphia. Mathematics and Applications, 2002.
- TOTH, P., VIGO, D. "The granular tabu search (and its application to the vehicle routing problem)", *INFORMS, Journal on Computing* v. 14, n. 4, pp. 333-348, 2003.
- WADE, A.C., SALHI, S. "An investigation into a new class of vehicle routing problem with backhauls", *Omega* v. 30, pp. 479-487, 2002.
- WREN, A. E HOLLIDAY A., "Computer scheduling of vehicles from one or more depots to a number of delivery points." *Operations Research*. 23, p333-344, 1972
- XU, J., KELLY, J.P. "A network flow-based tabu search heuristic for the vehicle routing problem", *Transportation Science* v. 30, pp. 379-393, 1996.
- ZACHARIADIS, E., TARANTILIS, C. E KIRANOUDIS, C., "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", *Expert Systems with Applications*, In Press, Uncorrected Proof, 2007.
- ZHONG, Y. e COLE, M. H. "A vehicle routing problem with backhauls and time windows: a guided local search solution", *Transportation Research Part E: Logistics and Transportation Review*, v. 41 n.2, pp. 131-144 , 2005.

## ANEXO 1

### PROBLEMAS DE ROTEAMENTO

O Problema de Roteamento de Veículos é enfrentado diariamente por milhares de empresas e organizações que precisam coletar ou entregar bens ou pessoas. As aplicações são as mais variadas e cada uma, por ter sua peculiaridade, traz sua própria restrição. A maioria das pesquisas focam em alguns casos genéricos do problema de roteamento de veículos, os quais vamos revisar neste Capítulo. Métodos exatos e heurísticos foram propostos para resolver tais problemas, sendo os metaheurísticos os que tem os melhores resultados em tempos computacionais aceitáveis.

A importância de sistemas de distribuição torna-se evidente quando se considera o impacto dos respectivos custos na operação das empresas. Existe uma variedade de problemas de decisão nessa área, em diferentes níveis: estratégico, tático e operacional. Decisões sobre a localização de facilidades (fábricas, depósitos, entre outras) são consideradas estratégicas; decisões que determinam por exemplo o tamanho e a composição da frota de veículos de uma empresa podem ser consideradas de nível estratégico/tático; a nível operacional temos por exemplo decisões sobre o roteamento e/ou seqüenciamento de frotas de veículos.

O Problema de Roteamento de Veículos (PRV) é um nome genérico dado a uma classe de problemas de distribuição. Um caso particular do problema geral constitui a chamada versão clássica do PRV e pode ser definida da seguinte maneira. Seja  $G = (U, A)$  um grafo (direcionado ou não) onde  $U = \{u_0, u_1, \dots, u_n\}$  é o conjunto de vértices (clientes) e  $A = \{(u_i, u_j): u_i, u_j \in U, i \neq j\}$  é o conjunto de arcos. O vértice  $u_0$  representa o depósito de uma frota homogênea de veículos com capacidade  $Q$  por veículo. Cada cliente tem uma demanda  $q_i$  e requer um tempo de serviço  $s_i$ . Uma matriz de custos de viagem  $C = [c_{ij}]$  é definida em  $A$ . O número de veículos é pré-determinado ( $m$  veículos) ou é considerado uma variável de decisão. O PRV clássico (com restrições de capacidade e distância máxima) consiste em designar um conjunto de  $m$  rotas de entrega e/ou coleta tal que: i) o custo total do conjunto de rotas percorrido pela frota seja minimizado; ii) cada rota se inicie e finalize em um depósito; iii) cada cliente tenha sua demanda suprida exatamente por um veículo; iv) a carga total de cada veículo não exceda sua capacidade  $Q$ ; e v) o tempo total necessário para completar qualquer rota

não exceda um limite pré-especificado  $T$ , que inclui tempos de viagem entre clientes e tempos de serviço em cada cliente.

O PRV é um problema NP-difícil, ver LENSTRA e RINNOOY KAN ( 1981). Devido à sua complexidade computacional, métodos exatos de solução são inviáveis para instâncias de grande porte, o que nos remete à utilização de métodos heurísticos/metaheurísticos. Existe um grande número de artigos que tratam do problema clássico descrito no parágrafo acima, usando diversas heurísticas e metaheurísticas, com predominância das metaheurísticas, mais recentes e apresentando resultados de melhor qualidade. Não é nosso objetivo fazer uma revisão detalhada da bibliografia existente sobre o assunto. Alguns artigos recentes relacionados ao tema são, por exemplo, CORDEAU et al. (2002), BAKER e AYECHEW (2003), TOTH e VIGO (2003), WASSAN et al. (2007). TOTH e VIGO (2002) escreveram um importante livro sobre o PRV, englobando tanto o problema clássico como variantes do mesmo. O PRV clássico adicionando-se restrições de capacidade aos veículos e demandas aos clientes também é conhecido como Problema de Roteamento de Veículos Capacitado, que também é um problema muito estudado. Ver CHRISTOFIDES et al. (1979), CHRISTOFIDES (1985), LAPORTE e NORBERT (1987), FISHER (1995), LAPORTE e OSMAN (1995), TOTH e VIGO (2002), MARTINHON et al. (2004), TARANTILIS (2005), TARANTILIS et al. (2005).

Os problemas de roteamento de veículos procuram em geral minimizar a distância percorrida pelos veículos, porém podem também: i) minimizar o custo total de transporte (função da distância percorrida, tempo e custos associados aos veículos. ii) minimizar o número de veículos utilizados. iii) minimizar penalidades associadas ao atendimento parcial de clientes.

As principais restrições que podem ser impostas a um problema de roteamento são em relação aos clientes ou aos veículos. A maioria das variações dos problemas de roteamento irá considerar ou não tais restrições. Em relação aos clientes temos as restrições relativas a: i) Distribuição Geográfica dos Clientes; ii) Demanda de coleta/entrega de bens ou pessoas para cada cliente; iii) Tempo de serviço para atendimento do cliente (quanto tempo o serviço ficará para deixar/coletar os bens); iv) Janela de Tempo em que o cliente pode ser atendido. v) Penalidades podem ser associadas ao não atendimento da Janela de Tempo ou por deixar de coletar ou recolher uma demanda.



Já em relação aos veículos podemos considerar: i) Restrição de capacidade de peso ou volume do veículo; ii) Podem estar disponíveis em um ou mais depósitos. iii) Restrição na armazenagem de pacotes tridimensionais; iv) Divisão em compartimentos para cada tipo de carga; v) Restrição de utilização de certas vias (por restrição de altura ou peso do veículo); vi) Custos associados à utilização de cada veículo em caso de frota heterogênea. vii) Restrições de precedência na ordem em que os clientes são atendidos.

## Variantes do PRV

Abordaremos aqui as principais variações do problema de roteamento de veículos encontradas na literatura.

### Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega

Neste problema os clientes podem ter demandas de coleta e entrega de bens. As demandas de entrega e coleta para cada cliente deve ser feita pelo mesmo veículo. O objetivo continua sendo minimizar a distância percorrida pelos veículos. Os trabalhos que se destacam para essa variante do PRV são: MIN (1989), GALVÃO e GUIMARÃES (1990), MOSHEIOV (1998), SALHI e NAGY (1999), DETHLOFF (2001), MONTANÉ (2004), NAGY e SALHI (2005), MONTANÉ e GALVÃO (2006).

### Problema de Roteamento de Veículos com Janelas de Tempo

Trata-se do problema clássico com a restrição adicional de uma janela (período) de tempo em que cada cliente pode ser atendido. Caso não seja atendido no período designado podem existir penalidades, no caso de problemas de janela de tempo flexíveis. Destacam-se os artigos de LENSTRA e RINNOOY (1981), SOLOMON (1987), POTVIN e ROUSSEAU (1993), THOMPSON e PSARAFTIS (1993), RUSSELL (1995), BRAMEL e SIMCHI-LEVI (1996), TAILLARD *et al.* (1997), GAMBARDELLA *et al.* (1999), HOMBERGER e GEHRING (1999), BERGER *et al.* (2003), LAU *et al.* (2003), BRÄYSY *et al.* (2004) e RUSSELL e CHIANG (2006).

### Problema de Roteamento de Veículos com Restrições de Capacidade e Tempo Máximo de Rotas

Neste problema temos restrições de capacidade do veículo e demandas de clientes e também restrição de distância máxima a ser percorrida, ver OSMAN (1993), ROCHAT e TAILLARD (1995), XU e KELLY (1996), TOTH e VIGO (2002), BALDACCI *et al.* (2004), PRINS (2004), TARANTILIS (2005) e TARANTILIS *et al.* (2005).

### Problema de Roteamento de Veículos com Entrega Particionada

Trata-se de uma variação do PRV clássico onde um cliente pode ser atendido por mais de um veículo, relaxando a restrição de que um cliente é sempre atendido por somente um veículo. DROR e TRUDEAU (1989,1990), FRIZZELL e GIFFIN (1992, 1995), DROR *et al.* (1994), MULLASERIL *et al.* (1997), BELENGUER *et al.* (2000).

### Problema de Roteamento de Veículos com Cargas de Retorno

Neste problema existem clientes apenas com carga para coleta e outros apenas com cargas para entrega. O veículo deve primeiro atender os clientes de entrega para depois coletar a carga nos demais clientes. Ver: DEIF e BODIN (1984), CASCO *et al.* (1988), GOETSCHALCKX e JACOBS-BLECHA (1989), GÉLINAS *et al.* (1995), DUHAMEL *et al.* (1997), TOTH e VIGO (1997,1999), MINGOZZI e GIORGI (1999), WADE e SALHI (2002) e ZHONG e COLE (2005).

### Problema de Roteamento de Veículos Periódico

Nesta variante o planejamento abrange mais de um dia, sendo que o cliente não pode retornar ao depósito no mesmo dia de sua saída. Ver: BELTRAMI e BODIN (1974), RUSSELL e IGO (1979), CHRISTOFIDES e BEASLEY (1984), TAN e BEASLEY (1984), RUSSELL e GRIBBIN (1991), CHAO *et al.* (1995) e BAPTISTA *et al.* (2002).

### Problema Estocástico de Roteamento de Veículos

Possui demandas estocásticas. As demandas são determinadas por uma distribuição de probabilidade e tempos de viagens são variáveis aleatórias. Ver:

STEWART e GOLDEN (1983), DROR e TRUDEAU (1986), DROR (1993), GENDREAU *et al.* (1996), SECOMANDI (2000), entre outros.

#### Problema Dinâmico de Roteamento de Veículos

Trata-se de uma abordagem dinâmica para atender alterações no roteamento em tempo real, devido a mudanças nas variáveis, seja causada por trânsito ou adição ou remoção de demandas em clientes. Ver: BERTSIMAS e VAN RYZIN (1991), DROR e POWELL (1993), POWELL *et al.* (1995), PSARAFTIS (1995), SÉGUIN *et a.* (1997), LARSEN (2000 ), TIMON *et al.* (2005), HAGHANI e JUNG (2005).

## ANEXO 2

### PESQUISA BIBLIOGRÁFICA PARA O PRV

O primeiro artigo da literatura do PRV data de 1954. DANTZIG et al (1954) estudou um problema de larga escala do problema do caixeiro viajante. A partir deste artigo diversos outros surgiram estudando o mesmo problema. O problema do caixeiro viajante se mostrou um caso do problema genérico do PRV. CLARKE e WRIGHT (1964) foram os primeiros a utilizar mais de um veículo na formulação do problema, sendo assim o primeiro artigo que aborda o VRP da maneira que conhecemos, embora o nome Roteamento de Veículos (“*Vehicle Routing*”) tenha surgido com GOLDEN et al (1977). SOLOMON (1987) introduziu as restrições de janela de tempo e criou as instâncias de solomon, hoje benchmark referência para a literatura.

O Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) é uma generalização do Problema de Roteamento de Veículos Clássico (PRV), onde rotas podem ser formadas a partir de múltiplos depósitos. Por se tratar de uma generalização do PRV, o PRVMD é NP-difícil, o que torna difícil sua solução ótima para instâncias que não sejam pequenas. Por este fato os algoritmos exatos resolvem em tempo viável apenas problemas com pequenas instâncias, ver BALDACCI *et al.* (2004). Por isso abordagens heurísticas são comuns para resolver problemas de roteamento, como veremos.

Este capítulo está dividido em três partes. A primeira trata das abordagens para o PRV Clássico, seu estudo é importante pois a idéia de diversos movimentos e heurísticas foram adaptados destas abordagens para o PRVMD. A segunda trata das abordagens para o PRVCES, que é o problema de roteamento considerando restrições de distância máxima de rota e capacidade dos veículos, ambas restrições que serão adotadas. A terceira trata de heurísticas e metaheurísticas para o PRVMD. Já a quarta trata da Busca Dispersa aplicada a Problemas de Roteamento de Veículos.

## Metaheurísticas para Problemas de Roteamento de Veículos

Metaheurísticas são baseadas em processos heurísticos, geralmente de estrutura padrão, que podem ser aplicadas a diversos tipos de problemas de otimização combinatória. Elas são capazes de, diferentemente de heurísticas simples, sair de ótimos locais para explorar outras regiões e encontrar outras zonas promissoras. Diversas metaheurísticas são utilizadas para resolver problemas de roteamento de veículos como Busca Tabu, Algoritmos Genéticos, Busca Dispersa, Simulated Annealing, Redes Neurais, Colônia de Formigas, Redes Neurais e memória adaptativa.

### Busca Tabu

A *Busca Tabu* teve suas primeiras implementações na década de 60. Trata-se de uma metaheurística que utiliza memória para guiar os próximos passos a serem tomados. Quando se atinge um ótimo local se utiliza da memória para realizar movimentos ainda não realizados, ou movimentos Tabu, já que eles possivelmente irão piorar a solução, tentando transcender o ótimo local em busca de outras regiões promissoras. A forma mais utilizada atualmente é a proposta por GLOVER (1986). A Busca Tabu é uma das metaheurísticas mais bem sucedidas na resolução dos mais diversos problemas de otimização combinatória. Uma análise detalhada da Busca Tabu também pode ser encontrada em GLOVER e LAGUNA (1997). Para problemas de roteamento de veículos temos as aplicações de OSMAN (1993), TAILLARD (1993), GENDREAU et al. (1994), XU e KELLY (1996), REGO e ROUCAIROI (1996), REGO (1998), BARBAROSOGLU e ÖGÜR (1999).

### Simulated Annealing

É uma metaheurística inspirada no comportamento físico do resfriamento de metais. É baseado em probabilidades que pode aceitar movimentos que melhoram ou pioram a Função Objetivo (FO). O procedimento inicia-se com uma temperatura inicial alta (função de probabilidade) e vai diminuindo a temperatura, deixando mais difícil aceitar movimentos que piorem a FO. O procedimento foi criado por KIRKPATRICK, GELATT e VECCHI (1983). OSMAN (1993) é uma referência no uso desta técnica para problemas de roteamento.

### Algoritmos Genéticos

Inspira-se na teoria evolutiva para criar uma metaheurística para problemas complexos de combinatória. Avaliação de cromossomo (análogo a representação estrutural da solução), aptidão (análogo a ter uma boa FO respeitando as restrições), mutação (movimentos) e reprodução (combinação de soluções) são elementos presentes neste método. Técnicas de busca local foram introduzidas gerando os Algoritmos Meméticos, que balanceiam cooperação e competição.

Na literatura, para resolver o PRV destaca-se BAKER e AYECHWE (2003), PRINS (2004), BERGER e BARKAOUI (2004) e MESTER e BRÄYSY (2004).

### Colônia de Formigas

Baseia-se em outro processo natural. Neste caso o comportamento das formigas em uma colônia e os procedimentos para se encontrar alimento. Verifica o feromônio deixado como rastro pelas formigas em busca de alimento. Estudos indicam que a tendência das formigas em seguir rastros de outras formigas, provoca, ao longo do tempo, a otimização dos trajetos percorridos pelas mesmas. BULLNHEIMER *et al* (1997) é referência no uso desta técnica para resolver problemas de roteamento

### GRASP

A metodologia GRASP é utilizada para a geração de soluções iniciais no processo de busca da metaheurística BD. A metaheurística GRASP - "Greedy Randomized Adaptive Search Procedure", foi desenvolvida ao final da década de 80, ver Feo e Resende (1995). GRASP é um procedimento de múltiplos inícios, no qual cada passo consiste em uma fase de construção e outra de melhoria. Na fase de construção aplica-se um procedimento heurístico construtivo para obter uma boa solução inicial. Esta solução é melhorada na segunda fase mediante um algoritmo de busca local. Esta metodologia foi escolhida por permitir a geração de soluções diversas, ou seja, dispersas no espaço de soluções. Tal característica é fundamental para o sucesso de uma implementação de um algoritmo de BD.

## Abordagens para o PRVCES

O PRVCES é uma variante do Problema de Roteamento de Veículos clássico (com restrições de capacidade e tempo máximo de rotas), consiste na definição de rotas de entrega e coleta de custo mínimo para uma frota de veículos, originando e terminando em um depósito central tal que todos os clientes sejam atendidos. Seu estudo será importante pois estas mesmas restrições estão presentes no PRVMD. Consideram-se situações em que os bens fornecidos do depósito aos clientes são diferentes dos bens fornecidos pelos clientes ao depósito. Entregas se iniciam a partir de um depósito e a coleta é conduzida até o mesmo depósito no final do trajeto. A frota homogênea deve servir um conjunto de clientes, cada qual caracterizado por demanda de coleta, demanda de entrega, e, em alguns dos casos, tempo de serviço. Cada cliente tem sua demanda (coleta e entrega) suprida exatamente por um veículo, isto é que a demanda dos clientes devem ser atendidos numa única visita por um veículo. A carga total de qualquer veículo não deve exceder a capacidade do mesmo, e o tempo total necessário para completar qualquer rota não pode exceder um limite pré-especificado, que inclui os tempos de viagem entre clientes e os tempos de serviço em cada cliente (A restrição de tempo máximo pode ser visto como uma extensão do conceito de distancia máxima). O objetivo é minimizar o tempo total de atendimento (distancia percorrida pelos veículos) assim como minimizar a frota de veículos; obedecendo a restrição de que o veículo deve ter suficiente capacidade para transportar os pedidos de entrega e coleta ao longo da rota.

MIN(1989) trabalhou no contexto de distribuição para bibliotecas públicas, sendo o primeiro a formular o PRVCES. A heurística proposta, baseada em agrupar e rotear é aplicada para um problema real com 22 bibliotecas e dois veículos homogêneos. GALVÃO e GUIMARÃES (1990) estendem o PRVCES adicionando restrição de número máximo de clientes por rota e o limite de distância máxima entre dois clientes adjacentes. Eles consideram uma frota heterogênea e aplicam para o problema de transportar funcionários na indústria de petróleo. MOSHEIOV (1994) propõe duas heurísticas simples para tratar o problema e também propõe uma formulação matemática. DETHLOFF (2001) propõe quatro heurísticas de inserção baseados em diferentes critérios de inserção. NAGY e SALHI (2005) desenvolveram uma busca local que permite certo grau de inviabilidade, introduzindo o conceito de inviabilidade fraca e forte. MONTANÉ e GALVÃO (2006) descrevem a aplicação de uma Busca Tabu para o PRVCES. O procedimento usa uma vizinhança com três movimentos. Memória de longo prazo baseada em frequência é utilizada para

balancear entre diversificação e intensificação. BIANCHESSI e RIGHINI (2007) também implementaram uma Busca Tabu com diversas vizinhança, de simples a mais complexas enquanto ZACHARIADIS et al (2009) propõe uma Busca Tabu híbrida com um procedimento de busca local.