

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

**ALGORITMO GENÉTICO APLICADO AO PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS**

**Autor:**

---

Dan Abensur Gandelman

**Orientador:**

---

Sergio Palma da Justa Medeiros

**Examinador:**

---

Flávio Luis de Mello

**Examinador:**

---

Aloysio de Castro Pinto Pedroza

**DEL**  
**Dezembro de 2007**

## **AGRADECIMENTO**

Agradeço ao Povo Brasileiro por patrocinar meus estudos através da Universidade Federal do Rio de Janeiro.

## **RESUMO**

Neste trabalho é apresentada uma heurística usando o conceito de Algoritmos Genéticos, mais especificamente a meta-heurística de Busca Dispersa (BD), para a solução do Problema de Roteamento de Veículos (PRV) Clássico. Experimentos computacionais foram realizados em quatro conjuntos de dados disponíveis na literatura. Os resultados mostram que o algoritmo proposto é robusto e competitivo em termos de qualidade das soluções obtidas e tempo computacional para o PRV Clássico, para os conjuntos de dados testados.

## **PALAVRAS-CHAVE**

Algoritmos Genéticos

Problema de Roteamento de Veículos

Meta-Heurística

Busca Dispersa

## Sumário

1	<b>INTRODUÇÃO</b>	7
1.1	Tema e Delimitação	7
1.2	Justificativa e Objetivos	7
1.3	Metodologia	8
1.4	Descrição	9
2	<b>FUNDAMENTAÇÃO TEÓRICA DO PRV</b>	10
2.1	Introdução	12
2.2	Variantes do PRV	12
2.2.1	Problema de Roteamento com Restrições de Capacidade	12
2.2.2	Problema de Roteamento de Veículos com restrições de capacidade e tempo máximo	12
2.2.2	Problema de Roteamento de Veículos com Janelas de Tempo	12
2.2.4	Problema de Roteamento de Veículos com Múltiplos Depósitos	13
2.2.5	Problema de Roteamento de Veículos Periódico	13
2.2.6	Problema de Roteamento de Veículos com Entrega Particionada	14
2.2.7	Problema de Roteamento de Veículos com Cargas de Retorno	14
2.2.8	Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega	14
2.2.9	Problema Estocástico de Roteamento de Veículos	15
2.2.10	Problema Dinâmico de Roteamento de Veículos	15
3	<b>METODOLOGIA UTILIZADA</b>	16
4	<b>O USO DE META-HEURISTICAS PARA RESOLVER O PRV</b>	18
4.1	Simulated Annealing	18
4.2	Busca Tabu	19
4.2	Algoritmos Genéticos	20
4.2	Colônia de Formigas	21
4.2	GRASP	23
5	<b>ALGORITMOS GENÉTICOS APLICADO AO PRV CLÁSSICO</b>	24
5.1	Etapa 1: Geração de Soluções Diversas	24
5.1.1	Algoritmo com base na heurística de Gillett e Miller	25
5.1.2	Algoritmo com base na heurística de Beasley	26
5.1.3	Comparação Computacional	26
5.2	Etapa 2: Melhoria das Soluções	27
5.2.1	Movimentos inter-rotas	27
5.2.2	Melhoria intra-rotas	27
5.3	Etapa 3: Construção de Refset	28
5.4	Etapa 4: Geração de Sub-conjuntos	28
5.5	Etapa 5: Combinação de Soluções	28
5.6	Atualização do Conjunto de Referência	30
6	<b>RESULTADOS COMPUTACIONAIS</b>	31
7	<b>CONCLUSÃO</b>	33
	<b>REFERÊNCIAS</b>	34
	<b>ANEXO</b>	41

## ÍNDICE DE FIGURAS

## Capítulo 1: Introdução

### 1.1 Tema e Delimitação

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

O PRV clássico (com restrições de capacidade e distância máxima) consiste em designar um conjunto de  $m$  rotas de entrega e/ou coleta tal que: i) o custo total do conjunto de rotas percorrido pela frota seja minimizado; ii) cada rota se inicie e finalize no depósito; iii) cada cliente tenha sua demanda suprida exatamente por um veículo; iv) a carga total de cada veículo não exceda sua capacidade  $Q$ ; e v) o tempo total necessário para completar qualquer rota não exceda um limite pré-especificado  $T$ , que inclui tempos de viagem entre clientes e tempos de serviço em cada cliente.

O PRV é um problema NP-difícil (Lenstra & Rinnooy Kan, 1981). Devido à sua complexidade computacional, métodos exatos de solução são inviáveis para instâncias de grande porte, o que nos remete à utilização de métodos heurísticos/meta-heurísticos. Recentemente os Algoritmos Genéticos, uma metodologia com base em populações, tem demonstrado ser muito efetiva na solução de problemas de otimização discreta.

Uma variação dos Algoritmos Genéticos, a Busca Dispersa (BD), proporciona um marco flexível que permite o desenvolvimento de diferentes algoritmos com distintos graus de complexidade para as diferentes etapas da busca. Os conceitos e princípios fundamentais do método foram propostos na década de 70 por Glover (1977), com base em estratégias para combinar regras de decisão e restrições. Os trabalhos de Laguna e Martí (2003) e Martí *et al.* (2006) são referências importantes sobre a BD. A Busca Dispersa tem por base o princípio de que, dadas duas soluções, pode-se obter novas soluções mediante a combinação delas, de modo a melhorar as que a originaram. Neste trabalho utilizaremos os conceitos da Busca Dispersa para solucionar o PRV Clássico

### 1.2 Justificativa e Objetivos

O foco deste trabalho é a utilização de Algoritmos Genéticos para resolver o Problema de Roteamento de Veículos (PRV). Existem diversas maneiras de resolver o PRV. Os métodos exatos, embora garantam a solução ótima, são demasiadamente demorados, inviabilizando sua utilização para a maioria das aplicações. Heurísticas são

utilizadas para achar rapidamente boas soluções, mesmo sem dar garantia de achar a solução ótima. O objetivo do trabalho é implementar um Algoritmo Genético que resolva instâncias clássicas do PRV em tempo razoável.

O Problema de Roteamento de Veículos (PRV) com restrições de capacidade e tempo máximo de rotas consiste em definir rotas de entrega de custo mínimo para uma frota de veículos, com origem e término em um depósito central. A frota deve servir um conjunto de clientes, cada qual caracterizado por uma demanda e tempo de serviço e por sua localização no espaço bidimensional. Cada cliente tem sua demanda suprida exatamente por um veículo. A carga total de qualquer veículo não deve exceder a capacidade do mesmo e o tempo total necessário para completar qualquer rota não pode exceder um limite pré-especificado, que inclui tempos de viagem entre clientes e tempos de serviço em cada cliente.

Devido à complexidade computacional dos problemas de roteamento, métodos exatos de solução são inviáveis para instâncias de grande porte, o que nos remete à utilização de métodos heurísticos. Recentemente a *Busca Dispersa* (BD), uma nova metodologia com base Algoritmos Genéticos, tem demonstrado que é muito efetiva na solução de problemas de otimização discreta.

### 1.3 Metodologia

Algoritmo Genético (AG) é um método evolutivo (baseado em populações) que é muito efetivo na solução de diversos problemas de otimização discreta. Tem por base combinar soluções pertencentes a um conjunto denominado *conjunto de referência* (*Refset*), com o intuito de capturar informação não contida nas soluções originais. O *Refset* guarda boas soluções encontradas durante o processo de busca. O AG, baseado na meta-heurística de Busca Dispersa, é composto basicamente por 5 etapas (métodos), descritos a seguir de forma resumida.

1. *Etapa Geradora de Soluções Iniciais*: esta etapa gera um conjunto P com *Psize* soluções, de onde extraímos um subconjunto menor que se denomina conjunto de referência *Refset*.

2. *Etapa de Melhoria*: tipicamente consiste de uma busca local para melhorar as *Psize* soluções inicialmente encontradas. Se a solução que está sendo avaliada é *inviável*, o método tenta transformá-la em uma solução viável. Se a solução é *viável*, o método tenta melhorá-la.

3. *Geração do Conjunto de Referência*: nesta etapa é construído ou atualizado o conjunto de referência *Refset*.

3.1 *Construção* - A partir do conjunto P se extrai *Refset*, combinando soluções de alta qualidade com soluções com diversidade; as soluções de *Refset* são usadas, durante o processo de busca, para gerar novas soluções mediante a combinação de soluções contidas nesse conjunto.

3.2 *Atualização* - A atualização de *Refset* acontece quando novas soluções que atendem os requisitos para ingressar em *Refset* são encontradas. Assim *Refset* mantém seu tamanho  $b=b_1+b_2$  constante, mas as soluções pertencentes ao mesmo vão melhorando durante o processo de busca.

Para atualizar *Refset* tem-se que considerar o *momento* em que a atualização deve ser realizada (estática ou dinâmica). A atualização se diz *estática* quando a mesma é realizada após testadas todas as possíveis combinações em *Refset*; se diz *dinâmica* quando imediatamente após alguma combinação a solução resultante é tentativamente inserida em *Refset*, sempre que atenda os requisitos necessários para ingressar no conjunto.

4. *Geração de Sub-conjuntos*: opera no *conjunto de referência*, consistindo em criar diferentes subconjuntos de soluções que serão utilizadas na etapa de combinação.

5. *Combinação de Soluções*: utiliza os subconjuntos de soluções gerados na Etapa 4, combinando as soluções em cada subconjunto com o objetivo de encontrar novas soluções. Esta etapa de combinação é um mecanismo específico para cada problema, uma vez que está diretamente relacionado com a representação da solução.

## 1.4 Descrição

No Capítulo 2 será apresentada uma descrição detalhada sobre o Problema de Roteamento de Veículos. No Capítulo 3 será abordada a metodologia de Algoritmos Genéticos e Busca Dispersa. No Capítulo 4 será feita uma breve análise das diversas meta-heurísticas existentes e sua bibliografia voltada ao PRV. No Capítulo 5 será apresentado o algoritmo proposto para resolver o PRV Clássico. No Capítulo 6 são apresentados os resultados computacionais e no Capítulo 7 é apresentada a conclusão.

## Capítulo 2: Fundamentação Teórica do PRV

### 2.1 Introdução

O Problema de Roteamento de Veículos (PRV) tem sido estudado desde a década de 60 na área de transportes. O PRV é um nome genérico que se dá a uma classe de problemas nos quais se tem que determinar um número de percursos (rotas para uma frota de veículos) que partem de um ou mais pontos (depósitos) para satisfazer a demanda de clientes geograficamente dispersos. O objetivo é atender eficientemente a demanda dos clientes minimizando o custo total. Em sistemas de distribuição em contexto real o conjunto de restrições pode complicar o modelo. A base para o estudo do Problema de Roteamento de Veículos é o Problema do Caixeiro Viajante (PCV), no qual um agente visita um conjunto de cidades e retorna à cidade inicial. O objetivo do PCV é minimizar a distancia percorrida pelo agente.

A importância do PRV reflete-se na grande variedade de aplicações correspondentes nas áreas de pesquisa operacional, logística, distribuição, transporte, por exemplo: sistemas de coleta e/ou entrega de mercadoria, roteamento de ônibus urbanos, de veículos automatizados, sistema de coleta de correspondência em caixas de correio, sistemas disque-transporte, coleta de lixo, entrega de periódicos a assinantes, entre outros. Recentemente, versões dinâmicas do PRV (*The Dynamic Vehicle Routing Problem*) têm recebido muita atenção por parte de pesquisadores.

Nesses problemas nem toda a informação relevante é conhecida no início do processo de roteamento, apenas durante o processo. Entre alguns exemplos de aplicação das versões dinâmicas do PRV temos o serviço de correio eletrônico, sistemas de disque transporte, serviços de emergência, entre outros.

O PRV é um problema que pertence à classe de problemas NP – difíceis (NP-hard). Isto é, o esforço computacional necessário para encontrar uma solução ótima por meio de um método exato cresce de forma exponencial com o tamanho do problema; por este motivo a maioria dos pesquisadores utilizam os métodos heurísticos (que não garantem encontrar a solução ótima) para achar soluções para o PRV, de modo que soluções suficientemente boas possam ser encontradas em tempo computacional razoável.

No PRV básico as rotas dos veículos se iniciam e finalizam em um depósito central para atender os clientes cuja demanda seja de coleta ou entrega de mercadorias. Cada cliente deve ser visitado exatamente uma vez por um único veículo. A função objetivo pode ser: minimizar o trajeto da frota, o custo dos serviços, o tempo de viagem, o número de veículos ou uma combinação destes, dependendo da aplicação em consideração. Partindo desta definição básica existem diferentes restrições que podem ser associadas ao PRV, dando origem desta maneira a variantes do PRV que recebem nomes específicos na literatura.

Existem importantes componentes dos problemas de roteamento nos sistemas de distribuição, tais como características dos clientes, dos veículos, restrições que podem

ser impostas na construção das rotas e possíveis objetivos para o processo da otimização.

#### Características dos clientes:

- O posicionamento do cliente no plano bidimensional;
- A cada cliente está associada uma demanda que pode ser de coleta e/ou entrega;
- Existe um tempo requerido para a entrega ou a coleta das mercadorias (*tempo de serviço*);
- O cliente pode ser atendido apenas durante um determinado período do dia (*janela de tempo*).

Em algumas situações não é possível satisfazer a demanda de cada cliente. Nestes casos algum subconjunto de clientes não será atendido, sendo associadas aos clientes prioridades ou penalidades.

#### Características dos veículos:

Cada veículo tem capacidade expressa como o máximo peso ou volume a ser transportado:

- Possível subdivisão do veículo em compartimentos, caracterizados pela capacidade e pelos tipos de cargas que podem ser transportadas;
- Estratégias para as operações de entrega e/ou coleta;
- Movimentação em rotas permitidas; por exemplo, veículos com muito peso não podem circular por determinadas vias;
- Custos associados à utilização do veículo (por distância, unidade de tempo, rota, entre outros).

As rotas devem ser construídas de modo a satisfazer as diferentes restrições que dependem da natureza das cargas que são transportadas, do nível de qualidade do serviço e de outras características dos clientes e veículos.

#### Restrições básicas do PRV:

- Durante a rota a carga do veículo não pode exceder a capacidade do mesmo;
- Os clientes a serem atendidos podem ter demanda de coleta, entrega ou ambas;
- Restrições de precedência na ordem em que os clientes serão atendidos podem ser impostas.

### Objetivos que podem ser considerados para os problemas de roteamento:

- Minimizar o custo total de transporte, função da distância total percorrida, tempo de viagem, e dos custos associados ao uso dos veículos;
- Minimizar o número de veículos que atendem os clientes;
- Minimizar as penalidades associadas ao atendimento parcial dos clientes;
- Uma combinação destes objetivos.

## **2.2 Variantes do PRV**

Nesta seção apresentamos dez variantes do PRV, com algumas referências bibliográficas de cada variante. Cabe ressaltar que para cada uma das variantes do PRV, que estão na literatura, pequenas modificações originam outros problemas de roteamento; isto é, partindo da definição básica existem diferentes restrições que são associadas ao PRV, originando desta maneira variantes do PRV que recebem nomes específicos na literatura.

### **2.2.1 Problema de Roteamento com Restrições de Capacidade – (PRVC)**

Em função da definição genérica do PRV dada acima, no PRVC os veículos iniciam e finalizam o trajeto das rotas em um depósito central; todos os clientes estão associados a demandas de entrega. Impõe-se a restrição adicional que a soma total das demandas dos clientes pertencentes a uma rota não deve exceder a capacidade do veículo. O objetivo pode ser minimizar o custo total da distancia percorrida. Entre os trabalhos encontrados na literatura para algoritmos exatos e aproximados temos: CHRISTOFIDES *et al.* (1979), CHRISTOFIDES (1985), LAPORTE e NORBERT (1987), FISHER (1995), LAPORTE e OSMAN (1995), TOTH e VIGO (2002), MARTINHON *et al.* (2004), TARANTILIS (2005), TARANTILIS *et al.* (2005).

### **2.2.2 Problema de Roteamento de Veículos com restrições de capacidade e tempo máximo de rotas - (PRVCTMR)**

No PRV com restrições de capacidade e tempo máximo de rotas (limite de distância) é incluída a restrição adicional que a distância total (tempo total) a ser percorrida por cada veículo não pode exceder um limite pré especificado que inclui as distâncias (tempos) de viagem entre clientes e o tempo de serviço em cada cliente. O objetivo deste problema pode ser: minimizar o comprimento do trajeto total, o tempo de viagem, o número de veículos. Entre os trabalhos que se destacam na literatura temos: OSMAN (1993), ROCHAT e TAILLARD (1995), XU e KELLY (1996), TOTH e VIGO (2002), BALDACCI *et al.* (2004), PRINS (2004), TARANTILIS (2005), TARANTILIS *et al.* (2005).

### **2.2.3 Problema de Roteamento de Veículos com Janelas de Tempo – (PRVJT)**

É um PRV com a restrição adicional que associa uma janela de tempo a cada cliente. Na definição clássica do problema se o veículo chega antes do instante de início da janela deve esperar até o momento estabelecido para iniciar o atendimento do cliente; se chegar no intervalo da janela de tempo, atende a demanda do cliente no momento da chegada; se chegar após o fechamento da janela o cliente pode ficar sem ser atendido. Existem dois tipos de janelas de tempo: (i) *janelas de tempo rígidas*, em que as restrições de janelas de tempo não podem ser violadas; e (ii) *janelas de tempo flexíveis*, quando é permitido o serviço adiantado ou atrasado em relação à janela de tempo, mas se isto acontecer é adicionada uma penalidade ao valor da função objetivo. O objetivo pode ser de minimizar o tamanho da frota de veículos, o tempo total de viagem ou o tempo total de espera para atender os clientes.

Entre os trabalhos encontrados na literatura temos: LENSTRA e RINNOOY (1981), SOLOMON (1987), POTVIN e ROUSSEAU (1993), THOMPSON e PSARAFTIS (1993), RUSSELL (1995), BRAMEL e SIMCHI-LEVI (1996), TAILLARD *et al.* (1997), GAMBARDELLA *et al.* (1999), HOMBERGER e GEHRING (1999), IOANNOU *et al.* (2001), BERGER *et al.* (2003), LI *et al.* (2003), LAU *et al.* (2003), BRÄYSY *et al.* (2004) e RUSSELL e CHIANG (2006).

### **2.2.4 Problema de Roteamento de Veículos com Múltiplos Depósitos - (PRVMD)**

Uma empresa pode possuir vários depósitos que atendem a demanda de seus clientes. Se os clientes estão agrupados ao redor dos depósitos, cada problema pode ser tratado de forma independente. O problema consiste em designar clientes aos depósitos e achar as rotas correspondentes. Cada depósito tem uma frota de veículos associada ao mesmo. Cada veículo parte de um depósito, serve os clientes designados a esse depósito e depois retorna ao mesmo. O objetivo do problema é servir a todos os clientes minimizando o número de veículos e a distância percorrida por eles. A aplicação do PRVMD é encontrada em diferentes contextos, como por exemplo, entrega de refeições, de produtos químicos, de produtos de petróleo, entre outros. Algoritmos exatos e heurísticos podem ser encontrados em RAFT (1965), TILLMAN (1969), TILLMAN e HERING (1971), TILLMAN e CAIN (1972), GOLDEN *et al.* (1977), LAPORTE *et al.* (1984), LAPORTE *et al.* (1988), CHAO *et al.* (1993), RENAUD *et al.* (1996) e finalmente NAGY e SALHI (2005).

### **2.2.5 Problema de Roteamento de Veículos Periódico - (PRVP)**

No PRV, o período de planejamento é um dia. No caso do PRVP o período de planejamento se estende a M dias. Um veículo pode não retornar ao depósito no mesmo dia de sua partida. Durante o período de M dias cada cliente deve ser visitado ao menos uma vez. O objetivo pode ser minimizar a frota de veículos ou o tempo total da viagem.

Entre os trabalhos encontrados na literatura temos: BELTRAMI e BODIN (1974), RUSSELL e IGO (1979), CHRISTOFIDES e BEASLEY (1984), TAN e BEASLEY (1984), RUSSELL e GRIBBIN (1991), CHÃO *et al.* (1995) e BAPTISTA *et al.* (2002)

### **2.2.6 Problema de Roteamento de Veículos com Entrega Particionada - (PRVEP)**

Nesta variante do PRV permite-se que o cliente possa ser atendido por vários veículos desde que o custo total seja reduzido por esse tipo de atendimento. O objetivo pode ser minimizar a frota de veículos ou o tempo total de viagem. Entre os trabalhos encontrados na literatura temos: DROR e TRUDEAU (1989,1990), FRIZZELL e GIFFIN (1992, 1995), DROR *et al.* (1994), MULLASERIL *et al.* (1997), BELENGUER *et al.* (2000). HO e HAUGLAND (2004) apresentam o PRV de entrega particionada com janelas de tempo.

### **2.2.7 Problema de Roteamento de Veículos com Cargas de Retorno – (PRVCR)**

Neste tipo de problema os clientes são divididos em dois subconjuntos; clientes de entrega e clientes de coleta. Os clientes de coleta somente são visitados depois que todos os clientes de entrega tiverem sido atendidos. Entre as características relevantes do PRVCR temos que durante cada percurso a carga do veículo é de apenas de um tipo, de entrega ou de coleta. O objetivo é encontrar um conjunto de rotas que minimizem a distancia total percorrida. Este problema é estudado, entre outros, por DEIF e BODIN (1984), GOLDEN *et al.* (1985), CASCO *et al.* (1988), GOETSCHALCKX e JACOBS-BLECHA (1989), GÉLINAS *et al.* (1995), DUHAMEL *et al.* (1997), TOTH e VIGO (1997,1999), MINGOZZI e GIORGI (1999), WADE e SALHI (2002) e ZHONG e COLE (2005).

### **2.2.8 Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega - (PRVCE)**

É uma variante do PRVCR na qual a demanda de cada cliente é composta de pedidos de entrega e/ou de coleta. Consideram-se situações em que as entregas se iniciam a partir de um depósito e a coleta é conduzida até o mesmo depósito no final do trajeto. Uma característica importante deste problema é que a carga do veículo em qualquer rota é uma mistura de pedidos de entrega e coleta. O objetivo pode ser minimizar a frota de veículos ou o tempo total de atendimento, com a restrição de que o veículo deve ter suficiente capacidade para transportar os pedidos de entrega e coleta ao longo da rota. Entre os trabalhos encontrados na literatura temos: MIN (1989), GALVÃO e GUIMARÃES (1990), MOSHEIOV (1998), SALHI e NAGY (1999), DETHLOFF (2001), MONTANÉ (2004), NAGY e SALHI (2005), MONTANÉ e GALVÃO (2006).

### 2.2.9 Problema Estocástico de Roteamento de Veículos - (PERV)

Refere-se a problemas em que um ou várias componentes dos dados são aleatórios. Por exemplo:

- Demandas estocásticas: a demanda do cliente  $i$  não é predefinida, , mas é definida por uma determinada distribuição de probabilidade.
- Tempos estocásticos: o tempo de serviço e os tempos de viagem são variáveis aleatórias

Entre os trabalhos encontrados na literatura temos os de STEWART e GOLDEN (1983), DROR e TRUDEAU (1986), DROR (1993), GENDREAU *et al.* (1996), SECOMANDI (2000), entre outros.

### 2.2.10 Problema Dinâmico de Roteamento de Veículos – (PDRV)

Problemas de decisão em tempo real têm um papel muito importante na economia. PSARAFTIS (1995) é o primeiro em introduzir o problema dinâmico de roteamento de veículos, classificando diferentes PRVs em estáticos ou dinâmicos. Para PSARAFTIS um problema de roteamento é *estático* se a solução de determinada formulação é um conjunto de rotas pré-planejadas que não serão re-otimizadas, ou seja, cujas operações são calculadas pela entrada de dados que não evoluem em tempo real. Um problema é dinâmico se a saída é um conjunto de rotas que forma parte de um plano de ação, que pode ser modificado em função dos dados de entrada que evoluem em tempo real. Problemas de roteamento dinâmico têm surgido como uma nova área de pesquisa.

### Capítulo 3: Metodologia Utilizada

A Busca Dispersa (*Scatter Search* em inglês) é um método evolutivo (baseado em populações) que é muito efetivo na solução de diversos problemas de otimização discreta. Tem por base combinar soluções pertencentes a um conjunto denominado *conjunto de referência (Refset)*, com o intuito de capturar informação não contida nas soluções originais. O *Refset* guarda “boas” soluções encontradas durante o processo de busca. Cabe destacar que o significado de “boa” não se restringe apenas à qualidade da solução, mas também a sua *diversidade* em relação a outras soluções deste conjunto. A BD é composta basicamente por 5 etapas (métodos), que descrevemos a seguir de forma resumida. Posteriormente explicaremos detalhadamente cada etapa no contexto de uma meta-heurística desenvolvida visando a solução do PRV Clássico. Para uma melhor explicação apresentamos inicialmente a nomenclatura que será utilizada neste trabalho.

$P$	Conjunto de soluções geradas com o método gerador de soluções iniciais;
$PSize$	Tamanho da população $P$ de soluções iniciais (i.e., $ P =Psize$ );
$Refset$	Conjunto de Referência;
$Pool$	Conjunto de soluções geradas pelo método da combinação;
$b$	Tamanho do conjunto de referência (i.e., $ RefSet =b$ );
$b_1$	Tamanho do subconjunto do <i>Refset</i> correspondente a soluções de qualidade;
$b_2$	Tamanho do subconjunto do <i>Refset</i> correspondente a soluções com diversidade;
$x^i$	A $i^{ésima}$ solução do conjunto de referência . $x^1$ é a melhor solução em <i>Refset</i> e $x^b$ a pior;
$d(x,y)$	Distância entre a solução $x$ e a solução $y$ ;

As etapas da BD são as seguintes:

1. *Etapa Geradora de Soluções Iniciais*: esta etapa gera um conjunto  $P$  com  $Psize$  soluções, de onde extraímos um subconjunto menor que se denomina conjunto de referência *Refset*.

2. *Etapa de Melhoria*: tipicamente consiste de uma busca local para melhorar as  $Psize$  soluções inicialmente encontradas. Se a solução que está sendo avaliada é *inviável*, o método tenta transformá-la em uma solução viável. Se a solução é *viável*, o método tenta melhorá-la.

3. *Geração do Conjunto de Referência*: nesta etapa é contruído ou atualizado o conjunto de referência *Refset*.

3.1 *Construção* - A partir do conjunto  $P$  se extrai *Refset*, combinando soluções de alta qualidade com soluções com diversidade; as soluções de *Refset* são usadas, durante o processo de busca, para gerar novas soluções mediante a combinação de soluções contidas nesse conjunto.

3.2 *Atualização*.- A atualização de *Refset* acontece quando novas soluções que atendem os requisitos para ingressar em *Refset* são encontradas. Assim *Refset* mantém seu tamanho  $b=b_1+b_2$  constante, mas as soluções pertencentes ao mesmo vão melhorando durante o processo de busca.

Para atualizar *Refset* tem-se que considerar o *momento* em que a atualização deve ser realizada (estática ou dinâmica). A atualização se diz *estática* quando a mesma é realizada após testadas todas as possíveis combinações em *Refset*; se diz *dinâmica* quando imediatamente após alguma combinação a solução resultante é tentativamente inserida em *Refset*, sempre que atenda os requisitos necessários para ingressar no conjunto.

4. *Geração de Sub-conjuntos*: opera no *conjunto de referência*, consistindo em criar diferentes subconjuntos de soluções que serão utilizadas na etapa de combinação.

5. *Combinação de Soluções*: utiliza os subconjuntos de soluções gerados na Etapa 4, combinando as soluções em cada subconjunto com o objetivo de encontrar novas soluções. Esta etapa de combinação é um mecanismo específico para cada problema, uma vez que está diretamente relacionado com a representação da solução.

O algoritmo da Figura 1 mostra como agem os elementos descritos em um esquema básico da *Busca Dispersa* para um problema de minimização (atualização de *Refset* estática).

---

**Passo 1.** Inicie com  $P \leftarrow \emptyset$ . Inicie a *etapa geradora de soluções iniciais* e construa uma solução  $y$ . Aplique o *método de melhoria* a  $y$  e encontre a solução melhorada  $x$ . Se  $x \notin P$  então, adicione  $x$  a  $P$ , caso contrário descarte  $x$ . Repita este passo até  $|P| = Psize$ .

**Passo 2.** Construa o *conjunto de referência*  $Refset = \{x^1, \dots, x^b\}$  com as  $b_1$  melhores soluções de  $P$  e as  $b_2$  soluções de  $P$  com maior diversidade em relação às soluções que estão em *Refset*. Ordene essas  $b=b_1+b_2$  soluções de forma crescente em relação à função objetivo. Faça  $newsolutions \leftarrow TRUE$ .

**Enquanto** ( $newsolutions = TRUE$ .)

$newsolutions \leftarrow FALSE$ .

**Passo 3.** Mediante a *geração de subconjuntos* gere subconjuntos a partir de *RefSet*.

**Enquanto** (existem subconjuntos sem examinar)

**Passo 4.** Aplique o *método de combinação* às soluções do subconjunto e guarde as soluções obtidas em uma lista denominada *Pool*.

**Passo 5.** Aplique o *método de melhoria* a cada solução de *Pool* obtida no Passo 4.

**Fim Enquanto**

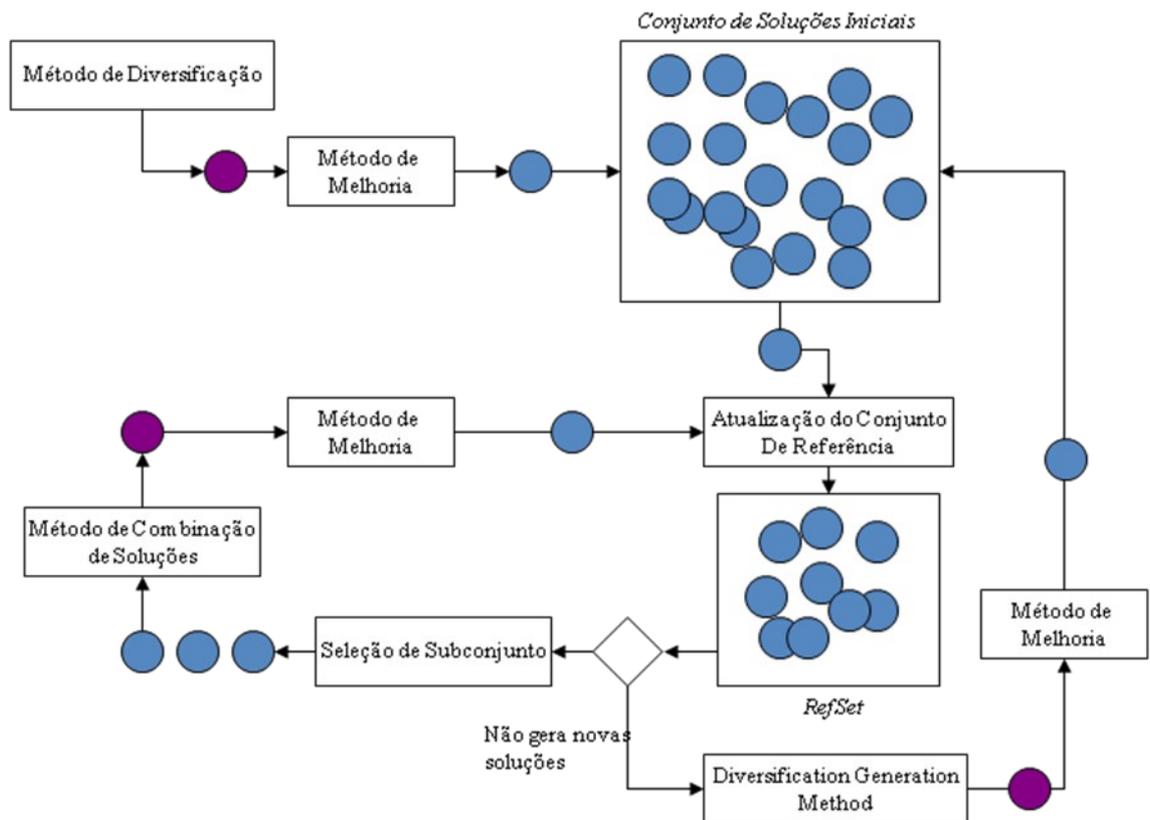
**Passo 6.** Caso possível atualize *Refset* ( $b$  melhores soluções em  $Refset \cup Pool$ ); Reordene *Refset*.

Se *Refset* atualizado,  $newsolutions \leftarrow TRUE$ .

**Fim Enquanto**

---

**Figura 1a** - Esquema Básico de um Algoritmo de Busca Dispersa.



**Figura 1b** - Esquema Básico de um Algoritmo de Busca Dispersa.

É interessante notar que o algoritmo básico de BD termina quando não existem novos elementos a serem combinados em *Refset*.

#### Capítulo 4: O Uso de Meta-Heurísticas para resolver o PRV

Uma metaheurística é um processo heurístico de alto nível para problemas de otimização combinatória. As metaheurísticas possuem a habilidade de sair de ótimos locais ao aceitar movimentos que não melhoram (ou mesmo pioram) o valor da função objetivo. Cada metaheurística tem um ou mais parâmetros que têm que ser ajustados. Isto permite flexibilidade, mas para cada aplicação específica de problemas requer calibração.

Varias metaheurísticas foram propostas para resolver PRVs. Entre as metaheurísticas que têm sido utilizadas para resolver PRVs temos: Simulated Annealing (SA), Busca Tabu (BT), Algoritmos Genéticos (AG), Colônia de Formigas, Redes neurais e algoritmos de memória adaptativa. Estas são explicadas de forma sucinta a seguir.

##### 4.1 Simulated Annealing

KIRKPATRICK, GELATT e VECCHI (1983) propõem um procedimento para obter soluções aproximadas a problemas de otimização, chamado de *Simulated Annealing* (SA). Este procedimento é baseado na analogia com o comportamento de um sistema físico no qual certo material é submetido ao resfriamento a partir de

determinadas temperaturas. É um procedimento de busca onde todo movimento de melhoria é aceito e também são permitidos movimentos que pioram o valor da função objetivo de acordo com determinada probabilidade. Essas probabilidades estão baseadas na analogia com o processo físico de esfriamento que se obtém como função da temperatura do sistema. A estratégia de SA começa com uma temperatura inicial alta, que proporciona uma alta probabilidade de se aceitar um movimento que piora a solução corrente. Em cada iteração se reduz a temperatura e portanto as probabilidades de aceitação de um movimento que piora a solução corrente são cada vez menores conforme avança o procedimento. Desta maneira, inicialmente realiza-se uma diversificação da busca sem controlar demasiadamente o custo das soluções visitadas. Em iterações posteriores resulta cada vez mais difícil aceitar movimentos de piora da solução corrente.

Desta forma, SA tem a habilidade de sair de ótimos locais ao aceitar movimentos que não melhoram a solução corrente nos estados intermédios. O algoritmo termina quando não há movimentos de melhoria.

A chave de uma implementação de SA é a maneira como é conduzida a seqüência do esfriamento (qual é a temperatura inicial e como diminui a temperatura em cada iteração). As recentes implementações de SA tendem a esquecer da analogia física e conduzir o esfriamento como uma estrutura de memória. Isto é, a probabilidade de aceitar ou rejeitar um movimento que não melhora a solução corrente depende não da iteração (tempo transcorrido), mas do que aconteceu durante a busca. O trabalho de OSMAN (1993) é uma referência de Simulated Annealing para resolver PRVs.

## 4.2 Busca Tabu

A metaheurística *Busca Tabu* (BT) é uma técnica que segue os princípios gerais de Inteligência Artificial (IA) e é utilizada para guiar o procedimento de busca local na busca do ótimo do problema. BT toma da IA o conceito de memória e a implementa mediante estruturas simples com o objetivo de conduzir a busca considerando o histórico da mesma. O procedimento trata de extrair informação do passado recente e atuar em função dessa informação. Neste sentido pode-se dizer que existe um certo aprendizado e que a busca é inteligente.

Busca Tabu tem suas raízes na década de 60, mas a forma como hoje é apresentada foi proposta por (GLOVER, 1986). BT é uma abordagem plenamente aceita para o tratamento de problemas de otimização combinatória e tem se mostrado como uma técnica muito promissora através de inúmeras aplicações bem sucedidas.

O método é uma técnica iterativa que explora o conjunto de soluções de um problema, movimentando-se de uma solução  $S$  para outra solução  $S'$ , numa determinada vizinhança  $N(S)$  de  $S$ . Esses movimentos são executados com o intuito de alcançar eficientemente uma solução qualificada como boa, que pode ser ótima ou próxima da ótima. Ao contrário de um algoritmo simples de descida, BT aceita soluções  $S'$  piores que  $S$ , evitando assim o problema de otimalidade local, fazendo uso de uma estrutura de memória capaz de suportar e até encorajar uma busca não monotonicamente decrescente (em problemas de minimização). BT percorre o espaço de busca, analisando os movimentos através de regras determinísticas. Quando BT aceita movimentos que pioram o valor da função objetivo, podem ocorrer ciclos, isto é, o retorno a soluções já

visitadas. Para evitar que ciclagens se estabeleçam, BT armazena os atributos das soluções já pesquisadas em uma lista chamada lista tabu. Os atributos armazenados são muitas vezes usados para impor condições, chamadas restrições tabu, que impedem a escolha de movimentos indesejáveis. Um movimento é chamado tabu se conduz a soluções cujos atributos estão contidos na lista tabu ou satisfazem restrições tabu. Após um determinado número de iterações a lista tabu é alterada.

Uma estrutura de geração de vizinhança de boa qualidade é fundamental para o sucesso da técnica. Se um movimento provoca uma variação muito pequena na função objetivo, é razoável supor que a busca por boas trajetórias para escapar de um mínimo local será problemática. Por outro lado, se essa amplitude é grande, certamente haverá dificuldades para encontrar um mínimo local cuja qualidade seja próxima do mínimo global.

Existem alguns refinamentos que podem ser implementados no procedimento padrão de BT de forma a torná-lo mais eficiente. Esses refinamentos tentam introduzir mais inteligência no processo de busca. Um deles se refere ao tamanho da lista tabu, que varia de acordo com o problema em estudo. O tamanho da lista depende também das restrições tabu empregadas. Em geral, restrições fortes implicam em listas pequenas. Uma maneira de identificar-se o tamanho da lista apropriada é simplesmente observar a ocorrência de ciclos (listas muito pequenas) e a deterioração da qualidade da solução (listas muito grandes). A melhor lista certamente estará entre esses extremos. Outro aprimoramento que se relaciona com o tamanho da vizinhança é a construção de uma lista de candidatos. Um exame completo de vizinhança geralmente proporciona soluções de boa qualidade, porém pode consumir muito tempo computacional. Por essa razão é importante a utilização de estratégias que isolam regiões da vizinhança contendo movimentos desejáveis e os colocam em uma lista de candidatos. Existem estudos sobre técnicas de decomposição de vizinhança, construção de listas de movimentos promissores, listas dos atributos preferidos e muitos outros.

Uma implementação bastante atraente consiste na variação da penalidade imposta à função objetivo para destruir a estrutura de otimalidade local da qual se deseja escapar. A variação da penalidade é exemplo de um procedimento conhecido como *oscilação estratégica*, que representa uma das abordagens básicas de diversificação da busca tabu. A oscilação pode cruzar os limites e penetrar em regiões de inviabilidade ou somente se aproximar e recuar, sem transpor esses limites. Uma maneira simples de se implementar essa penalidade é alternar uma série de movimentos que melhoram a função objetivo com outra série de movimentos que a pioram.

Existem outros aprimoramentos da BT que podem ser encontrados com mais detalhe em GLOVER e LAGUNA (1997). Trabalhos que usam a Busca Tabu para resolver PRVS temos OSMAN (1993), TAILLARD (1993), GENDREAU *et al.* (1994), XU e KELLY (1996), REGO e ROUCAIROI (1996), REGO (1998), BARBAROSOGLU e ÖGÜR (1999) e CORDEAU *et al.* (2001) entre outros.

### 4.3 Algoritmos Genéticos

A metaheurística de *Algoritmos Genéticos* (AG) foi introduzida por HOLLAND (1975) inspirado no processo observado da evolução natural dos seres vivos, tentando imitar os mecanismos de reprodução existentes na natureza. Os AG estabelecem uma

analogia entre o conjunto de soluções de um problema e o conjunto de indivíduos de uma população, codificando a informação de cada solução num vetor chamado de cromossomo. Para isso se introduz uma função de avaliação dos cromossomos, isto é a função objetivo (*função aptidão*). Igualmente se introduz um mecanismo de seleção de tal modo que os cromossomos com melhor avaliação sejam escolhidos para que se reproduzam.

Quando se trabalha com os AG tem-se que levar em consideração a necessidade de se integrar e implementar eficientemente duas idéias fundamentais: as representações simples como vetores das soluções do problema e a realização de transformações simples para modificar e melhorar essas soluções. Para implementar os AG tem-se que considerar vários elementos principais tais como: a representação do cromossomo, uma população inicial, a função que avalia o cromossomo, os critérios de seleção e eliminação do cromossomo, uma ou varias operações de recombinação (*cruzamento*), uma ou varias operações de *mutação*.

Em geral a população inicial é gerada aleatoriamente. No entanto, ultimamente se estão utilizando métodos heurísticos para gerar soluções iniciais com boa qualidade. Neste caso é importante garantir a diversidade estrutural das soluções para evitar a convergência prematura. Em relação à avaliação dos cromossomos em geral utiliza-se o valor da função objetivo do problema. A seleção dos pais habitualmente é dada mediante probabilidades segundo sua aptidão. Os operadores de cruzamento mais utilizados são: (i) *de um ponto*, no qual escolhe-se aleatoriamente um ponto de ruptura nos pais e seus bits são intercambiados; (ii) *de dois pontos*, no qual são escolhidos aleatoriamente dois pontos de ruptura para realizar o intercambio de bits; (iii) *uniforme* no qual para cada bit escolhe-se aleatoriamente um pai para que ceda seu bit ao filho, enquanto que o segundo filho recebe o bit do outro pai.

A operação da mutação mais simples e uma das mais utilizadas consiste em substituir com certa probabilidade o valor de um bit. O papel que cumpre a mutação é de introduzir um fator de diversificação, pois em determinadas ocasiões a convergência do procedimento a boas soluções pode ser prematura e ficar preso em ótimo locais. Outra forma de introduzir novos elementos numa população é combinar elementos tomados aleatoriamente sem considerar a função aptidão.

Incorporando algumas outras técnicas de busca local no esquema dos AG, surgem os algoritmos denominados Algoritmos Meméticos, que são um caso particular sob o nome de cooperação e competição. Em metaheurísticas com base em populações para resolver PRVs, temos BAKER e AYECHEW (2003), PRINS (2004), BERGER e BARKAOUI (2004), MESTER e BRÄYSY (2004);

#### **4.4 Colônia de Formigas**

A metaheurística Colônia de Formigas (CF) foi introduzido por DORIGO et al. (1996). CF é inspirada no comportamento que rege as formigas de diversas espécies para encontrar os caminhos mais curtos entre a fonte da comida como o formigueiro. As formigas são insetos que vivem em colônias e que, devido à sua colaboração mutua, são capazes de mostrar comportamentos complexos e realizar tarefas difíceis desde o ponto de vista de uma formiga individual. Um aspecto interessante do comportamento é a sua habilidade para encontrar os caminhos mais curtos entre o formigueiro e as fontes

de alimento. Este fato é especialmente interessante quando se considera que muitas espécies de formigas são cegas, o que evita o uso de pistas visuais.

Enquanto que as formigas se movimentam entre o formigueiro e a fonte de alimento, algumas espécies de formigas depositam uma substância química chamada *feromônio* (substância que se pode cheirar). Se não se encontra nenhum vestígio de feromônio, as formigas se movimentam de maneira basicamente aleatória, mas quando existe o feromônio elas tendem a seguir o rastro. Experimentos realizados por biólogos tem demonstrado que formigas preferem de maneira probabilística os caminhos marcados com uma concentração superior de feromônio. Na prática, a escolha entre distintos caminhos toma lugar quando vários caminhos se cruzam. Então, as formigas escolhem o caminho com uma decisão probabilística sesgada pela quantidade de feromônio; isto é quanto mais forte é o rastro de feromônio maior a probabilidade de ser escolhido. Como as formigas depositam feromônio no caminho que percorre, este comportamento leva a um processo de auto-reforço que conclui com uma concentração elevada de feromônio. Este comportamento permite que as formigas encontrem os caminhos mais curtos entre o formigueiro e a fonte de alimento.

Nos algoritmos de CF, formigas artificiais, pertencentes a uma colônia de tamanho finito, procuram coletivamente por soluções de boa qualidade. Cada formiga constrói uma solução começando de um estado inicial, selecionado de acordo com o problema de otimização. Enquanto a formiga constrói a solução vai colecionando informações sobre as características do problema e sobre seu próprio desempenho; e usa esta informação para modificar a representação do problema, quando é visto pelas outras formigas. As formigas podem agir coletivamente ou independentemente, mostrando desta maneira um comportamento cooperativo.

De acordo com a noção de vizinhança, cada formiga constrói uma solução realizando determinados movimentos através de uma seqüência finita de estados vizinhos. Os movimentos são selecionados de acordo a busca local que usa (i) a informação do estado interno da formiga (a memória) e (ii) a trilha de feromônio para a informação local. O estado interno armazena a informação sobre a história passada podendo evitar soluções inviáveis. No estado local, onde a formiga se encontra, ela dispõe de informação e conhecimento que estão codificados em trilhas de feromônio acumulado por todas as formigas desde o início do processo de busca. Este conhecimento influencia nas decisões tomadas pelas formigas. As decisões sobre quando as formigas devem liberar feromônio no “ambiente” e quanto feromônio deveria ser depositado dependem das características do problema e da implementação do planejamento (algoritmo). As formigas podem liberar feromônio enquanto constroem a solução, depois de construir a solução ou de ambos os modos. Quanto mais um movimento é escolhido pelas formigas, mais feromônio é adicionado nesse movimento e mais interessante ele se torna para a formiga seguinte. Em geral, a quantidade de feromônio depositada pela formiga é proporcional à “qualidade” da solução que ela construiu ou está construindo.

A união da informação heurística local com a quantidade de feromônio avaliada localmente define uma *tabela de decisão da formiga*, isto é, uma tabela probabilística usada pela formiga para tomar a decisão de ir a regiões mais interessantes do espaço de busca. A componente estocástica da política de decisão e o mecanismo de evaporação de feromônio evitam um rápido desvio de todas as formigas para a mesma parte do

espaço de busca. Uma vez que a formiga completou sua tarefa, consistindo na construção da solução e no depósito de feromônio, ela “morre”, isto é, é eliminada do sistema. Os algoritmos de CF, como consequência de sua natureza adaptativa e colaborativa, são particularmente apropriados para problemas estocásticos, onde a presença de origens exógenas determina a não-estacionaridade na representação do problema. Exemplos deste tipo são problemas relacionados com redes de comunicação ou transporte, distribuídos de modo intrínseco e não-estacionário.

Existem numerosas implementações, com sucesso, aplicadas a diferentes problemas de otimização combinatória. Através dessas implementações é possível distinguir entre duas classes de aplicações: aquela cujos problemas de otimização combinatória são estáticos e aquela em que eles são dinâmicos. Problemas estáticos são aqueles nos quais as características do problema são dadas quando o problema é definido e não mudam enquanto o problema está sendo resolvido. Ao contrário, problemas dinâmicos são definidos como uma função de algumas quantidades que variam de acordo com um sistema dinâmico. Nesse caso o problema muda durante o tempo de execução do algoritmo de otimização. Neste caso, o algoritmo deve ser capaz de se adaptar a essas mudanças ambientais. Como referência do uso do Algoritmo de Colônia de Formigas para resolver PRVs temos por exemplo BULLNHEIMER *et al* (1997).

#### 4.5 GRASP

A metodologia GRASP será utilizada neste trabalho para a geração de soluções iniciais no processo de busca da metaheurística Busca Dispersa. A metaheurística “*Greedy Randomized Adaptive Search Procedure*”- GRASP foi desenvolvida ao final da década dos 80, ver FEO e RESENDE (1989, 1995). GRASP é um procedimento de múltiplos inícios onde cada passo consiste numa fase de construção e em outra de melhoria. Na fase de construção aplica-se um procedimento heurístico construtivo para obter uma boa solução inicial. Esta solução é melhorada na segunda fase mediante um algoritmo de busca local. A melhor de todas as soluções examinadas é guardada como resultado final.

Na fase de construção se constrói iterativamente uma possível solução, considerando um elemento em cada passo. Em cada iteração a seleção do próximo elemento para ser inserido à solução parcial é determinada por uma função gulosa. Esta função mede o benefício de inserir cada um dos elementos segundo a função objetivo e escolhe a melhor. Esta medida é míope no sentido que não considera o que ocorrerá em iterações posteriores ao realizar uma seleção. Isto é, a avaliação que se realize num determinado elemento na iteração  $j$ , não coincidiria necessariamente com aquela que se tenha na iteração  $j+1$ . GRASP é aleatória porque não seleciona o melhor candidato segundo a função gulosa; com o objetivo de diversificar e não repetir soluções em duas construções diferentes constrói-se uma lista denominada lista restrita de candidatos (LRC) com os melhores candidatos, dos quais um é escolhido aleatoriamente. Dado que a fase inicial não garante a otimalidade local em relação à estrutura da vizinhança na qual se está trabalhando, aplica-se um procedimento de busca local como pós-processamento para melhorar a solução obtida.

## Capítulo 5: Algoritmos Genéticos Aplicado ao PRV Clássico

Nesta seção são apresentadas as diferentes etapas de uma aplicação da variação de Algoritmos Genético, conhecido como Busca Dispersa, ao PRV clássico. Para a *etapa geradora de soluções* foram testadas duas heurísticas construtivas, apresentadas a seguir. São apresentados também três métodos de melhoria inter-rotas e um método de melhoria intra-rota para a *etapa de melhoria* da BD. As etapas de *construção (atualização) de Refset*, *geração de sub-conjuntos* e *combinação de soluções* são finalmente descritos.

A construção de *Refset* se inicia a partir do conjunto  $P$  de  $Psize$  soluções geradas após as duas etapas iniciais da busca. As soluções de *Refset* são ordenadas em ordem crescente do valor da função objetivo, isto é de acordo com sua qualidade. Seleciona-se os pares de soluções que serão combinadas e o processo de combinação de soluções de *Refset* é iniciado. *Refset* é atualizado com novas soluções que forem geradas por esse processo; a atualização de *Refset* pode ser *estática* ou *dinâmica*. O critério de parada para a BD é quando for atingido o valor da solução ótima da instância em consideração (quando disponível na literatura), ou quando *Refset* não for atualizado na última etapa da BD.

### 5.1 Etapa 1: Geração de Soluções Diversas

As heurísticas desenvolvidas para o PRV Clássico podem ser classificadas em três categorias distintas: heurísticas de inserção, heurísticas de duas fases e heurísticas de melhoria. As heurísticas de inserção constroem gradualmente uma solução pela inserção de clientes de acordo com algum critério pré-estabelecido. Nas heurísticas de duas fases o problema é decomposto em duas fases: agrupamento e roteamento. Pode ser feito inicialmente o agrupamento de clientes, seguido pelo seu roteamento; ou as duas fases podem ser invertidas (primeiro o roteamento, seguido do agrupamento). As heurísticas de melhoria podem ser de melhoria inter-rotas ou intra-rotas.

Apresentamos a seguir dois métodos para gerar as  $Psize$  soluções iniciais (população inicial) da BD: o primeiro uma adaptação da Heurística de Varredura de Gillett e Miller (1974); o segundo com base na heurística de Beasley (1983). Os resultados produzidos por esses algoritmos são melhorados por meio de diversos procedimentos simples na etapa seguinte da BD.

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

Esta heurística corresponde a um algoritmo de *agrupamento-roteamento*. Na heurística de varredura com base no algoritmo proposto por Gillett e Miller (1974), os grupamentos são formados girando uma semi-reta com origem no depósito e incorporando os clientes “varridos” por dita semi-reta na mesma rota até que as restrições do PRV Clássico não sejam satisfeitas. Para este algoritmo cada cliente  $i$  está dado por suas coordenadas polares  $(\rho_i, \theta_i)$ , em um sistema que tem o depósito como origem. Uma versão simplificada da heurística é definida pelos passos descritos a seguir.

Passo 1: Ordenar os clientes segundo o ângulo polar  $\theta$  que cada cliente faz com a semi-reta.. Se dois clientes possuem o mesmo valor de  $\theta$ , o cliente com menor valor de  $\rho$  é selecionado. Selecione um cliente  $u_i$  como cliente inicial e faça  $i=1$ ,  $k=1$ ;  $C_1 = \{u_i\}$  é o *cluster de ordem 1*.

Passo 2: Se todos os clientes pertencem a algum *cluster*, ir ao Passo 4. Caso contrário ir ao Passo 3.

Passo 3: Fazer  $i=i+1$  e selecionar o próximo cliente  $u_i$ . Se  $u_i$  puder ser adicionado a  $C_k$ , fazer  $C_k = C_k \cup \{u_i\}$ . Caso contrário, fazer  $k=k+1$  e criar um novo agrupamento  $C_k = \{u_i\}$ . Ir ao Passo 2.

Passo 4: Para cada agrupamento  $C_k$ , resolver um Problema do Caixeiro Viajante.

Uma solução viável do PRV é assim construída. O processo se repete  $n$  vezes (onde  $n$  é o número de clientes,  $n = Psize$ ), iniciando cada execução do algoritmo por um cliente diferente.

### 5.1.2 Algoritmo com base na heurística de Beasley

Heurística proposta por Beasley (1983), cuja estratégia é rotear primeiro, agrupar depois. Sem considerar as restrições do PRV, na fase de roteamento contrói-se inicialmente um “tour gigante” iniciado no depósito, que percorre todos os clientes e retorna ao depósito. A este “tour gigante” aplica-se a heurística de melhoria *2-opt*, na qual duas arestas do tour são trocadas por outras duas não pertencentes ao mesmo, sempre que o resultado da troca diminui o comprimento do “tour” (ver 3.2.2). Na fase de agrupamento as restrições do PRV são consideradas; o “tour gigante” é dividido em um número de rotas viáveis em que os clientes são designados às rotas conforme sua seqüência no mesmo. Uma solução viável do PRV é assim construída.

### 5.1.3 Comparação Computacional

Diversas experiências computacionais foram realizadas a fim de comparar o desempenho dos algoritmos construtivos com base nas heurísticas de Gillett e Miller e Beasley. Experimentos foram realizados em dois conjuntos de dados, formados respectivamente pelas 14 instâncias descritas em Christofides *et al.* (1979), e pelas 27 instâncias descritas em Augerat *et al.* (1995).

A Tabela 1 abaixo mostra os resultados obtidos para as duas heurísticas de construção sem melhoria (linha 1) e depois de aplicar a heurística de melhoria (linha 2) às instâncias. Os procedimentos de melhoria (explicados na seção 3.2) foram propostos por diferentes autores, conforme descrito por exemplo em Toth e Vigo (2002). Todos os dados da Tabela 1 são os Desvios Percentuais Médios em Relação às melhores soluções encontradas na literatura (limite superior ou solução ótima). Observamos na Tabela 1 que ambas heurísticas apresentam desempenhos semelhantes antes e após o procedimento de melhoria. Optamos por trabalhar com o procedimento com base na heurística de Gillett e Miller.

**Tabela 1** - Comparação das heurísticas de Construção.

DPRM	Gillett e Miller	Beasley
Sem melhoria	19,33	19,15
Com melhoria	9,31	9,13

## 5.2 Etapa 2: Melhoria das Soluções

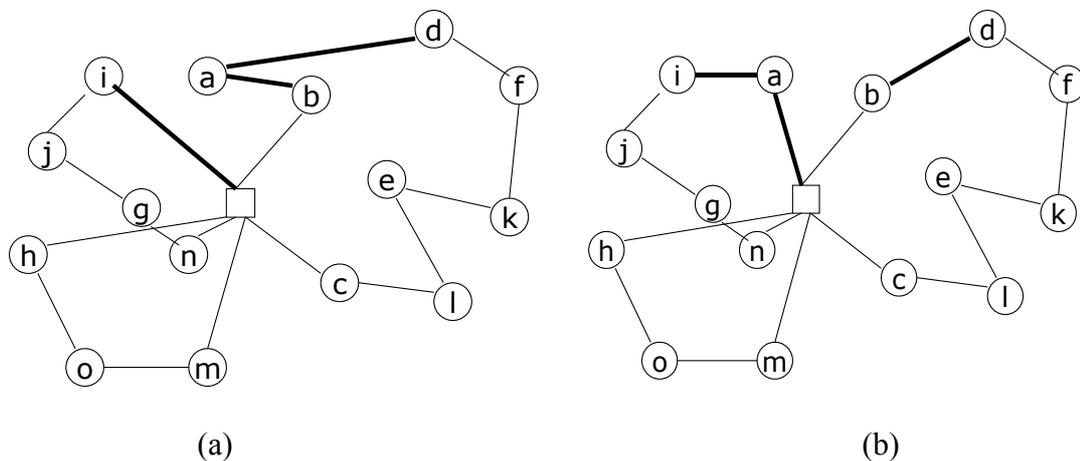
Nesta seção apresentamos as heurísticas de melhoria utilizadas na Fase 2 de nossa heurística de BD para o PRV clássico. Apresentamos três movimentos inter-rotas e um movimento intra-rota.

### 5.2.1 Movimentos inter-rotas

Apresentamos a seguir três tipos de movimentos inter-rotas: re-alocação, intercâmbio e cruzamento.

#### Re-alocação

Consiste em remover um cliente de uma das rotas e inseri-lo em uma outra rota. Nas Figuras 2(a) e 2(b) apresentamos o movimento de re-alocação. Para este exemplo o depósito é denominado por 0 e na figura está representado por um quadrado; nestas figuras o cliente  $a$  é mudado de rota; ou, se desejamos considerar a re-alocação em função de arcos, os arcos  $(i,0)$ ,  $(a,b)$  e  $(a,d)$  são substituídos respectivamente pelos arcos  $(i,a)$ ,  $(a,0)$  e  $(b,d)$ .



**Figura 2** - Movimento de Re-alocação.

#### Intercâmbio

Consiste em intercambiar clientes entre duas rotas. Nas Figuras 3(a) e 3(b) os clientes  $g$  e  $j$  são intercambiados; ou, se desejamos expressar o intercâmbio em função de arcos temos que os arcos  $(i,g)$ ,  $(g,0)$ ,  $(h,j)$  e  $(j,0)$  são substituídos respectivamente pelos arcos  $(i,j)$ ,  $(j,0)$ ,  $(h,g)$  e  $(g,0)$ .

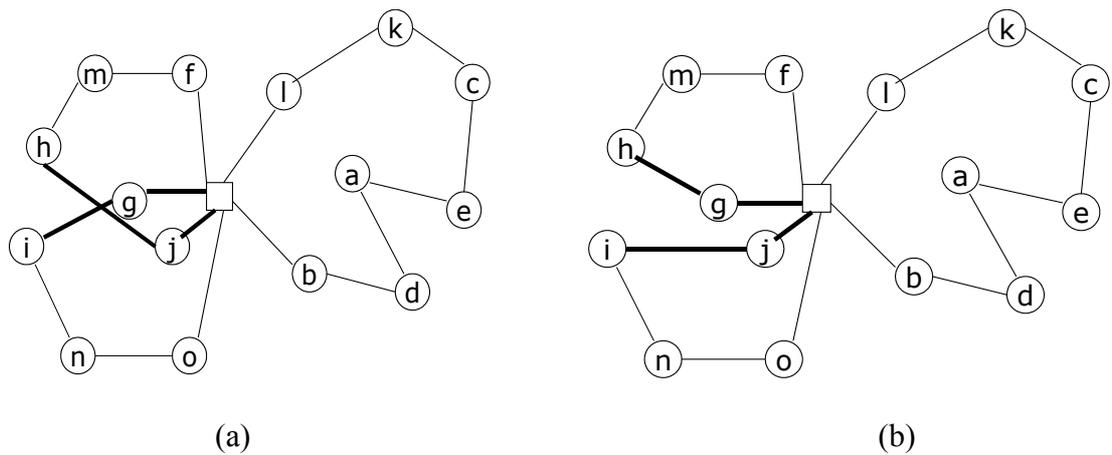


Figura 3 - Movimento de Intercâmbio.

### Cruzamento

Neste movimento seleciona-se um par de rotas. Cada rota é dividida em duas seções que são re-conectadas. Consegue-se isso removendo um arco de cada rota e substituindo-os por dois novos arcos que conectam respectivamente a seção inicial da primeira rota com a seção final da segunda rota e a seção inicial da segunda com a seção final da primeira. Na Figura 4 ilustramos este movimento para um par de rotas; a rota  $p_0-p_{i-1}-p_i-p_0$  é substituída pela rota  $p_0-p_{i-1}-q_j-p_0$  e a rota  $q_0-q_{j-1}-q_j-q_0$  é substituída pela rota  $q_0-q_{j-1}-p_i-q_0$ .

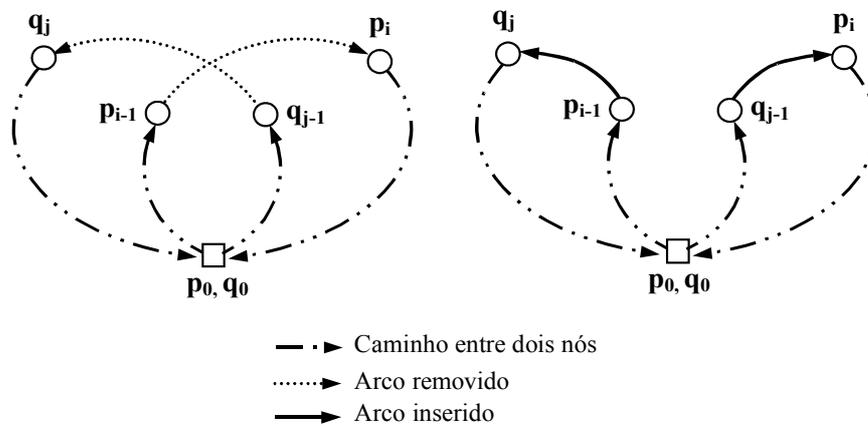


Figura 4 - Movimento de Cruzamento

### 5.2.2 Melhoria intra-rotas

Na literatura são encontradas diferentes formas de realizar o movimento intra-rotas; em nosso algoritmo utilizamos o movimento *2-opt*.

#### Operador $\lambda$ -intercambio ( $\lambda = 2-opt$ )

Para a vizinhança intra-rotas utilizamos a heurística *2-opt*, inicialmente proposta por Lin (1965); os arcos  $(i,o)$  e  $(b,j)$  [Figura 5(a)] são trocados respectivamente pelos arcos  $(i,j)$  e  $(b,o)$  [Figura 5(b)].

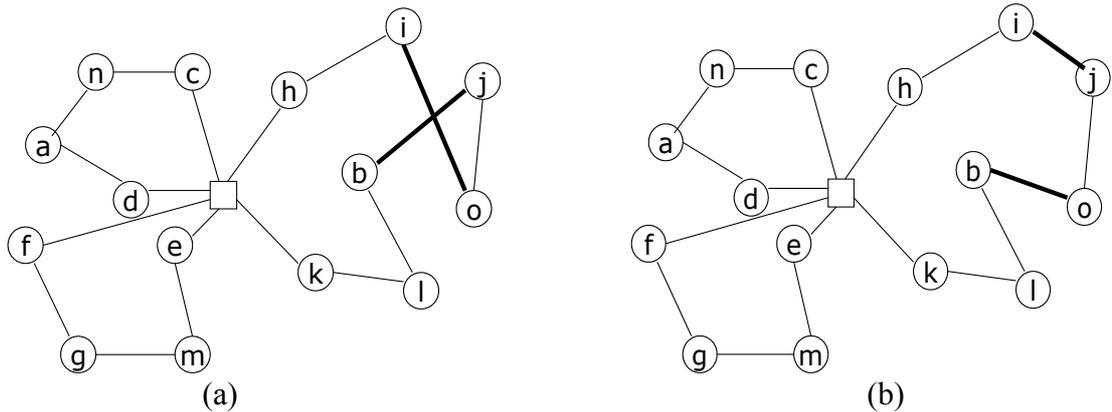


Figura 5 - Movimento 2-opt.

### 5.3 Etapa 3: Construção de *Refset*

As melhores  $b_1$  soluções do conjunto  $P$  são incluídas em *Refset*. Outras  $b_2$  soluções são acrescentadas a *Refset* tal que a diversidade de *Refset* seja a maior possível. É definida uma distância entre duas soluções  $x$  e  $y$ , para determinar a diversidade entre as mesmas:  $d(x, y) = \sum_{k=1}^{k=n} h_k$ , onde  $h_k$  é 0 se o cliente  $k$  pertence à mesma rota nas soluções  $x$  e  $y$ ; é 1 caso contrário. Seleciona-se de  $P$  a solução que maximiza a distância mínima a todas as soluções já incluídas em *Refset*; o processo é repetido até serem inseridas deste modo  $b_2$  soluções nesse conjunto.

### 5.4 Etapa 4: Geração de Sub-conjuntos

Terminada a construção de *Refset*, consideramos a implementação mais simples do *Método de geração de subconjuntos*, que consiste em realizar unicamente combinações de soluções duas a duas. Isto é, para a geração de cada sub-conjunto consideramos pares de soluções de *Refset* que não tenham sido ainda combinadas.

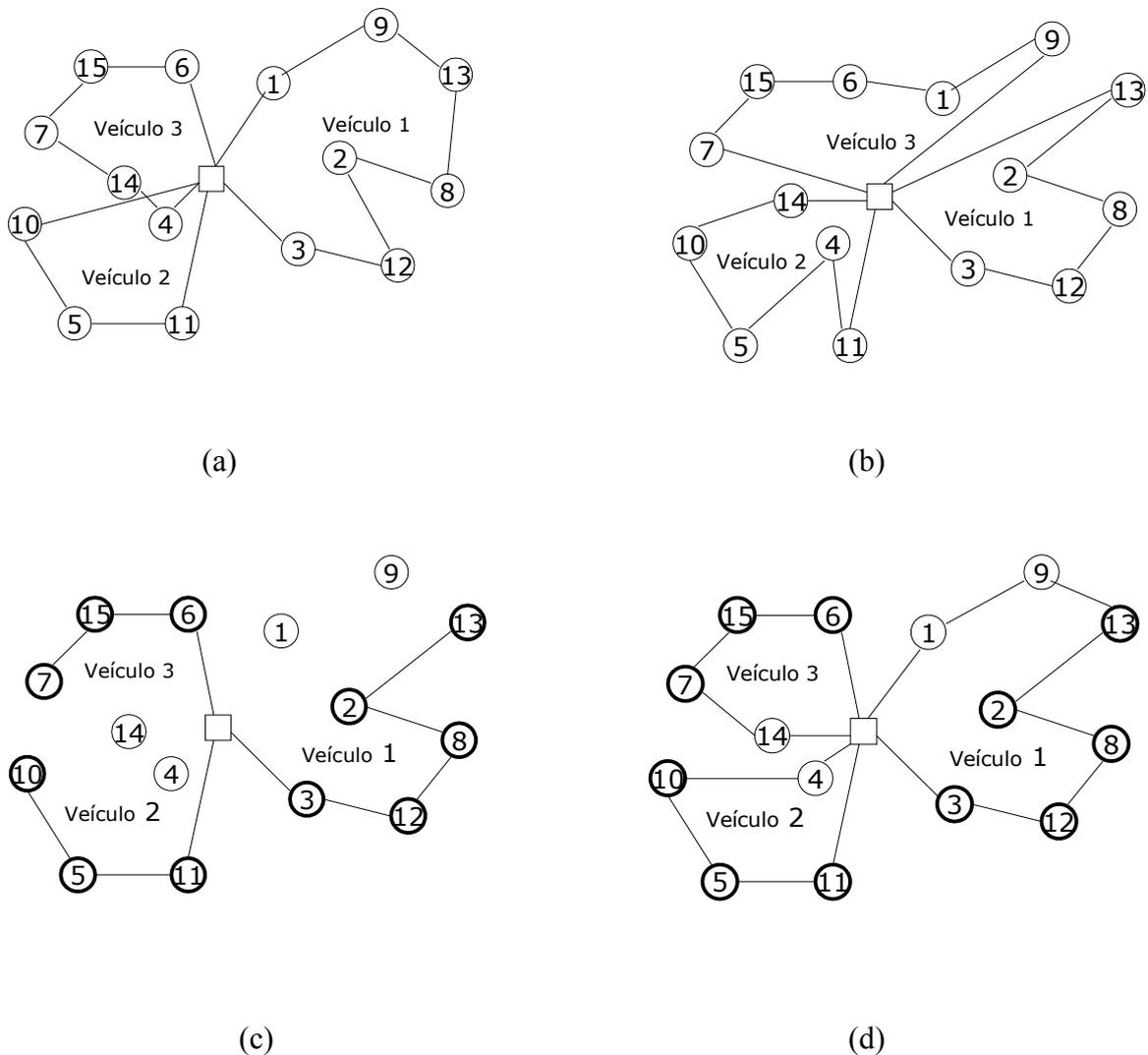
### 5.5 Etapa 5: Combinação de Soluções

Após de selecionar os pares de soluções a serem combinadas aplica-se o *Método de Combinação de Soluções*. Este método funciona da seguinte maneira: i) Identifica-se os clientes que pertencem às mesmas rotas em soluções a serem combinadas. Esses clientes são fixados às rotas; os clientes não fixados são desconectados das mesmas; ii) Constrói-se uma nova solução conforme explicado abaixo e iii) encontrada a nova solução, aplica-se à mesma o *Método de Melhoria*.

Pode acontecer que após de aplicado o método de combinação acima existam clientes desconectados das rotas, em cujo caso é necessária a viabilização da solução correspondente. Para isto é proposta uma heurística que considera os dados de distância e peso relativos do cliente que foi desconectado. Para cada cliente  $i$  (*cliente\_candidato*) não fixado encontra-se a rota mais próxima. Uma rota se diz próxima do *cliente\_candidato* se a distância do último cliente da mesma ao *cliente\_candidato*, mais a distância do *cliente\_candidato* ao depósito, tem o menor valor possível e satisfaz as restrições do PRV Clássico. A soma dessas duas distâncias é dividida pela demanda do

*cliente\_candidato*. O *cliente\_candidato* com o menor valor deste quociente é inserido na rota mais próxima e os dados são atualizados. Quando a inserção de um novo cliente torna a solução inviável, uma nova rota é iniciada. O processo termina quando todos os clientes forem roteados.

Na Figura 6 ilustra-se o método de combinação de um problema com 15 clientes em que as duas soluções a serem combinadas têm três veículos cada. As Figuras 6(a) e 6(b) mostram as duas soluções que serão combinadas. Na Figura 6(c) clientes comuns das soluções 1 e 2 são identificados e fixados nas rotas respectivas; a obtenção da nova solução é feita de modo que sejam formadas rotas segundo o algoritmo descrito acima, conforme mostramos na Figura 6(d).



**Figura 6** - (a) Solução 1; (b) Solução 2; (c) Clientes comuns; (d) Solução resultante.

## 5.6 Atualização do Conjunto de Referência

Se a solução encontrada pelo método de combinação for de qualidade esta pode imediatamente substituir a pior solução em *Refset* (atualização dinâmica) ou ser armazenada em uma lista denominada *Pool* (para que futuramente seja efetuada uma atualização estática). No caso de uma atualização estática *Refset* consistirá das  $b$  melhores soluções das soluções armazenadas em  $Refset \cup Pool$ . A estratégia dinâmica modifica *Refset* rapidamente e usualmente produz soluções em menor tempo computacional, enquanto a estratégia estática usualmente produz melhores soluções em mais tempo computacional.

Se na atualização de *Refset* novas soluções são introduzidas, ativa-se novamente o método da combinação de soluções. O critério de parada para a BD acontece quando

foi atingido o valor ótimo da instância respectiva (quando disponível na literatura), ou quando *Reset* não foi modificado, isto é, não foi atualizado na última etapa da BD.

## Capítulo 6: Resultados Computacionais

Com a finalidade de permitir uma comparação apropriada, cabe esclarecer que a distância entre pares de vértices (clientes) é dada pela distância Euclidiana (sem nenhum arredondamento) para os Conjuntos I e IV, enquanto que para as instâncias dos Conjuntos II e III a distância entre pares de vértices é acrescentado o valor de 0.5, sendo então o valor obtido dessa soma arredondado. Além disso, conforme explicam Martinhon *et al.* (2004), o número de veículos para estas instâncias é um dado de entrada quando os problemas são resolvidos por meio de algoritmos exatos, pode ser uma variável de decisão no caso de heurísticas/meta-heurísticas.

Os códigos para as diferentes implementações da BD foram programados na linguagem C++, em um microcomputador PC com processador Pentium 4 de 2.26 Ghz. Com o objetivo de avaliar o grau de proximidade entre as soluções encontradas nas diferentes implementações da BD em relação às melhores soluções encontradas na literatura, foi definida uma medida de *desvio*. Ela é dada pelo *desvio percentual relativo* (DPR),  $DPR = 100 * ((BD - Fbest) / Fbest)$ , onde *BD* é o valor da função objetivo encontrado pela Busca Dispersa e *Fbest* é a *melhor solução disponível* na literatura para um dado problema-teste.

Na Tabela 2 mostramos os resultados computacionais obtidos pelo algoritmo de Busca Dispersa com diferentes valores de tamanho para *Refset*. A Tabela 2 mostra os resultados para diferentes estratégias na geração de *Refset*, utilizando dois tamanhos para o mesmo: 10 e 20 soluções. Varias combinações para ambos os tamanhos foram testadas. Os valores de  $b_1$  e  $b_2$  representam o número de soluções escolhidas por qualidade e diversidade, respectivamente. Para cada conjunto de instâncias a Tabela 2 mostra o desvio percentual médio (DPRM) em relação às melhores soluções disponíveis na literatura; o tempo computacional (T) da BD em segundos; o tempo necessário para encontrar a melhor solução da BD (TE), em segundos; e o tempo computacional em segundos (TMD) para construir as soluções iniciais da BD.

**Tabela 2 - Resultados Computacionais da Busca Dispersa: Atualização Estática.**

<b>Instâncias</b>		<b>BD 5 5</b>	<b>BD 2 8</b>	<b>BD 3 7</b>	<b>BD 7 3</b>	<b>BD 10 10</b>
Conjunto I (14 inst.)	DPRM	0,86	0,98	1,09	1,08	0,78
	T (s)	127,55	128	124,46	142,91	213,94
	TE(s)	66,61	87,53	88,79	72,20	123,20
	TMD(s)	33,32	32,35	35,30	34,46	47,83
Conjunto II (27 inst.)	DPRM	0,45	0,37	0,37	0,38	0,38
	T (s)	12,93	2,86	2,84	3,09	3,66
	TE(s)	1,96	0,70	0,48	0,60	1,22
	TMD(s)	1,66	0,87	0,87	0,86	0,83
Conjunto III (11 inst.)	DPRM	0,14	0,08	0,03	0,05	0,12
	T (s)	8,68	11,16	7,90	8,95	12,09
	TE(s)	2,88	2,97	1,94	2,59	4,34
	TMD(s)	3,91	5,38	3,85	3,84	4,19
Conjunto IV (3 inst.)	DPRM	0,61	0,05	0,80	0,41	0,35
	T (s)	32,27	21,91	26,76	32,01	42,75
	TE(s)	18,47	12,92	16,87	21,53	32,95
	TMD(s)	7,69	7,50	7,43	8,19	8,11

Foram realizados testes adicionais aumentando o tamanho de *Refset*, mas a melhoria dos resultados é pequena (quando o tamanho da instância é grande) ou não existe (instâncias pequenas), enquanto que o tempo computacional cresce bastante. As soluções obtidas pelo algoritmo da BD em todas as instâncias quando o tamanho de *Refset* é 10 e  $b_1 = 5$  e  $b_2 = 5$  são muito boas em termos de DPRM e tempo computacional, com um desvio percentual menor que 1% em relação às melhores soluções disponíveis na literatura. É interessante notar que razoável porcentagem do tempo computacional, na maioria das instâncias, é consumida ao gerar o conjunto de soluções iniciais (conjunto *P*).

## Capítulo 7: Conclusão

Foi apresentado um algoritmo desenvolvido para o PRV Clássico, testando seu desempenho para quatro conjuntos de dados disponíveis na literatura. Resultados computacionais mostram que as soluções encontradas pelo algoritmo são próximas às melhores soluções reportadas na literatura para essas instâncias, tendo sido obtidas em tempo computacional reduzido.

A técnica apresentada não é a mais bem sucedida para tratar o PRV. A metaheurística de Busca Tabu apresenta melhores resultados em menor tempo computacional. Porém só é possível fazer tal afirmação pela constatação obtida pelos resultados computacionais deste projeto, sendo esta a contribuição do autor.

Para este projeto foram utilizados os conhecimentos adquiridos nos seguintes cursos do Departamento de Engenharia Eletrônica e de Computação:

- Computação I e II
- Linguagens de Computação
- Integração e Interface de Sistemas
- Sistemas Operacionais
- Inteligência Artificial
- Algoritmos e Estruturas de Dados

Também foram utilizados conhecimentos adquiridos em cursos do Programa de Engenharia de Produção da COPPE/UFRJ:

- Meta-Heurísticas e Complexidade de Algoritmos
- Programação Linear e Inteira
- Teoria dos Grafos
- Logística
- Métodos Matemáticos em Logística

Diversas são as possibilidades de pesquisas futuras. AG baseados em Busca Dispersa é uma metaheurística relativamente nova e com poucas aplicações. Fora o PRV Clássico, como apresentado neste projeto, somente a variação com Janela de Tempo foi resolvida. Todas as outras variações do PRV mostradas no Capítulo 2, podem ser resolvidas para verificar a eficiência do método a cada problema. Embora para o PRV Clássico não se tenha obtido os melhores resultados eles não foram insatisfatórios, logo para um outro sub-problema é possível que se obtenha bons resultados.

Melhorias para o problema proposto também são possíveis. Entre elas destacam-se:

- Implementação de Movimento baseado em Busca Tabu, tornando a metaheurística híbrida
- Implementação de outros movimentos como 3-opt e movimentos intra-rotas como inserção e troca

## Referências Bibliográficas

- [1] AUGERAT, J. R., BELENGUER, J. M., BENEVANT, E. et al., *Computational results with a branch and cut code for the capacitated vehicle routing problem*, Technical Report RR940-M, University Joseph Fourier, Grenoble, 1995.
- [2] BALDACCI, R., HADJICONSTANTINO, E. A., e MINGOZZI, A., "An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation, *Operations Research* v. 52, pp. 723-738, 2004.
- [3] BAKER, B.M., AYECHIEW, M.A. "A genetic algorithm for the vehicle routing problem", *Computers & Operations Research* v. 30, pp. 787-800, 2003.
- [4] BARBAROSOGLU, G., ÖGÜR, D. "A tabu search algorithm for the vehicle routing problem", *Computers & Operations Research* v. 26, pp. 255-270, 1999.
- [5] BAPTISTA, S., CARVALHO, R.O., ZÚQUETE, E. "A period vehicle routing case study", *European Journal of Operational Research* v. 139, n.2, pp. 220-229, 2002.
- [6] BEASLEY, J. E. "Route-first cluster-second methods for vehicle routing", *Omega* v. 11, pp.403-408, 1983.
- [7] BEASLEY J. E., CHU, P. C. "Constraint handling in genetic algorithms: the set partitioning problem", *Journal of Heuristics* v. 4, pp. 323-357, 1998.
- [8] BELENGUER, J. M., MARTINEZ, M.C., MOTA, E. "A lower bound for the split delivery vehicle routing problem", *Operations Research* v. 48, No. 5, pp. 801-810, 2000.
- [9] BELTRAMI, E., BODIN, L. "Networks and vehicle routing for municipal waste collection", *Networks* v. 4, pp. 65-94, 1974.
- [10] BERGER, J., BARKAOUI, M., BRÄYSY, O. "A route-directed hybrid genetic approach for the vehicle routing problem with time windows", *Information Systems and Operations Research* v. 41, pp. 179-194, 2003.  
BERGER, J., BARKAOUI, M. "A new hybrid genetic algorithm for the capacitated vehicle routing problem", *Journal of the Operational Research Society* v. 54, pp. 1254-1262, 2004.
- [11] BERTSIMAS, D.J., VAN RYZIN, G. "Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane", *Operations Research* v. 39, pp. 601-615, 1991.
- [12] BRAMEL, J., SIMCHI-LEVI, D. "A location based heuristic for general routing problems", *Operations Research* v. 43, n. 4, pp. 649-660, 1995.
- [13] BRAMEL, J., SIMCHI-LEVI, D. "Probabilistic analysis and practical algorithms for the vehicle routing problem with time windows", *Operations Research* v. 44, pp. 501-509, 1996.
- [14] BRÄYSI, O., HASLE, G., DULLAERT, W. "A multi-star local search algorithm for the vehicle routing problem with time windows", *European Journal of Operational Research* v. 159, pp. 586-605, 2004.
- [15] BULLNHEIMER, B., HARTL, R., STRAUSS, C. "An improved ant system algorithm for the vehicle routing problem", *Annals Operations Research* v. 89, pp. 319-328, 1999.
- [16] CAMPOS V., LAGUNA M., MARTÍ R. "Scatter Search for the Linear Ordering Problem". In: *New Ideas in optimisation*, D. Corne, M. Dorigo e F. Glover (Eds.), McGraw-Hill, pp. 331-341, 1999.
- [17] CAMPOS V., LAGUNA M., MARTÍ R. *Context-Independent Scatter and Tabu Search for Permutation Problems*. In: *Technical report TR03-2001*, Departamento de Estadística e I.O., Universidad de Valencia, 2001.

- [18]CAMPOS V., GLOVER, F., LAGUNA M., MARTÍ R. "An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem", *Journal of Global Optimization* v. 21, pp., 397-414, 2001.
- [19]CASCO, D. O., GOLDEN, B.L., WASIL, E. A. "Vehicle routing with backhauls". In: Golden BL, Assad A.(ed). *Vehicle Routing: Methods and Studies*, North Holland, Amsterdam, pp. 127-147, 1988.
- [20]CLARKE, G., WRIGHT, J.W. "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research* v. 12, pp. 568-581, 1964.
- [21]CHAO, I. M., GOLDEN, B. L., WASIL, E. "A new heuristic for the multi-depot vehicle routing problem that improves upon best-know solutions", *American Journal of Mathematical Management Sciences*. v. 13, n. 3-4,pp. 371-406, 1993.
- [22]CHAO, I. M., GOLDEN, B. L., WASIL, E. "An improved heuristic for the period vehicle routing problem", *Networks* v. 26, pp. 25-44, 1995.
- [23]CHIANG, W.C., RUSSELL, R.A. "Simulated annealing metaheuristics for the vehicle routing problem with time windows", *Annals of Operations Research* v. 63, pp. 3-27, 1996.
- [24]CHRISTOFIDES, N., "Vehicle Routing", In : E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), *The Traveling Salesman Problem*, J. Wiley & Sons, Chichester, 1985.
- [25]CHRISTOFIDES, N., BEASLEY, J.E. "The period vehicle routing", *Networks* v. 14, pp. 237-256, 1984.
- [26]CHRISTOFIDES, N., EILON, S., "An algorithm for the vehicle-dispatching problem", *Operational Research Quarterly* 20, pp. 309-318, 1969.
- [27]CHRISTOFIDES, N., MINGOZZI, A., TOTH, P., "The vehicle routing problem". In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, Wiley, Chichester, UK, pp. 315-338, 1979.
- [28]CORBERÁN, A., FERNÁNDEZ, E., LAGUNA, M., MARTÍ, R. "Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives", *Journal of the Operational Research Society* v. 53, n. 4, pp. 427-435. 2002.
- [29]CORDEAU, J., GENDREAU, M., LAPORTE, G., et al. "A guide to vehicle routing heuristics", *Journal of the Operational Research Society* v. 53, pp. 512-522, 2002.
- [30]DEIF, I., BODIN, L.D., "Extension of the Clarke and Wright algorithm for solving the vehicle routing with backhauling". In: *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, Mass, 1984.
- [31]DETHLOFF, J. "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up", *OR Spektrum* v. 23, pp. 79-96, 2001.
- [32]DORIGO M., MANIEZZO V., COLORNI A., "The Ant System: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man and Cybernetics – Part B*, 26: 1, pp. 29-41, 1996.
- [33]DROR, M. "Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution", *European Journal of Operational Research* v. 64, Issue 3, pp. 432-441, 1993.
- [34]DROR M., TRUDEAU, P. "Stochastic vehicle routing with modified savings algorithm", *European Journal of Operational Research* v. 23, n. 2, pp. 228-235, 1986.
- [35]DROR, M., POWELL, W. "Special Issue on Stochastic and Dynamic Models in Transportation", *Operations Research* v. 41, 1993.
- [36]DROR, M., TRUDEAU, P. "Savings by split delivery routing", *Transportations Science* v. 23, n. 2, pp. 141-145, 1989.
- [37]DROR, M., TRUDEAU, P. "Split delivery routing", *Naval Research Logistics* v. 37. n. 3, pp. 383-402, 1990.

- [38]DROR, M., LAPORTE, G., TRUDEAU, P. "Vehicle routing with split deliveries", *Discrete Applied Mathematics* v. 50, n. 3, pp. 239-254, 1994.
- [39]DUHAMEL, C., POTVIN, J.Y., ROUSSEAU, J.M. "A tabu search heuristic for the vehicle routing problem with backhauls and time windows", *Transportation Science* v. 31, pp. 49-49, 1997.
- [40]DORIGO, M., CARO, G. D., "The Ant Colony Optimization Meta-heuristic". In D. Crone, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*, Mc Graw-Hill, pp. 11-32, 1999.
- [41]FEO, T., RESENDE, M.G.C. "A probabilistic heuristic for a computational difficult set covering problems", *Operations research letters* v. 8, pp., 67-71, 1989.
- [42]FEO, T., RESENDE, M.G.C. "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization* v. 2, pp. 1-27, 1995.
- [43]FESTA, P., RESENDE, M.G.C., "GRASP: An annotated bibliography". In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, pp. 325-367, 2002.
- [44]FISHER M. L., "Optimal solution of behivle routing problems using minimum K-trees", *Operations Research*, v. 42, pp. 626-642, 1994.
- [45]FISHER M. L., JAIKUMAR R. "A generalized assignment heuristic for vehicle routing", *Networks* v. 11, pp. 109-124, 1981.
- [46]FISHER, M. L., "Vehicle routing problem". In : T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), *Networks Routing, Handbooks in Operations Research and Management Science*, v. 8, pp. 1-33, Elsevier Publisher BV, Amsterdam, 1995.
- [47]FRIZZELL, P. W., GIFFIN, J.W. "The bounded split delivery vehicle routing problem with grid network distances", *Asia-Pacific Journal of Operational Research* v. 9, No. 1, pp. 101-116, 1992.
- [48]FRIZZELL, P. W., GIFFIN, J.W. "The split delivery vehicle scheduling problem GAMBARDELLA, L., TAILLARD, E., AGAZZI, G., "Macs VRPTW : A multiple ant colony system for vehicle routing problems with time windows", In: Corne, D.,
- [49]Dorigo, M., Glover, F., (Eds.), *New Ideas in Optimization*, McGraw-Hill, London, pp.63-76, 1999.
- [50]GAREY, M.R., JOHNSON, D.S., "Computers and Intractability: A Guide to the Theory of NP-completeness", Freeman, San Francisco, 1979.
- [51]GÉLINAS, S., DESROCHERS, M., DESROSIERS, J., SOLOMON, M.M. "A new branching strategy for time constrained routing problems with application to backhauling", *Annals of Operations Research* v. 61, pp. 91-110, 1995.
- [52]GENDREAU, M., HERTZ, A., LAPORTE, G. "A tabu search heuristic for the vehicle routing problem", *Management Science* v. 40, pp. 1276-1290, 1994.
- [53]GENDREAU, M., LAPORTE, G., SÉGUIN, S. "Stochastic vehicle routing", *European Journal of Operational Research*, v. 88, n. 1, pp. 3-12, 1996.
- [54]GHAMLOUCHE, I., CRAINIC, T.G. E GENDREAU, M. "Path relinking, cyclebased neighbourhoods and capacitated multicommodity network design", *Annals of Operations Research* v.131 (1-4), pp. 109-133, 2004.
- [55]GILLET, B., MILLER, L. R. "A heuristic algorithm for the vehicle dispatch problem", *Operations Research* v. 22, pp. 340-349, 1974.
- [56]GLOVER, F. "Heuristic for integer programming using surrogate constraints", *Decision Sciences* 8, pp. 156-166, 1977.
- [57]GLOVER, F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research* v. 13, pp. 533-549, 1986.
- [58]GLOVER, F., "A Template for Scatter Search and Path Relinking, Artificial Evolution". In: *Lecture Notes in Computer Science* 1363, J. K. Hao, E. Lutton, E. Ronald, M. Schoenauer e D. Snyers (Eds.), Springer-Verlag, pp. 13-54, 1998.

- [59]GLOVER, F. "Tabu search for non-linear and parametric optimisation (with links to genetic algorithms)", *Discrete Applied Mathematics* v. 49, pp. 231-255, 1994.
- [60]GLOVER, F., LAGUNA, M. "Tabu search". Kluwer Academic Publishers, 1997.
- [61]GLOVER, F., LAGUNA, M., MARTÍ, R. "Fundamentals of Scatter Search and Path Relinking", *Control and Cybernetics* v. 29 n. 3, pp. 653-684, 2000.
- [62]GOETSCHALCKX, M., JACOBS-BLECHA, C. "The vehicle routing problem with backhauls", *European Journal of Operational Research*, v. 42, pp. 39-51, 1989.
- [63]GOLDEN, B. L., MAGNANTI, T. L., NGUYEN, H. Q. "Implementing vehicle routing algorithms", *Networks* v. 7, pp. 113-148, 1977.
- [64]HAGHANI, A. e JUNG, S. "A dynamic vehicle routing problem with time-dependent travel times", *Computers & Operations Research* v. 32, n. 11, pp. 2959-2986, 2005.
- [65]HERRERA, F., LOZANO, M. e MOLINA, D. "Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies", *European Journal of Operational Research*, v. 169, pp. 450-476, 2006.
- [66]HO, S. C., HAUGLAND, D. "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries", *Computers & Operations Research* v. 31, pp. 1947-1964, 2004.
- [67]HOLLAND, J., "Adaptation in Natural and Artificial Systems", *University of Michigan*, Press, 1975.
- [68]HOMBERGER, J., GEHRING, H., "Two evolutionary metaheuristics for the vehicle routing problem with time windows", *Information Systems and Operational Research-Special issue: Metaheuristics for Location and Routing Problems*, v. 37, pp. 297-318, 1999.
- [69]JASZKIEWICZ, A., KOMINEK, P. "Genetic local search with distance preserving recombination operator for a vehicle routing problem", *European Journal of Operational Research* v. 151, pp. 352-364, 2003.
- [70]KINDERWATER, G.A.P., SAVELSBERGH, M.W.P., "Vehicle routing: Handling edge exchanges". In : Aarts, E.H.L., and Lenstra, J.K. (eds), *Local Search in Combinatorial Optimization*, pp. 337-360, Wiley, Chichester, 1997.
- [71]KIRKPATRICK, S., GELATT, C.D.,VECCHI, P.M. "Optimization by simulated annealing", *Science* v. 220, pp.671-680, 1983.
- [72]LAGUNA, M., MARTÍ, R., *Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions*. Technical report TR11-2000, Departamento de Estadística e I.O., Universidad de Valencia, 2000.
- [73]LAGUNA, M. MARTI, R. "Scatter Search Methodology and Implementations in C", Kluwer Academic Publishers, 2003.
- [74]LAGUNA, M., ARMENTANO V., "Lessons from Applying and Experimenting with Scatter Search". In: *Metaheuristic Optimization Via Adaptive Memory and Evolution: Tabu Search and Scatter Search*, C. Rego and B. Alidaee (eds.), Norwell, MA: Kluwer Academic Publishers, pp. 229-246, 2005.
- [75]LAPORTE, G., LOUVEAUX, F., MERCURE, H. "Models and exact solutions for a class of stochastic location-routing problems", *European Journal of Operational Research* v. 9, n. 1, pp. 71-78, 1989.
- [76]LAPORTE, G., NORBERT, Y., " A branch and bound algorithm for the capacitated vehicle routing problem", *Operation Research Spektrum*, v. 5, pp. 77-85, 1983.
- [77]LAPORTE, G., NORBERT, Y., "Exact algorithms for the vehicle routing problem". In: S. Martello, G. Laporte, M. Minoux, C. Ribeiro (Eds.), *Surveys in*

- Combinatorial Optimization, Annals of Discrete Mathematics*, v. 31, pp. 147-184, 1987.
- [78] LAPORTE, G., NOBERT, Y., ARPIN, D., "Optimal solutions to capacitated multidepot vehicle routing problems", *Congressus Numerantium* v. 44, pp. 283-292, 1984.
- [79] LAPORTE, G., OSMAN, I.H., "Routing problems: A bibliography", *Annals of operations research* v. 61, n. 1, pp. 227-262, 1995.
- [80] LARSEN, A., The Dynamic Vehicle Routing Problem, Ph.D. Thesis, Department of Mathematical Modelling, Technical University of Denmark, 2000.
- [81] LAU, H.CH., SIM, M., MENG T., K. "Vehicle routing problem with time windows and a limited number of vehicles", *European Journal of Operational Research*, 2003.
- [82] LENSTRA, J., RINNOOY K. A. "Complexity of vehicle routing and scheduling problems", *Networks* v. 11, pp. 221-227, 1981.
- [83] LIN, S. "Computer solution of the traveling salesman problem", *Bell System Technical Journal*, v. 44, pp. 2245-2269, 1965.
- [84] LIN, S., KERNIGHAN, B. "An effective heuristic algorithm for the traveling salesman problem", *Operations Research* v. 21, pp. 498-516, 1973.
- [85] MARTÍ, R., LAGUNA, M. e GLOVER, F. "Principles of scatter search", *European Journal of Operational Research* v.169, n. 2, pp. 359-372, 2006.
- [86] MARTINHON, C., LUCENA, A., MACULAN, N. "Stronger K-tree relaxations for the vehicle routing problem", *European Journal of Operational Research* v. 158, n. 1, pp. 56-71, 2004.
- [87] MIN, H., "The multiple vehicle routing problem with simultaneous delivery and pickup points", *Transportation Research*, v. 23A, pp. 377-386, 1989.
- [88] MESTER, D., BRÄYSY, O. "Active guided evolution strategies for the large scale vehicle routing problem with time windows", *Computers & Operations*, v. 32, n. 6, pp. 1593-1614, 2005.
- [89] MINGOZZI, A., GIORGI, S. "An exact method for the vehicle routing problem with backhauls", *Transportation Science* v. 33, n. 3, pp. 315-329, 1999.
- [90] MOLE, R. H., JAMESON S.R. "A sequential route-building algorithm employing a generalized savings criterion", *Operational Research Quaterly* v. 27, pp. 503-511, 1976.
- [91] MONTANÉ, F.A.T. *Problemas de roteamento de veículos com pedidos de coleta e entrega*, Tese de doutorado, Engenharia de Produção, Univ. Federal de Rio de Janeiro, 2004.
- [92] MONTANÉ, F.A.T., GALVÃO, R.D. "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers & Operations Research* v. 33, n. 3, pp. 595-619, 2006.
- [93] MOSCATO, P., COTTA, C., "A gentle introduction to memetic algorithms". In: Glover, F., and Kochenberger, G.A. (eds), *Handbook of Metaheuristics*, pp. 105-144, Kluwer, Boston, 2003.
- [94] MOSHEIOV, G. "The traveling salesman problem with pick-up and delivery", *European Journal of Operational Research*, v. 79, pp. 299-310, 1994.
- [95] MULLASERIL, P.A., DROR, M., LEUNG, J. "Split-delivery routing heuristics in livestock feed distribution", *Journal of the Operational Research Society*, v. 48, n. 2, pp. 107-116, 1997.
- [96] NADDEF, D., RINALDI, G., "Branch-and-cut algorithms for the capacitated VRP". In: Toth, P., and Vigo, D. (eds), *The Vehicle Routing Problem*, pp. 53-84, SIAM Monographs on Discrete Mathematics and Application, Philadelphia, 2002.

- [97]NAGY, G., SALHI, S., "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, v. 162, n. 1, pp. 126-141, 2005.
- [98]NELSON, M.D., NYGARD, K.E., GRIFFIN, J.H., SHREVE, W.E. "Implementation techniques for the vehicle routing problem", *Computers & Operations Research*, V. 12, pp. 273-283, 1985.
- [99]OR, I., *Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking*, Ph. D. Thesis, Northwestern University, Evanston, IL, 1976.
- [100]OSMAN, I.H. "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem", *Annals of Operations Research*, v. 41, n. 4, pp. 421-451, 1993.
- [101]PITSOULIS, L.S., RESENDE, M.G.C., "Greedy randomized adaptive search procedures". In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization*, Oxford University Press, pp. 168-183, 2002.
- [102]POTVIN, J., ROUSSEAU, J. "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows", *European Journal of Operational Research*, v. 66, pp. 331-340, 1993.
- [103]POTVIN, J.-Y., ROUSSEAU, J.-M. "An exchange heuristic for routing problems with time windows", *Journal of the Operational Research Society* v. 46, pp. 1433-1446, 1995.
- [104]POWELL, W.B., JAILLET, P., ODONI, A. "Stochastic and Dynamic Networks and Routing". In: Network Routing, v. 8 of Handbooks in Operations Research and Management Science, chapter 3, pp. 141--295. North--Holland, Amsterdam, the Netherlands, 1995.
- [105]PRINS, C. "A simple and effective evolutionary algorithm for the vehicle routing problem", *Computers & Operations Research*, v. 31, n. 12, pp. 1985-2002, 2004.
- [106]PSARAFTIS, H. N. "Dynamic Vehicle routing: status and prospects", *Annals of Operations Research* v. 61, 143-164, 1995.
- [107]PUREZA, V. *Abordagens adaptativas de metaheurísticas tabu*, Tese de doutorado, Engenharia Elétrica e de Computação, Univ. Estadual de Campinas, 1996.
- [108]RALPHS, T. K., KOPMAN, L., PULLEYBLANK, W. R., et al. "On the Capacitated Vehicle Routing Problem", *Mathematical Programming*, Series B, 94, 343, 2003.
- [109]REIMANN, M., DOERNER, K., HARTL, R.F., "D-Ants: savings based ants divide and conquer the vehicle routing problem", *Computers & Operations Research* v. 31, n. 4, pp. 563-591, 2004.
- [110]REGO, C., "Node-ejection chains for the vehicle routing problem: sequential and parallel algorithms", *Parallel Computing* v. 27, n. 3, pp. 201-222, 2001.
- [111]REGO, C., "A subpath ejection method for the vehicle routing problem", *Management Science* v.44, pp. 1447-1459, 1998.
- [112]REGO, C., ROUCAIROL, C., "A parallel tabu search algorithm using ejection chains for the vehicle routing problem", In: *Meta-Heuristics: Theory and Applications*, Kluwer, Boston, pp. 661-675, 1996.
- [113]RENAUD, J., LAPORTE, G., BOCTOR, F.F. "A tabu search heuristic for the multi-depot vehicle routing problem", *Computers & Operations Research* v. 23, n. 3, pp. 229-235, 1996.
- [114]RESENDE, M.G.C., RIBEIRO, C.C., "Greedy randomized adaptive search procedures". In: F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, Kluwer Academic Publishers, pp. 219-249, 2002.

- [115]ROCHAT, Y., TAILLARD, E.D. "Probabilistic Diversification and intensification in local search for vehicle routing", *Journal of Heuristics*, v. 1, pp. 147-167, 1995.
- [116]RUSSELL, R., IGO, W. "An assignment routing problem", *Networks* v. 9, pp. 1-17, 1979.
- [117]RUSSELL, R., GRIBBIN, D. "A multiphase approach to the period routing problem", *Networks* v. 21, pp.747-765, 1991.
- [118]RUSSELL, R. "Hybrid heuristics for the vehicle routing problem with time windows", *Transportation Science*, v. 29, pp. 156-166, 1995.
- [119]RUSSELL, R. A., CHIANG, W. Ch. "Scatter search for the vehicle routing problem with time windows", *European Journal of Operational Research* v. 169, n. 2, pp. 606-622, 2006.
- [120]SALHI, S., NAGY, G. "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling", *Journal of the Operational Research Society* v. 50, pp. 1034-1042, 1999.
- [121]SECOMANDI, N. "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands", *Computers & Operations Research* v. 27, n. 5, pp. 1171-1200 , 2000.
- [122]SÉGUIN, R., POTVIN, J.Y., GENDREAU, M., et al., "Real-Time Decision Problems: An Operational Perspective", *Journal of the Operational Research Society* v. 48, 162-174, 1997.
- [123]SOLOMON, M. "Algorithms for the vehicle routing and scheduling problem with time windows constraints", *Operations Research* v. 35, n. 2, pp. 254-265, 1987.
- [124]STEWART, W. R., GOLDEN, J.B. "Stochastic vehicle routing: A comprehensive approach", *European Journal of Operational Research*, v. 14 n. 4, pp. 371-385 , 1983.
- [125]TAILLARD, E. D. "Parallel iterative search methods for vehicle routing problem", *Networks* v. 23, pp. 661-673, 1993.
- [126]TAILLARD, E.D., BADEAU, P., GENDREAU, P., GUERTIN, F., POTVIN, J.Y. "A tabu search heuristic for the vehicle routing problem with soft time windows", *Transportation Science* v. 31, pp. 170 - 186, 1997.
- [127]TAN, C.C.R., BEASLEY J. E. "A heuristic algorithm for the period vehicle routing problem", *Omega* v. 12, n. 5, pp. 497-504, 1984.
- [128]TARANTILIS, C. D., IOANNOU, G. e PRASTACOS, G. "Advanced vehicle routing algorithms for complex operations management problems", *Journal of food engineering* v. 70, pp. 455-471, 2005.
- [129]TARANTILIS, C. D. "Solving the vehicle routing problem with adaptive memory programming methodology", *Computers & Operations Research* v. 32, pp. 2309 – 2327, 2005.
- [130]TARANTILIS, C.D., KIRANOUDIS, C.T. "Bone Route: an adaptive memory-based method for effective fleet management", *Annals of Operations Research* v. 115, pp. 227-241, 2002.
- [131]THOMPSON, P., PSARAFTIS, H. "Cyclic transfer algorithms for multivehicle routing and scheduling problem", *Operations Research* v. 41, pp. 935-946, 1993.
- [132]TILLMAN, F. A. "The multiple terminal delivery problem with probabilistic demands", *Transportation Science* v. 3, pp. 192-204, 1969.
- [133]TILLMAN, F. A., HERING, R. W. "A study of a look-ahead procedure for solving the multiterminal delivery problem", *Transportation Research* v. 5, pp. 225-229, 1971.

- [134]TILLMAN, F. A., CAIN, T. M. "An upperbound algorithm for the single and multiple terminal delivery problem", *Management Science* v. 18, pp. 664-682, 1972.
- [135]TIMON C. D., ELTON Y. L. , DEFROSE CH. "Dynamic vehicle routing for online B2C delivery", *Omega* v. 33, n.1, pp. 33-45, 2005.
- [136]TOTH, P., VIGO, D. "An exact algorithm for the vehicle routing problem with backhauls", *Transportation Science* v. 31, pp. 372-385, 1997.
- [137]TOTH, P., VIGO, D. "A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls", *European Journal of Operational Research*, v. 113, pp. 528-543, 1999.
- [138]TOTH, P., VIGO, D. "The Vehicle Routing Problem". SIAM Monographs on Discrete . Philadelphia. Mathematics and Applications, 2002.
- [139]TOTH, P., VIGO, D. "The granular tabu search (and its application to the vehicle routing problem)", INFORMS, *Journal on Computing* v. 14, n. 4, pp. 333-348, 2003.

## Anexo

Aqui é apresentado resultados para algumas instâncias testadas, visualizadas de maneira gráfica.

Figura 1: Solução para instância de Salhi and Nagy: vrpncBD1

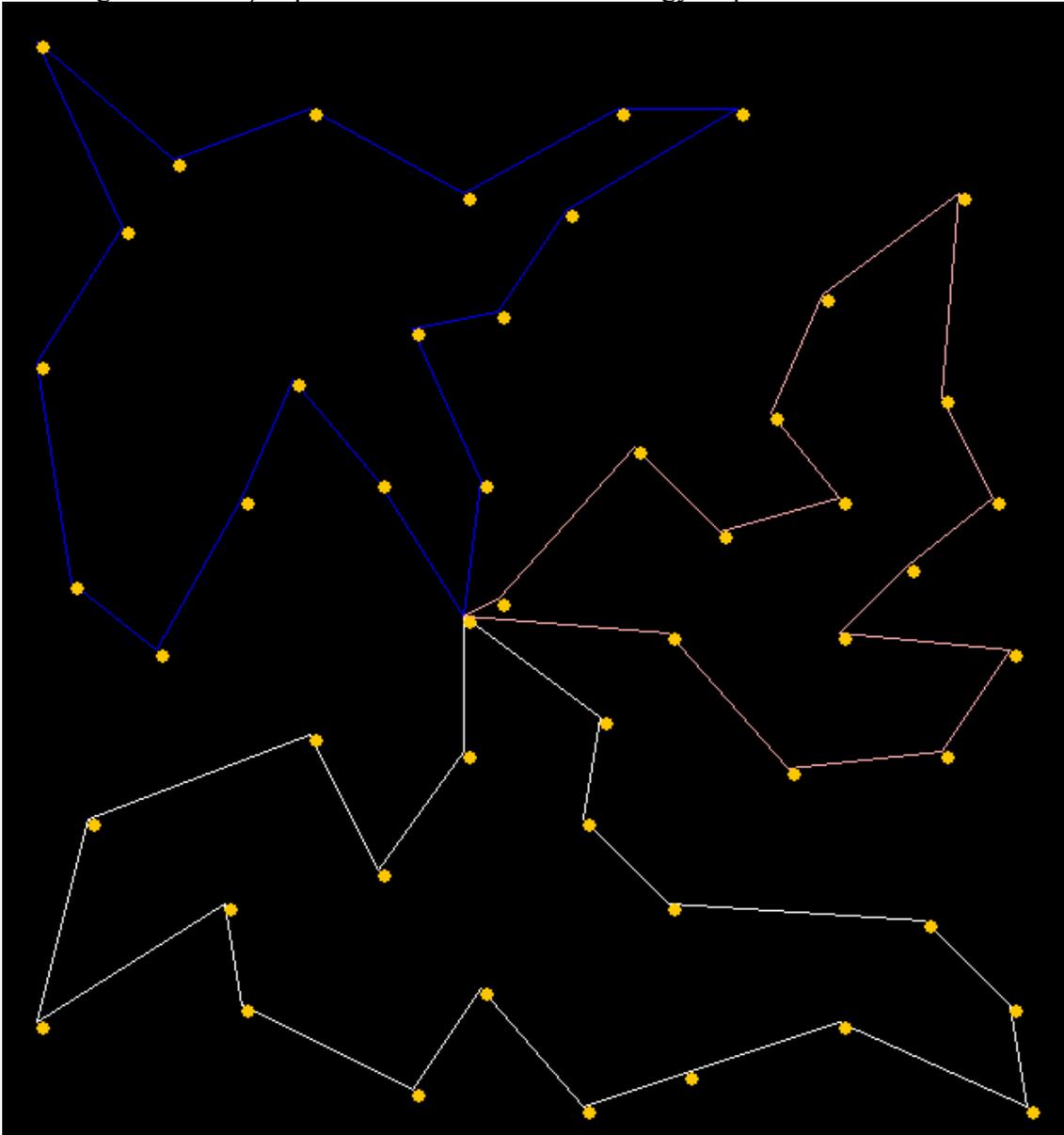


Figura 2: Solução para instância de Salhi and Nagy: vrpncBD3

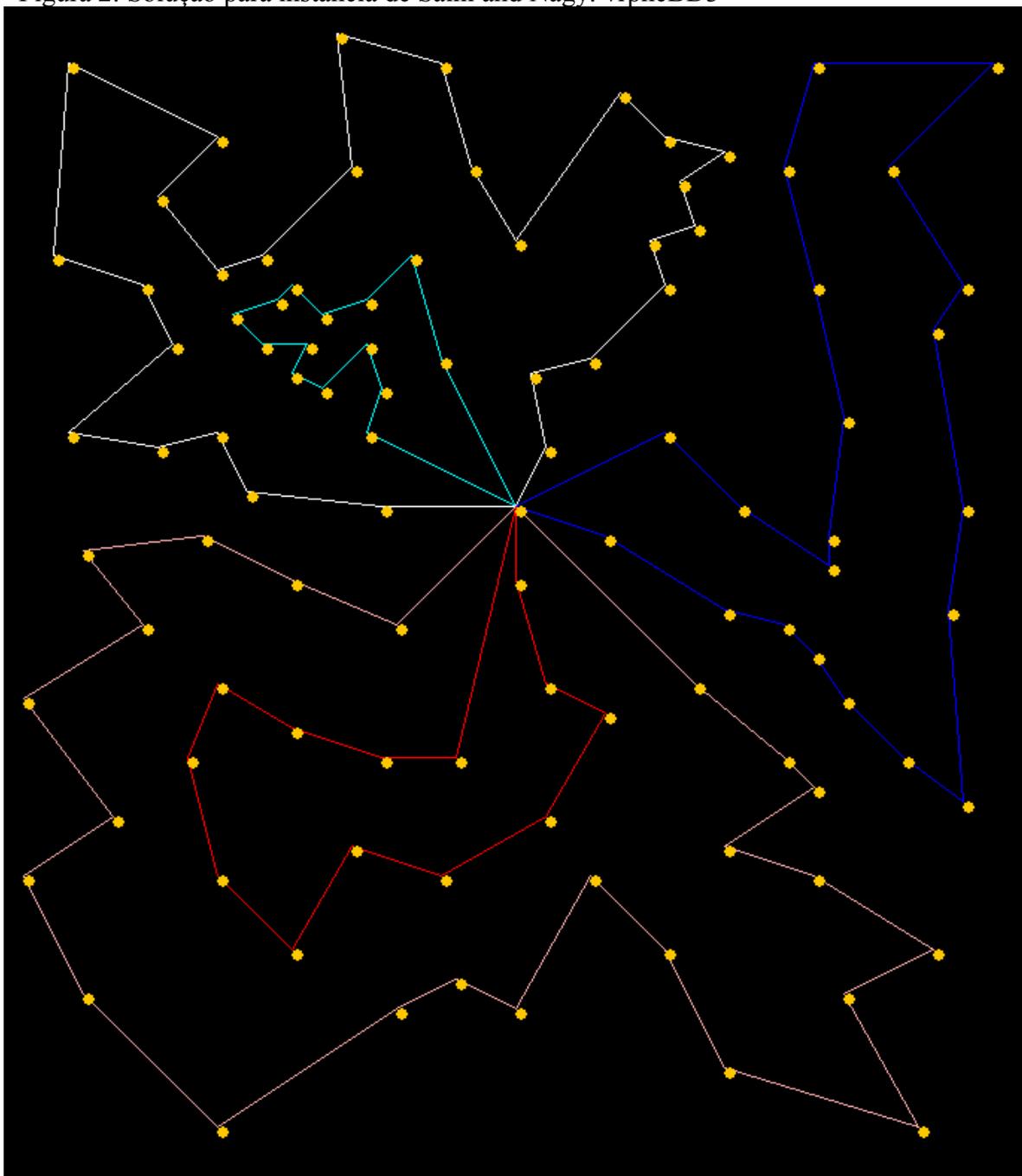


Figura 2: Solução para instância de Salhi and Nagy: vrpncBD11

