

**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

ANAND SUBRAMANIAN

**METAHEURÍSTICA *ITERATED LOCAL SEARCH*
APLICADA AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEA**

**JOÃO PESSOA - PB
2008**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ANAND SUBRAMANIAN

**METAHEURÍSTICA *ITERATED LOCAL SEARCH* APLICADA AO
PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E
ENTREGA SIMULTÂNEA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção do Centro de Tecnologia da Universidade Federal da Paraíba, como requisito parcial para obtenção do grau de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Lucídio dos Anjos Formiga Cabral

**JOÃO PESSOA - PB
2008**

S941m Subramanian, Anand
Metaheurística *Iterated Local Search* aplicada ao Problema de Roteamento de Veículos com Coleta e Entrega Simultânea / Anand Subramanian – João Pessoa: UFPB, 2008.
86 f. il.:

Orientador: Prof. Lucídio dos Anjos Formiga Cabral Dr.

Dissertação (Mestrado em Engenharia de Produção) – DEP /CT / Universidade Federal da Paraíba – UFPB.

1. Roteamento de Veículos 2. Coleta e Entrega Simultânea
3. *Iterated Local Search* 4. VND I. Título.

CDU: 519.8 (043)

ANAND SUBRAMANIAN

**METAHEURÍSTICA *ITERATED LOCAL SEARCH* APLICADA AO
PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E
ENTREGA SIMULTÂNEA**

Dissertação apresentada e aprovada, em 22 de Fevereiro de 2008, como requisito parcial à obtenção do grau de mestre em Engenharia de Produção do Programa de Pós-Graduação em Engenharia de Produção (PPGEP) da Universidade Federal da Paraíba, pela comissão formada pelos seguintes membros:

BANCA EXAMINADORA

Prof. Dr. Lucídio dos Anjos Formiga Cabral - Orientador
PPGEP - UFPB

Prof. Dr. Roberto Quirino do Nascimento - Examinador
PPGEP - UFPB

Prof. Dr. Luiz Satoru Ochi – Examinador
Instituto de Computação - UFF

Aos meus pais.

AGRADECIMENTOS

Aos meus pais, por todo apoio, ensinamentos e investimento em minha formação.

Ao Prof. Dr. Lucídio dos Anjos Formiga Cabral (PPGEP/UFPB), pessoa extraordinária, a quem devo muito pela orientação, confiança e amizade.

Ao Prof. Dr. Ulisses Umbelino dos Anjos (DE/UFPB) e ao Prof. MSc. Jozemar Pereira dos Santos (DE/UFPB), pelas discussões e colaborações referentes à análise estatística dos resultados da pesquisa.

Ao Prof. Dr. Luiz Satoru Ochi (IC/UFF) e ao Prof. Dr. Roberto Quirino do Nascimento (DE/UFPB), pelas importantes sugestões e contribuições relativas a esta dissertação

Ao meu irmão Aravind, minha cunhada Ramya e meu grande amigo Madhavan, que, apesar da distância, sempre me apoiaram em todos os momentos.

Aos colegas Marcel de Góis Pinto, Liane Márcia Freitas e Silva e Hugo Harry F. R. Kramer, pela amizade e trabalhos desenvolvidos nesses últimos dois anos.

A Alan Kelon Oliveira de Moraes, pela ajuda na aquisição de alguns artigos.

Aos amigos José Maurício Fernandes Medeiros e Gustavo Rezende Carvalho, pelas dicas de C/C++ concedidas há dois anos, mas que foram de bastante valia.

A Philipe Moura, pelas correções de português efetuadas nesta dissertação.

A CAPES pelo apoio financeiro através da concessão de uma bolsa de mestrado.

RESUMO

Esta dissertação trata do Problema de Roteamento de Veículos com Coleta e Entrega Simultânea. Um procedimento baseado na metaheurística *Iterated Local Search* (ILS) que utiliza o método de descida em vizinhança variável (VND) para realização da busca local é proposto. A estratégia utilizada para geração da solução inicial é fundamentada no princípio da “inserção viável mais barata” associada a um critério de bonificação de clientes longínquos que, por sua vez, possui um parâmetro, $\gamma \in [0,1]$, para mensurar a intensidade deste favorecimento. Dez estruturas de vizinhança são empregadas na fase de busca. Destas, seis realizam movimentos entre rotas (*Shift(1,0)*, *Crossover*, *Swap(1,1)*, *Shift(2,0)*, *Swap(2,1)*, *Swap(2,2)*) e quatro dentro das rotas (*Or-opt*, *2-opt*, *Exchange*, *Reverse*). Três mecanismos de perturbação são utilizados, a saber: *Ejection Chain*, *Double-Swap* e *Double-Bridge*. O algoritmo desenvolvido foi testado em cerca de cem instâncias encontradas na literatura relativas ao PRVCEs, tendo sido capaz de melhorar 50% das soluções conhecidas. Além disso, uma nova formulação matemática baseada em fluxo em redes com duas comodidades é apresentada.

Palavras-chave: Roteamento de Veículos. Coleta e Entrega Simultânea. *Iterated Local Search*. VND.

ABSTRACT

This dissertation deals with the Vehicle Routing Problem with Simultaneous Pickup and Delivery. A procedure based on the Iterated Local Search (ILS) metaheuristic which uses the Variable Neighborhood Descent (VND) method for performing the local search is proposed. The strategy utilized for generating the initial solution relies on the “cheapest feasible insertion” principle, associated with a benefit criterion of remotely located clients, which in turn, has a parameter, $\gamma \in [0,1]$, to measure its weight. Ten neighborhood structures are used in the local search phase. Six of them perform movements between the routes (Shift(1,0), Crossover, Swap(1,1), Shift(2,0), Swap(2,1), Swap(2,2)) and four of them inside the routes (Or-opt, 2-opt, Exchange, Reverse). Three perturbation mechanisms are used, namely: Ejection Chain, Double-Swap and Double-Bridge. The algorithm developed was tested in about hundred instances and it was capable of improving 50% of the known solutions. In addition, a new mathematical model based on the two-commodity network flow formulation is presented.

Keywords: Vehicle Routing. Simultaneous Pickup and Delivery. Iterated Local Search. VND.

LISTA DE FIGURAS

Figura 1 – Ilustração de um exemplo do PRV.....	18
Figura 2 – PRV com <i>backhauls</i>	22
Figura 3 – PRV com Coleta e Entrega Mista	24
Figura 4 – Problema <i>Dial-a-ride</i>	25
Figura 5 – PRV com Coleta e Entrega Simultânea	27
Figura 6 – Procedimento ILS	32
Figura 7 – Ilustração do procedimento ILS.....	33
Figura 8 – Exemplo esquemático da formulação proposta.....	40
Figura 9 – Pseudocódigo do algoritmo ILS-VND.....	45
Figura 10 – Pseudocódigo do procedimento construtivo	46
Figura 11 – Movimento <i>Shift</i> (1,0).....	48
Figura 12 – Movimento <i>Crossover</i>	49
Figura 13 – Movimento <i>Swap</i> (1,1).....	49
Figura 14 – Pseudocódigo do algoritmo VND	50
Figura 15 – Movimento <i>Ejection Chain</i>	52
Figura 16 – Movimento <i>Double-Bridge</i>	52
Figura 17 – Influência do parâmetro γ nas instâncias SCA3.....	67
Figura 18 – Influência do parâmetro γ nas instâncias SCA8.....	68
Figura 19 – Influência do parâmetro γ nas instâncias CON3	68
Figura 20 – Influência do parâmetro γ nas instâncias CON8	69
Figura 21 – Influência do parâmetro γ nas instâncias CMT	70
Figura 22 – Influência do parâmetro γ nas instâncias CMT com <i>drop time</i>	71
Figura 23 – Influência do parâmetro γ nas instâncias CMT sem <i>drop time</i>	71
Figura 24 – Influência do parâmetro γ nas instâncias de Montané e Galvão	73

LISTA DE TABELAS

Tabela 1 – Resultados obtidos pelo ILS-VND nas instâncias de Dethloff.....	58
Tabela 2 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Dethloff.....	58
Tabela 3 – Comparação entre o ILS-VND e os resultados de Bianchessi e Righini nas instâncias de Dethloff.....	59
Tabela 4 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy.....	60
Tabela 5 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy.....	61
Tabela 6 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy com restrições de limite de tempo (com <i>drop time</i>)	62
Tabela 7 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy com restrições de limite de tempo (com <i>drop time</i>).....	63
Tabela 8 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy com restrições de limite de tempo (sem <i>drop time</i>).....	64
Tabela 9 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy com restrições de limite de tempo (sem <i>drop time</i>)	64
Tabela 10 – Resultados obtidos pelo ILS-VND nas instâncias de Montané e Galvão.....	65
Tabela 11 – Comparação entre o ILS-VND e os resultados de Montané e Galvão nas instâncias dos mesmos.....	65
Tabela 12 – Resultados obtidos pelo ILS-VND na instância de Min.....	66
Tabela 13 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Min	66
Tabela 14 – Valores de γ associados às melhores soluções médias nas instâncias de Dethloff	69
Tabela 15 – Valores de γ associados às melhores soluções médias nas instâncias de Salhi e Nagy	72
Tabela 16 – Valores de γ associados às melhores soluções médias nas instâncias de Montané e Galvão.....	73

LISTA DE SIGLAS

PRV	Problema de Roteamento de Veículos
PRVCE	Problema de Roteamento de Veículos com Coleta e Entrega
PRVCEs	Problema de Roteamento de Veículos com Coleta e Entrega Simultânea
PRVCEM	Problema de Roteamento de Veículos com Coleta e Entrega Mista
PRVB	Problema de Roteamento de Veículos com <i>Backhauls</i>
DARP	<i>Dial-a-ride Problem</i>
VNS	<i>Variable Neighborhood Search</i>
VND	<i>Variable Neighborhood Descent</i>
ILS	<i>Iterated Local Search</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
BT	Busca Tabu
LC	Lista de Candidatos
NP	<i>Non-Polynomial</i>
ANOVA	Análise de Variância

SUMÁRIO

1 INTRODUÇÃO	13
1.1 DEFINIÇÃO DO TEMA.....	13
1.2 JUSTIFICATIVA	14
1.3 OBJETIVOS.....	17
1.3.1 Objetivo Geral	17
1.3.2 Objetivos Específicos	17
2 REVISÃO BIBLIOGRÁFICA	18
2.1 PROBLEMAS DE ROTEAMENTO DE VEÍCULO COM COLETA E ENTREGA	21
2.1.1 PRV com <i>Backhauls</i> (PRVB).....	21
2.1.2 PRV com Coleta e Entrega Mista (PRVCEM).....	23
2.1.3 O Problema <i>Dial-a-Ride</i> (DARP)	25
2.1.4 PRV com Coleta e Entrega Simultânea.....	27
2.2 METAHEURÍSTICAS.....	30
2.3 <i>ITERATED LOCAL SEARCH</i>	31
2.4 MODELOS MATEMÁTICOS PROPOSTOS PARA O PRVCES	34
3. FORMULAÇÃO MATEMÁTICA PROPOSTA	39
4 PROCEDIMENTOS METODOLÓGICOS	44
4.1 ALGORITMO PROPOSTO.....	44
4.1.1 Procedimento Construtivo	45
4.1.2 Procedimento de Busca Local	47
4.1.3 Mecanismos de Perturbação	51
4.2 CONJUNTO DE INSTÂNCIAS	53
4.2.1 Min (1989).....	53
4.2.2 Salhi e Nagy (1999).....	53
4.2.3 Dethloff (2001)	54
4.2.4 Montané e Galvão (2006).....	55
4.3 PARÂMETROS DO ALGORITMO	55
5 RESULTADOS COMPUTACIONAIS	57
5.1 COMPARAÇÕES ENTRE ABORDAGENS.....	57

5.2 INFLUÊNCIA DO PARÂMETRO γ	67
6 CONCLUSÕES	75
REFERÊNCIAS BIBLIOGRÁFICAS	77

1 INTRODUÇÃO

1.1 DEFINIÇÃO DO TEMA

O Problema de Roteamento de Veículos (PRV) é um problema clássico de otimização combinatória proposto no final da década de 50, sendo ainda um dos mais estudados na área de Pesquisa Operacional. O grande interesse no PRV se deve a sua importância prática, bem como ao alto grau de dificuldade em resolvê-lo.

Segundo Oliveira e Arenales (2003), os termos *roteirização* ou *roteamento* de veículos, não são encontrados nos dicionários de língua portuguesa. Na verdade, são formas que vêm sendo utilizadas como equivalentes ao inglês *routing* (ou *routeing*) para designar o processo de determinação de um ou mais roteiros ou seqüências de paradas a serem cumpridos por veículos de uma frota, objetivando visitar um conjunto de pontos geograficamente dispersos, em locais pré-determinados, que necessitem de atendimento.

De acordo com Berbeglia *et al.* (2006), o Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE), isto é, no qual pessoas ou objetos devem ser recolhidos e distribuídos, constituem uma importante classe do PRV. No final dos anos 80, Min (1989) propôs uma nova vertente para o PRVCE, onde em determinadas situações, os procedimentos de coleta e entrega são efetuados simultaneamente em cada cliente, caracterizando uma das principais variantes desta classe: o Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES).

A partir dos anos 90 houve um crescimento de aplicações de metaheurísticas às diversas variantes do PRV. As metaheurísticas consistem em um processo iterativo que conduzem e modificam as operações de uma heurística subordinada, de tal forma a produzir, eficientemente, soluções de alta qualidade (OSMAN, 2002). O termo metaheurística foi

inicialmente proposto por Glover (1986) e teve sua origem a partir da composição de duas palavras gregas. *Heurística* é derivada do verbo *heuriskein* que significa “procurar” e o sufixo *meta* significa “além, em um nível superior” (BLUM; ROLI, 2003). Dentre as existentes, pode-se citar Busca Tabu, *Simulated Annealing*, Algoritmos Genéticos, Colônia de Formigas, VNS (*Variable Neighborhood Search*), GRASP (*Greedy Randomized Adaptive Search Procedure*), ILS (*Iterated Local Search*), etc.

Nesta dissertação é apresentada uma aplicação da metaheurística ILS, que pode ser definida como sendo um método de busca local estocástico que procura focar a pesquisa em certo subespaço definido por soluções que são ótimas locais de determinado procedimento de otimização (LOURENÇO *et al.*, 2002).

À luz de Cordeau *et al.* (2005), algoritmos de busca local são tipicamente utilizados para tentar refinar soluções iniciais geradas por outras heurísticas. Partindo de uma dada solução, um método de busca local aplica modificações (movimentos) simples, como troca de arcos ou realocação de clientes, com o intuito de se tentar obter soluções vizinhas de qualidade superior.

Neste contexto, será utilizada a metaheurística ILS para solucionar o PRVCES, isto é: **dado um conjunto de clientes, cada qual com uma demanda p_i por coleta e q_i por entrega, e um depósito com veículos de capacidade Q , quais os conjuntos de rotas capazes de minimizar os custos de transporte?**

1.2 JUSTIFICATIVA

Pode-se afirmar que o PRV, em geral, é uma questão deveras importante na gestão da cadeia de suprimentos de uma empresa, e é tratado diariamente por diversos tipos de organizações envolvidas com serviços de distribuição.

Toth e Vigo (2002) afirmam que o uso de métodos computacionais em processos de distribuição freqüentemente resulta em economia da ordem de 5% a 20% nos custos de transporte. Golden *et al.* (2002) e Barker (2002) descrevem vários estudos de caso, nos quais, a aplicação de algoritmos ao PRV tem conduzido a reduções de custo substanciais.

No caso do PRVCES, verifica-se, além da distribuição, a necessidade de coleta de bens ou pessoas. Alguns exemplos podem ser observados na indústria de bebidas, onde as garrafas cheias são entregues ao passo que as vazias são recolhidas; em mercearias, nas quais os *pallets* ou recipientes são reutilizados para o transporte de mercadorias, etc. É importante salientar que pode haver a exigência, da parte de alguns clientes, de que os serviços de coleta e entrega não devam ser prestados separadamente, pois isto pode acarretar, em determinados casos, num acréscimo dos custos e esforços operacionais para os mesmos.

Assim sendo, deve-se existir uma preocupação não só com a Logística de Distribuição, mas também com o gerenciamento do fluxo reverso. É neste contexto que emerge o conceito de Logística Reversa, que pode ser definido como sendo o processo de planejar, implementar e controlar o retorno de matérias-primas, inventários em processo, produtos acabados e informações relativas ao ponto de consumo até o ponto de origem (ROGERS; TIBBEN-LEMBKE, 1999). Em seu sentido mais amplo, a Logística Reversa abrange todas as operações relacionadas à reutilização de produtos e materiais. Refere-se a todas as atividades logísticas de coletar, desmontar e processar produtos e/ou materiais e peças utilizados com o objetivo de garantir uma recuperação sustentável.

A prática da Logística Reversa vem tomando proporções maiores nos últimos anos. Segundo De Brito *et al.* (2003), as razões pelas quais as empresas adotam este tipo de estratégia podem estar relacionadas a três fatores, a saber: (i) econômico, que inclui redução de custos, melhoria da relação entre cliente/consumidor, entre outros; (ii) legislação, ou seja,

direitos do consumidor, leis ambientais, etc.; (iii) “cidadania corporativa”, quando a empresa busca passar uma boa imagem perante a sociedade (responsabilidade social).

Desta forma, a Logística de Distribuição e a Logística Reversa devem atuar em conjunto com o objetivo de garantir a sincronia entre as operações de coleta e entrega, bem como o impacto destas na gestão da cadeia de suprimentos da empresa, culminando na satisfação do cliente e minimização do esforço operacional. Para tanto, deve-se atentar para algumas problemáticas a serem solucionadas e uma delas é justamente a otimização do conjunto das rotas a serem percorridas pela frota de veículos existente.

Porém, não se trata de uma tarefa fácil, visto que o PRVCES é um problema pertencente à classe NP-*hard* (ANILY, 1996, DETHLOFF, 2001), onde a determinação da solução ótima em um tempo aceitável é praticamente inviável, ainda que as instâncias sejam de dimensão reduzida. Diversas abordagens exatas e heurísticas foram propostas aos problemas derivados do PRVCE, contudo uma atenção especial merece ser dada as metaheurísticas. As aplicações destas a problemas relacionados ao PRV (GENDREAU *et al.*, 1999) estão se tornando cada vez mais freqüentes em virtude de seu sucesso na obtenção de soluções de qualidade com baixo esforço computacional.

A literatura apresenta várias aplicações, relativamente bem sucedidas, da metaheurística ILS. Entretanto, não houve, até o presente momento, a aplicação desta ao PRVCES, que, segundo Lourenço *et al.* (2002), é uma metaheurística conceitualmente simples, de fácil implementação, robusta e eficaz.

A busca por métodos eficientes na resolução de problemas referentes ao PRVCE é uma área de intensa pesquisa, tendo em vista seu empenho na melhoria/automatização do processo de gestão das empresas prestadoras de serviços de coleta e entrega de mercadorias ou pessoas. A utilização da metaheurística ILS, praticamente inexplorada na resolução do

PRVCES até o presente momento, parece ser uma alternativa interessante para se gerar soluções de qualidade com baixo custo computacional.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Aplicar a metaheurística ILS para resolver o Problema de Roteamento de Veículos com Coleta e Entrega Simultânea.

1.3.2 Objetivos Específicos

- Definir um procedimento construtivo para gerar uma solução inicial.
- Propor uma nova formulação matemática para o PRVCES.
- Desenvolver um algoritmo baseado na metaheurística ILS para refinar a solução inicialmente obtida.
- Averiguar diversas possibilidades de movimentos (estruturas de vizinhanças) para melhoria das soluções.
- Examinar diferentes mecanismos de perturbação de uma solução que se encontra em um ponto de mínimo local.
- Testar o algoritmo proposto em diferentes instâncias constantes na literatura.
- Comparar os resultados obtidos com os encontrados na literatura até então.

2 REVISÃO BIBLIOGRÁFICA

O PRV foi proposto por Dantzig e Ramser (1959), onde os autores descrevem uma aplicação real concernente à distribuição de gasolina a estações de serviços. Desde então, uma imensa quantidade de trabalhos relativos ao PRV foram tratados na literatura. O problema pode ser descrito da seguinte forma. Dado um conjunto de clientes, cada qual com uma demanda q_i previamente conhecida, e um depósito onde estão localizados veículos de capacidade Q , determinar quais os conjuntos de rotas capazes de minimizar os custos de transporte. A Figura 1 ilustra um exemplo envolvendo três rotas (ou pétalas).

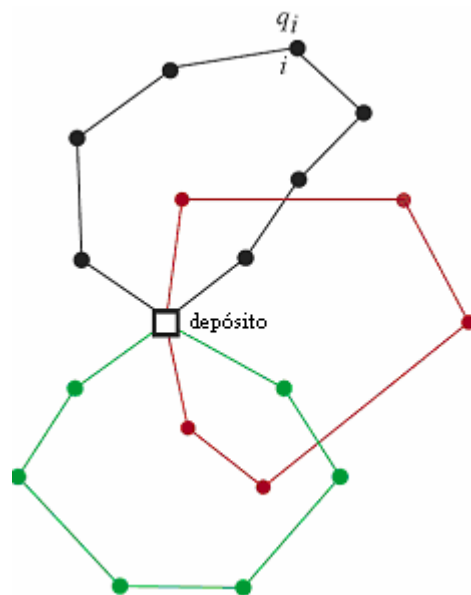


Figura 1 – Ilustração de um exemplo do PRV

Um panorama geral sobre o PRV, incluindo metodologias para resolução, aplicações práticas, além de uma extensa revisão literária acerca do problema, pode ser encontrado em Golden e Assad (1988), Toth e Vigo (2002), Cordeau *et al.* (2005) e Laporte (2007).

Reconhecido como NP-*hard* (LENSTRA, 1981), o PRV possui inúmeros algoritmos exatos propostos na literatura (veja CORDEAU *et al.*, 2005), contudo, são geralmente viáveis apenas para pequenas instâncias. De modo geral, as heurísticas têm se mostrado mais adequadas para o PRV. Laporte e Semet (1999) as dividem em dois grupos distintos: heurísticas clássicas, desenvolvidas entre a década de 60 e 90, e as metaheurísticas, cujo crescimento se deu no final da década de 80. Os autores fazem uma revisão das heurísticas clássicas e a aplicação de metaheurísticas ao PRV. Bodin e Golden (1981) classificam as estratégias heurísticas destinadas ao PRV da seguinte forma: (i) primeiro agrupar depois rotear (*cluster-first / route-second*); (ii) primeiro rotear depois agrupar (*route-frist / cluster-second*); (iii) economias / inserções; (iv) refinamento / trocas; (v) baseadas em programação matemática; (vi) otimização interativa. Cordeau *et al.* (2004) publicaram um trabalho contendo as mais recentes heurísticas dirigidas ao PRV.

Alguns complicadores podem ser adicionados através da inserção de diferentes restrições, resultando assim em diversas variantes relacionadas do PRV. As restrições encontradas com maior frequência na literatura são as de:

- **Capacidade:** cada cliente possui uma demanda não-negativa q_i por entrega, onde o somatório das demandas não deve exceder a capacidade Q do veículo, ou seja, $\sum q_i \leq Q$.
- **Janelas de tempo:** em cada cliente i , o serviço de coleta ou entrega deve iniciar entre um dado intervalo de tempo (janela) $[a_i, b_i]$. Um veículo é autorizado a chegar antes de a_i e esperar até o cliente ficar disponível, porém chegadas após b_i são proibidas. O PRV com janelas de tempo possui numerosas aplicações no gerenciamento da distribuição. Exemplos comuns são entregas de comida, entregas de jornais, coletas de lixo, etc. Uma revisão sobre

os métodos utilizados para lidar com esta variante pode ser encontrado em Bräysy e Gendreau (2005a, 2005b).

- **Componentes estocásticos:** os principais casos envolvem (i) clientes aleatórios, onde o cada um possui probabilidade p_i de estar presente e $1 - p_i$ de estar ausente; (ii) demandas (por coleta ou entrega) estocásticas, em que a demanda do cliente segue certa distribuição de probabilidade; (iii) tempos estocásticos, onde os tempos de serviço t_i e os tempos de viagens t_{ij} são variáveis aleatórias (CORDEAU *et al.*, 2005). Gendreau *et al.* (1996) levantaram o estado da arte das principais abordagens do PRV estocástico.
- **Múltiplos depósitos:** nesta variante, os veículos podem partir de mais de um depósito.
- **Heterogeneidade da frota (Frota mista):** neste caso, os veículos de uma frota apresentam capacidades Q_i diferentes.
- **Periodicidade:** o planejamento é efetuado em um horizonte temporal que engloba T dias. Quando $T = 1$, o problema se resume ao PRV clássico.
- **Entrega dividida:** veículos realizam o atendimento do cliente dividindo a demanda.
- **Coleta e entrega:** os serviços de coleta e entrega são prestados no mesmo *tour*. Os principais problemas pertencentes a esta variante são: PRV com Entrega prioritária (*Backhauls*), PRV com Coleta e Entrega Mista, o Problema *Dial-a-ride*, PRV com Coleta e Entrega Simultânea. A Seção 2.1 trata em detalhes destas vertentes.

2.1 PROBLEMAS DE ROTEAMENTO DE VEÍCULO COM COLETA E ENTREGA

Esta Seção apresenta os problemas de roteamento de veículos que envolvem coleta e entrega. As variantes aqui abordadas são aquelas que apresentam uma relação mais direta com o PRVCE. Revisões sobre as variantes do PRVCE e métodos utilizados para resolvê-los podem ser encontradas em Savelsbergh e Sol (1995) e Berbeglia *et al.* (2006).

De maneira geral, os problemas envolvendo coleta e entrega podem ser descritos da maneira que se segue. Seja $G = (V, E)$ um grafo completo e direcionado com um conjunto de vértices $V = \{0, \dots, n\}$, onde o vértice 0 representa o depósito ($V_0 = \{0\}$) e os vértices restantes representam os clientes. Cada aresta $(i, j) \in E$ possui um custo não-negativo c_{ij} que satisfaz a desigualdade triangular. Cada cliente $i \in V - V_0$ possui demanda $q_i \in D$ por entrega e $p_i \in P$ por coleta, onde D e P são os conjuntos contendo as quantidades de um determinado bem (ou pessoas) a serem distribuídos e recolhidos respectivamente e $C = \{1, \dots, m\}$ o conjunto de veículos disponíveis, cada qual com capacidade Q .

O PRVCE consiste em construir um conjunto de no máximo m rotas que satisfaçam aos seguintes requisitos: (i) todas as demandas por coleta e entrega devem ser atendidas; (ii) a capacidade do veículo não deve ser extrapolada; (iii) a soma dos custos deve ser minimizada.

2.1.1 PRV com *Backhauls* (PRVB)

Dentre os problemas que envolvem coleta e entrega, o PRVB é o mais abordado na literatura. Nesta variante, também conhecida como *linehaul-backhaul*, todas as entregas

(*linehauls*) são realizadas antes de qualquer coleta (*backhauls*) ser efetuada (Figura 2). Salienta-se que uma rota não pode ser formada apenas por *backhauls*, ao passo que é permitido que uma rota contenha somente *linehauls*. Goetschalckx e Jacobs-Blecha (1989) destacam que a importância do PRVB está relacionada ao esforço contínuo na redução de custos aproveitando-se da capacidade inutilizada de um veículo vazio durante seu retorno ao depósito. Recentemente, Toth e Vigo (2002) apresentaram uma ampla revisão sobre o PRVB.

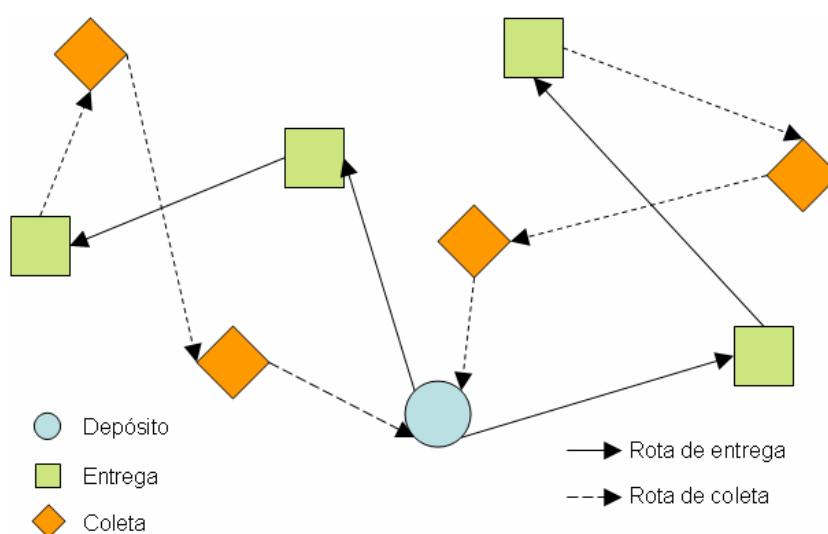


Figura 2 – PRV com *backhauls*

No caso particular em que se considera um único veículo, o PRVB se transforma em um PCVB (Problema do Caixeiro Viajante com *Backhauls*). Para esta situação, Anily e Mosheiov (1994) sugerem uma heurística inspirada no problema da árvore geradora mínima. Gendreau *et al.* (1996) propõem seis procedimentos heurísticos de duas fases. Mladenovic e Hansen (1997) utilizam a metaheurística VNS combinada com alguns algoritmos sugeridos em Gendreau *et al.* (1996), enquanto Gharizi e Osman (2002) desenvolveram um procedimento heurístico baseado em redes neurais.

Para a situação nas quais múltiplos veículos são considerados, Golden *et al.* (1985) e Casco *et al.* (1988) propõem diversas heurísticas inspiradas no algoritmo clássico de

Clarke e Wright (1964). Uma heurística de duas fases é sugerida por Goetschalckx e Jacobs-Blecha (1989). Outros algoritmos heurísticos foram desenvolvidos por Anily (1996), Toth e Vigo (1999), Ropke e Pisinger (2004). Procedimentos baseados na metaheurística Busca Tabu foram sugeridos por Brandão (1996) e Osman e Wassan (2002). Carreto e Baker (2002) fizeram uso da metaheurística GRASP, cuja fase de construção é baseada em uma heurística de clusterização e a etapa de busca local utiliza uma estrutura de vizinhança do tipo *3-opt*. Thangiah *et al.* (1996), Duhamel *et al.* (1997), Zhong e Cole (2005) e Reimann e Ulrich (2006) desenvolveram heurísticas levando-se em conta as restrições de janela de tempo, onde neste último trabalho foi utilizada a metaheurística Colônia de Formigas. Algoritmos exatos foram propostos por Toth e Vigo (1997b) e Mingozzi *et al.* (1999).

2.1.2 PRV com Coleta e Entrega Mista (PRVCEM)

Nesta situação, as coletas e entregas podem ocorrer de forma alternada e os veículos partem e chegam ao mesmo depósito, conforme mostra a Figura 3. Para o caso especial onde somente um veículo é considerado (PCVCEM), Mosheiov (1994) propõe dois procedimentos heurísticos, onde um deles, denominado “inserção viável mais barata”, é fundamentado na reconhecida heurística de inserção mais barata desenvolvida para a versão clássica do PCV. Gendreau *et al.* (1999) desenvolveram duas heurísticas: a primeira é um algoritmo linear e exato, aplicada quando o grafo é um ciclo, e a segunda é baseada na metaheurística Busca Tabu. Baldacci *et al.* (2003) fizeram uso de uma abordagem exata por meio de um algoritmo *branch-and-cut*, utilizando uma formulação de fluxo em redes.

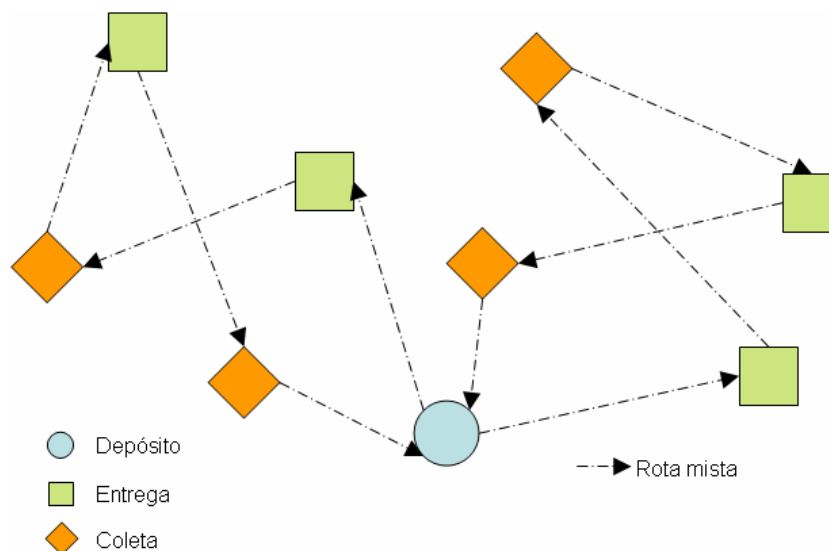


Figura 3 – PRV com Coleta e Entrega Mista

No caso em que é admitido mais de um veículo, Golden *et al.* (1985) desenvolveram uma heurística, cujos *backhauls* são inseridos em rotas formadas por *linehauls*. Casco *et al.* (1988) também propõem heurísticas de inserção (*load base*) às quais são atribuídas penalizações sobre o custo de inserção em função da carga restante a ser entregue após a inclusão de determinado cliente na rota, ao invés de se levar em conta o número de clientes restantes a serem visitados. Mosheiov (1998) desenvolveu duas heurísticas de particionamento de rotas que consistem na quebra de uma rota em segmentos disjuntos, onde cada um destes é servido por veículos diferentes.

Alguns pesquisadores exploraram o PRVCEM com restrições de janela de tempo. Dumas *et al.* (1991) desenvolveram um algoritmo exato que utiliza um esquema de geração de colunas. Bruggen *et al.* (1993) sugerem um método de busca local baseada em uma busca em profundidade variável. Nanry e Barnes (2000) sugerem um algoritmo Busca Tabu reativo com três estruturas de vizinhança distintas. A mesma metaheurística também é utilizada por Landrieu *et al.* (2001), Duchenne *et al.* (2005), ao passo que Caricato *et al.* (2003) propõem um algoritmo tabu paralelo.

2.1.3 O Problema *Dial-a-Ride* (DARP)

O DARP é um problema de coleta e entrega geralmente relacionado a situações que envolvem o transporte de pessoas. O escopo do DARP está relacionado à determinação de um conjunto de rotas, a custo mínimo, capazes de acomodar o maior número de passageiros sob certas condições restritivas (CORDEAU; LAPORTE, 2003b). A Figura 4 demonstra uma situação em que quatro clientes são recolhidos em localidades distintas e entregues em diferentes destinos. Observa-se que os veículos partem e retornam ao depósito sem passageiros.

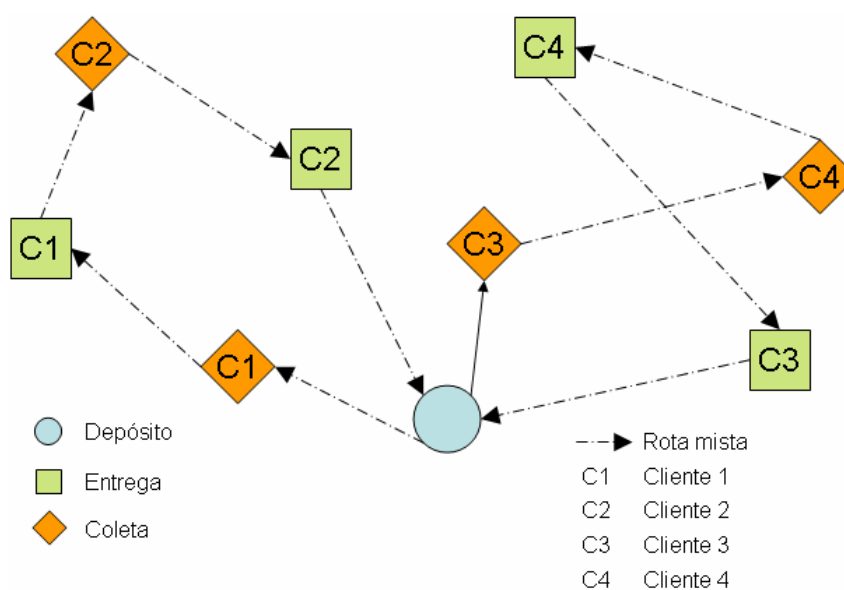


Figura 4 – Problema *Dial-a-ride*

O estado da arte acerca do DARP pode ser encontrado em Cordeau e Laporte (2003a). Os autores afirmam que, em países ocidentais, várias autoridades locais estão implantando serviços do tipo *dial-a-ride*, ou revisando aqueles que estão sendo utilizados, em função da crescente demanda. Este fenômeno pode estar atribuído em parte ao envelhecimento populacional, bem como à tendência de desenvolvimento de serviços de cuidado à saúde.

Psaraftis (1980) modelou e resolveu o DARP, quando apenas um veículo é utilizado, como um problema de programação dinâmica. Posteriormente, o autor adaptou o algoritmo previamente proposto levando-se em conta restrições de janela de tempo (PSARAFTIS, 1983). O problema também é abordado por Sexton e Bodin (1986a, 1986b). Um algoritmo exato baseado em programação dinâmica é sugerido por Desrosiers *et al.* (1986).

Jaw *et al.* (1986) desenvolveram uma heurística de inserção seqüencial para a situação em que múltiplos veículos e restrições de janela de tempo são considerados. Bodin e Sexton (1986) propõem um algoritmo do tipo “primeiro agrupar, depois rotear”, posteriormente refinado por Dumas *et al.* (1989), que sugerem um método de formação “*mini-clusters*” na fase inicial do algoritmo, onde este foi aprimorado por Desrosiers *et al.* (1991). A utilização deste método também é explorada por Ioachim *et al.* (1995). Uma heurística de busca local foi implementada por Toth e Vigo (1996) para lidar com uma situação real em Bologna e depois a mesma foi aprimorada pelos autores por meio de um algoritmo tabu (TOTH; VIGO, 1997a).

Cordeau e Laporte (2003b) utilizam a metaheurística Busca Tabu e testaram o procedimento proposto tanto em instâncias geradas artificialmente como em seis casos reais de uma transportadora dinamarquesa. Baba *et al.* (2004) fizeram uso da metaheurística Colônia de Formigas em um estudo de caso realizado na Empresa de Desenvolvimento Urbano e Social (URBES), no município de Sorocaba-SP. Mauri e Lorena (2006) descrevem um modelo multi-objetivo e aplicam um procedimento baseado na metaheurística *Simulated Annealing* para solucioná-lo. Procedimentos exatos foram desenvolvidos por Cordeau (2006) e Ropke *et al.* (2007) onde ambos têm por base um algoritmo do tipo *branch-and-cut*.

2.1.4 PRV com Coleta e Entrega Simultânea

Nesta variante, assim como no PRVB e no PRVCEM, os veículos partem do depósito com os itens a serem entregues e retornam ao depósito com itens coletados ao longo da rota. O que difere o PRVCES das demais situações é o fato de os serviços de coleta e entrega serem efetuados numa única visita ao cliente, como ilustra a Figura 5.

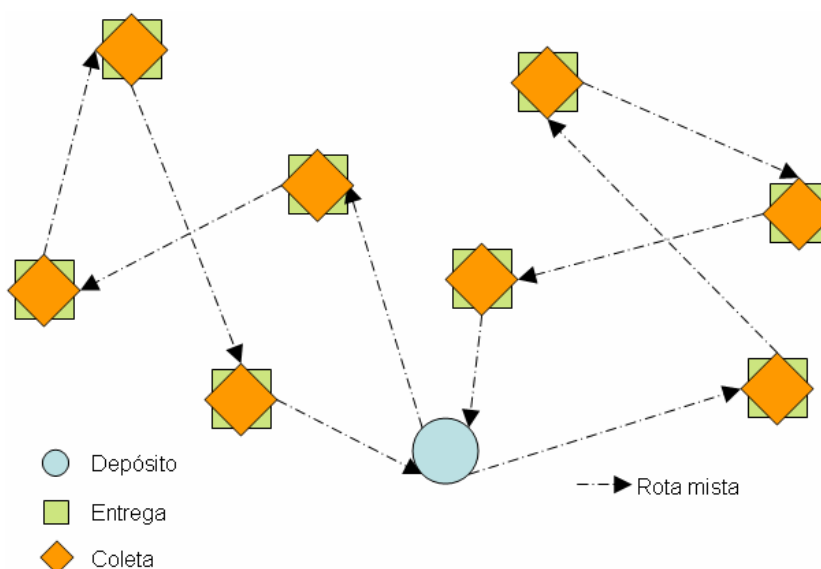


Figura 5 – PRV com Coleta e Entrega Simultânea

O PRVCES foi proposto por Min (1989), onde o autor ilustra uma aplicação prática do problema através de um estudo de caso realizado no sistema de distribuição de uma biblioteca pública. O procedimento utilizado para resolução deste problema envolveu as seguintes etapas: (i) agrupamento de clientes (nós) em *clusters* de tal forma que cada grupo não exceda a capacidade do veículo designado para cada *cluster*; (ii) designação dos veículos para os *clusters*; (iii) determinação da seqüência do serviço de coleta e entrega simultânea de maneira que a carga acumulada em cada parada não exceda a capacidade do veículo, através de uma heurística destinada à resolução do PCV.

Desde então, poucos trabalhos (de conhecimento do autor) se debruçaram na abordagem desta variante. Apenas uma década, depois Salhi e Nagy (1999) sugeriram algumas heurísticas de inserção levando em consideração múltiplos depósitos. Duas delas são basicamente heurísticas gulosas inspiradas nos procedimentos de inserção propostos por Golden *et al.* (1985) e Casco *et al.* (1988). Uma outra consiste em um procedimento de inserção de *clusters*, que é uma extensão dos outros dois procedimentos gulosos. Recentemente, os mesmos autores desenvolveram uma heurística (NAGY; SALHI, 2005) que envolve certos níveis de viabilidade das soluções.

Dethloff (2001) trata o PRVCES sob os diversos aspectos da logística reversa e propõe um procedimento construtivo (heurística de inserção), fundamentado nos conceitos de inserção mais barata; de beneficiamento em função da distância radial; e da capacidade residual, onde este último é uma adaptação da abordagem *load-base* (CASCO *et al.*, 1988), além de investigar as relações deste problema com outras variantes do PRV.

Vural (2003) faz uso de dois métodos fundamentados em Algoritmos Genéticos. O primeiro está relacionado ao método de chaves aleatórias (*Random Key Method*), enquanto o segundo consiste em uma heurística de melhoramento. Gökçe (2004) utiliza a metaheurística Colônia de Formigas, cujo algoritmo encontra-se dividido em quatro etapas, a saber: (i) primeiramente, uma lista de candidatos é formada para cada cliente com o objetivo de reduzir o tempo computacional; (ii) em seguida, uma solução viável é encontrada e as quantidades de feromônio são determinadas através desta; (iii) posteriormente, o nível de feromônio é atualizado localmente e globalmente; (iv) finalmente, as rotas são melhoradas através da utilização de um algoritmo *2-opt*.

Crispim e Brandão (2005) apresentam um procedimento híbrido onde metheurísticas Busca Tabu e VND (*Variable Neighbourhood Descent*) são combinadas. Ropke e Pisinger (2004) desenvolveram uma heurística que faz uso de extensas vizinhanças

(*Large Neighborhood Search*) associado a um procedimento similar a metaheurística VNS. Além do PRVCES, o algoritmo desenvolvido foi testado em outras variações do PRVCE.

Montané e Galvão (2006) propõem uma heurística Busca Tabu associado a diversos tipos de vizinhança, sendo três destas relacionados aos movimentos de realocação, troca e *crossover* (entre as rotas) e um quarto movimento consiste na combinação dos mesmos. Um procedimento *2-opt* (dentro das rotas) foi utilizado com o intuito de se obter soluções alternativas para o problema. Duas estratégias de buscas foram empregadas, onde a primeira delas considera o primeiro movimento admissível enquanto a outra considera o melhor movimento admissível.

Uma heurística Busca Tabu também foi implementada por Gribkovskaia *et al.* (2006), para o caso em que apenas um veículo é considerado. Os autores propõem um modelo de programação inteira mista que admite que as coletas e entregas sejam realizadas tanto de forma mista como simultaneamente ao longo da rota. O algoritmo gerador da solução inicial desenvolvido é fundamentado nas heurísticas clássicas de construção e melhoramento.

Chen e Wu (2006) utilizam uma heurística de inserção para gerar soluções e procedimentos de refinamento baseado na aproximação *record-to-record travel* e listas tabu, e de melhoramento de rotas. Chen (2006) desenvolveu uma heurística híbrida que combina os princípios da metaheurísticas *Simulated Annealing* e Busca Tabu.

Bianchessi e Righini (2007) sugerem alguns algoritmos construtivos, heurísticas de busca local e um procedimento Busca Tabu associado a uma estrutura de vizinhança variável. Para tanto, se combinou os movimentos de permutação de nós (*node-exchange-based*) e permutação de arcos (*arc-exchange-based*) e variação do tamanho da lista tabu. Wassan *et al.* (2007) fazem uso de um procedimento construtivo baseado no algoritmo varredura (GILLET; MILLER, 1974) e propõem um algoritmo tabu reativo que utiliza as

seguintes estruturas de vizinhança: realocação de um cliente (*shift*), permuta de clientes de duas rotas distintas (*exchange*) e inversão do sentido da rota (*reverse*).

Dell’Amico *et al.* (2005) fazem uso de uma abordagem exata, por meio do método *branch-and-price*. O problema é tratado através de duas formas, sendo ambas sob a ótica da programação dinâmica. A mesma técnica é aplicada por Angelelli e Mansini (2003), onde os autores consideram o PRVCES com restrições de janelas de tempo.

2.2 METAHEURÍSTICAS

As metheurísticas consistem em métodos de resolução que regem a interação entre procedimentos de melhoria local e estratégias de alto nível para criar processos capazes de escapar de ótimos locais e executar uma busca robusta ao longo do espaço de busca. (GLOVER; KOCHENBERGER, 2002).

Dentre as metaheurísticas mais conhecidas, tem-se: Busca Tabu, *Simulated Annealing*, Algoritmos Genéticos, Colônia de Formigas, VNS e GRASP.

Blum e Roli (2003) resumem as propriedades fundamentais das metaheurísticas da seguinte forma:

- São estratégias que conduzem os processos de busca;
- exploraram, de forma eficiente, o espaço de busca de tal forma a encontrar soluções sub-ótimas;
- são técnicas que constituem algoritmos que vão desde um simples procedimento de busca local até complexos processos de “aprendizagem”;
- os algoritmos metaheurísticos são aproximações e, usualmente, são não-determinísticos;

- os algoritmos incorporam mecanismos que impedem o confinamento em áreas do espaço de busca;
- os conceitos básicos de busca permitem uma descrição em um nível abstrato;
- não são problemas de caráter específicos;
- podem fazer uso de uma área do conhecimento específica na forma de uma heurística controlada por uma estratégia de nível superior;
- as metaheurísticas atuais mais avançadas utilizam a experiência de busca (armazenada em algum tipo de memória) para conduzir a busca.

Segundo Glover e Kochenberger (2002), inúmeras ferramentas e mecanismos que emergiram a partir da criação de metaheurísticas têm demonstrado ser notadamente eficazes, tanto que, nos últimos anos, passaram a ser o centro das atenções como principal estratégia adotada para solucionar diversos tipos de problemas de otimização complexos, particularmente aqueles de natureza combinatória.

O sucesso relativo à aplicação de metaheurísticas pode ser observado na prática. Dullaert *et al.* (2007) elencam uma série de trabalhos relativos à aplicações de metaheurísticas para solucionar problemas reais.

Para um maior detalhamento sobre este assunto, consulte Glover e Kochenberger (2002), Blum e Roli (2003), Michalewicz e Fogel (2004) e Dréo *et al.* (2006).

2.3 ITERATED LOCAL SEARCH

Suponha que uma solução ótima local tenha sido encontrada por um algoritmo de busca local. Ao invés de reiniciar o mesmo procedimento a partir de uma solução completamente nova, a metaheurística ILS aplica repetidamente uma busca local às soluções

iniciais obtidas através de perturbações das soluções ótimas locais previamente visitadas (STÜTZLE, 1998).

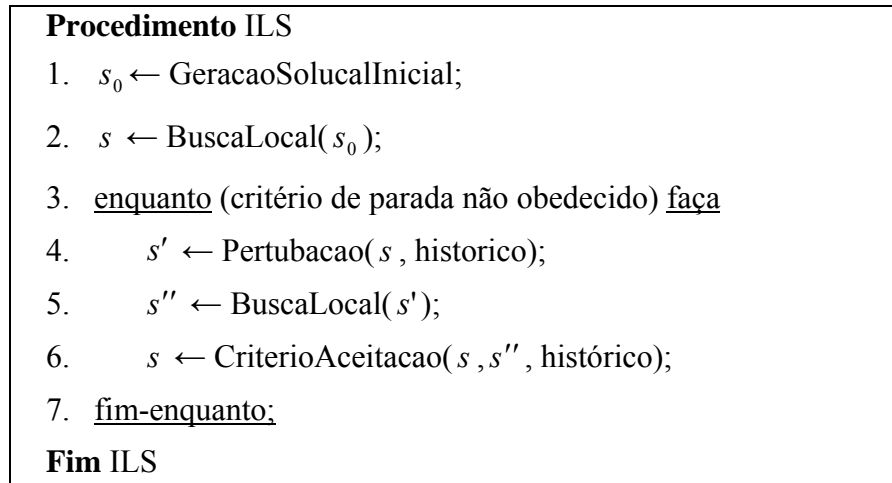


Figura 6 – Procedimento ILS
Fonte: Lourenço *et al.* (2002).

Segundo Lourenço *et al.* (2002), a idéia essencial da ILS está atribuída ao fato desta metaheurística realçar seu foco em um subespaço menor, ao invés de considerar o espaço completo de soluções, definido por aquelas que são ótimos locais de um dado procedimento de otimização.

Para se aplicar a metaheurística ILS, quatro procedimentos devem ser especificados, conforme ilustra a Figura 6. São eles: (i) “GeracaoSolucaoInicial”, no qual uma solução inicial é construída; (ii) “BuscaLocal”, que refina a solução inicialmente obtida; (iii) “Perturbacao”, onde um novo ponto de partida é gerado, por meio de uma perturbação de uma solução encontrada na busca local; (iv) “CritérioAceitacao”, que determina a partir de qual solução deve-se prosseguir a busca. Uma ilustração gráfica do procedimento ILS pode ser observada na Figura 7.

De acordo com Stützle (1998), a modificação efetuada na etapa de perturbação é empregada com o objetivo de escapar de uma solução ótima local ainda distante de um ótimo global. Frequentemente, o movimento é escolhido aleatoriamente dentro de uma vizinhança

de maior ordem em relação à utilizada no procedimento de busca local, ou um movimento que o algoritmo de busca local não possa reverter em apenas um passo.

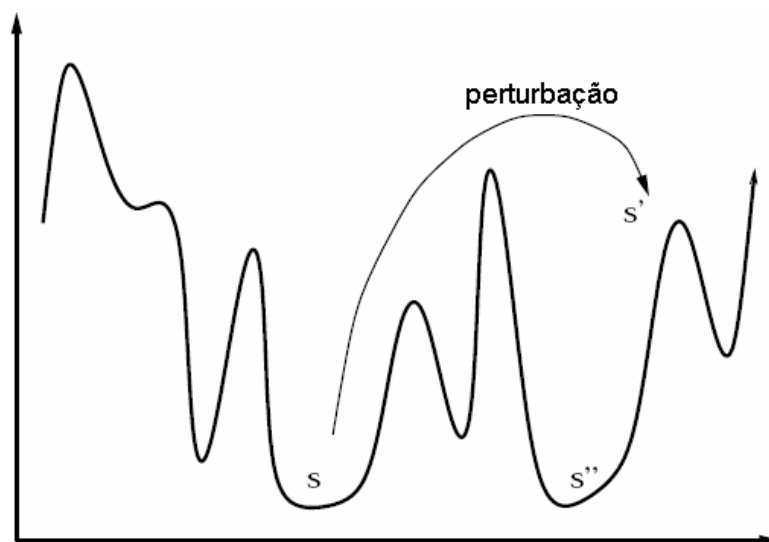


Figura 7 – Ilustração do procedimento ILS
Fonte: Adaptado de Lourenço *et al.* (2002).

A princípio, qualquer método de busca local pode ser utilizado, porém seu desempenho, no tocante a qualidade da solução e esforço computacional, depende fortemente do algoritmo escolhido. Geralmente uma heurística de descida (em problemas de minimização) é usada, contudo a adoção de um procedimento mais complexo também é admissível.

O critério de aceitação é utilizado para decidir qual a próxima solução a ser perturbada. A escolha deste critério é importante, haja vista que controla o balanço entre a intensificação e diversificação. O histórico de busca é usado para decidir se alguma solução ótima local, encontrada anteriormente, deve ser escolhida ou se a mesma exerce algum tipo de influência na decisão entre s e s'' .

O procedimento ILS deve conduzir a boas amostragens do espaço de busca contanto que as perturbações não sejam nem tão grandes nem demasiadamente pequenas. Se forem pequenas, poucas novas soluções serão exploradas, ao passo que se forem muito grandes, o algoritmo tenderá adotar pontos de partidas praticamente aleatórios.

2.4 MODELOS MATEMÁTICOS PROPOSTOS PARA O PRVCES

Algumas formulações matemáticas foram propostas para representar o PRVCES.

A formulação a seguir foi desenvolvida por Dethloff (2001).

Notação

Conjuntos

J Conjunto de todos os clientes

J_0 Conjunto de todos os nós, ou seja, clientes e o depósito, $J_0 = J \cup \{0\}$

V Conjunto de todos os veículos

Parâmetros

C Capacidade dos veículos

C_{ij} Distância entre os nós $i \in J_0, j \in J_0, i \neq j$; $C_{ii} := M, i \in J, C_{00} := 0$

D_j Quantidade a ser entregue ao cliente $j \in J$

n Número de nós, isto é, $n = |J_0|$

P_j Quantidade a ser coletada do cliente $j \in J$

M Número grande, ex: $M = \max \left\{ \sum_{j \in J} (D_j + P_j), \sum_{i \in J_0} \sum_{j \in J_0, j \neq i} C_{ij} \right\}$

Variáveis de decisão

l'_v Carga do veículo $v \in V$ quando sai do depósito

l_j Carga do veículo depois de prestado serviço ao cliente $j \in J$.

π_j Variável utilizada para proibir subrotas. Pode ser interpretada como sendo a posição do

nó $j \in J$ na rota.

x_{ijv} Variável binária que indica se o veículo $v \in V$ transita diretamente do nó $i \in J_0$ para o nó $j \in J_0$ ($x_{ijv} = 1$) ou não ($x_{ijv} = 0$)

Formulação Matemática 1

$$\text{Minimizar } \sum_{i \in J_0} \sum_{j \in J_0} \sum_{v \in V} C_{ij} x_{ijv} \quad (1)$$

Sujeito a:

$$\sum_{i \in J_0} \sum_{v \in V} x_{ijv} = 1, (j \in J) \quad (2)$$

$$\sum_{i \in J_0} x_{isv} = \sum_{j \in J_0} x_{sjv}, (s \in J, v \in V) \quad (3)$$

$$l'_v = \sum_{i \in J_0} \sum_{j \in J} D_j x_{ijv}, (v \in V) \quad (4)$$

$$l_j \geq l'_v - D_j + P_j - M(1 - x_{0jv}), (j \in J, v \in V) \quad (5)$$

$$l_j \geq l_i - D_j + P_j - M \left(1 - \sum_{v \in V} x_{ijv} \right), (i \in J, j \in J, j \neq i) \quad (6)$$

$$l'_v \leq C, (v \in V) \quad (7)$$

$$l_j \leq C, (j \in J) \quad (8)$$

$$\pi_j \geq \pi_i + 1 - n \left(1 - \sum_{v \in V} x_{ijv} \right), (i \in J, j \in J, j \neq i) \quad (9)$$

$$\pi_j \geq 0, (j \in J) \quad (10)$$

$$\pi_{ijv} \in \{0,1\}, (i \in J_0, j \in J, v \in V) \quad (11)$$

A função objetivo (1) indica que a distância total a ser percorrida pelos veículos deve ser minimizada. As restrições (2) garantem que o serviço deve ser prestado somente uma vez por cliente. As restrições (3) indicam que o veículo que chega e sai de cada cliente é o mesmo. As restrições (4) correspondem às cargas iniciais de cada veículo. As restrições (5) indicam a carga de cada veículo depois de ter visitado o primeiro cliente. As restrições (6) são relativas às cargas dos veículos em rota. As restrições (7) e (8) estão relacionadas à capacidade de cada veículo depois de ter visitado o primeiro cliente e em rota. As restrições (9) são responsáveis pela eliminação das subrotas. Por fim, (10) e (11) representam as variáveis do problema.

Montané e Galvão (2006) construíram um modelo baseado no trabalho de Mosheiov (1998), no qual os autores levam em consideração a restrição de máxima distância a ser percorrida por um veículo. A modelagem encontra-se descrita a seguir:

Notação

Conjuntos

V Conjunto de clientes

V_0 Conjunto de clientes incluindo o depósito (cliente 0): $V_0 = V \cup \{0\}$

Parâmetros

n Número total de clientes: $n = |V|$

c_{ij} Distância entre os clientes i e j

p_j Demanda por coleta do cliente j , $j = 1, \dots, n$

d_j Demanda por entrega do cliente j , $j = 1, \dots, n$

Q Capacidade do veículo

MD Máxima distância permitida para qualquer rota k

\bar{k} Máximo número de veículos

Variáveis de decisão

$$x_{ij}^k = \begin{cases} 1, & \text{se o arco}(i, j) \text{ pertencer a rota operada pelo veículo } k \\ 0, & \text{caso contrário} \end{cases}$$

y_{ij} Demanda coletada nos clientes roteados até o nó i (incluindo o nó i) e transportados no arco (i, j)

z_{ij} Demanda a ser entregue para clientes roteados após o nó i e transportados no arco (i, j)

Formulação Matemática 2

$$\text{Minimizar } \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k \quad (12)$$

Sujeito a:

$$\sum_{i=0}^n \sum_{k=1}^{\bar{k}} x_{ij}^k = 1, \quad (j = 1, \dots, n) \quad (13)$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{i=0}^n x_{ji}^k = 0, \quad (j = 0, \dots, n), \quad (k = 0, \dots, \bar{k}) \quad (14)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \quad (k = 0, \dots, \bar{k}) \quad (15)$$

$$\sum_{i=0}^n \sum_{j=1}^n c_{ij} x_{ij}^k \leq MD, \quad (k = 0, \dots, \bar{k}) \quad (16)$$

$$\sum_{i=0}^n y_{ji} - \sum_{i=0}^n y_{ij} = p_j, \quad \forall j \neq 0 \quad (17)$$

$$\sum_{i=0}^n z_{ij} - \sum_{j=0}^n z_{ji} = d_j, \quad \forall j \neq 0 \quad (18)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{\bar{k}} x_{ij}^k, \quad (i = 1, \dots, n), \quad (j = 1, \dots, n) \quad (19)$$

$$x_{ij}^k \in \{0, 1\}, \quad (i = 1, \dots, n), \quad (j = 1, \dots, n) \quad (20)$$

$$y_{ij} \geq 0, \quad (i = 1, \dots, n), \quad (j = 1, \dots, n) \quad (21)$$

$$z_{ij} \geq 0, \quad (i = 1, \dots, n), \quad (j = 1, \dots, n) \quad (22)$$

A função objetivo (12) representa a função objetivo que visa minimizar a soma das distâncias percorridas pelos veículos ao longo das rotas. As restrições (13) garantem que cada cliente é visitado uma vez. As restrições (14) indicam que o mesmo veículo deve chegar e partir de um mesmo cliente. As restrições (15) ilustram que podem ser utilizados até \bar{k} veículos enquanto a expressão (16) é relativa à distância máxima permitida em cada rota. As restrições (17) e (18) são as equações de fluxo referentes à coleta e entrega respectivamente. As restrições (19) estabelecem que as coletas e entregas sejam efetuadas utilizando-se os arcos inclusos na solução, respeitando a capacidade do veículo. Finalmente, as restrições (20), (21) e (22) definem a natureza das variáveis de decisão.

3. FORMULAÇÃO MATEMÁTICA PROPOSTA

A formulação matemática proposta neste trabalho foi inspirada no modelo desenvolvido por Baldacci *et al.* (2004) para a versão clássica do PRV, onde os autores apresentam uma formulação em fluxo com duas comodidades (*two-commodity flow formulation*).

Considere inicialmente um grafo $G(V, E)$ completo e não-direcionado, onde o vértice $i = 0$ representa o depósito e os demais n vértices, os clientes. Para melhor caracterização do modelo, faz-se necessário acrescentar um vértice $n + 1$ que representa uma cópia do depósito. No modelo aqui proposto, três problemas são considerados, a saber: (i) PRV somente com entrega; (ii) PRV somente com coleta; (iii) PRV com coleta e entrega simultânea. Nos três casos, dois fluxos são considerados, sendo um partindo do vértice 0 em direção ao vértice $n + 1$ e vice-versa. O fluxo partindo do vértice $n + 1$ percorre exatamente o caminho inverso.

Os problemas (i) e (ii) são independentes entre si e, na verdade, correspondem a problemas auxiliares utilizados para garantir a viabilidade relativa à carga dos veículos no problema (iii). Ressalta-se que toda solução viável de (iii) é necessariamente solução viável de (i) e (ii), porém a recíproca não é verdadeira. Este fato é simples de verificar, pois a carga dos veículos no problema (iii) será sempre maior ou igual a dos problemas (i) e (ii), considerando uma ordem lexicográfica das arestas do grafo.

A Figura 8 ilustra um exemplo esquemático da abordagem do modelo proposto. Observa-se que nesta situação que os três problemas apresentam a mesma solução, porém os valores das cargas variam de acordo com a situação. No exemplo, dois veículos de capacidade $Q = 20$ são considerados e as rotas são: $0-1-2-3-4-N+1$ (r_1) e $0-5-6-7-8-N+1$ (r_2).

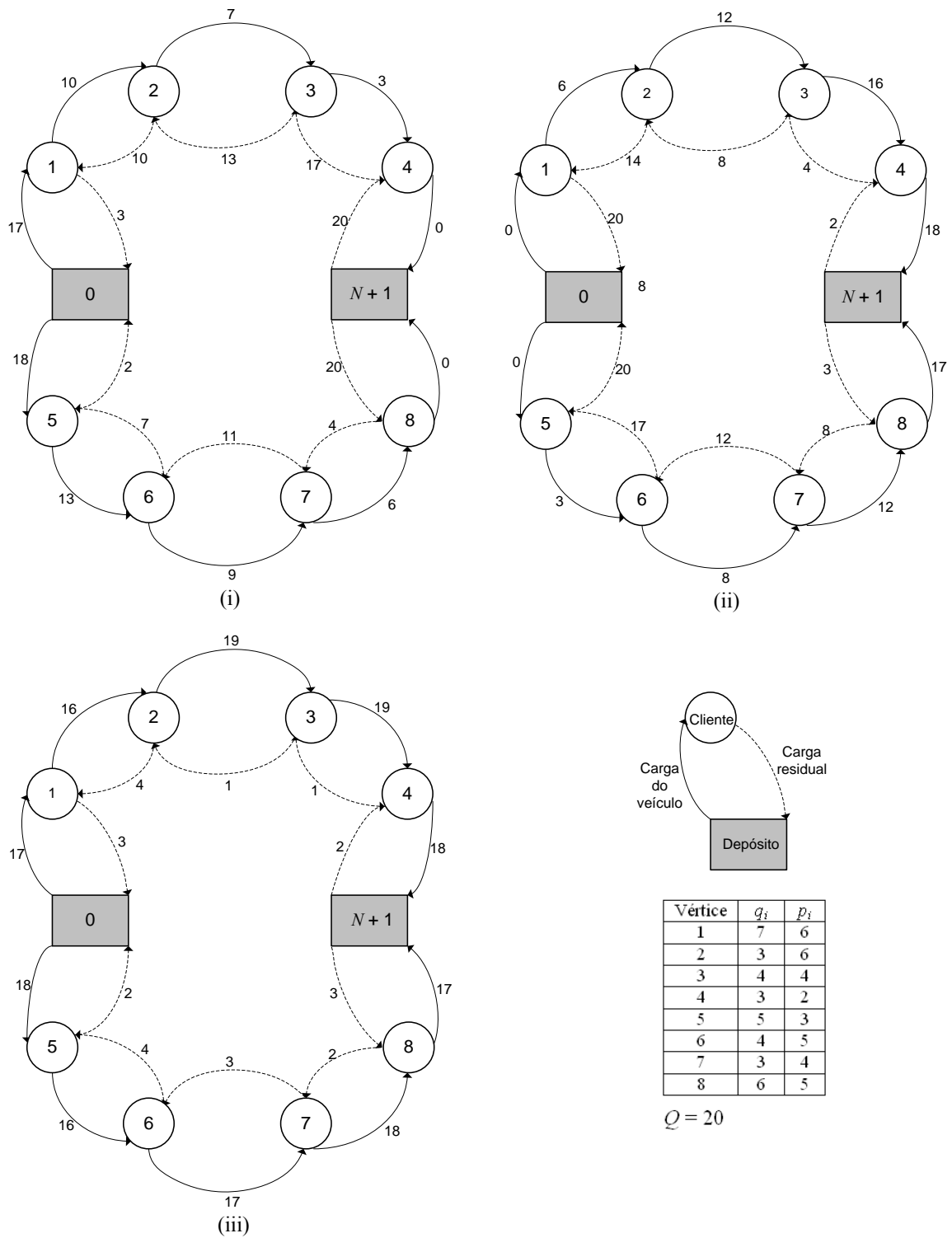


Figura 8 – Exemplo esquemático da formulação proposta

Ainda na Figura 8, observa-se que a cargas dos veículos ao saírem do depósito nos problemas (i) e (iii) são iguais e correspondem a $\sum_{i=1}^{i=4} q_i = 7 + 3 + 4 + 3 = 17$ na rota r_1 e

$\sum_{i=5}^{i=8} q_i = 5 + 4 + 3 + 6 = 18$ na rota $r2$. Nos problemas (ii) e (iii) as cargas dos veículos ao retornarem ao depósito são iguais e correspondem a $\sum_{i=1}^{i=4} p_i = 6 + 6 + 4 + 2 = 18$ na rota $r1$ e $\sum_{i=5}^{i=8} p_i = 3 + 5 + 4 + 5 = 17$ na rota $r2$.

Notação

Conjuntos

V	Conjunto de todos os clientes e o depósito
$\bar{V} = V \cup \{n+1\}$	Conjunto V agregado de uma cópia do depósito
$V' = V \setminus \{0, n+1\}$	Conjunto de todos os clientes
\bar{E}	Conjunto das arestas (i, j) , onde $i \in \bar{V}$ e $j \in \bar{V}$

Parâmetros

v	Número de rotas
q_i	Demanda por entrega do cliente $i \in V'$
p_i	Demanda por coleta do cliente $i \in V'$
c_{ij}	Custo entre os clientes $(i, j) \in \bar{E}$
Q	Capacidade do veículo

Variáveis de decisão

$$x_{ij} = \begin{cases} 1, & \text{se a aresta } (i, j) \text{ estiver inclusa na solução} \\ 0, & \text{caso contrário} \end{cases}$$

$loadD_{ij}$ = carga do veículo, referente a entrega, na aresta $(i, j) \in \bar{E}$

$loadP_{ij}$ = carga do veículo, referente a coleta, na aresta $(i, j) \in \bar{E}$

$load_{ij}$ = carga do veículo, referente a coleta e entrega simultânea, na aresta $(i, j) \in \bar{E}$

Formulação Matemática 3

$$\text{Min} \sum_{i \in \bar{V}} \sum_{j \in \bar{V}} c_{ij} x_{ij} \quad (23)$$

Sujeito a:

$$\sum_{j \in V'} (loadD_{ji} - loadD_{ij}) = 2q_i \quad \forall i \in V' \quad (24)$$

$$\sum_{j \in V'} (loadD_{0j}) = \sum_{i \in V'} q_i \quad (25)$$

$$\sum_{j \in V'} (loadD_{j0}) = v \cdot Q - \sum_{i \in V'} q_i \quad (26)$$

$$\sum_{j \in V'} (loadP_{ij} - loadP_{ji}) = 2p_i \quad \forall i \in V' \quad (27)$$

$$\sum_{j \in V'} (loadD_{jn+1}) = \sum_{i \in V'} p_i \quad (28)$$

$$\sum_{j \in V'} (loadD_{n+1j}) = v \cdot Q - \sum_{i \in V'} p_i \quad (29)$$

$$\sum_{j \in V'} (load_{ji} - load_{ij}) = 2(q_i - p_i) \quad \forall i \in V' \quad (30)$$

$$load_{0j} = loadD_{0j} \quad \forall j \in V' \quad (31)$$

$$load_{j0} = loadD_{j0} \quad \forall j \in V' \quad (32)$$

$$load_{jn+1} = loadP_{jn+1} \quad \forall j \in V' \quad (33)$$

$$load_{n+1j} = loadP_{n+1j} \quad \forall j \in V' \quad (34)$$

$$loadD_{ij} + loadD_{ji} = Q \cdot x_{ij} \quad \forall (i, j) \in \bar{E} \quad (35)$$

$$loadP_{ij} + loadP_{ji} = Q \cdot x_{ij} \quad \forall (i, j) \in \bar{E} \quad (36)$$

$$load_{ij} + load_{ji} = Q \cdot x_{ij} \quad \forall (i, j) \in \bar{E} \quad (37)$$

$$\sum_{\substack{j \in V' \\ i < k}} x_{ik} + \sum_{\substack{j \in V' \\ j > k}} x_{kj} = 2 \quad \forall k \in V' \quad (38)$$

$$\sum_{j \in V'} x_{0j} \leq v \quad (39)$$

$$\sum_{j \in V'} x_{jn+1} \leq v \quad (40)$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in \bar{E} \quad (41)$$

$$loadD_{ij} \geq 0, \quad \forall (i,j) \in \bar{E} \quad (42)$$

$$loadP_{ij} \geq 0, \quad \forall (i,j) \in \bar{E} \quad (43)$$

$$load_{ij} \geq 0, \quad \forall (i,j) \in \bar{E} \quad (44)$$

A função objetivo (23) minimiza o somatório das distâncias a serem percorridas. As restrições (24)-(26) estão relacionadas ao problema (i). A restrição (24) garante que todas as demandas por entrega sejam atendidas. A restrição (25) faz com que a soma da carga que sai do depósito seja igual à soma das demandas de todos os clientes. A restrição (26) assegura que a soma das cargas dos veículos que chegam no vértice 0 seja igual à soma da carga residual dos veículos ao sair do depósito. As restrições (27)-(29) são relativas ao problema (ii) e têm significado análogo às restrições (24)-(26). As restrições (30)-(34) correspondem ao problema (iii). A restrição (30) garante que as demandas por coleta e entrega sejam atendidas simultaneamente. As restrições (31) e (32) indicam que as cargas que saem e chegam do vértice 0 devem ser respectivamente iguais às do problema (i). As restrições (33) e (34) asseguram que as cargas que entram e saem do vértice $n + 1$ sejam respectivamente iguais às do problema (ii). As restrições (35), (36) e (37) apontam que a soma das cargas que entram e saem de cada cliente, nos problemas (i), (ii) e (iii) respectivamente, deve ser igual à capacidade do veículo. A restrição (38) obriga que toda solução viável contenha duas arestas incidentes em cada cliente. As restrições (39) e (40) correspondem ao limite superior de veículos. As restrições (41)-(44) estão relacionadas à natureza das variáveis de decisão.

4 PROCEDIMENTOS METODOLÓGICOS

O presente capítulo apresenta os métodos e procedimentos utilizados nesta dissertação, descrevendo respectivamente: o algoritmo proposto; o conjunto de instâncias utilizadas; e os parâmetros adotados para o algoritmo proposto.

4.1 ALGORITMO PROPOSTO

O algoritmo desenvolvido para o PRVCES funciona da maneira que se segue. O procedimento é executado $MaxIter$ vezes, onde a solução inicial é gerada por uma heurística gulosa e, em seguida, é refinada por um procedimento baseado na metaheurística ILS que efetua a busca local através de um procedimento baseado no algoritmo VND. O pseudocódigo encontra-se ilustrado na Figura 9, onde s^* representa a melhor solução, v corresponde ao número de veículos (ou rotas) inicialmente consideradas e γ um parâmetro que é tratado em detalhes na Subseção 4.1.1.

Procedimento ILS-VND($MaxIter$, $MaxIterILS$, semente, γ , v , $N(\cdot)$, $f(\cdot)$, r)

1. CarregarDados();
2. $f(s^*) \leftarrow \infty$;
3. para $k = 1, \dots, MaxIter$ faça
4. $s \leftarrow$ GeracaoSolucaoInicial(γ , semente, v);
5. $s' \leftarrow s$;
6. $iterILS \leftarrow 0$;
7. enquanto $iterILS < MaxIterILS$ faça
8. $s \leftarrow$ VND($N(\cdot)$, $f(\cdot)$, r , s); { $r = n^o$ de vizinhanças}
9. se $f(s) < f(s')$
10. $s' \leftarrow s$;
11. $f(s') \leftarrow f(s)$;
12. $iterILS \leftarrow 0$;
13. fim se

```

14.      $s \leftarrow \text{Perturba}(s')$ ;
15.      $iterILS \leftarrow iterILS + 1$ ;
16.     fim enquanto;
17.     se  $f(s') < f(s^*)$ 
18.          $s^* \leftarrow s'$ ;
19.          $f(s^*) \leftarrow f(s')$ ;
20.     fim se;
21. fim;
22. retornar  $s^*$ ;
Fim ILS-VND

```

Figura 9 – Pseudocódigo do algoritmo ILS-VND

4.1.1 Procedimento Construtivo

O método utilizado para construir uma solução inicial viável possui uma estratégia gulosa, podendo ser classificado, à luz de Bodin e Golden (1981), como sendo do tipo “inserção”. Trata-se de uma adaptação da heurística de inserção proposta por Dethloff (2001), mas sem considerar a capacidade residual do veículo. O pseudocódigo deste procedimento pode ser visualizado na Figura 10.

Inicialmente, todas as rotas são preenchidas com um cliente e , selecionado aleatoriamente da lista de candidatos (LC). Posteriormente, dá-se início à etapa de construção propriamente dita. Os clientes pertencentes à LC são avaliados segundo o critério de inserção expresso pela equação (45):

$$g(e^v) = (C_{ik} + C_{kj} - C_{ij}) - \gamma(C_{0k} + C_{k0}) \quad (45)$$

A primeira parcela da expressão (45) está relacionada ao critério da “inserção viável mais barata”, que consiste em uma abordagem gulosa, que leva em consideração o

menor custo adicional no tocante à inserção do cliente k entre os clientes i e j da rota v . Naturalmente, apenas as inserções viáveis são admitidas.

A segunda parcela corresponde a uma bonificação atribuída a um cliente que esteja em uma localização desprivilegiada. A idéia é favorecer a inserção deste cliente de maneira que o mesmo não seja tardiamente adicionado à rota. A distância de ida e volta do cliente k em relação ao depósito é multiplicada por um fator $\gamma \in [0,1]$.

O cliente e associado à g^{min} é, então, adicionado à solução s . O procedimento construtivo termina quando todos os clientes tiverem sido adicionados à solução s .

Procedimento GeracaoSolucaoInicial (semente, γ , v)

1. $s \leftarrow \emptyset$;
2. Inicializar Lista de Candidatos (LC);
3. Seja $s = \{s^1, s^2, \dots, s^v\}$ o conjunto solução formado por v rotas vazias;
4. $t \leftarrow 1$;
5. enquanto $t \leq v$
6. $s^t \leftarrow e \in LC$ selecionado aleatoriamente;
7. Atualizar LC;
8. $t \leftarrow t + 1$;
9. fim enquanto
10. enquanto $LC \neq \emptyset$ faça
11. Avaliar o valor de cada custo $g(e)$ para $e \in LC$;
12. $g^{min} \leftarrow \min \{g(e) \mid e \in LC\}$;
13. $n \leftarrow$ cliente e associado a g^{min} ;
14. $s \leftarrow s \cup \{n\}$;
15. Atualizar a LC;
16. fim enquanto;
17. retornar s ;

Fim GeracaoSolucaoInicial

Figura 10 – Pseudocódigo do procedimento construtivo

4.1.2 Procedimento de Busca Local

Conforme mencionado anteriormente, a fase de busca local é efetuada por um procedimento baseado na metaheurística VND. Proposto por Mladenović e Hansen (1997), o método de descida em vizinhança variável é essencialmente caracterizado por realizar mudanças sistemáticas de estruturas de vizinhança pertencentes a um conjunto $N = \{N^{(1)}, N^{(2)}, N^{(3)}, \dots, N^{(r)}\}$, de forma determinística. A vizinhança $k' = k + 1 \in N$ é acionada apenas se não tiver havido melhora nas k vizinhanças pesquisadas anteriormente no que se refere ao melhoramento da solução corrente.

No algoritmo proposto, utilizou-se um conjunto de dez estruturas de vizinhanças. A grande maioria destas foi empregada por Chen (2006), à exceção do movimento *Reverse*, que foi utilizado por Nagy e Salhi (2005) e Wassan *et al.* (2006). Apenas os movimentos viáveis, isto é, aqueles que não violam as restrições de carga máxima ou, quando for o caso, de tempo máximo exigido por rota, são permitidos. Desta forma, toda vez em que ocorrer, devido a algum movimento, melhora na função objetivo, deve-se verificar se houve violação de alguma restrição. Das dez vizinhanças aqui adotadas de forma exaustiva, seis executam movimentos entre rotas e quatro dentro das rotas. O conjunto N de vizinhanças (entre rotas) encontra-se descrito a seguir. As outras quatro são apresentadas ao final desta subseção.

Shift(1,0) – $N^{(1)}$ – Um cliente c é transferido de uma rota $r1$ para outra rota $r2$. A carga do veículo é verificada da seguinte forma. Todos os clientes que se encontram antes da posição em que houve a inserção em $r2$ têm a carga somada de q_c (valor da demanda por entrega de c), enquanto aqueles localizados após a posição de inserção têm a carga acrescida de p_c (valor da demanda por coleta de c). Vale salientar que alguns artifícios para evitar que o custo de movimentos inviáveis seja calculado de forma desnecessária podem ser empregados. Por

exemplo, antes de testar a inserção de c em uma determinada rota, é realizada uma checagem para averiguar se a carga do veículo ao sair ou voltar para o depósito extrapolou o limite permitido. Em caso afirmativo, não há necessidade de testar as demais possibilidades de inserção de c nesta rota, pois a carga sempre será violada. A Figura 11 ilustra um exemplo onde o cliente “7” é transferido de uma rota para outra.

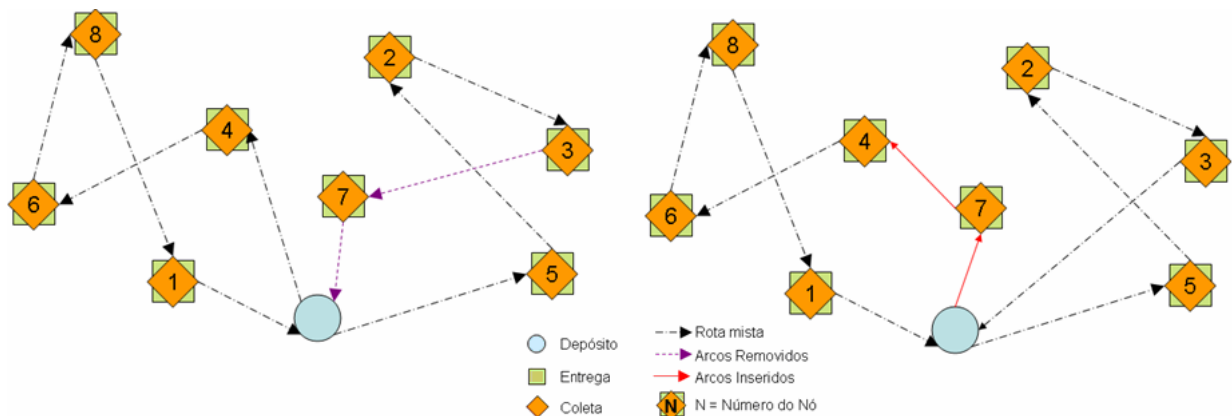


Figura 11 – Movimento $Shift(1,0)$

Crossover – $N^{(2)}$ – Um aresta é removida entre os clientes adjacentes c_1 e c_2 pertencentes a uma rota r_1 e outra é removida entre os clientes adjacentes c_3 e c_4 pertencentes a uma rota r_2 . Em seguida, uma aresta é inserida entre c_1 e c_4 e outra é inserida entre c_3 e c_2 . No tocante à verificação da carga, o *Crossover* é mais caro computacionalmente quando comparado ao *Shift(1,0)*. Inicialmente, calcula-se a carga inicial (l_0) do depósito e a carga final (l_f), isto é,

$l_0 = \sum_{i \in V - V_0} q_i$ e $l_f = \sum_{i \in V' - V_0} p_i$, respectivamente. Caso os valores de l_0 e l_f não ultrapassem

o limite permitido, calcula-se o restante das cargas l_i através da seguinte expressão:

$l_i = l_{i-1} + p_i - q_i$. Assim, se l_i extrapolar a máxima capacidade suportada pelo veículo, o movimento é inviável. Um exemplo pode ser observado na Figura 12, em que uma aresta de

cada rota é removida e duas arestas que ligam ambas as rotas são inseridas.

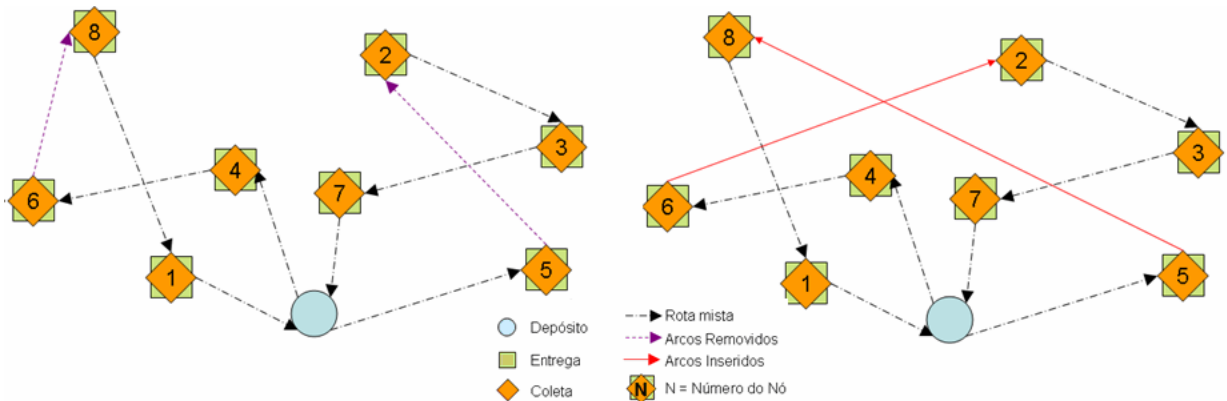


Figura 12 – Movimento *Crossover*

Swap(1,1) – $N^{(3)}$ – Permutação entre um cliente c_1 de uma rota r_1 por um cliente c_2 de uma rota r_2 . A carga dos veículos de ambas as rotas é verificada da mesma forma. Por exemplo, no caso de r_2 , todos os clientes situados antes da posição em que c_2 se encontrava (agora ocupada por c_1) têm o valor da carga somada de q_{c_1} e subtraída de q_{c_2} , ao passo que a carga de todos os clientes localizados após c_1 sofrem acréscimo de p_{c_1} e decréscimo de p_{c_2} . A Figura 13 ilustra uma situação em que os clientes “4” e “7” trocam de posição.

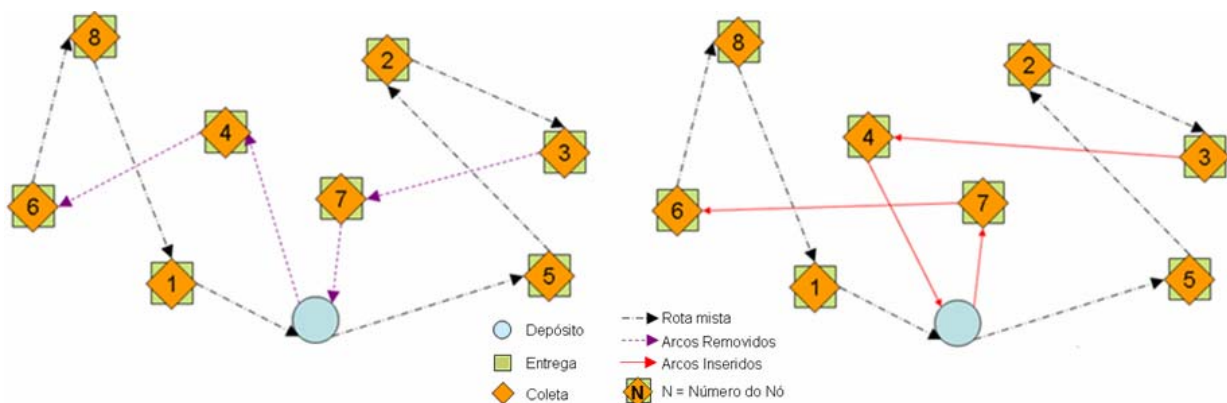


Figura 13 – Movimento *Swap(1,1)*

Shift(2,0) – $N^{(4)}$ – Dois clientes consecutivos, c_1 e c_2 , são transferidos de uma rota para outra. A carga do veículo é testada de forma análoga à vizinhança *Shift(1,0)*.

Swap(2,1) – $N^{(5)}$ – Permutação de dois clientes consecutivos, $c1$ e $c2$, de uma rota $r1$ por um cliente $c3$ de uma rota $r2$. A carga é verificada através da extensão da abordagem utilizada nas vizinhanças *Shift*(1,0) e *Swap*(1,1).

Swap(2,2) – $N^{(6)}$ – Permutação de dois clientes consecutivos, $c1$ e $c2$, de uma rota $r1$ por dois consecutivos, $c3$ e $c4$, de uma rota $r2$. A verificação da carga é efetuada de maneira análoga a vizinhança *Swap*(1,1).

<p>Procedimento VND($N(\cdot), f(\cdot), r, s$)</p> <ol style="list-style-type: none"> 1. Seja r o número de estruturas diferentes de vizinhança; 2. $k \leftarrow 1$; {Tipo de estrutura de vizinhança corrente} 3. <u>enquanto</u> ($k \leq r$) <u>faça</u> 4. Encontre o melhor vizinho s' de $s \in N^k(s)$; 5. <u>se</u> $f(s') < f(s)$ 6. <u>então</u> 7. $s \leftarrow s'$; 8. $f(s) \leftarrow f(s')$; 9. $k \leftarrow 1$; 10. {intensificação nas rotas alteradas} 11. $s' \leftarrow Or-opt(s)$; 12. $s'' \leftarrow 2-opt(s')$; 13. $s''' \leftarrow Exchange(s'')$; 14. $s'''' \leftarrow Reverse(s''')$; 15. <u>se</u> $f(s'''') < f(s)$ 16. <u>então</u> 17. $s \leftarrow s''''$; 18. $f(s) \leftarrow f(s'''')$; 19. <u>fim se</u>; 20. <u>senão</u> 21. $k \leftarrow k + 1$; 22. <u>fim se</u>; 23. <u>fim enquanto</u>; 24. <u>retorne</u> s; <p>Fim VND</p>
--

Figura 14 – Pseudocódigo do algoritmo VND

Ressalta-se que a complexidade de todas as vizinhanças supracitadas é da ordem de $O(n^2)$. A Figura 14 ilustra o pseudocódigo do procedimento VND.

Caso haja alguma melhora na solução corrente, realiza-se uma pesquisa com o objetivo de melhorar a qualidade da solução das rotas que contribuíram para a redução da função objetivo, ou seja, aquelas que participaram do último movimento de melhora. Para tanto, quatro movimentos são efetuados de forma exaustiva.

Or-opt – Proposto por Or (1976), onde um, dois ou três clientes consecutivos são removidos e posteriormente inseridos em outra posição. A complexidade deste movimento é $O(n^2)$.

2-opt – Um par de arcos é removido e outro par é inserido. Neste movimento, a complexidade corresponde a $O(n^2)$.

Exchange – Permutação entre dois clientes. A complexidade desta vizinhança é $O(n^2)$.

Reverse – Inverte o sentido da rota. O movimento é aceito se houver diminuição no valor da carga máxima. Este apresenta complexidade $O(n)$.

4.1.3 Mecanismos de Perturbação

Um conjunto de três mecanismos de perturbação, $P = \{P^{(1)}, P^{(2)}, P^{(3)}\}$, foram adotados nesta pesquisa. A cada chamada da função “Perturba”, um dos movimentos descritos a seguir é selecionado aleatoriamente. Admitem-se, apenas, aqueles que mantenham a viabilidade da solução perturbada.

Ejection Chain – $P^{(1)}$ – Inicialmente proposto por Rego e Roucairol (1996) para a versão clássica do PRV, este movimento foi aqui empregado como um mecanismo de perturbação e funciona da forma que se segue. Um cliente da rota 1 é transferido para a rota 2, em seguida um cliente da rota 2 é transferido para a rota 3 e assim sucessivamente. O movimento termina

quando um cliente da última rota é transferido para a rota 1. Os clientes são selecionados de maneira aleatória. A Figura 15 ilustra um exemplo envolvendo quatro rotas.

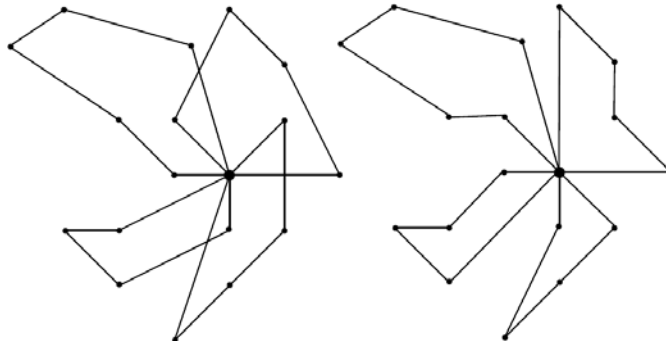


Figura 15 – Movimento *Ejection Chain*
Fonte: Dréo *et al.* (2006)

Double Swap(1,1) – $P^{(2)}$ – Dois movimentos *Swap(1,1)* são efetuados em seqüência de forma aleatória.

Double Bridge – $P^{(3)}$ – Introduzido por Martin *et al.* (1991), esta perturbação foi originalmente desenvolvida para o PCV e consiste na eliminação de quatro arestas e inserção de outras quatro, conforme ilustra a Figura 16. No caso em questão, o movimento é aplicado de forma aleatória em todas as rotas. Para um número elevado destas, a perturbação é aplicada em somente em algumas, selecionadas arbitrariamente. Lourenço *et al.* (2002) afirma que diversas aplicações da metaheurística ILS ao PCV fizeram uso deste tipo de perturbação e esta tem se mostrado eficiente para instâncias de diferentes dimensões.

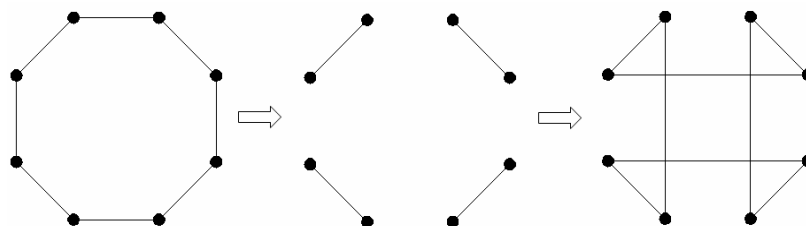


Figura 16 – Movimento *Double-Bridge*

É importante frisar que nem todos os mecanismos foram necessariamente empregados nos diferentes conjuntos de instâncias utilizados para testar o algoritmo proposto. Estes foram escolhidos após a realização de experimentos preliminares que tiveram o intuito de averiguar quais perturbações se mostraram mais adequadas aos respectivos problemas-testes.

4.2 CONJUNTO DE INSTÂNCIAS

Os conjuntos de instâncias utilizados foram formulados por Min (1989), Salhi e Nagy (1999), Dethloff (2001) e Montané e Galvão (2006). A seguir é detalhado como estes problemas foram gerados.

4.2.1 Min (1989)

Como já exposto anteriormente na Subseção 2.1.4, Min (1989), trata-se de um problema real realizado no sistema de distribuição de uma biblioteca pública. Este consiste em um total de 22 clientes, cada qual com determinadas demandas por coleta e entrega e 2 veículos com capacidades idênticas.

4.2.2 Salhi e Nagy (1999)

As 28 instâncias formuladas por Salhi e Nagy (1999) são derivadas dos problemas-testes propostos por Christofides *et al.* (1979). O número de clientes varia entre 50 e 199. As demandas por coleta foram geradas da seguinte maneira. Para cada cliente a calcula-se um raio r_a como sendo $\min((x_a/y_a), (y_a/x_a))$. Assim, a nova demanda relativa à

entrega do cliente a é determinada por $q_a = r_a \cdot t_a$ e a demanda relacionada à coleta é representada por $p_a = (1 - r_a) \cdot t_a$, onde t_a corresponde à demanda por entrega do problema original.

Outro conjunto de demandas por entrega e coleta foi gerado através da troca entre ambas as demandas de cada cliente. Os resultados médios foram computados e referidos como sendo X e Y para os dois conjuntos respectivamente.

Importa destacar que dentre os 28 problemas-testes, 14 consideram uma restrição adicional de tempo máximo permitido por rota. Isso pode ser tratado de forma similar à restrição de distância máxima. O tempo total consumido por um dado veículo é dado pelo tempo total de percurso mais o tempo despendido para atender os clientes. A distância total é obtida multiplicando-se a velocidade média do veículo pelo tempo total de percurso. Assim, o tempo total de percurso e a distância total percorrida são iguais numericamente quando se considera a velocidade média como sendo unitária. Desta forma, o tempo total pode ser calculado através da soma entre a distância total mais o tempo de serviço (*drop time*). Quando o *drop time* é constante para todos os clientes, o tempo total de serviço pode ser calculado multiplicando-se o número de clientes pelo *drop time*. Todas as instâncias apresentam *drop time* constante e admitiu-se uma velocidade média unitária dos veículos.

4.2.3 Dethloff (2001)

As instâncias propostas por Dethloff (2001) consistem em um total de 40 problemas-testes gerados aleatoriamente, cujo número de clientes corresponde a 50 em todos os casos. Em metade das instâncias (SCA3 e SCA8), as coordenadas estão uniformemente distribuídas ao longo do intervalo $[0, 100]$, que corresponde a um quadrado. Na outra metade (CON3 e CON8), as coordenadas estão mais concentradas e encontram-se uniformemente

distribuídas no intervalo $[100/3, 200/3]$, levando, assim, a uma configuração mais urbana em $1/9$ da área. As distâncias foram calculadas utilizando a métrica euclidiana em ambos os casos.

As demandas por entrega D_j estão distribuídas de maneira uniforme no intervalo $[0,100]$. Já as demandas por coleta P_j são computadas utilizando-se um número aleatório r_i , que se encontra distribuído uniformemente no intervalo $[0,1]$, onde $P_j = (0,5 + r_i)D_j$.

A capacidade dos veículos também é levada em consideração. As instâncias geradas possuem diferentes capacidades por meio da escolha do número mínimo de veículos necessários (μ). Assim, a capacidade correspondente é dada por $C = \sum_{s \in J} D_s / \mu$.

Vale salientar que a notação considerada nos parágrafos anteriores é a mesma proposta no modelo de Dethloff (2001), exposto na Seção 2.3.

4.2.4 Montané e Galvão (2006)

As instâncias geradas por Montané e Galvão para o PRVCES foram adaptadas dos problemas-testes de Solomon (1987) e de uma versão estendida destes (GEHRING; HOMBERGER, 1999) respectivamente. As coordenadas e demandas por entrega foram mantidas e as demandas por coleta foram geradas aleatoriamente a partir do mesmo intervalo utilizado por Solomon (1987) e Gehring e Homberger (1999).

4.3 PARÂMETROS DO ALGORITMO

A heurística ILS-VND proposta foi executada 30 vezes, em cada problema-teste, para diferentes valores de $\gamma \in [0,1]$. Os parâmetros foram calibrados empiricamente após

diversos testes com diferentes valores. O número de iterações (*MaxIter*) efetuadas para todos os cenários foi de 15, enquanto o número máximo de perturbações permitidas sem que houvesse melhora da solução corrente (*MaxIterILS*) foi de 30. Nas instâncias de Dethloff (2001), Montané e Galvão (2006) e Min (1989), variou-se o valor de γ ao longo do intervalo $[0,0; 1,0]$ com incrementos de 0,10. Nas instâncias de Salhi e Nagy (1999), o intervalo adotado foi $[0,00; 0,50]$, com incrementos de 0,05.

Visando a mostrar a influência do parâmetro γ nos resultados alcançados em cada conjunto de instâncias (I), duas abordagens foram empregadas. A primeira buscou determinar o desvio médio de cada γ , para um dado conjunto de problemas-testes, entre a solução média (S_{media_i}) e a melhor solução encontrada pelo algoritmo (S_{best_i}), em cada problema i , considerando todos os valores de γ . Gráficos de barra foram construídos ilustrando os valores dos desvios médios percentuais de cada valor de γ . Estes foram calculados da seguinte maneira:

$$\text{desvio médio} = \frac{\sum_{i \in I} \left(\frac{S_{media_i} - S_{best_i}}{S_{best_i}} \right) \times 100}{|I|}, \quad \forall \gamma \quad (46)$$

A segunda abordagem procurou verificar quais valores de γ obtiveram melhor desempenho médio para determinado grupo de problemas-testes. Esta análise se deu em duas fases. Inicialmente, efetuou-se um teste de comparação entre médias (ANOVA *one-way*) para cada instância com o intuito de averiguar se o parâmetro γ influenciava o resultado. Em caso positivo, selecionou-se o maior grupo possível contendo as menores médias para os quais o teste ANOVA mostrava que não havia diferenças. Posteriormente, verificou-se, através da interseção dos grupos de valores de γ encontrados na fase anterior, quais os valores de γ que levaram, em média, a soluções de melhor qualidade.

5 RESULTADOS COMPUTACIONAIS

O algoritmo proposto foi implementado na linguagem de programação C++ utilizando o compilador Borland C++ Builder 6.0 e foi executado em um PC Intel Core 2 Duo 2.13 GHz com 1024MB de memória RAM e sistema operacional Windows XP – Professional Edition.

5.1 COMPARAÇÕES ENTRE ABORDAGENS

A Tabela 1 ilustra os resultados do algoritmo ILS-VND obtidos nas instâncias de Dethloff (2001), onde v_i corresponde ao número de veículos inicialmente considerados e v_f ao número de veículos determinados pela solução final. A Tabela 2, por sua vez, exibe uma comparação entre as soluções apontadas pela literatura e as encontradas pelo procedimento ILS-VND. As perturbações utilizadas foram $P^{(1)}$, $P^{(2)}$ e $P^{(3)}$.

A partir dos resultados expostos na Tabela 1, é possível afirmar que o ILS-VND teve um desempenho consistente, uma vez que o *gap* médio entre as melhores soluções obtidas e a média das soluções encontradas foi de apenas 0,13%, tendo seu maior valor igual a 0,72% na instância CON3-9. Através da Tabela 2, observa-se que dos 40 problemas em questão, o procedimento ILS-VND igualou a solução de 31 destes e melhorou o resultado dos outros 9, levando a um ganho médio de -0,12% em relação às melhores soluções apontadas na literatura. Uma comparação relativa ao número de veículos utilizados acaba se tornando complicada, pois Ropke e Pisinger (2004) não disponibilizaram em seu trabalho. Ainda assim é possível notar que o número de veículos utilizados pelo ILS-VND foi sempre menor ou igual aos utilizados por Montané e Galvão (2006).

Tabela 1 – Resultados obtidos pelo ILS-VND nas instâncias de Dethloff

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
SCA3-0	50	4	4	636,06	636,25	0,03	0,5	4,82
SCA3-1	50	4	4	697,84	697,84	0,00	0,0	5,20
SCA3-2	50	4	4	659,34	659,34	0,00	0,0	5,33
SCA3-3	50	4	4	680,04	680,04	0,00	0,0	5,08
SCA3-4	50	4	4	690,50	690,50	0,00	0,0	6,03
SCA3-5	50	4	4	659,90	659,90	0,00	0,2	5,14
SCA3-6	50	4	4	651,09	651,09	0,00	0,0	5,83
SCA3-7	50	4	4	659,17	663,50	0,66	0,9	6,04
SCA3-8	50	4	4	719,47	719,47	0,00	0,0	4,95
SCA3-9	50	4	4	681,00	681,00	0,13	0,0	4,69
SCA8-0	50	9	9	961,50	964,97	0,36	0,0	7,58
SCA8-1	50	9	9	1049,65	1050,07	0,04	0,8	7,29
SCA8-2	50	9	9	1039,64	1040,63	0,10	0,2	11,11
SCA8-3	50	9	9	983,34	984,26	0,09	1,0	7,50
SCA8-4	50	9	9	1065,49	1066,09	0,06	0,5	6,28
SCA8-5	50	9	9	1027,08	1031,20	0,40	0,9	6,72
SCA8-6	50	9	9	971,82	972,29	0,05	0,5	6,97
SCA8-7	50	9	9	1051,28	1057,23	0,57	0,5	19,31
SCA8-8	50	9	9	1071,18	1071,18	0,00	0,0	5,77
SCA8-9	50	9	9	1060,50	1062,01	0,14	0,6	8,00
CON3-0	50	4	4	616,52	617,67	0,19	1,0	6,15
CON3-1	50	4	4	554,47	554,52	0,01	0,9	4,99
CON3-2	50	4	4	519,11	520,13	0,20	0,6	5,58
CON3-3	50	4	4	591,19	591,19	0,00	0,0	5,80
CON3-4	50	4	4	588,79	589,76	0,16	0,9	5,71
CON3-5	50	4	4	563,70	563,77	0,01	1,0	5,66
CON3-6	50	4	4	499,05	500,25	0,24	0,8	5,79
CON3-7	50	4	4	576,48	577,09	0,11	0,6	5,72
CON3-8	50	4	4	523,05	523,05	0,00	0,9	6,25
CON3-9	50	4	4	578,24	582,43	0,72	0,8	5,45
CON8-0	50	9	9	857,17	857,65	0,06	0,7	8,30
CON8-1	50	9	9	740,85	740,87	0,00	0,9	6,52
CON8-2	50	9	9	712,89	713,31	0,06	0,6	6,74
CON8-3	50	10	10	811,07	812,37	0,16	0,7	2,99
CON8-4	50	9	9	772,25	772,25	0,00	0,5	7,29
CON8-5	50	9	9	754,88	756,92	0,27	0,2	6,51
CON8-6	50	9	9	678,92	680,32	0,21	0,9	6,62
CON8-7	50	9	9	811,96	813,07	0,14	0,3	5,92
CON8-8	50	9	9	767,53	768,34	0,11	0,9	6,46
CON8-9	50	9	9	809,00	809,66	0,08	0,7	6,88

Tabela 2 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Dethloff

Problema	Nº de clientes	Montané e Galvão			Ropke e Pisinger			ILS-VND			Gap (%)
		Sol.	ν	t^*	Sol.	ν	t^{**}	Sol.	ν	t	
SCA3-0	50	640,55	4	3,37	636,1	-	232	636,06	4	3,87	0,00
SCA3-1	50	697,84	4	3,25	697,8	-	170	697,84	4	4,36	0,00
SCA3-2	50	659,34	4	3,52	659,3	-	160	659,34	4	4,70	0,00
SCA3-3	50	680,04	4	3,31	680,6	-	182	680,04	4	4,11	0,00
SCA3-4	50	690,50	4	3,43	690,5	-	160	690,50	4	5,03	0,00
SCA3-5	50	659,90	4	3,67	659,9	-	178	659,90	4	4,48	0,00
SCA3-6	50	653,81	4	3,35	651,09	-	171	651,09	4	4,53	0,00
SCA3-7	50	659,17	4	3,33	666,1	-	162	659,17	4	5,48	0,00
SCA3-8	50	719,47	4	3,40	719,5	-	157	719,47	4	4,23	0,00
SCA3-9	50	681,00	4	3,41	681,0	-	167	681,00	4	4,36	0,00
SCA8-0	50	981,47	9	4,14	975,1	-	98	961,50	9	6,41	-1,39
SCA8-1	50	1077,44	9	4,27	1052,4	-	95	1049,65	9	5,67	-0,26

SCA8-2	50	1050,98	10	4,20	1039,6	-	83	1039,64	9	9,63	0,00
SCA8-3	50	983,34	9	4,17	991,1	-	94	983,34	9	6,61	0,00
SCA8-4	50	1073,46	9	4,13	1065,5	-	84	1065,49	9	5,33	0,00
SCA8-5	50	1047,24	9	4,02	1027,1	-	96	1027,08	9	6,06	0,00
SCA8-6	50	995,59	9	3,85	972,5	-	93	971,82	9	6,11	-0,07
SCA8-7	50	1068,56	10	4,22	1061,0	-	92	1051,28	9	17,19	-0,92
SCA8-8	50	1080,58	9	3,85	1071,2	-	85	1071,18	9	4,80	0,00
SCA8-9	50	1084,80	9	4,20	1060,5	-	86	1060,50	9	7,30	0,00
CON3-0	50	631,39	4	3,64	616,5	-	171	616,52	9	5,30	0,00
CON3-1	50	554,47	4	3,31	554,5	-	190	554,47	9	4,05	0,00
CON3-2	50	522,86	4	3,45	521,4	-	176	519,11	9	5,45	-0,44
CON3-3	50	591,19	4	3,28	591,2	-	177	591,19	9	4,44	0,00
CON3-4	50	591,12	4	3,47	588,8	-	173	588,79	9	4,98	0,00
CON3-5	50	563,70	4	3,38	563,7	-	179	563,70	9	4,53	0,00
CON3-6	50	506,19	4	3,32	499,1	-	195	499,05	9	5,03	0,00
CON3-7	50	577,68	4	3,51	576,5	-	226	576,48	9	4,64	0,00
CON3-8	50	523,05	4	3,66	523,1	-	174	523,05	9	4,92	0,00
CON3-9	50	580,05	4	3,36	578,2	-	163	578,24	9	3,92	0,00
CON8-0	50	860,48	9	4,19	857,2	-	86	857,17	9	7,03	0,00
CON8-1	50	740,85	9	3,89	740,9	-	81	740,85	9	5,48	0,00
CON8-2	50	723,32	9	3,76	716,0	-	84	712,89	9	6,23	-0,43
CON8-3	50	811,23	10	4,12	811,1	-	91	811,07	10	2,42	0,00
CON8-4	50	772,25	9	3,75	772,3	-	87	772,25	9	6,05	0,00
CON8-5	50	756,91	9	3,99	755,7	-	94	754,88	9	6,20	-0,11
CON8-6	50	678,92	9	4,04	693,1	-	96	678,92	9	5,78	0,00
CON8-7	50	814,50	9	4,00	814,8	-	94	811,96	9	5,22	-0,31
CON8-8	50	775,59	9	3,74	774,0	-	94	767,53	9	5,66	-0,84
CON8-9	50	809,00	9	4,13	809,3	-	92	809,00	9	5,50	0,00

(*) Tempo de CPU em segundos em um PC Athlon XP 2.0GHz.

(**) Tempo de CPU em segundos em um PC Pentium IV 1.5GHz.

Os tempos apresentados na Tabela 2 dão uma idéia do esforço computacional demandando, mas como se refere a máquinas com configurações distintas, não é possível efetuar uma comparação direta entre os respectivos algoritmos.

Tabela 3 – Comparação entre o ILS-VND e os resultados de Bianchessi e Righini nas instâncias de Dethloff

Instância	Bianchessi e Righini		ILS-VND		Gap (%)
	Solução Média	t^*	Solução Média	t	
SCA3	684,6	59,47	673,35	5,33	-1,64
SCA8	1035,7	17,05	1024,55	8,38	-1,08
CON3	568,5	59,36	561,06	5,71	-1,31
CON8	776,4	15,14	771,65	6,34	-0,61

(*) Tempo de CPU em segundos em um PC 1.6GHz.

Bianchessi e Righini (2007) também testaram suas heurísticas nas instâncias geradas por Dethloff (2001), onde o algoritmo baseado na metaheurística Busca Tabu

alcançou melhores soluções. No entanto, os resultados apresentados em seu trabalho apenas consideram a média das soluções encontradas em cada um dos quatro grupos de problemas-testes. A Tabela 3 mostra estes resultados, onde é possível constatar que as soluções encontradas pelo ILS-VND foram, em média, melhores, apresentando um ganho de -1,16%.

A Tabela 4 apresenta os resultados alcançados nos problemas-testes de Salhi e Nagy (1999), ao passo que a Tabela 5 mostra uma comparação do ILS-VND com outros algoritmos da literatura. As mesmas perturbações utilizadas nas instâncias de Dethloff (2001) foram empregadas nestes problemas-testes, isto é, $P^{(1)}$, $P^{(2)}$ e $P^{(3)}$.

Tabela 4 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
CMT1X	50	3	3	466,77	467,32	0,12	0,25	4,80
CMT1Y	50	3	3	466,77	467,44	0,14	0,30	5,20
CMT2X	75	6	6	684,21	687,67	0,51	0,10	26,21
CMT2Y	75	6	6	684,60	687,72	0,46	0,15	26,22
CMT3X	100	5	5	721,40	726,76	0,74	0,50	33,00
CMT3Y	100	5	5	721,27	726,26	0,69	0,50	33,53
CMT12X	100	6	5	662,22	671,04	1,33	0,35	34,80
CMT12Y	100	6	6	663,50	673,03	1,44	0,00	34,67
CMT11X	120	5	4	842,23	879,31	4,40	0,50	58,51
CMT11Y	120	5	4	846,23	880,77	4,08	0,35	57,75
CMT4X	150	7	7	852,83	864,64	1,38	0,35	209,97
CMT4Y	150	7	7	852,83	866,93	1,65	0,35	208,44
CMT5X	199	11	10	1031,17	1065,51	3,33	0,50	240,87
CMT5Y	199	11	10	1030,93	1059,49	2,77	0,10	240,59

Analisando a Tabela 4, verifica-se que o *gap* médio entre as melhores soluções obtidas e a média das soluções foi de 1,65%, tendo seu maior valor igual a 4,40% na instância CMT5X. Já a Tabela 5 mostra que dentre os 14 problemas listados, o algoritmo ILS-VND obteve um *gap* médio de 0,66% em relação às melhores soluções obtidas na literatura até então, tendo conseguido melhorar o resultado de 3 problemas e igualado à solução de outros 2. Contudo, na maioria dos casos, as soluções encontradas foram inferiores às da literatura. À exceção das instâncias de 199 clientes, os números de veículos foram sempre iguais aos melhores resultados da literatura.

Tabela 5 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy

Problema	Nº de clientes	Montané e Galvão			Ropke e Pisinger			Chen			Wassan <i>et al.</i>			ILS-VND			Gap (%)
		Sol.	ν	t^*	Sol.	ν	t^{**}	Sol.	ν	t^{***}	Sol.	ν	t^{****}	Sol.	ν	t	
CMT1X	50	472	3	3,73	467	-	171	478	3	6,23	468,30	3	48	466,77	3	4,25	1,70
CMT1Y	50	470	3	4,37	467	-	192	481	3	6,75	458,96	3	69	466,77	3	3,86	0,00
CMT2X	75	695	7	6,91	702	-	255	689	6	21,78	668,77	6	94	684,21	6	30,31	2,31
CMT2Y	75	700	7	7,61	685	-	315	679	6	11,63	663,25	6	102	684,60	6	24,67	3,22
CMT3X	100	721	5	11,04	727	-	768	745	5	83,77	729,63	5	294	721,40	5	28,17	0,00
CMT3Y	100	719	5	12,01	734	-	632	723	5	112,98	745,46	5	285	721,27	5	34,25	0,32
CMT12X	100	675	6	12,23	683	-	574	675	5	44,78	644,70	5	242	662,22	5	34,69	2,72
CMT12Y	100	689	5	12,80	673	-	531	673	5	53,67	659,52	6	254	663,50	6	39,81	0,60
CMT11X	120	900	4	18,17	837	-	1821	859	4	309,72	861,97	4	504	842,23	4	67,63	0,62
CMT11Y	120	910	4	18,08	920	-	1376	860	4	221,64	830,39	4	325	846,23	4	62,23	1,96
CMT4X	150	880	7	24,60	877	-	1459	867	7	469,78	876,50	7	558	852,83	7	183,14	-1,63
CMT4Y	150	878	7	29,09	854	-	1866	852	7	417,44	870,44	7	405	852,83	7	225,06	0,10
CMT5X	199	1098	11	51,50	1108	-	2057	1067	10	775,83	1044,51	9	483	1031,17	10	272,30	-1,28
CMT5Y	199	1083	10	56,21	1131	-	2030	1046	10	731,71	1054,46	9	533	1030,93	10	271,56	-1,44

(*) Tempo de CPU em segundos em um PC Athlon XP 2.0GHz.

(**) Tempo de CPU em segundos em um PC Pentium IV 1.5GHz.

(***) Tempo de CPU em segundos em um PC Pentium IV 1.6GHz.

(****) Tempo de CPU em segundos em um Sun-Fire-V440 com processador UltraSPARC-IIIi 1062 MHz.

No tocante às instâncias que incluem a restrição de tempo máximo por rota, alguns autores (Salhi and Nagy (1999), Dethloff (2001), Ropke e Pisinger (2004)) trataram esses problemas-testes considerando a presença do *drop time*, ao passo que Montané e Galvão (2006) não o incluíram. Recentemente, Wassan *et al.* (2007) avaliou ambas as situações. Nagy e Salhi (2005) estudaram o problema levando em conta o *drop time*, mas permitindo duas visitas a um mesmo cliente. Neste caso, uma comparação direta não pôde ser realizada com os resultados alcançados por estes últimos autores, uma vez que esta situação não foi abordada no presente trabalho.

A Tabela 6 exhibe os resultados obtidos para a situação em que o *drop time* foi considerado, ao passo que a Tabela 7 apresenta uma comparação com os demais resultados apontados pela literatura. Neste caso, apenas a perturbação *double-bridge* ($P^{(3)}$) foi utilizada. Embora os resultados, em média, tenham sido similares, quando o *double-swap* também era utilizado, ressaltam-se que as melhores soluções foram obtidas quando se fez uso apenas do *double-bridge*. Já a perturbação *Ejection Chain* não foi capaz de atuar de forma eficiente nestas instâncias. A principal razão para a ocorrência deste fato se deve à dificuldade em se efetuar uma perturbação viável, principalmente devido à restrição de tempo máximo.

Tabela 6 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy com restrições de limite de tempo (com *drop time*)

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
CMT6X	50	7	6	555,43	558,71	0,59	0,35	2,61
CMT6Y	50	7	6	555,43	558,93	0,63	0,45	2,48
CMT7X	75	14	11	901,22	911,35	1,12	0,35	3,83
CMT7Y	75	14	11	901,22	910,88	1,07	0,40	3,79
CMT8X	100	10	9	865,50	870,72	0,60	0,50	20,13
CMT8Y	100	10	9	865,50	870,42	0,57	0,20	25,28
CMT14X	100	11	10	821,75	821,75	0,00	0,00	32,85
CMT14Y	100	11	10	821,75	821,75	0,00	0,00	31,87
CMT13X	120	12	11	1545,96	1566,97	1,36	0,20	102,57
CMT13Y	120	12	11	1542,86	1557,83	0,97	0,35	99,29
CMT9X	150	16	14	1167,23	1196,81	2,53	0,20	67,87
CMT9Y	150	17	14	1167,69	1194,08	2,26	0,10	66,89
CMT10X	199	21	18	1407,66	1437,31	2,11	0,20	157,55
CMT10Y	199	20	18	1413,88	1434,05	1,43	0,45	158,21

Observando a Tabela 6, verifica-se que o *gap* médio entre as soluções médias e as melhores soluções foi de 1,23%, com maior valor na instância CMT9X (2,53%). Analisando a Tabela 7, é possível observar que, dos 14 problemas-testes, o ILS-VND foi capaz de melhorar o resultado de 13 e igualou o valor de outros 2, com um *gap* médio de -1,85%. Novamente a comparação entre o número de veículos não pode ser efetuada de maneira clara, já que estes valores não foram explicitados por Ropke e Pisinger (2004). Em relação aos resultados de Dethloff (2001) e Wassan *et al.* (2007), o número de veículos determinados pelo ILS-VND sempre foi menor ou igual.

Tabela 7 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy com restrições de limite de tempo (com *drop time*)

Problema	Nº de clientes	Dethloff			Ropke e Pisinger			Wassan <i>et al.</i>			ILS-VND			Gap (%)
		Sol.	ν	t	Sol.	ν	t^*	Sol.	ν	t^{**}	Sol.	ν	t^{***}	
CMT6X	50	584	6	-	559	-	97	556,06	6	32	555,43	6	2,47	-0,11
CMT6Y	50	584	6	-	559	-	93	558,17	6	33	555,43	6	2,44	-0,49
CMT7X	75	961	11	-	901	-	167	903,05	11	49	901,22	11	3,72	0,00
CMT7Y	75	961	11	-	952	-	166	903,36	11	47	901,22	11	3,88	-0,24
CMT8X	100	928	9	-	866	-	384	879,60	9	131	865,50	9	19,53	-0,06
CMT8Y	100	936	9	-	873	-	398	917,42	10	113	865,50	9	23,14	-0,86
CMT14X	100	871	10	-	863	-	367	823,95	10	134	821,75	10	29,62	-0,26
CMT14Y	100	871	10	-	-	-	-	823,34	10	122	821,75	10	28,33	-0,19
CMT13X	120	1576	11	-	1578	-	563	1647,51	12	163	1545,96	11	95,00	-1,91
CMT13Y	120	1576	11	-	1602	-	547	1647,04	11	173	1542,86	11	105,66	-2,10
CMT9X	150	1299	15	-	1197	-	726	1220,00	15	201	1167,23	14	69,20	-2,49
CMT9Y	150	1299	15	-	1256	-	720	1213,11	15	171	1167,69	14	65,36	-3,74
CMT10X	199	1571	19	-	1462	-	1275	1464,58	19	301	1407,66	18	151,41	-3,72
CMT10Y	199	1571	19	-	1552	-	1255	1419,79	18	318	1413,88	18	148,11	-0,42

(*) Tempo de CPU em segundos em um PC Pentium IV 1.5GHz.

(**)Tempo de CPU em segundos em um Sun-Fire-V440 com processador UltraSPARC-IIIi 1062 MHz.

A Tabela 8 ilustra os resultados obtidos pelo ILS-VND quando o *drop time* foi desprezado. Assim como no caso anterior, apenas a perturbação *double bridge* ($P^{(3)}$) foi utilizada. Neste caso, o *gap* médio entre as soluções médias e as melhores soluções foi de 1,83%, com o maior *gap* na instância CMT13Y (3,86%). A partir dos resultados expostos na Tabela 9, observa-se que, dos 14 problemas-testes, o ILS-VND não conseguiu superar a literatura em 5 situações, porém, se mostrou capaz de melhorar 8 resultados e igualar 1, com

gap médio de -0,54%. Dentre os problemas em que houve melhora, foi possível reduzir o número de veículos na instância CMT13X. Para os demais casos, os valores foram equivalentes, à exceção das instâncias CMT14X e CMT14Y.

Tabela 8 – Resultados obtidos pelo ILS-VND nas instâncias de Salhi e Nagy com restrições de limite de tempo (sem *drop time*)

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
CMT6X	50	3	3	466,77	467,62	0,18	0,30	4,13
CMT6Y	50	3	3	466,77	466,84	0,01	0,25	4,19
CMT7X	75	7	6	686,52	698,03	1,68	0,45	12,38
CMT7Y	75	7	6	688,46	698,97	1,53	0,10	12,93
CMT8X	100	5	5	721,40	727,60	0,86	0,35	31,04
CMT8Y	100	5	5	721,40	728,09	0,93	0,50	31,82
CMT14X	100	6	6	663,50	672,82	1,40	0,20	34,06
CMT14Y	100	6	5	662,22	672,56	1,56	0,40	32,48
CMT13X	120	5	4	846,85	878,27	3,71	0,50	61,50
CMT13Y	120	5	4	848,45	881,23	3,86	0,40	57,11
CMT9X	150	8	7	855,74	870,77	1,76	0,35	102,88
CMT9Y	150	8	7	856,74	871,21	1,69	0,30	103,71
CMT10X	199	11	10	1037,37	1058,31	2,02	0,05	284,52
CMT10Y	199	11	10	1036,59	1059,56	2,22	0,15	242,11

Tabela 9 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Salhi e Nagy com restrições de limite de tempo (sem *drop time*)

Problema	Nº de clientes	Montané e Galvão			Wassan <i>et al.</i>			ILS-VND			Gap (%)
		Sol.	v	t^*	Sol.	v	t^{**}	Sol.	v	t	
CMT6X	50	476	3	3.7	471,89	3	65	466,77	3	3,56	-1,08
CMT6Y	50	474	3	4.37	467,70	3	60	466,77	3	3,20	-0,20
CMT7X	75	695	7	6.91	663,95	6	86	686,52	6	12,59	3,40
CMT7Y	75	700	6	7.61	662,50	6	80	688,46	6	11,62	3,92
CMT8X	100	720	5	11.04	726,88	5	254	721,40	5	27,33	0,19
CMT8Y	100	721	5	12.01	741,96	5	315	721,40	5	34,64	0,06
CMT14X	100	675	6	12.23	644,70	5	244	663,50	6	34,08	2,92
CMT14Y	100	689	6	12.80	659,52	6	255	662,22	5	37,09	0,41
CMT13X	120	918	5	18.17	858,48	5	319	846,85	4	63,95	-1,35
CMT13Y	120	910	5	18.08	880,56	4	546	848,45	4	71,19	-3,65
CMT9X	150	885	7	24.60	880,61	7	561	855,74	7	100,64	-2,82
CMT9Y	150	900	8	29.09	886,84	7	562	856,74	7	111,47	-3,39
CMT10X	199	1100	11	51.50	1079,99	10	529	1037,37	10	271,45	-3,95
CMT10Y	199	1083	11	56.21	1058,09	10	524	1036,59	10	225,45	-2,03

(*) Tempo de CPU em segundos em um PC Athlon XP 2.0GHz.

(**) Tempo de CPU em segundos em um Sun-Fire-V440 with UltraSPARC-IIIi processor, CPU speed 1062 MHz, em um Solaris 9.

A Tabela 10 apresenta os resultados encontrados nas instâncias de Montané e Galvão (2006), enquanto a Tabela 11 mostra uma comparação entre os ILS-VND e o algoritmo proposto por estes autores. As perturbações empregadas nestas instâncias foram:

double swap ($P^{(2)}$) e *double-bridge* ($P^{(3)}$). Salienta-se que a perturbação *Ejection Chain* não se mostrou eficiente para estas instâncias, em particular naquelas envolvendo um elevado número de veículos, pois as soluções eram modificadas em demasia, além de frequentemente produzirem soluções inviáveis.

Tabela 10 – Resultados obtidos pelo ILS-VND nas instâncias de Montané e Galvão

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
r101	100	12	12	1012,96	1024,66	1,16	0,2	40,14
r201	100	3	3	666,20	666,54	0,05	0,3	31,60
c101	100	16	16	1220,99	1228,88	0,65	0,9	15,90
c201	100	5	5	662,07	662,32	0,04	0,1	23,56
rc101	100	10	10	1059,32	1063,80	0,42	0,4	48,21
rc201	100	3	3	672,92	672,95	0,00	0,8	27,54
r1_2_1	200	24	23	3393,02	3433,12	1,18	0,4	236,72
r2_2_1	200	6	5	1666,43	1682,35	0,96	0,1	167,18
c1_2_1	200	29	28	3644,30	3668,31	0,66	1,0	193,27
c2_2_1	200	9	9	1729,59	1737,85	0,48	0,5	227,57
rc1_2_1	200	24	24	3340,04	3381,15	1,23	0,2	315,93
rc2_2_1	200	6	5	1560,00	1565,03	0,32	0,2	158,12
r1_4_1	400	55	54	9761,34	9869,86	1,11	0,7	1308,32
r2_4_1	400	11	10	3594,90	3646,47	1,43	0,6	1464,61
c1_4_1	400	64	63	11179,23	11271,89	0,83	0,6	1390,00
c2_4_1	400	16	15	3577,56	3633,26	1,56	0,8	1592,26
rc1_4_1	400	53	52	9677,44	9757,13	0,82	0,7	1602,63
rc2_4_1	400	12	11	3426,60	3484,95	1,70	0,3	1238,98

Tabela 11 – Comparação entre o ILS-VND e os resultados de Montané e Galvão nas instâncias dos mesmos

Problema	Nº de clientes	Montané e Galvão	v	t^*	ILS-VND	v	t	Gap (%)
r101	100	1042,62	12	13,20	1012,96	12	42,03	-2,84
r201	100	671,03	3	12,02	666,20	3	26,00	-0,72
c101	100	1259,79	17	12,07	1220,99	16	15,52	-3,08
c201	100	666,01	5	12,40	662,07	5	21,45	-0,59
rc101	100	1094,15	11	12,30	1059,32	10	41,22	-3,18
rc201	100	674,46	3	12,07	672,92	3	21,80	-0,23
r1_2_1	200	3447,20	23	55,56	3393,02	23	236,00	-1,57
r2_2_1	200	1690,67	5	50,95	1666,43	5	185,25	-1,43
c1_2_1	200	3792,62	29	52,21	3644,30	28	221,84	-3,91
c2_2_1	200	1767,58	9	65,79	1729,59	9	254,44	-2,15
rc1_2_1	200	3427,19	24	58,39	3340,04	24	292,91	-2,54
rc2_2_1	200	1645,94	5	52,93	1560,00	5	147,25	-5,22
r1_4_1	400	10027,81	54	330,42	9761,34	54	1455,47	-2,66
r2_4_1	400	3695,26	10	324,44	3594,90	10	1549,99	-2,72
c1_4_1	400	11676,27	65	287,12	11179,23	63	1707,76	-4,26
c2_4_1	400	3732,00	15	330,20	3577,56	15	1456,14	-4,14
rc1_4_1	400	9883,31	52	286,66	9677,44	52	1782,87	-2,08
rc2_4_1	400	3603,53	11	328,16	3426,60	11	1468,34	-4,91

(*) Tempo de CPU em segundos em um Athlon XP 2.0

Pela Tabela 10, tem-se, para estas instâncias, que o *gap* médio entre as soluções médias e as melhores soluções foi de 0,81%, com o maior valor na instância rc2_4_1 (1,70%). Comparando os resultados de Montané e Galvão com os alcançados pelo ILS-VND (Tabela 11), observa-se que houve uma melhora em todos os 12 problemas-testes, com um *gap* médio expressivo de -2,68%. O número de veículos foi reduzido em 4 instâncias, igualando-se ao valor de Montané e Galvão (2006) nas restantes.

Os resultados obtidos na instância de Min (1989) encontram-se apresentados na Tabela 12. Assim como nos problemas de Dethloff (2001) e Salhi e Nagy (1999), as perturbações empregadas foram $P^{(1)}$, $P^{(2)}$ e $P^{(3)}$. A Tabela 13 mostra que a solução encontrada pelo ILS-VND igualou-se às demais da literatura. Vale ressaltar que este resultado (88) corresponde à solução ótima do problema, conforme determinado por Montané e Galvão (2006).

Por se tratar de um problema de dimensão pequena, o modelo apresentado no Capítulo 3 também foi aplicado, através do uso do *software* LINGO, versão 7.0, para resolução deste problema. A solução ótima foi encontrada em aproximadamente 33 min.

Tabela 12 – Resultados obtidos pelo ILS-VND na instância de Min

Problema	Nº de clientes	v_i	v_f	Melhor Solução	Solução Média	Gap (%)	γ	Tempo Médio (s)
MIN22	22	2	2	88	88	0,00	0,0	0,54

Tabela 13 – Comparação entre o ILS-VND e os resultados da literatura nas instâncias de Min

Problema	Nº de clientes	Montané e Galvão			Ropke e Pisinger			ILS-VND			Gap (%)
		Sol.	v	t^*	Sol.	v	t^{**}	Sol.	v	t	
MIN22	50	88	2	1,20	88	-	37	88	2	0,54	0,00

(*) Tempo de CPU em segundos em um PC Athlon XP 2.0GHz.

(**) Tempo de CPU em segundos em um PC Pentium IV 1.5GHz.

5.2 INFLUÊNCIA DO PARÂMETRO γ

Conforme mencionado na Seção 4.3, duas abordagens foram utilizadas para examinar a influência do fator γ nos resultados alcançados pelo ILS-VND. O objetivo da análise através do desvio médio é avaliar a qualidade das soluções médias obtidas para cada valor de γ adotado, ao passo que a segunda abordagem procura verificar os valores de γ mais expressivos em cada problema teste. A idéia é determinar empiricamente, a partir de ambos os estudos, quais são os valores de γ indicados para cada grupo de instâncias.

As Figuras 17-20 mostram os valores dos desvios médios associados a cada valor do parâmetro γ nas instâncias geradas por Dethloff (2001). A Tabela 14 mostra o resultado das análises de variância (ANOVA) realizadas em cada grupo de problemas-testes.

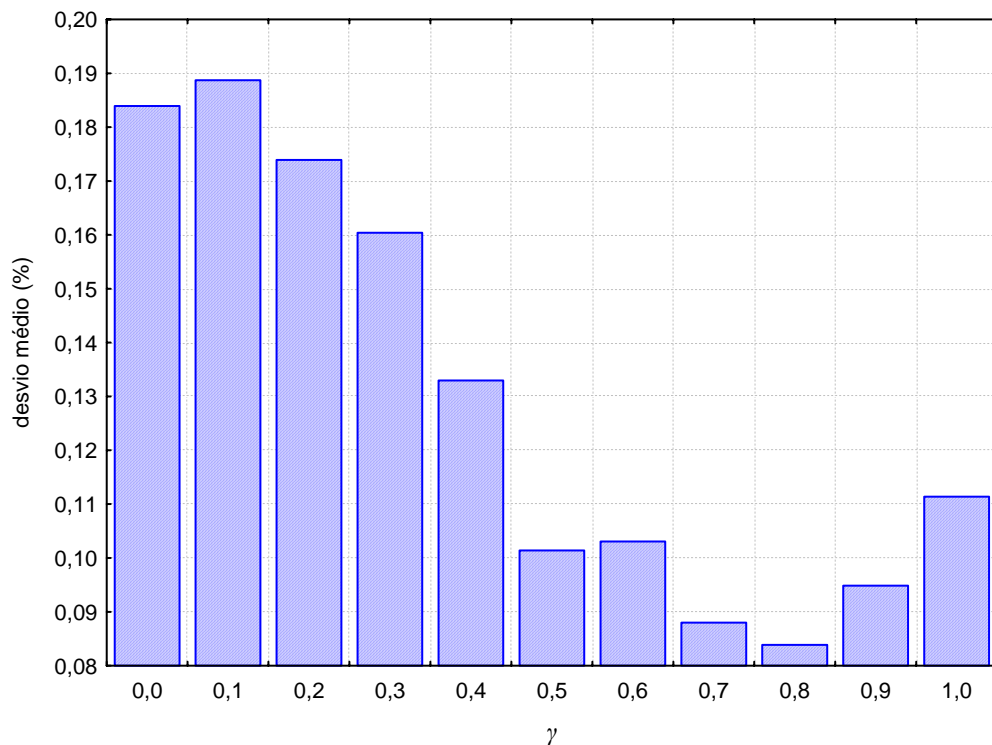


Figura 17 – Influência do parâmetro γ nas instâncias SCA3

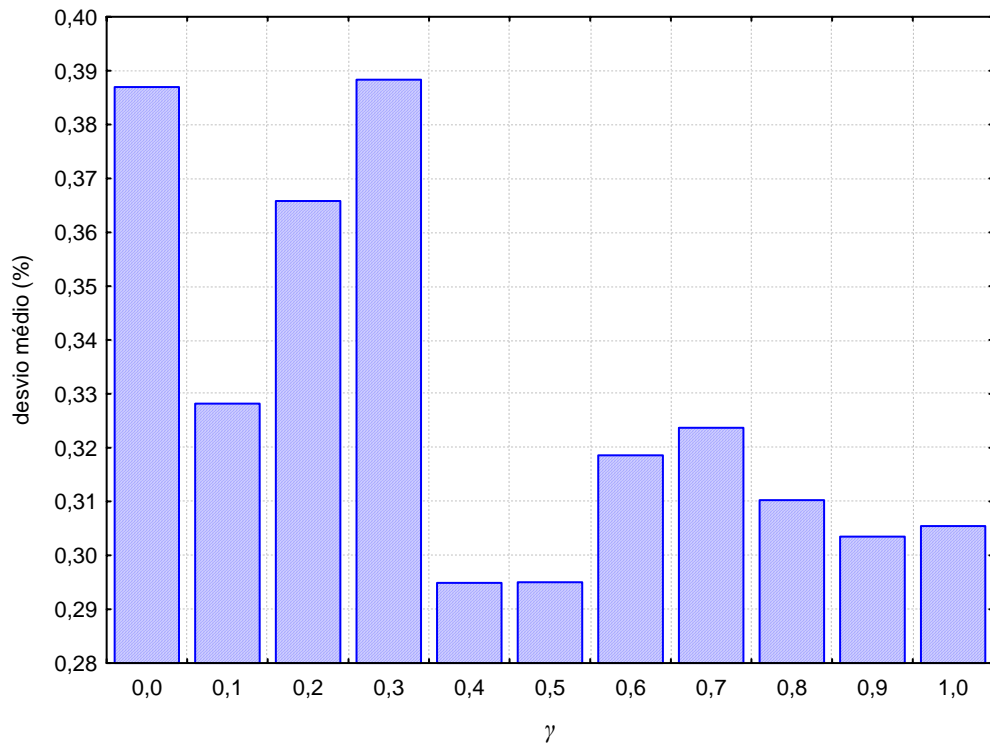


Figura 18 – Influência do parâmetro γ nas instâncias SCA8

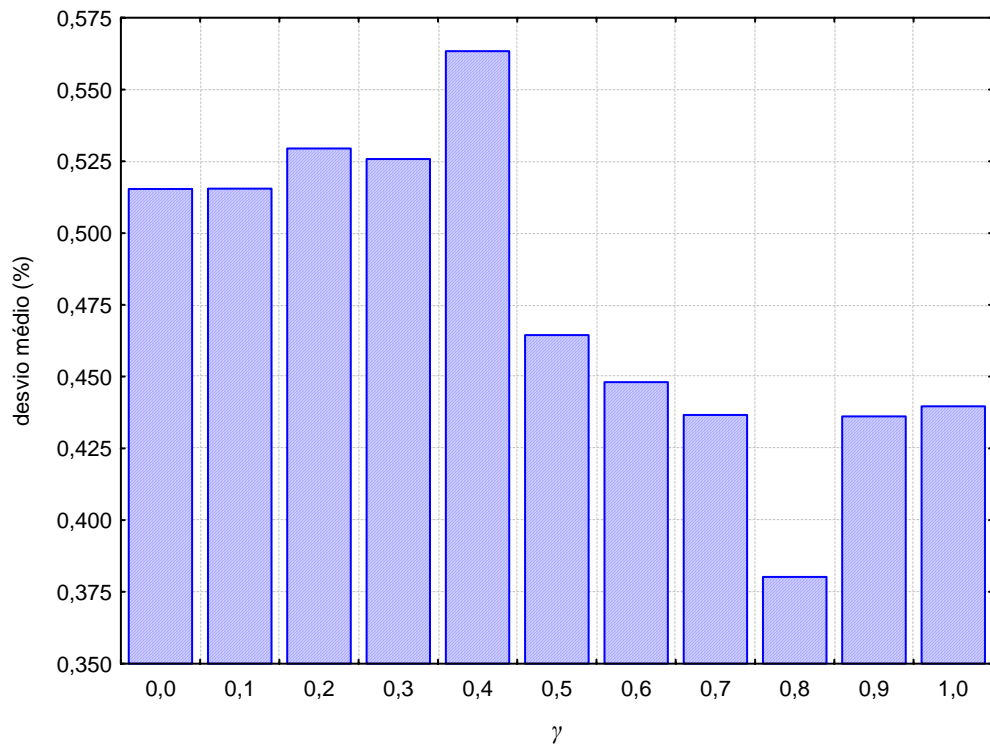


Figura 19 – Influência do parâmetro γ nas instâncias CON3

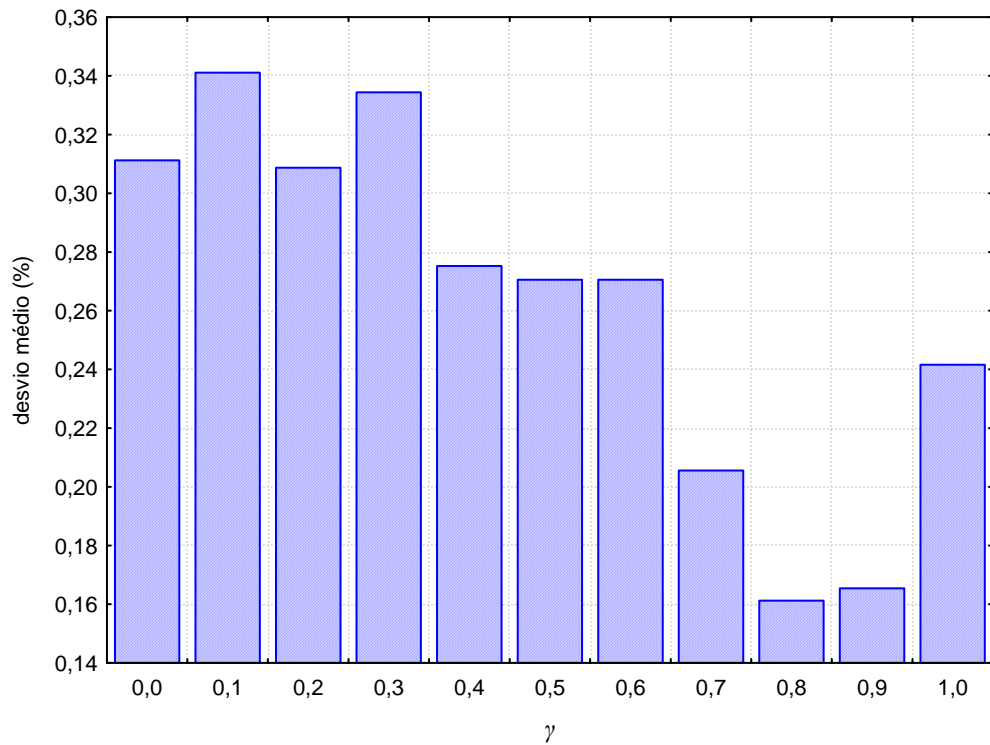


Figura 20 – Influência do parâmetro γ nas instâncias CON8

Tabela 14 – Valores de γ associados às melhores soluções médias nas instâncias de Dethloff

SCA3		SCA8		CON3		CON8	
Problema	γ	Problema	γ	Problema	γ	Problema	γ
SCA3-0	0,3;...; 0,8	SCA8-0	0,2;...; 1,0	CON3-0	0,8;...; 1,0	CON8-0	0,1; 0,4;...; 1,0
SCA3-1	0,0;...; 1,0	SCA8-1	0,0;...; 1,0	CON3-1	0,8;...; 1,0	CON8-1	0,6;...; 1,0
SCA3-2	0,0;...; 1,0	SCA8-2	0,0;...; 1,0	CON3-2	0,0;...; 1,0	CON8-2	0,0;...; 1,0
SCA3-3	0,0;...; 0,6	SCA8-3	0,4;...; 1,0	CON3-3	0,0;...; 1,0	CON8-3	0,6;...; 0,9
SCA3-4	0,0;...; 1,0	SCA8-4	0,4;...; 1,0	CON3-4	0,3;...; 0,6 0,8;...; 1,0	CON8-4	0,0;...; 1,0
SCA3-5	0,0;...; 1,0	SCA8-5	0,2;...; 0,3 0,5;...; 1,0	CON3-5	0,0;...; 1,0	CON8-5	0,0;...; 1,0
SCA3-6	0,0;...; 1,0	SCA8-6	0,0;...; 1,0	CON3-6	0,6;...; 1,0	CON8-6	0,8;...; 0,9
SCA3-7	0,0;...; 1,0	SCA8-7	0,0;...; 1,0	CON3-7	0,5;...; 1,0	CON8-7	0,0;...; 1,0
SCA3-8	0,0;...; 1,0	SCA8-8	0,0;...; 1,0	CON3-8	0,0;...; 1,0	CON8-8	0,0;...; 0,1 0,3;...; 0,5 0,8;...; 0,9
SCA3-9	0,6;...; 1,0	SCA8-9	0,0;...; 0,7 0,9;...; 1,0	CON3-9	0,5;...; 0,8	CON8-9	0,7;...; 0,9
Interseção: 0,6		Interseção: 0,5;...; 0,7 0,9;...; 1,0		Interseção: 0,8		Interseção: 0,8;...; 0,9	

Observando a Figura 17, percebe-se que o desvio médio tende a ser menor nas instâncias SCA3 quando $\gamma = 0,5;...; 0,9$. Analisando a Tabela 14, verifica-se que para este mesmo grupo, o valor de γ que sempre esteve associado às melhores soluções médias de cada

problema foi 0,6. Já nas instâncias SCA8, observa-se que os valores de γ que levaram aos menores desvios médios (Figura 18) pertencem ao intervalo $\gamma = 0,4; \dots; 0,5$, enquanto as melhores soluções médias estiveram sempre associadas a $\gamma = 0,5; \dots; 0,7$ e $\gamma = 0,9; \dots; 1,0$. Nas instâncias CON3, o valor mais expressivo de ambas as abordagens coincidiram (Figura 19 e Tabela 14), sendo $\gamma = 0,8$. O mesmo ocorreu nas instâncias CON8 (Figura 20 e Tabela 14), sendo $\gamma = 0,8; \dots; 0,9$.

As Figuras 21-23 ilustram os desvios médios relativos às soluções obtidas nas instâncias propostas por Salhi e Nagy (1999). A Tabela 15 apresenta os valores de γ associados às melhores soluções médias nestes problemas-testes.

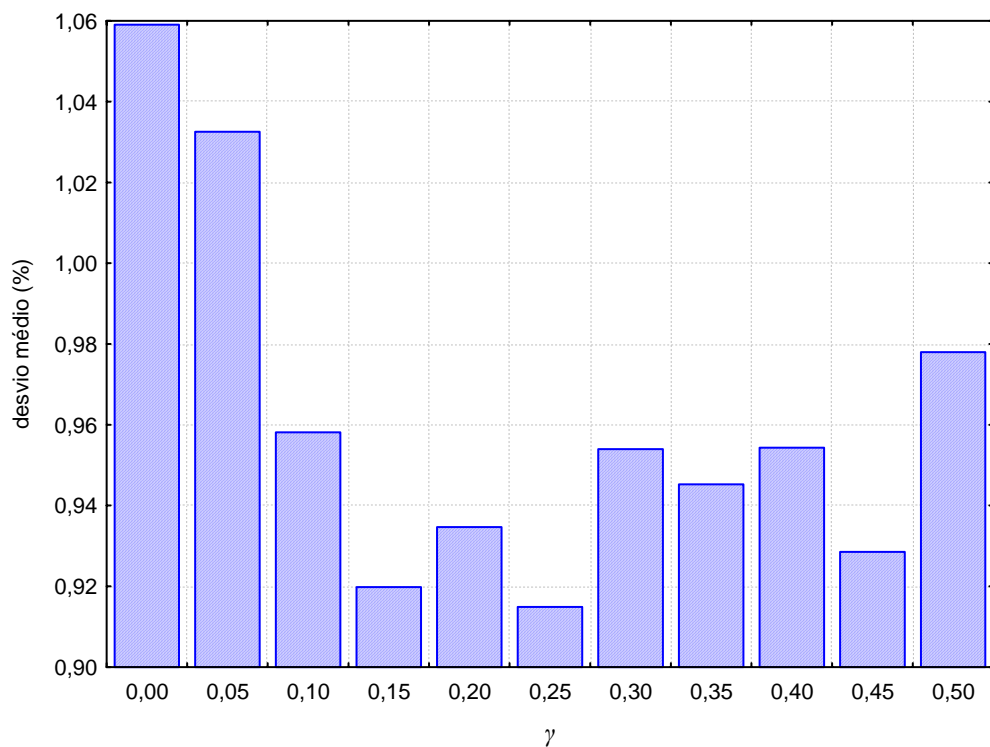


Figura 21 – Influência do parâmetro γ nas instâncias CMT

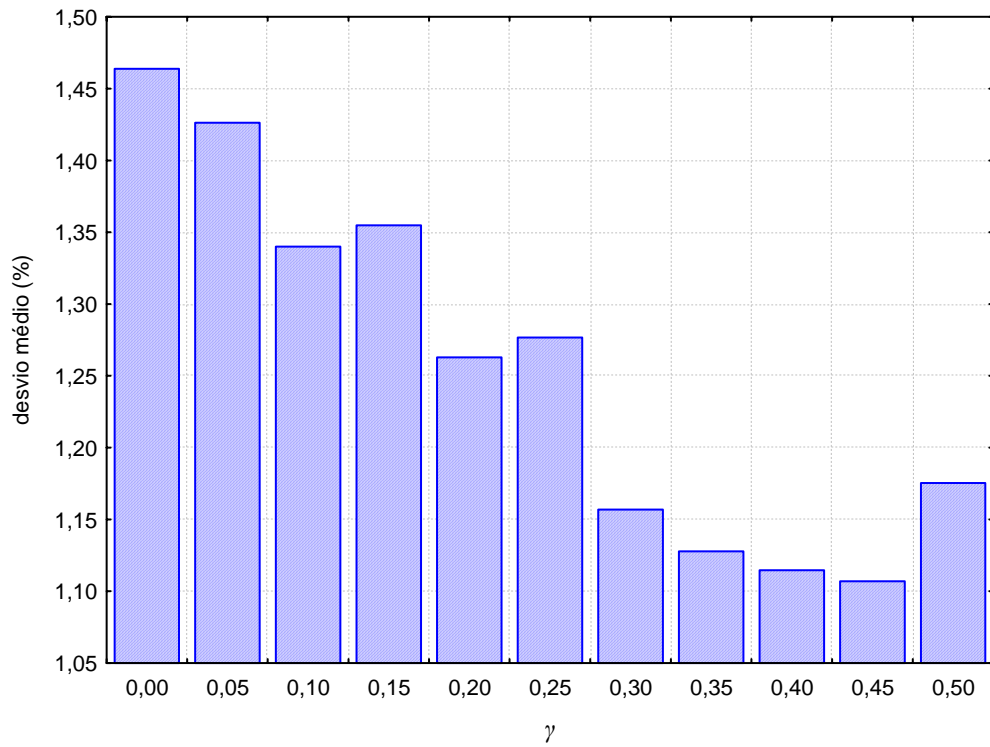


Figura 22 – Influência do parâmetro γ nas instâncias CMT com drop time

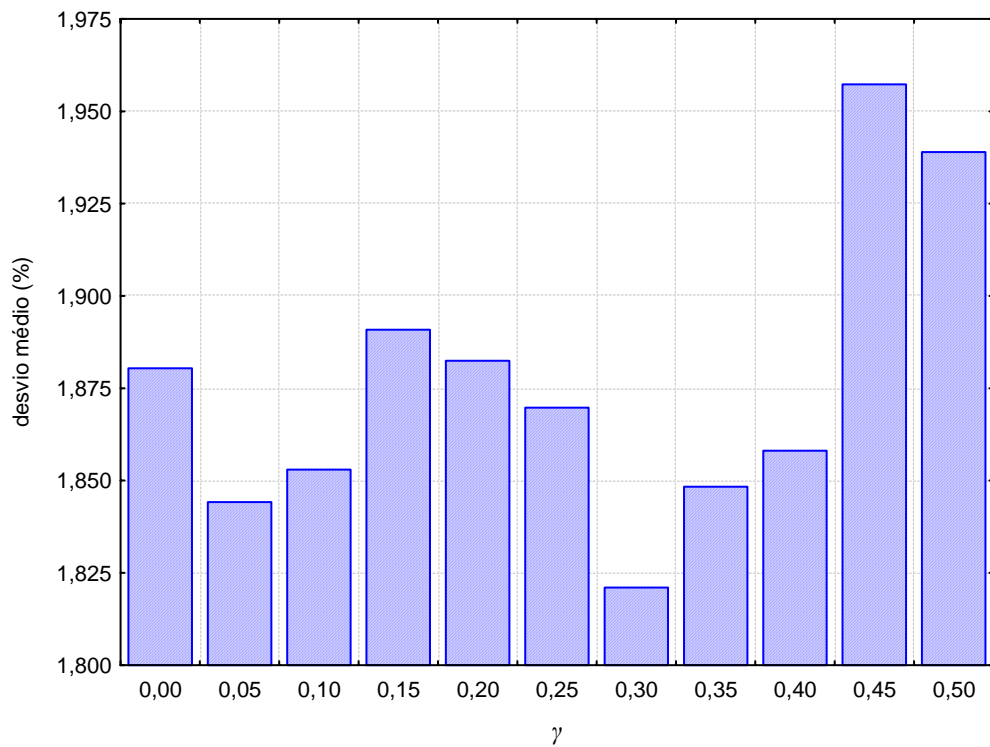


Figura 23 – Influência do parâmetro γ nas instâncias CMT sem *drop time*

Tabela 15 – Valores de γ associados às melhores soluções médias nas instâncias de Salhi e Nagy

CMT		CMT c/ Drop Time		CMT s/ Drop Time	
Problema	γ	Problema	γ	Problema	γ
CMT1X	0,25;...; 0,40	CMT6X	0,00;...; 0,45	CMT6X	0,25;...; 0,40
CMT1Y	0,25;...; 0,40	CMT6Y	0,00;...; 0,50	CMT6Y	0,25;...; 0,35
CMT2X	0,00;...; 0,30	CMT7X	0,00; 0,20;...; 0,50	CMT7X	0,00;...; 0,50
CMT2Y	0,00;...; 0,30	CMT7Y	0,15; 0,25;...; 0,50	CMT7Y	0,00;... 0,50
CMT3X	0,00;...; 0,50	CMT8X	0,00;...; 0,50	CMT8X	0,00;...; 0,50
CMT3Y	0,00;...; 0,50	CMT8Y	0,00;...; 0,50	CMT8Y	0,00;...; 0,50
CMT12X	0,00;...; 0,50	CMT14X	0,00;...; 0,50	CMT14X	0,00;...; 0,10 0,20; 0,35;...; 0,50
CMT12Y	0,00;...; 0,50	CMT14Y	0,00;...; 0,50	CMT14Y	0,00;...; 0,50
CMT11X	0,00;...; 0,50	CMT13X	0,00;...; 0,50	CMT13X	0,00;...; 0,50
CMT11Y	0,00;...; 0,50	CMT13Y	0,30;...; 0,45	CMT13Y	0,00;...; 0,50
CMT4X	0,00;...; 0,30	CMT9X	0,00;...; 0,50	CMT9X	0,00;...; 0,50
CMT4Y	0,00;...; 0,50	CMT9Y	0,00;...; 0,50	CMT9Y	0,00;...; 0,50
CMT5X	0,00;...; 0,30	CMT10X	0,00;...; 0,50	CMT10X	0,00;...; 0,30
CMT5Y	0,00;...; 0,25	CMT10Y	0,00;...; 0,50	CMT10Y	0,00;...; 0,30
Interseção: 0,25		Interseção: 0,35;...; 0,45		Interseção: 0,30	

A Figura 21 mostra que o valor de γ responsável pelo menor desvio médio nas instâncias CMT foi 0,25, sendo igual ao valor da interseção exposto na Tabela 15. Fato semelhante ocorreu nas instâncias CMT com *drop time*, onde os valores mais consistentes (Figura 22 e Tabela 15) deste parâmetro foram os mesmos, isto é, $\gamma = 0,35;...; 0,45$. Nas instâncias CMT sem *drop time*, verifica-se que o desvio médio assume valores menores quando $\gamma = 0,30$ (Figura 23). Pela Tabela 15, observa-se que o valor de γ que sempre esteve presente entre as melhores soluções médias também correspondeu a 0,30.

A Figura 24 mostra que o desvio médio, nas instâncias de Montané e Galvão, tende a ser inferior entre 0,2 e 0,6, sendo mais expressivo ao assumir valores nos extremos deste intervalo. Por outro lado, a Tabela 16 mostra que, à exceção das instâncias r201 e rc1_4_1, o valor de γ que esteve consecutivamente associado às melhores soluções médias foi 0,6.

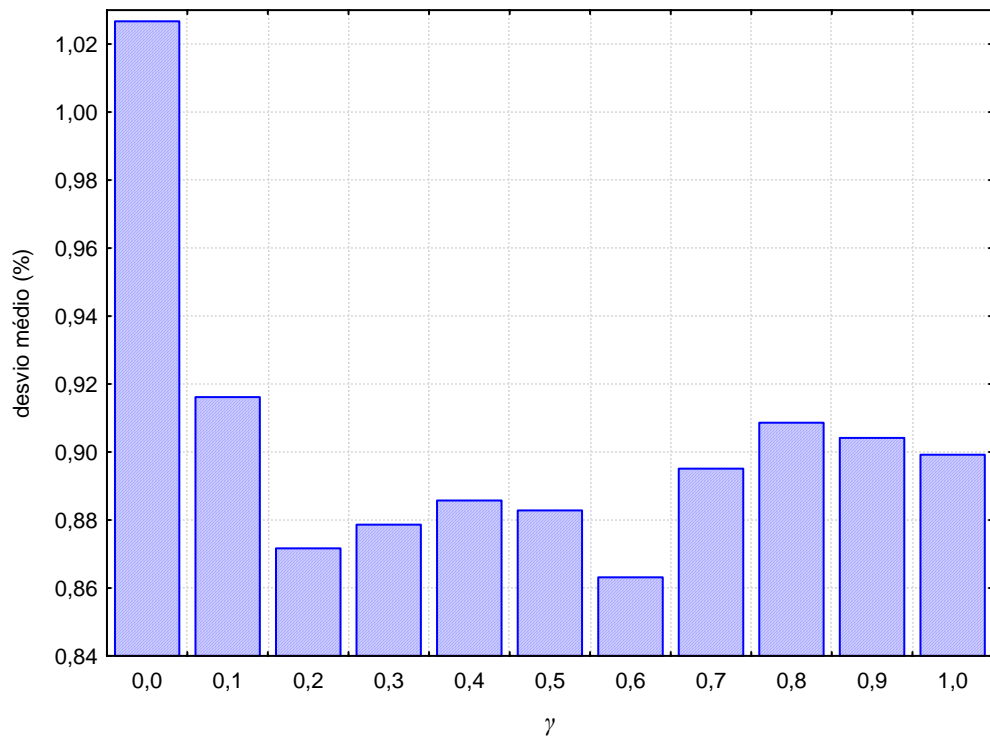


Figura 24 – Influência do parâmetro γ nas instâncias de Montané e Galvão

Tabela 16 – Valores de γ associados às melhores soluções médias nas instâncias de Montané e Galvão

Montané e Galvão	
Problema	γ
r101	0,0;...; 1,0
r201	0,0;...; 0,5 0,8;...; 1,0
c101	0,0;...; 1,0
c201	0,0;...; 1,0
rc101	0,1;...; 1,0
rc201	0,2;...; 1,0
r1_2_1	0,0;...; 0,9
r2_2_1	0,1;...; 0,4 0,6;...; 0,7
c1_2_1	0,4;...; 1,0
c2_2_1	0,0;...; 1,0
rc1_2_1	0,0;...; 1,0
rc2_2_1	0,0;...; 0,5 0,8;...; 1,0
r1_4_1	0,3;...; 1,0
r2_4_1	0,0;...; 0,7
c1_4_1	0,5; ...; 1,0
c2_4_1	0,0;...; 1,0
rc1_4_1	0,6;...; 1,0
rc2_4_1	0,0;...; 0,4 0,6; 0,7; 1,0
Interseção: \emptyset	

Por fim, na instância de Min (1989), não houve necessidade de se elaborar um gráfico para avaliar a influência do parâmetro γ , pois a solução média foi igual à solução ótima (Tabela 12) para todos os valores de γ adotados.

6 CONCLUSÕES

Esta dissertação tratou do Problema de Roteamento de Veículos com Coleta e Entrega Simultânea. Para resolvê-lo, um algoritmo baseado na metaheurística *Iterated Local Search*, que usa uma abordagem inspirada no procedimento VND na fase de busca local, foi proposto. É a primeira vez que esta metaheurística foi utilizada para solucionar este problema (de conhecimento do autor).

Para construção da solução inicial, fez-se uso de uma heurística de inserção fundamentada no princípio da “inserção viável mais barata” associada a um critério de bonificação de clientes longínquos que, por sua vez, possui um parâmetro $\gamma \in [0,1]$ para mensurar a intensidade deste favorecimento.

Na etapa de busca local, seis estruturas de vizinhança (*Shift(1,0)*, *Crossover*, *Swap(1,0)*, *Shift(2,0)*, *Swap(2,1)*, *Swap(2,2)*) que executam movimentos entre rotas foram empregadas. Outras quatro vizinhanças (*Or-opt*, *2-opt*, *Exchange*, *Reverse*) foram incorporadas para efetuar uma pesquisa de intensificação dentro das rotas. Já a fase de perturbação valeu-se de três mecanismos, a saber: *Ejection Chain*, *Double Swap* e *Double-Bridge*.

O algoritmo desenvolvido foi testado em 101 problemas-testes disponíveis na literatura, sendo 40 gerados por Dethloff (2001), 18 por Montané e Galvão (2006), 1 por Min (1989) e 28 por Salhi e Nagy (1999). Das 28 instâncias geradas por esses últimos autores, 14 foram testadas em dois cenários distintos.

Nas instâncias de Dethloff (2001), o ILS-VND foi capaz de melhorar o resultado de 9 instâncias e igualar à solução das outras 31, tendo um *gap* médio de -0,12% em relação às melhores soluções apontadas pela literatura. Nos problemas-testes propostos por Montané e Galvão (2006), houve melhora em todas as 18 instâncias e o *gap* médio foi de -2,68%. Nos

problemas formulados por Salhi e Nagy (1999) que não envolvem a restrição de tempo limite, verificou-se uma melhora em 3 destes e uma equivalência em outros 2, tendo 0,66% como *gap* médio. Na situação em que se considerou a restrição de tempo máximo e o tempo de serviço (*drop time*), ocorreram uma melhora em 12 problemas e duas igualdades, acarretando em um *gap* medio de -1,18%. No caso em que se desconsiderou o *drop time*, observaram-se 8 melhoras e uma igualdade e o *gap* médio foi de -0,54%. Finalmente, na instância de Min (1989) a solução ótima foi alcançada. Em suma, o algoritmo ILS-VND foi capaz de melhorar o resultado de 50 instâncias e igualar a solução de outras 37. Por outro lado, o mesmo não conseguiu superar a solução de 14 problemas-testes.

Embora o algoritmo desenvolvido neste trabalho apresente diversos componentes estocásticos, e, de certa forma, seja relativamente dependente tanto do conjunto de perturbações utilizadas em cada grupo de instâncias, bem como do parâmetro γ , o mesmo se mostrou bastante competitivo frente aos demais procedimentos propostos na literatura, melhorando cerca de 50% das soluções previamente obtidas.

Ademais, uma nova formulação matemática, baseada em fluxo com duas comodidades, foi proposta para o PRVCES. Esta consiste em uma adaptação do modelo desenvolvido por Baldacci *et al.* (2004) para a versão clássica do PRV.

Como trabalhos futuros sugerem-se: a incorporação de procedimentos mais eficientes para reduzir a dependência do fator γ para geração da solução inicial; averiguar alternativas para diminuir o esforço computacional em algumas vizinhanças de tal forma a não comprometer o desempenho da busca local; implementação de outros mecanismos de perturbação; efetuar hibridizações conjugando métodos exatos e heurísticos; e o desenvolvimento de estratégias paralelas para o algoritmo proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANGELELLI, E.; MANSINI, R. A branch-and-price algorithm for a simultaneous pick-up and delivery problem, Working Paper, Article presented at the **EURO/INFORMS Meeting**, 2003.
- ANILY, S. The vehicle-routing problem with delivery and back-haul options. **Naval Research Logistics**, v. 43, p. 415–434, 1996.
- ANILY, S. MOSHEIOV, G. The traveling salesman problem with delivery and backhauls. **Operations Research Letters**, v. 16, p. 11-18, 1994.
- BALDACCI, R.; HADJICONSTANTINO, E. A.; MINGOZZI, A. An exact algorithm for the traveling salesman problem with deliveries and collections. **Networks**, v. 42, p. 26-41, 2003.
- BALDACCI, R.; HADJICONSTANTINO, E. A.; MINGOZZI, A. An exact algorithm for the capacited vehicle routing problem based on a two-commodity network flow formulation. **Operations Research**, v. 52, n. 5, p. 723-738, 2004.
- BABA, C. M.; CORRÊA, F. H. M.; SOUZA, S. A. C.; WAHBA, T. M.; MEDINA, A. C. Otimização da colônia de formigas aplicada ao problema da programação e roteirização de veículos para o transporte de pessoas portadoras de deficiência. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO - ENEGEP2004, 24, 2006, Florianópolis, **Anais ...** Florianópolis, 2006.
- BIANCHESSI, N.; RIGHINI, G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, **Computers & Operations Research**, v. 34, n. 2, p. 578-594, 2007.
- BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms, **Transportation Science**, v. 39, n. 1, p. 104-118 2005.
- BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics, **Transportation Science**, v. 39, n. 1, p. 119-139 2005.
- BARKER, E. K. Evolution of Microcomputer-Based Vehicle Routing Software: Case Studies

in United States. In: Paolo Toth e Daniele Vigo (eds.), **The Vehicle Routing Problem**, SIAM Monographs on Discrete Mathematics and Applications 9, Philadelphia, p. 353–362, 2002.

BERBEGLIA, G.; CORDEAU, J-F.; GRIBKOVSKAIA, I.; LAPORTE, G. Static Pickup and Delivery Problems: A Classification Scheme and Survey. **Top**, v. 15, p. 1-31, 2007.

BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Computer Surveys**, v. 35, n. 3, p. 268 – 308, 2003.

BODIN, L. D., GOLDEN, B. L. Classification in Vehicle Routing and Scheduling. **Networks**, v. 11, p. 97-108, 1981.

BODIN, L. D.; SEXTON, T. R. The multi-vehicle subscriber dial-a-ride problem, **TIMS Studies in the Management Sciences**, v. 22, p. 73-86, 1986.

BRANDÃO, J. A new tabu search algorithm for the vehicle routing problem with backhauls. **European Journal of Operational Research**, v. 173, p. 540-555, 1996

CASCO, D. O.; GOLDEN, B. L.; WASIL, E. A. **Vehicle routing with backhauls: Models, algorithms, and case studies**. In: Golden BL, Assad AA (eds) *Vehicle routing: Methods and studies*, Elsevier, Amsterdam, p. 127-147, 1988.

CHRISTOFIDES, N; MINGOZZI, A; TOTH, P. The vehicle routing problem. In: N. Christofides, P. Mingozzi, P. Toth e C. Sandi (eds). **Combinatorial Optimization**, p. 315-338. Wiley: Chichester, 1979.

CARICATO, P.; GHIANI, G.; GRIECO, A.; GUERRIERO, E. Parallel tabu search for a pickup and delivery problem under track contention. **Parallel Computing**, v. 29, p. 631-639, 2003.

CARRETO, C.; BAKER, B. In C.C. Ribeiro e P. Hansen, (eds). **Essays and surveys in metaheuristics**, p.185-200. Kluwer Academic Publishers, 2002.

CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points, **Operations Research**, v. 12, p. 568-581, 1964.

CHEN, J. F. Approaches for the Vehicle Routing Problem with Simultaneous Deliveries and Pickups. **Journal of the Chinese Institute of Industrial Engineers**, v. 23, n. 2, p. 141-150, 2006.

CHEN, J. F.; WU, T. H., Vehicle routing problem with simultaneous deliveries and pickups, **Journal of the Operational Research Society**, v. 57, n. 5, p. 579-587, 2006.

CORDEAU, J-F.; LAPORTE, G. The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. **4OR - Quaterly Journal of the Belgian, French and Italian Operations Research Societies**, v. 1, p. 89-101, 2003a.

CORDEAU, J-F.; LAPORTE, G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. **Transportation Research Part B**, v. 37. p. 579-594, 2003b.

CORDEAU, J-F.; GENDREAU, M.; HERTZ, A.; LAPORTE, G.; SORMANY, J-S. New Heuristics for the Vehicle Routing Problem. **Les Cahiers du GERAD G-2004-33**, Montréal, Canada, 2004.

CORDEAU, J-F.; LAPORTE, G.; SAVELSBERGH, M. W. P.; VIGO, D. Vehicle Routing. Disponível em <<http://neumann.hec.ca/chairedistributique/common/vrp.pdf>>, acesso em 23/01/2007, 2005.

CORDEAU, J-F. A branch-and-cut algorithm for the dial-a-ride problem, **Operations Research**, v. 54. p. 573-586, 2006.

CRISPIM, J.; BRANDÃO, J. Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls, **Journal of the Operational Research Society**, v. 56, n. 7, p. 1296–1302, 2005.

DANTZIG, G. B.; RAMSER, J. H. The Truck Dispatching Problem. **Management Science**, v. 6, p. 80–91, 1959.

DE BRITO, M. P. **Managing Reverse Logistics or Reversing Logistics Management?**. Holanda, 2002. Tese (Doutorado), Erasmus University Rotterdam, Holanda, 2002.

DELL'AMICO, M.; RIGHINI, G.; SALANIM, M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, **Transportation Science**, v. 40, n. 2, p. 235-247, 2006.

DESROSIERS, J.; DUMAS, Y.; SOUMIS, F.; TAILLEFER, S.; VILLENEUVE, D. An algorithm for mini-clustering in handicapped transport, **Les Cahiers du GERAD**, G-91-02, Montréal, Group for Research in Decision Analysis, 1991.

DETHLOFF, J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up, **OR Spektrum**, v. 23, p. 79-96, 2001.

DRÉO, J.; SIARRY, P.; PÉTROWSKI, A.; TAILLARD, E. **Metaheuristics for hard-optimization: Methods and Case Studies**. Springer-Verlag, Berlin, 2006.

DUCHENNE, É.; MALCA, F.; SEMET F. Real-time management of a fleet of trucks : A case study in France. **Workshop on solving rich vehicle routing problems**, Molde, Norvège, juin, 2005.

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, p. 49-59, 1997

DULLAERT, W.; SEVAUX, M.; SÖRENSEN, K.; SPRINGAEL, J. Applications of metaheuristics. **European Journal of Operational Research**, v. 179, n. 3, p. 601-604, 2007.

DUMAS, Y.; DESROSIERS, J.; SOUMIS, F. Large scale multi-vehicle dial-a-ride problems, **Les Cahiers du GERAD**, G-89-30, Montréal, 1989.

DUMAS, Y.; DESROSIERS, J.; SOUMIS, F. The pickup and delivery problem with time windows. **European Journal of Operational Research**, v. 54, p. 7-22, 1991.

FESTA, P.; RESENDE, M.G.C. GRASP: an annotated bibliography. In: C.C. Ribeiro and P. Hansen (eds.), **Essays and Surveys in Metaheuristics**. Kluwer Academic Publishers, p. 325-367, 2002.

GEHRING, H, HOMBERGER, J. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: Miettinen K, Mäkelä M, Toivanen J, editors. **Proceedings of EUROGEN99**, v. A2(S), Springer, Berlin, p. 57-64, 1999.

GENDREAU, M.; LAPORTE, G.; SÉGUIN, R. Stochastic vehicle routing. **European Journal of Operational Research**, v. 88, 1996.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. The Traveling Salesman Problem with Backhauls. **Computers & Operations Research**, v. 23, n. 5, 1996.

GENDREAU, M.; LAPORTE, G.; VIGO, D. Heuristics for the traveling salesman problem with pickup and delivery. **Computers & Operations Research**, v. 26, 699-714, 1999.

GENDREAU, M.; LAPORTE, G.; POTVIN, J.-Y. Metaheuristics for the Vehicle Routing Problem. **Les Cahiers du GERAD G-98-52**, Montréal, Canada, 1999.

GHAZIRI, H.; OSMAN, I. H. A neural network algorithm for the traveling salesman problem with backhauls. **Computers and Industrial Engineering**, v. 44, p. 267–281, 2003.

GLOVER, F. Future paths in Integer Programming and links to Artificial Intelligence. **Computers & Operations Research**, v. 13, n.5, p. 533-549, 1986.

GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**, Kluwer Academic Publishers, Norwell, MA, 2002.

GOETSCHALCKX, M.; JACOBS-BLECHA, C. The vehicle routing problem with backhauls, **European Journal of Operational Research**, v. 42, p. 39-51, 1989.

GÖKÇE, E. I. **A revised ant colony system approach to vehicle routing problems**, Dissertação de Mestrado, Graduate School of Engineering and Natural Sciences, Sabanci University, Turquia, 2004.

GOLDEN B. L.; BAKER, E.K.; ALFARO, J. L.; SCHAFFER, J. R. The vehicle routing problem with backhauling: two approaches. **Proceedings of the Twenty-First Annual Meeting of the S. E. TMS**, Myrtle Beach, SC, USA, 1985.

GOLDEN, B.L.; ASSAD, A. A. **Vehicle Routing: Methods and Studies**, North-Holland, Amsterdam, 1988.

GOLDEN, B. L.; WASIL, E. A.; KELLY, J. P.; CHAO, I-M. Metaheuristics in vehicle routing. In: T.G Crainic e G. Laporte (eds.), **Fleet Management and Logistics**, Kluwer, Boston, p. 33–56, 1998.

GOLDEN, B.L.; ASSAD, A. A.; WASIL, E. A. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Paolo Toth e Daniele Vigo (eds.), **The Vehicle Routing Problem**, SIAM Monographs on Discrete

Mathematics and Applications 9, Philadelphia, p. 245–286, 2002.

GRIBKOVSKAIA, I.; HALSKAU, Ø.; LAPORTE, G.; VLCEK, M. General solutions to the single vehicle routing problem with pickups and deliveries, **European Journal of Operational Research**, v. 180, p. 568-584, 2006.

IOACHIM, I.; DESROSIERS, J.; DUMAS, Y.; SOLOMON, M. M.; VILLENEUVE, D. A request clustering algorithm for door-to-door handicapped transportation, **Transportation Science**, v. 29, p. 63-78, 1995.

JAW, J. J.; ODoni, A. R.; PSARAFTIS, H. N.; WILSON, N. H. M. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, **Transportation Research Part B**, v. 20, p. 243-257, 1986.

LANDRIEU, A.; MATI, Y.; BINDER, Z. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. **Journal of Intelligent Manufacturing**, v. 12, p. 497–508, 2001.

LAPORTE, G. What You Should Know about the Vehicle Routing Problem. **Les Cahiers du GERAD**, G-2007-59, Montréal, Canada, 2007.

LAPORTE, G. SEMET, L. Classical Heuristics for the Vehicle Routing Problem. **Les Cahiers du GERAD** G-98-54, Montréal, Canada, 1999.

LENSTRA, J.; RINNOOY KAN, A. Complexity of vehicle routing and scheduling problems. **Networks**, v. 11, n. 2, p. 221-227, 1981.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated Local Search. In: Fred Glover e Gary A. Kochenberger (eds.), **Handbook of Metaheuristics**, p. 321-353. Kluwer Academic Publishers, Norwell, MA, 2002.

MARTIN, O.; OTTO, S. W.; FELTEN, E. W. Large-step Markov chains for the traveling salesman problem. **Complex Systems**, v. 5, p. 299-326, 1991.

MAURI, G. R.; LORENA, L. A. N. **Uma nova abordagem para o problema dial-a-ride**. Disponível em <<http://www.lac.inpe.br/~lorena/mauri/mauri-lorena-producao2006.pdf>>, acesso em 02/02/2007.

MICHALEWICZ, Z.; FOGEL, D. B. **How to Solve It: Modern Heuristics**. 2 ed. Springer, Berlin, 2004.

MIN, H. The multiple vehicle routing problem with simultaneous delivery and pick-up points, **Transportation Research**, v. 23, n. 5, p. 377-386, 1989.

MINGOZZI, A.; GIORGI, S.; BALDACCI, R. An exact method for the vehicle routing problem with backhauls. **Transportation Science**, v. 33, p. 315-329, 1999.

MLADENOVIC, N.; HANSEN, P. Variable Neighborhood Search. **Computers & Operations Research**, v. 24, n. 11, p. 1097-1100, 1997.

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, p. 595-619, 2006.

MOSHEIOV, G. The traveling salesman problem with pick-up and delivery. **European Journal of Operational Research**, v. 79, p. 299-310, 1994

MOSHEIOV, G. Vehicle routing with pick-up and delivery: tour partitioning heuristics. **Computers and Industrial Engineering**, v. 34, p. 669-684, 1998.

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, p. 126-141, 2005.

NANRY, W. P.; BARNES, J. W. Solving the pickup and delivery problem with time windows using reactive tabu search. **Transportation Research Part B**, v. 34, p. 107-121, 2000.

OLIVEIRA, R. M.; ARENALES, M. N. A Técnica de Geração de Colunas Aplicada a Problemas de Roteamento. V SIMPÓSIO DE TESES E DISSERTAÇÕES - USP, 2002, São Carlos. **Anais ...** São Carlos: USP, 2002.

OR, I. **Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking**. Tese de doutorado, Northwestern University, USA, 1976.

OSMAN, I. H. Focused issue on applied meta-heuristics. **Computers and Industrial Engineering**, v. 44, p. 205-207, 2002.

OSMAN, I. H.; WASSAN, N. A. A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. **Journal of Scheduling**, v. 5, p. 263-285, 2002.

PSARAFTIS, H. N. A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. **Transportation Science**, v. 14, p. 130-154, 1980.

PSARAFTIS, H. N. An exact algorithm for the single-vehicle, many-to-many dial-a-ride problem with time windows, **Transportation Science**, v. 17, 351-357, 1983.

REGO, C.; ROUCAIROL, C.; A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem. I. Osman and J. Kelly (eds), **Meta-heuristics Theory and Applications**, Kluwer, Dordrecht, p. 253–295, 1996.

REIMANN, M.; ULRICH, H. Comparing backhauling strategies in vehicle routing using Ant Colony Optimization. **Central European Journal of Operations Research**, v. 14, n. 2, p. 105-123, 2006.

ROGERS, D. S.; TIBBEN-LEMBKE, R. S. Going Backwards: reverse logistics trends and practices. **Reverse Logistics Executive Council**, Pittsburgh, PA, 1999.

ROPKE, S.; PISINGER, D. **A unified heuristic for a large class of vehicle routing problems with backhauls**. Technical Report n. 2004/14, University of Copenhagen, Denmark, 2004.

ROPKE, S.; CORDEAU, J-F.; LAPORTE, G. **Models and a branch-and-cut algorithm for pickup and delivery problems with time windows**. Disponível em <<http://www.hec.ca/chairedistributique/common/pdptw.pdf>>, acesso em 03/02/2007.

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, p. 1034-1042, 1999.

SAVELSBERGH, M. W. P. SOL, M. The General Pickup and Delivery Problem. **Transportation Science**, v. 29, p. 17-29, 1995.

SEXTON, T. R.; BODIN, L. D. Optimizing single vehicle many-to-many dial-a-ride problem with desired delivery times: I. Scheduling, **Transportation Science**, v. 19, p. 378-410, 1985a.

SEXTON, T. R.; BODIN, L. D. Optimizing single vehicle many-to-many dial-a-ride problem with desired delivery times: II. Routing, **Transportation Science**, v. 19, p. 411-435, 1985b.

SOLOMON, M. M. **Algorithms for the vehicle routing and scheduling problems with time window constraints**. Operations Research, v. 35, p. 254-65, 1987.

STÜTZLE, T. G. **Local Search Algorithms for Combinatorial Problems**. Tese (doutorado) Technische Universität Darmstadt, Alemanha, 1998.

THANGIAH, S. R.; POTVIN, J.-Y.; SUN, T. Heuristic approaches to vehicle routing with backhauls and time windows. **Computers & Operations Research**, v. 23, p.1043-1057, 1996

TOTH, P.; VIGO, D. Fast local search algorithms for the handicapped persons transportation problem, in I. Osman and J. Kelly (eds), **Meta-heuristics Theory and Applications**, Kluwer, Boston, p. 677-690, 1996.

TOTH, P.; VIGO, D. Heuristic algorithms for the handicapped persons transportation problem, **Transportation Science**, v. 31, p. 60-71, 1997.

TOTH, P.; VIGO, D. An Exact Algorithm for the Vehicle Routing Problem with Backhauls. **Transportation Science**, v. 31, n. 4, p.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, p. 528-543, 1999

TOTH, P.; VIGO, D. VRP with backhauls. In: Paolo Toth e Daniele Vigo (eds.), **The Vehicle Routing Problem**, SIAM Monographs on Discrete Mathematics and Applications 9, Philadelphia, p. 195–224, 2002.

TOTH, P.; VIGO, D. **Vehicle Routing Problem**, SIAM Monographs on Discrete Mathematics and Applications 9, Philadelphia, 2002.

VAN der BRUGGEN, L. J. J.; LENSTRA, J. K., SCHUUR, P. C. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. **Transportation Science**, v. 27, p.298–311, 1993.

VURAL, A. V. **A GA based meta-heuristic for capacited vehicle routing problem with simultaneous pick-up and deliveries**, Dissertação de Mestrado, Graduate School of Engineering and Natural Sciences, Sabanci University, Turquia, 2003.

WASSAN, N. A.; WASSAN, A. H.; NAGY, G.: A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. **Journal of Combinatorial Optimization**, 2007.

ZHONG, Y.; COLE, M. H. A vehicle routing problem with backhauls and time windows: a guided local search solution. **Transportation Research Part E**, v. 41, p. 131-144, 2005.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)