

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**APLICAÇÃO DE UMA ABORDAGEM ADAPTATIVA DE BUSCA  
TABU A PROBLEMAS DE ROTEIRIZAÇÃO E PROGRAMAÇÃO  
DE VEÍCULOS**

Aluna: Juliana Maria Rangel Barbosa

DISSERTAÇÃO DE MESTRADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**APLICAÇÃO DE UMA ABORDAGEM ADAPTATIVA DE BUSCA  
TABU A PROBLEMAS DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE  
VEÍCULOS**

Aluna: Juliana Maria Rangel Barbosa

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Produção.

Orientadora: Prof. Dra. Vitória M. Pureza

**São Carlos  
2005**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

B238aa

Barbosa, Juliana Maria Rangel.

Aplicação de uma abordagem adaptativa de busca tabu a problemas de roteirização e programação de veículos / Juliana Maria Rangel Barbosa. -- São Carlos : UFSCar, 2006.

100 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2005.

1. Logística empresarial. 2. Roteirização. 3. Busca - tabu. 4. Otimização combinatória. I. Título.

CDD: 658.5 (20<sup>a</sup>)



## FOLHA DE APROVAÇÃO

Aluno(a): Juliana Maria Rangel Barbosa

DISSERTAÇÃO DE MESTRADO DEFENDIDA E APROVADA EM 23/06/2005 PELA  
COMISSÃO JULGADORA:

Profª Drª Vitória Maria Miranda Pureza  
Orientador(a) PPGEP/UFSCar

Prof. Dr. Cláudio Barbieri da Cunha  
POLI/USP

Profª Drª Débora Pretti Ronconi  
POLI/USP

---

Prof. Dr. Dário Henrique Alliprandini  
Coordenador do PPGEP

## **DEDICATÓRIA**

Á Deus e a minha querida família:  
Vera, José Roberto, José Augusto e  
Jose Roberto Filho.

## **AGRADECIMENTOS**

Esta página é destinada a fazer uma singela homenagem a determinadas pessoas que de alguma forma ajudaram e acreditaram no atual trabalho.

A agradeço primeiramente a meus pais que me deram todo apoio para a realização deste trabalho.

Agradeço a minha querida orientadora profa. Dra. Vitória Pureza pela dedicação, esmero e por acreditar no meu trabalho.

Aos professores Reinaldo Morabito e Marcos Arenales pelos ensinamentos adquiridos durante o curso de mestrado.

Ao Marcio Rogério dos Santos que tanto me incentivou nas piores etapas do trabalho.

A todos os meus amigos do laboratório de Modelagem do departamento de Engenharia de Produção da UFSCar.

Aos meus amigos Isaías e José Laércio pelos preciosos comentários.

## RESUMO

O corrente projeto tem como objetivo o refinamento da abordagem adaptativa de busca tabu HTSA (PUREZA, 1996) e a verificação de seu desempenho quando aplicada ao Problema de Roteirização de Veículos clássico e ao Problema de Roteirização com Janelas de Tempo. A abordagem HTSA tem como objetivo a integração de estratégias de intensificação e diversificação, consistindo na variação sistemática de valores de parâmetros tabu selecionados e apoiada principalmente na análise de padrões da trajetória da busca. O desenvolvimento de novas abordagens baseadas na meta-heurística busca tabu (GLOVER, 1989; GLOVER & LAGUNA, 1997) é uma linha de pesquisa interessante uma vez que a busca tabu tem oferecido novas marcas em qualidade da solução em problemas de Roteirização de veículos e suas variantes, geralmente superando outros métodos. Os resultados obtidos com a aplicação da abordagem HTSA a instâncias de roteirização de veículos clássicas e com janela de tempo indicam soluções de qualidade em tempos computacionais razoáveis quando comparadas aos resultados de métodos competitivos da literatura.

**Palavras-chave:** Roteirização e Programação de Veículos. Otimização Combinatória. Heurísticas. Busca Tabu.

## **ABSTRACT**

This project consists in the refinement of the tabu search adaptive approach HTSA (PUREZA, 1996) and the analysis of its performance when applied to the classical Vehicle Routing Problem and to the Vehicle Routing Problem with Time Windows. HTSA promotes the integration of intensification and diversification strategies through the systematic variation of the values of selected tabu parameters, mostly based on the analysis of search trajectory patterns. The development of new implementations based on tabu search (GLOVER, 1989; GLOVER & LAGUNA, 1997) is an interesting avenue of research since tabu search has offered new marks on solution quality in routing problems, usually outperforming other methods. The results obtained with the application of HTSA approach to a set of classical routing instances and to a set of routing with times windows instances indicate quality solutions within reasonable computational times when compared to the results provided by competitive methods in the literature.

**Keywords:** Vehicle Routing and Scheduling. Combinatorial Optimization. Heuristics. Tabu Search.



## LISTA DE FIGURAS

FIGURA 2.1 - (a) Rotas 1 e 2 com um único cliente (distância total = $d(i, 0) + d(0, i) + d(j, 0) + d(0, j)$ ). (b) Rota 1' resultante da junção das rotas 1 e 2 e candidata à efetivação (distância total = $d(0, j) + d(i, 0) + d(j, i)$ ).....	25
FIGURA 2.2 – Passos da heurística (i) de Solomon - Passo 1: Escolha do cliente f como semente da rota. Passo 2: Avaliação das posições de inserção dos clientes com inserção viável na rota (clientes e e d). Passo 3 e 4: Escolha do cliente com melhor posição de inserção segundo a avaliação (cliente e), e sua inserção na rota (entre o cliente f e o depósito). Passo 5: Roteirização de todos os clientes. ....	28
FIGURA 2.3 - Aplicação de movimento 2-opt à rota 1. Rota 2 é resultante da eliminação das arestas $a-0$ e $c-d$ e adição das arestas $d-a$ e $c-0$ . ....	31
FIGURA 2.4 - Aplicação de movimentos de troca e inserção de nós entre as rotas .....	32
FIGURA 3.1 - Procedimento geral de métodos de busca local. ....	36
FIGURA 3.2 - Procedimento geral de busca tabu sumples. ....	40
FIGURA 4.1 - Descrição dos padrões de trajetória .....	59
FIGURA 4.2 - Descrição das médias anterior e corrente e da variável hoz .....	61
FIGURA 4.3 – Fluxograma da Abordagem Adaptativa HTA.....	62
FIGURA 5.1 - Fluxograma da Abordagem HTSA.....	67
FIGURA 5.2 - Lista de armazenagem de custos de soluções.....	68

FIGURA 5.3 - Gráfico 5.1 de convergência dos algoritmos PF + HTSA e PF + HTSA + MF aplicados à instância P8. ....	82
FIGURA 5.4 - Gráfico 5.2 de convergência dos algoritmos PF + HTSA e PF + HTSA + MF aplicados à instância P12. ....	83
FIGURA 5.5 – Gráfico 5.3 de convergência média dos algoritmos PF + HTSA e PF + HTSA + MF para as 14 instâncias de Christofides et al.,1979.....	83

## LISTA DE TABELAS

TABELA 1.1 - Demonstrativo dos custos da empresa Refrescos Ipiranga S/A – 1999.....	16
TABELA 1.2 - Resultados alcançados com a implantação dos softwares de Roterização ..	19
TABELA 5.1 - Instâncias em CHRISTOFIDES <i>et al.</i> (1979).....	72
TABELA 5.2 - Resultados de PF + HTSA para as 14 instâncias de Christofides <i>et al.</i> 1979 (PRV).....	73
TABELA 5.3 - Resultados de PF + HTSA e 5 algoritmos competitivos para as 14 instâncias de Christofides <i>et al.</i> 1979 (PRV).....	74
TABELA 5.4 - Resultados de PF + HTSA + MF para as 14 instâncias de Christofides <i>et al.</i> , 1979 (PRV).....	80
TABELA 5.5 – Comparação entre os resultados dos algoritmos PF + HTSA e PF + HTSA + MF para as 14 instâncias de Cristofides <i>et al.</i> , 1979 (PRV).....	81
TABELA 5.6 – Resultados computacionais PFJT + HTSA e PFJT + HTSA + 2-troca* para os 6 conjuntos das instâncias de Solomon, 1987 (PRVJT).....	86
TABELA 5.7 - Comparação entre os resultados computacionais de 5 algoritmos e as melhores soluções reportadas na literatura para os 6 conjuntos de instâncias de Solomon .....	87

TABELA A.1 - Resultados computacionais obtidos pelo algoritmo PFJT + HTSA para as 56 instancias de instâncias de Solomon, 1987 (PRVJT). .....98

TABELA A.2 - Resultados computacionais obtidos pelo algoritmo PFJT + HTSA + 2-troca\* para as 56 instancias de instâncias de Solomon, 1987 (PRVJT). ...99

## SUMÁRIO

<b>CAPÍTULO 1 INTRODUÇÃO.....</b>	<b>14</b>
<b>CAPÍTULO 2 O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS .....</b>	<b>21</b>
2.1 Definição e Modelagem.....	21
2.2 Métodos de Construção de Rotas.....	25
2.3 Métodos de Duas Fases.....	29
2.4 Métodos de Otimização Incompleta .....	30
2.5 Métodos de Melhoria da Rota.....	30
<b>CAPÍTULO 3 META - HEURISTICA BUSCA TABU - CONCEITOS BÁSICOS ..</b>	<b>35</b>
3.1 Memória de Curto Prazo.....	37
3.2 Memória de Longo Prazo .....	40
3.3 Estratégias de Intensificação da Busca .....	42
3.4 Estratégias de Diversificação da Busca .....	43
<b>CAPÍTULO 4 ALGORITMOS DE BUSCA TABU PARA O PROBLEMA DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS.....</b>	<b>46</b>
4.1 Algoritmo de OSMAN (1993).....	46
4.2 Algoritmo de TAILLARD (1993) .....	48
4.3 Algoritmo de ROCHAT & TAILLARD (1995).....	49
4.4 Algoritmo TABUROUTE (GENDREAU <i>et al.</i> , 1994).....	51
4.5 Algoritmo UTSA (CORDEAU <i>et al.</i> , 2001) .....	53
4.6 Algoritmo de TAN <i>et al.</i> , 2001 .....	54

4.7 Algoritmo LC03 (BOUTHILLIER & CRAINIC, 2005).....	55
4.8 Algoritmo PF (PUREZA & FRANÇA, 1990).....	57
4.9 Algoritmo HTA (PUREZA & FRANÇA, 2001).....	58
<b>CAPÍTULO 5 ABORDAGEM DE BUSCA TABU HTSA E SUA APLICAÇÃO AO PRV E AO PRVJT.....</b>	<b>63</b>
5.1 Princípios da Abordagem HTSA.....	64
5.2 Aplicação da Abordagem HTSA ao Problema de Roteirização de Veículos.....	68
5.2.1 Passos do Algoritmo PF + HTSA.....	69
5.2.1.1 Experimentos Computacionais para Calibragem de Parâmetros.....	71
5.2.1.2 Combinação da Abordagem HTSA com Diversificação Baseada em Frequência.....	75
5.2.1.2.1 Passos do Algoritmo PF + HTSA + MF.....	77
5.2.1.2.2 Resultados Computacionais.....	79
5.3 Aplicação da Abordagem HTSA ao Problema de Roteirização de Veículos com Janelas Tempo (PRVJT).....	84
5.3.1 Experimentos e Resultados Computacionais.....	85
<b>CAPÍTULO 6 CONCLUSÕES E PERSPECTIVAS DE PESQUISA.....</b>	<b>89</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>91</b>
<b>APÊNDICE A.....</b>	<b>97</b>

## CAPÍTULO 1 INTRODUÇÃO

Nos tempos atuais, empresas de diferentes setores operam em cenários cada vez mais competitivos. Isto se deve, dentre outros fatores, à queda das barreiras alfandegárias e à globalização da economia, somadas ao maior grau de qualidade dos bens e serviços exigidos pelos clientes. Como resultado, muitas empresas possuem recursos e clientes espalhados em uma ampla área geográfica, o que, por sua vez, gera um hiato entre a produção da maioria dos bens e o momento em que eles são consumidos. Esse hiato pode ser entendido como sendo de tempo e espaço, ou seja, a distância entre a produção e consumo a ser percorrida e que demandará certo tempo (PELIZARO, 2000).

Diminuir a distância existente entre a produção e a demanda, de modo que os consumidores tenham bens e serviços quando e onde quiserem, e na condição física que desejarem, passou a ser um problema vital para as empresas. Na tentativa de solucionar este problema, as empresas passaram a dar maior ênfase ao chamado *Sistema Logístico* que hoje se apresenta não somente como uma ciência ou centro de custos das empresas, mas também como uma ferramenta fundamental para aumentar os níveis de competitividade da empresa.

“Desde a Segunda Guerra Mundial, a logística vem apresentando evoluções contínuas, sendo hoje considerado um dos elementos-chave na estratégia competitiva das empresas. No início, era confundida com o transporte e armazenagem de produtos, hoje é o ponto principal da cadeia produtiva integrada, atuando com o moderno conceito de *Supply Chain Management (SCM)*” (NOVAES, 2001).

Segundo *Council of Supply Chain Management Professionals - CSCMP*, 2004 a logística pode ser definida como “à parte do processo da cadeia de suprimentos que planeja, implementa e controla o fluxo efetivo e eficaz, a armazenagem de bens, serviços e informações relacionadas, a partir da origem até o ponto de consumo desses bens, de maneira a satisfazer as necessidades dos consumidores”

Outros autores como EILON (1971), definem a logística simplesmente como “a provisão de bens e serviços de um ponto de oferta para um ponto de demanda”.

Os processos logísticos de algumas empresas estão sujeitos a algum tipo de sazonalidade. A humanidade e seus grupos sociais, desde os tempos mais remotos, sempre tiveram suas atividades controladas por algum tipo de evento periódico: inverno, verão, meses do ano, período semanal e mesmo ao longo das horas do dia. Esta variação rítmica de atividades tem inúmeras implicações, dentre elas um forte impacto nas operações logísticas. A demanda por produtos e serviços é geralmente influenciada por componentes sazonais, que devem ser levados em conta para a utilização mais eficiente dos recursos e das oportunidades disponíveis (SALIBY, 1999). Um exemplo desta sazonalidade pode ser verificado em empresas do setor de bebidas, uma vez que estes são produtos à mercê de variações climáticas, econômicas e sociais.

Assim, tal variação da demanda juntamente com a diversidade dos pontos de venda dos produtos e/ou serviços, horário restrito de entrega destes, capacidade do veículo de entrega, e outros aspectos, geram um problema de distribuição física de produtos e/ou serviços.

“Nas empresas do setor de bebidas, a aquisição de matérias-primas e de recursos necessários para produzir as bebidas tornou-se questão secundária, pois os fornecedores, a localização e custos decorrentes já foram razoavelmente equacionados. Em face à grande concorrência e devido às características da geografia do país, a maior dificuldade atual deste setor é a *Distribuição Física das Mercadorias aos Clientes*, que disponibilizarão essas mercadorias para os consumidores finais” (FERNANDES, 2001).

“A *Distribuição Física* é o ramo da logística que trata da movimentação, estocagem e entrega dos pedidos ao consumidor final da empresa, ou seja, leva os produtos certos, para os lugares certos, no momento certo e com o nível de serviço desejado, pelo menor custo” (BALLOU, 1993).

“As atividades relacionadas ao transporte e à distribuição física buscam cada vez mais o aprimoramento da qualidade e da produtividade, de forma a garantir um melhor aproveitamento da frota e diminuição do percurso dos veículos. O aumento do número de entregas e sua dispersão geográfica, em decorrência da política de redução de estoque das empresas, que as leva a efetuar pedidos menores e com maior frequência, causam um impacto significativo nas operações e nos custos associados aos sistemas de distribuição.



Concomitantemente, aumentam as exigências dos clientes com relação a prazos, datas e horários de entrega” (LOPES, 1996).

De acordo com informações obtidas junto ao departamento de custos da empresa Refrescos Ipiranga S.A., em Ribeirão Preto – SP, empresa engarrafadora e distribuidora de bebidas da marca Coca-Cola na região de Ribeirão Preto e sul do estado de Minas Gerais, os custos de distribuição (entrega dos produtos aos revendedores) em 1999 corresponderam a 8% do custo total do produto (veja TABELA 1.1).

**TABELA 1.1 - Demonstrativo dos custos da empresa Refrescos Ipiranga S/A – 1999.**

Descrição	Valor (R\$ milhões)	Porcentagem
Custos de produção	114	69%
Despesas operacionais (administração + vendas)	37	23%
Custos com distribuição	12	8%
Total	163	100%

Fonte: Departamento de Custos da empresa Refrescos Ipiranga S.A. (2000).

Administrar os custos de distribuição física de uma empresa é de grande importância, pois as empresas precisam oferecer aos seus clientes produtos com preços compatíveis com o mercado. Segundo FERNANDES (2001), “o custo de distribuição física ainda é um dos fatores que permitem maior influência de administração e adequações, visto que a forma de gerir e operar o sistema de distribuição possibilita variações que podem levar a reduções significativas ao final do processo”.

Por este motivo, houve a necessidade de geração de sistemas informatizados que resolvessem de forma eficaz os chamados *Problemas de Roteirização e Programação de Veículos*, onde se busca a elaboração de rotas de veículos para o atendimento das necessidades de clientes, utilizando-se metodologias científicas. A roteirização consiste na definição da seqüência de clientes a serem atendidos por cada veículo. A programação define os instantes de tempo em que o atendimento a cada cliente ocorrerá.

Estes sistemas informatizados são denominados *roteirizadores*, e tiveram um rápido desenvolvimento durante as décadas de 80 e 90, devido ao grande impulso dado

pela Pesquisa Operacional de natureza acadêmica, bem como pela disponibilização de recursos computacionais abundantes e de baixo custo.

De acordo com FERNANDES (2001), “O roteirizador é um instrumento de planejamento e de simulação de situações na área de logística. Trata-se especificamente de um *software* capaz de montar um modelo matemático que seja fácil de manipular e parametrizar, e represente de maneira fiel às condições externas de trabalho (distância entre pontos, condições de trânsito, tempos de deslocamento e de trabalho, etc.), possibilitando assim a simulação do que ocorrerá nas ruas e estradas durante o trabalho de vendas, coletas ou entrega de materiais”.

Um roteirizador estabelece o itinerário ideal de entregas/visitas considerando:

- ✓ Um mapa urbano/rodoviário;
- ✓ Um conjunto de clientes (pontos de coleta ou entrega);
- ✓ Um conjunto de veículos - caminhão, carros, motos, e outros - disponíveis;
- ✓ Um conjunto de restrições para o mapa, clientes e veículos;
- ✓ Uma demanda de entrega/coleta que deve ser satisfeita pelos veículos.
- ✓ Um ou mais depósitos ou garagens, de onde os veículos partem e retornam após a operação.

Em situações reais, várias restrições podem ser impostas e tratadas por roteirizadores. Dentre as mais comuns, destacam-se:

- 1) Capacidade máxima de cada veículo;
- 2) Tempo máximo de duração da rota;
- 3) Tempos de serviço em cada ponto de demanda;
- 4) Intervalo de tempo no qual o cliente impõe o início do atendimento (janelas de tempo para clientes);
- 5) Intervalos de descanso dos motoristas (janelas de tempo para veículos).

Um caso particular de problemas de roteirização de veículos é aquele onde existe uma única garagem e que contempla as restrições (1), (2) e (3). Este problema é chamado de *Problema de Roteirização de Veículos clássico* (PRV). Uma extensão do PRV é o chamado *Problema de Roteirização e Programação de Veículos* ou *Problema de*

*Roteirização de Veículos com Janelas de Tempo (PRVJT)*, o qual inclui, além das restrições do PRV, a restrição (4).

O resultado do trabalho desenvolvido por um roteirizador é a seqüência de entregas/coletas a serem realizadas pelos veículos nos clientes especificados, segundo um critério definido pelo usuário, dentre os quais se inclui:

- ✓ Menor distância a ser percorrida pela frota;
- ✓ Menor tempo para se realizar todas as entregas;
- ✓ Menor custo total;
- ✓ Conjunção de um ou mais critérios anteriores.

Alguns exemplos da aplicação dos roteirizadores podem ser vistos, de acordo com HALL & PARTUKA (2000), nos setores de:

- ✓ Transporte de numerário e coleta de documentos em caixas eletrônicos;
- ✓ Transporte coletivo municipal e intermunicipal;
- ✓ Abastecimento de postos de combustíveis;
- ✓ Entrega de encomendas solicitadas via internet;
- ✓ Distribuição de produtos do atacado até o varejo;
- ✓ Distribuição de serviços postais;

Vários roteirizadores comerciais desenvolvidos para atender a diversas necessidades estão hoje disponíveis no mercado. De acordo com FERNANDES (2001), os dois software de roteirização com maior número de licenças no Brasil são *Trucks* e *Roadshow*. A tabela 1.2, logo a seguir, mostra alguns resultados obtidos por empresas, de diversos setores, que implantaram os softwares de roteirização citados acima.

**TABELA 1.2 - Resultados alcançados com a implantação dos softwares de Roteirização**

<b>Empresa</b>	<b>Ramo de atuação</b>	<b>Software de roteirização</b>	<b>Resultados</b>
Sadia S/A	Alimentos	Roadshow 1.5	Redução de 5% do custo operacional
Security couriers	Distribuidora de vale-refeição	Roadshow	Redução da frota e otimização das entregas
Cofesa	Atacadista	Trucks 8.2	Rotas mais otimizadas e maior controle da frota
Marilan biscoitos	Alimentos	Trucks	Redução de 5% nos custos de entrega
Grupo Martins	Distribuidora de atacado	Trucks	Redução e agilidade no tempo de entrega
Kibon	Sorvetes	Trucks	Redução em 25% km rodado e maior otimização da frota
Protege	Transporte de valores	Trucks 9.0	15% de redução da frota, maior rigor nos horários de entrega e retirada de valores

Fonte: Melo (2000).

Na literatura sobre os problemas de roteirização é possível encontrar vários métodos que obtêm soluções para o PRV e suas variantes. Em um extremo, encontram-se os métodos exatos que provêem soluções ótimas. Como o PRV tem natureza combinatória e pertence à classe *NP-hard* (GAREY & JOHNSON, 1979), o esforço computacional cresce exponencialmente com o tamanho da instância. Por esta razão, métodos ótimos são impraticáveis em situações de interesse.

Em tais situações, utilizam-se métodos heurísticos, pois apesar de não oferecerem garantia de otimalidade, são capazes de gerar soluções subótimas de alta qualidade com um custo computacional aceitável. Dentre os métodos heurísticos, as *meta-heurísticas* apresentam um desempenho que geralmente supera os demais métodos em um grande número de problemas. Em particular, a meta-heurística *Busca Tabu* (GLOVER, 1989; GLOVER & LAGUNA, 1997) vem oferecendo novas marcas em qualidade da solução em problemas de roteirização de veículos. Neste sentido, o corrente projeto tem como objetivo o refinamento na abordagem adaptativa de busca tabu HTSA (PUREZA,

1996), através de uma análise paramétrica e inclusão de mecanismos mais sofisticados de busca em vizinhança e de procedimentos da meta-heurística Busca Tabu, e posteriormente, a verificação de seu desempenho quando aplicada ao PRV e ao PRVJT.

O texto desta dissertação está estruturado da seguinte forma. O capítulo 2 apresenta uma descrição do PRV e alguns métodos heurísticos de resolução, juntamente com uma introdução sobre as heurísticas e meta-heurísticas descritas na literatura. No capítulo 3, são apresentados os conceitos da meta-heurística *Busca Tabu*. O capítulo 4 descreve alguns métodos heurísticos da literatura para o PRV e PRVJT e a abordagem adaptativa de busca tabu HTA (PUREZA & FRANÇA, 2001), e os resultados decorrentes de sua aplicação a um conjunto de exemplos clássicos do PRV. No capítulo 5 são apresentados, inicialmente, os resultados da análise paramétrica realizada na abordagem adaptativa tabu HTSA (PUREZA, 1996) e os resultados obtidos com sua aplicação a um conjunto de instâncias de PRV clássicos e ao conjunto de instâncias de PRV acrescidos da restrição janela de tempo. São também descritos, no capítulo 5, a implementação desta abordagem, adicionada de estratégias de memória de longo prazo da meta-heurística Tabu e de uma modificação na estrutura do mecanismo de busca em vizinhança, e os resultados computacionais associados para análise comparativa. O capítulo 6 descreve as conclusões obtidas e as perspectivas futuras desta pesquisa.

## CAPÍTULO 2 O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS

### 2.1 Definição e Modelagem

O Problema de Roteirização de Veículos clássico (PRV) consiste na definição de rotas de entrega (produtos) a custo mínimo por uma frota de veículos heterogêneos, originando e terminando em uma garagem ou depósito central. A frota deve servir um conjunto de clientes, cada qual caracterizado por sua localização no espaço bidimensional, demanda e tempo de serviço. Cada cliente tem sua demanda suprida por exatamente um veículo. A demanda total dos clientes não deve exceder a capacidade total do veículo designado a suprir a demanda destes clientes. O tempo total de qualquer rota não pode exceder um limitante pré-especificado e inclui os tempos de viagem entre clientes e tempos de serviço em cada cliente.

Seja:

$n$  - Número de clientes (nós) a serem visitados por uma frota de veículos heterogêneos, identificados pelo índice  $v$ , que originam e terminam em um depósito identificado pelo índice 0;

$nv$  - Número de veículos  $v$ ,  $v = 1, 2, \dots, nv$ ;

$d_i$  - Demanda do cliente  $i$ ,  $i = 1, 2, \dots, n$ ;

$t_i^v$  - Tempo de serviço do veículo  $v$  no cliente  $i$ ,  $i = 1, 2, \dots, n$ .

$t_{ij}^v$  - Tempo de viagem do veículo  $v$  entre os clientes  $i$  e  $j$ ,  $i, j = 1, 2, \dots, n$ .

$K_v$  - Capacidade do veículo  $v$ ,  $v = 1, 2, \dots, nv$ .

$c_{ij}^v$  - Custo de viagem do veículo  $v$  entre os clientes  $i$  e  $j$ ;

$T_v$  - Tempo máximo de viagem para o veículo  $v$ ;

Definidas as variáveis de decisão  $x_{ij}^v$ ,  $i, j = 1, 2, \dots, n$  e  $v = 1, 2, \dots, nv$ , tais que:

$$x_{ij}^v = \begin{cases} 1, & \text{se houver a viagem entre os clientes } i \text{ e } j \\ 0, & \text{caso contrário} \end{cases}$$

No PRV clássico, o objetivo mais comumente utilizado é a obtenção de uma solução que minimize o comprimento total do conjunto de rotas, mas para algumas variantes do PRV, como por exemplo o problema de roteirização de veículos com janela de tempo, objetiva-se também a minimização do número de veículos utilizados. Desta forma, o modelo matemático que expressa de maneira mais compreensiva os diversos aspectos do PRV clássico pode ser expresso pela seguinte formulação. (BODIN *et al.*, 1983):

$$\text{Min} \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^{nv} c_{ij}^v x_{ij}^v \quad (2.0)$$

Sujeita a:

$$\sum_{v=1}^{nv} \sum_{i=0}^n x_{ij}^v = 1, \quad j = 1, 2, \dots, n \quad (2.1)$$

$$\sum_{v=1}^{nv} \sum_{k=0}^n x_{jk}^v = 1, \quad i = 1, 2, \dots, n \quad (2.2)$$

$$\sum_{i=0}^n x_{ik}^v - \sum_{j=0}^n x_{kj}^v = 0, \quad v = 1, 2, \dots, nv; \quad k = 0, 1, 2, \dots, n \quad (2.3)$$

$$\sum_{j=1}^n x_{0j}^v \leq 1, \quad v = 1, 2, \dots, nv \quad (2.4)$$

$$\sum_{j=1}^n x_{j0}^v \leq 1, \quad v = 1, 2, \dots, nv \quad (2.5)$$

$$\sum_{i=1}^n d_i \left( \sum_{j=0}^n x_{ij}^v \right) \leq K_v, \quad v = 1, 2, \dots, nv \quad (2.6)$$

$$\sum_{i=1}^n t_i^v \sum_{j=0}^n x_{ij}^v + \sum_{i=1}^n \sum_{j=0}^n t_j^v x_{ij}^v \leq T_v, \quad v = 1, 2, \dots, nv \quad (2.7)$$

$$x_{ij}^v = 0 \text{ ou } 1 \text{ para todo } i, j, v \quad (2.8)$$

$$x \in S \tag{2.9}$$

As equações (2.1) e (2.2) garantem, respectivamente, que exatamente um veículo chega e sai do nó  $j$  ( $j = 1, \dots, n$ ). A equação (2.3) assegura que o veículo que chega e sai de cada nó  $k$  ( $j = 1, \dots, n$ ) é o mesmo. As inequações (2.4) e (2.5) asseguram exatamente um veículo sai e chega ao depósito (0). As inequações (2.6) e (2.7) representam respectivamente as restrições de capacidade dos veículos e tempo total de viagem. Admitese que a demanda de qualquer cliente  $j$  possa ser atendida por um único veículo e, neste modelo, atendimento parcial da demanda de um cliente por um veículo não é permitido. Note que condicionantes temporais não são importantes para a definição das rotas e das seqüências de atendimento; as estratégias de solução são, portanto, direcionadas aos aspectos espaciais da localização dos pontos de demanda (CUNHA, 2000).

A restrição (2.9) imposta sobre a matriz  $X$  das variáveis  $x_{ij}$  proíbe subrotas (ciclos) não contendo a base (depósito). O conjunto  $S$  é constituído por restrições impostas para cada veículo  $v$ , isto é,  $S$  pode ser expresso como a união de conjuntos  $S_v$ , definidos por:

$$S = \left\{ x_{ij}^v : \sum_{i \in Q} \sum_{j \in Q} x_{ij}^v \leq |Q| - 1 \right\}$$

para todo subconjunto não vazio  $Q$  de  $(1, \dots, n)$  onde  $|Q|$  representa o número de nós do conjunto  $Q$ .

Em muitos casos os clientes estabelecem limites inferiores e superiores,  $a_i$  e  $b_i$ ,  $i = 1, \dots, n$ , para o instante de chegada e saída do veículo que irá prestar o serviço. Este intervalo gerado por  $a_i$  e  $b_i$ ,  $i = 1, \dots, n$  é chamado de *janela de tempo do cliente*. Note que com a inclusão de janelas de tempo, o problema passa a ser de *Roteirização e Programação dos veículos*, uma vez que além dos aspectos espaciais de localização dos pontos, o atendimento destas restrições temporais é determinante para a definição da rotas (CUNHA, 2000)<sup>1</sup>.

Para incorporar estas restrições temporais ao modelo matemático do PRV clássico, descrito no item 2.1, é necessário, inicialmente, considerar o objetivo primário da função objetivo a minimização do número de veículos que suprem a demanda dos clientes e

---

<sup>1</sup>Por simplicidade, o problema de roteirização e programação de veículos será referido ao longo do texto por problema de roteirização de veículos com janela de tempo.



como objetivo secundário a minimização da distância total percorrida, posteriormente considera-se que o instante de chegada do veículo  $v$  ao cliente  $j$  é dado por  $s_j$  assim, restrições de *janela de tempo* podem ser representadas no modelo matemático como:

$$s_j = \sum_{v=1}^{NV} \sum_{i=0}^n (s_i + t_i^v + t_{ij}^v) x_{ij}^v \quad j = 1, \dots, n \quad (3.0)$$

$$a_j \leq s_j \leq b_j \quad s_0 = 0 \quad (3.1)$$

onde  $s_j$  é calculado em função do instante de chegada no nó  $i$  ( $s_i$ ), tempo de serviço no nó  $i$  ( $t_i^v$ ) e tempo de viagem entre os nós  $i$  e  $j$  ( $t_{ij}^v$ ). Caso  $s_j < a_j$ , então  $s_j = \max\{a_j, s_i + t_i^v + t_{ij}^v\}$  ou caso  $s_j > b_j$ , a viagem entre os nós  $i$  e  $j$  não ocorrerá;

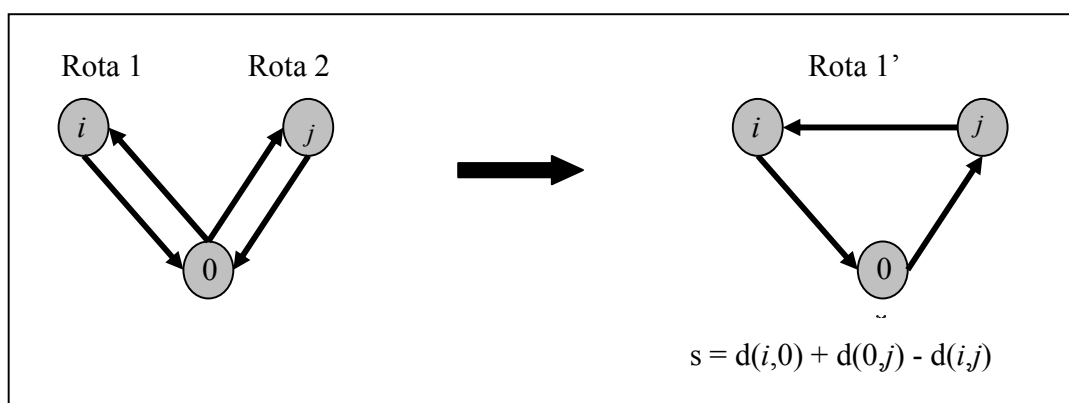
O PRV (e suas extensões) é classificado como um problema de natureza combinatória, onde o espaço de soluções é construído através de um arranjo, agrupamento, ordenação ou seleção de objetos, usualmente finitos. Com respeito a sua complexidade computacional, ele faz parte da classe *NP-hard* (GAREY & JOHNSON, 1979), ou seja, não são conhecidos algoritmos exatos de tempo polinomial que o resolva. Devido à dificuldade intrínseca da aplicação de algoritmos exatos em problemas de interesse prático, “métodos heurísticos são aplicados a este problema para que se obtenham soluções não necessariamente ótimas, porém de boa qualidade” (PUREZA, 1996).

Métodos heurísticos podem ser vistos como sendo métodos de solução de problemas desenvolvidos racionalmente, com enfoque intuitivo, apoiando-se em métodos computacionais que visam à escolha de um caminho, dentre os vários possíveis, para a obtenção de soluções viáveis ou não e com esforço computacional aceitável.

A efetividade de métodos heurísticos depende de sua habilidade em se adaptar a uma instância em particular, de evitar o aprisionamento em soluções sub-ótimas e de explorar a estrutura básica dos problemas. Uma classificação de métodos heurísticos para o PRV clássico e suas extensões é apresentada nas próximas seções. Uma recente revisão pode ser obtida em CORDEAU *et al.*, (2002,2002a)

## 2.2 Métodos de Construção de Rotas

Os métodos de construção de rotas se desenvolveram, em sua maioria, a partir do trabalho de CLARKE & WRIGHT (1964), pois é um procedimento simples e flexível. Este método assume como solução de partida  $n$  rotas distintas constituídas por um único cliente (tal configuração é considerada inviável, pois é geralmente impraticável designar um veículo da frota para cada cliente). O algoritmo avalia então a economia em distância que resulta em servir os clientes de cada par de rotas em uma única rota, unindo as duas rotas a partir de seus nós iniciais ou finais (veja FIGURA 2.1).



**FIGURA 2.1 - (a) Rotas 1 e 2 com um único cliente (distância total =  $d(i, 0) + d(0, i) + d(j,0) + d(0, j)$ ). (b) Rota 1' resultante da junção das rotas 1 e 2 e candidata à efetivação (distância total =  $d(0, j) + d(i, 0) + d(j, i)$ ).**

Uma dada combinação de rotas é passível de efetivação se a combinação resultar em uma economia positiva. CLARKE & WRIGHT (1964) sugerem uma estratégia gulosa, isto é, a cada estágio é escolhida a junção viável que provoca a maior economia. Este método mostra-se eficiente, ou seja, com pequeno esforço computacional se obtém soluções com boas características.

O acréscimo de restrições ao algoritmo de Clarke e Wright gera, usualmente, deteriorização na qualidade das soluções, isto pode ser explicado pelo fato que este algoritmo está baseado sobre um princípio guloso. Em SOLOMON (1987) são relatados os

resultados desapontadores que se obteve com o acréscimo da restrição de janela de tempo ao algoritmo de Clarke e Wright.

Dentre os métodos de construção da solução inicial um dos métodos mais simples é o *Método do Vizinho mais Próximo* (BELLMORE & NEMHAUSER, 1968), esta heurística parte de um nó arbitrariamente escolhido, selecionam-se como o próximo nó a ser visitado aquele com a menor distância do nó corrente. Ao final deste processo, todos os nós estarão roteirizados. Esta é uma heurística essencialmente gulosa, pois procura escolher a melhor inclusão local. Outro método de construção bastante conhecido é o *Método da Varredura* (GILLET & MILLER, 1974) onde se procura obter a solução inicial da instância em duas etapas distintas. A primeira visa agrupar os pontos de demanda através de um procedimento de varredura, que consiste em girar em dado sentido, uma semi-reta com origem na base e assim identificar os nós viáveis que formaram grupos que em uma segunda etapa serão roteirizados por um algoritmo elaborado para o tipo de problema. O princípio desta heurística está no fato de que os trajetos entre nós serão desenvolvidos preferencialmente entre vizinhos.

Uma heurística de construção muito aplicada para o tratamento do PRV clássico e adicionado das restrições de janela de tempo - PRVJT é a heurística de inserção sequencial *Push-Forward* de SOLOMON (1987). Seu nome deriva do fato de que a solução é construída através de inserções viáveis de nós não roteirizados na rota corrente. A viabilidade temporal do resultado da inserção é verificada, calculando-se de forma engenhosa, o efeito que tal inserção teria sobre os tempos de início de serviço dos nós já presentes na rota e que sucederiam o nó candidato à inserção (*Push-forward*). São propostas três variantes desta heurística, denominadas em SOLOMON (1987) como (i), (ii) e (iii). Os passos da heurística (i), considerada a mais bem sucedida para as instâncias tratadas no trabalho, são os seguintes:

1. Seleciona-se um nó não roteirizado para iniciar a rota corrente segundo um dos dois critérios: (1) nó não roteirizado mais distante da garagem, (2) nó não roteirizado com menor prazo de início do serviço.

2. Seja  $(i_0, i_1, \dots, i_m)$  a rota corrente, com  $i_0 = i_m = 0$ . Para cada nó não roteirizado, obtém-se sua posição de inserção viável que provê a melhor avaliação segundo o critério:

$$c_1(i(u), u, j(u)) = \text{Min} [c_1(i_{p-1}, u, i_p)], \quad p = 1, \dots, m$$

Onde

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \quad \alpha_1 + \alpha_2 = 1, \quad \alpha_1, \alpha_2 \geq 0$$

tal que

$$c_{11}(i, u, j) = d(i, u) + d(u, j) - \mu d(i, j), \quad \mu \geq 0$$

$$\text{e } c_{12}(i, u, j) = b_{ju} - b_j$$

onde  $b_j$  é o tempo atual de início do serviço no nó  $j$  e  $b_{ju}$  é o tempo de início do serviço no nó  $j$  com a inserção de  $u$ .

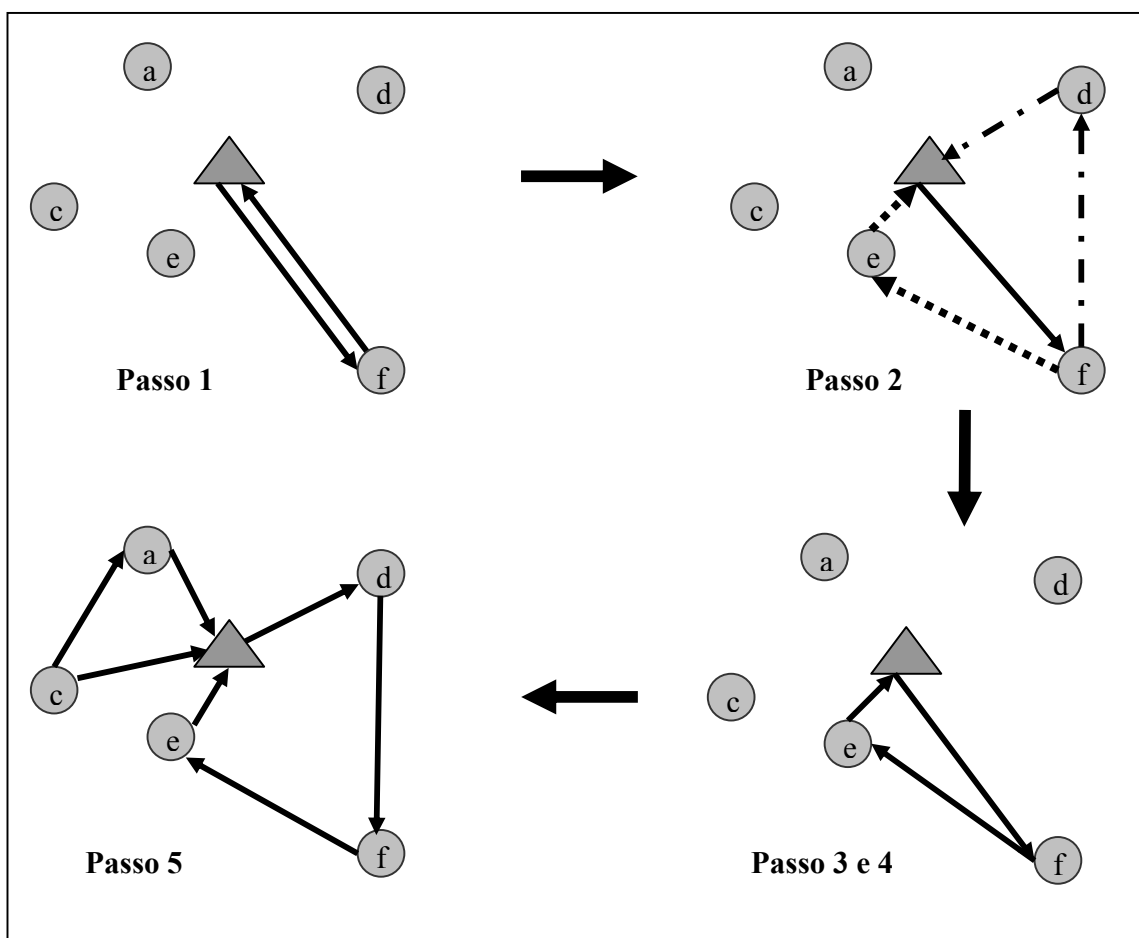
3. Seleciona-se  $u^*$  dentre os nós não roteirizados com melhor posição de inserção, tal que:

$$c_2(i(u^*), u^*, j(u^*)) = \text{Max}[c_2(i(u), u, j(u))],$$

$$\text{onde: } c_2(i, u, j) = \delta d(0, u) - c_1(i, u, j), \quad \delta \geq 0;$$

4. Insere-se o nó  $u^*$  na rota  $r$  entre  $i(u^*)$  e  $j(u^*)$ , atualizando-se os tempos de início de serviço de todos os nós a partir de  $j(u^*)$ , e a capacidade restante do veículo.
5. Quando não há mais nós cuja inserção seja viável, uma nova rota é iniciada. A construção é finalizada quando todos os nós tiverem sido designados.

Os passos da heurística (i) apresentada acima podem ser visualizados na FIGURA 2.2 abaixo:



**FIGURA 2.2 – Passos da heurística (i) de Solomon - Passo 1: Escolha do cliente f como semente da rota. Passo 2: Avaliação das posições de inserção dos clientes com inserção viável na rota (clientes e e d). Passo 3 e 4: Escolha do cliente com melhor posição de inserção segundo a avaliação (cliente e), e sua inserção na rota (entre o cliente f e o depósito). Passo 5: Roteirização de todos os clientes.**

Note que esta heurística procura maximizar o benefício (tempo e/ou distância) de se servir um nó em uma rota parcial em lugar de atendê-lo diretamente a partir do depósito. Por exemplo, quando  $\mu = \alpha_1 = \delta = 1$  e  $\alpha_2 = 0$ , então  $c_2(i, u, j)$  representa a economia em distância ao servir o nó  $u$  na mesma rota que serve os nós  $i$  e  $j$ , em oposição ao serviço direto e individual de  $u$ . Se, por outro lado,  $\mu = \alpha_2 = \delta = 1$  e  $\alpha_1 = 0$ ,  $c_2(i, u, j)$  representará a economia em tempo. O incremento de  $\delta$  de 1 para 2 funciona como uma

penalização sobre as rotas compostas de um único nó. Esta e demais variantes de heurísticas de inserção *Push-forward* apresentam resultados de boa qualidade e são até hoje utilizadas em algoritmos mais sofisticados para o tratamento do PRVJT (veja, por exemplo, THANGIAH *et al.* 1994). Sua maior dificuldade consiste no fato de não apresentar limitações em relação ao número de rotas, por sua vez, o principal objetivo em problemas de roteirização com janelas de tempo.

Uma variante desta heurística de inserção seqüencial de SOLOMON (1987) para o PRVJT pode ser vista em POTVIN *et al.*, (1996).

### 2.3 Métodos de Duas Fases

Métodos de duas fases recebem este nome porque as operações de designação de clientes a veículos e a roteirização não são feitas simultaneamente, mas sim em duas etapas. Na primeira etapa são aplicados critérios para a formação de grupos de clientes, cujas restrições podem ser satisfeitas por um único veículo. A formação dos grupos reduz o problema original à  $m$  problemas menores e independentes. Há diversos métodos que formam estes grupos, dentre eles:

- ✓ TYAGI (1968) apresentou um método que agrupa pontos de demanda baseados no conceito do vizinho mais próximo. Ou seja, o próximo cliente não roteirizado a ser adicionado ao grupo é o mais próximo do último cliente adicionado;
- ✓ CHRISTOFIDES *et al.*, (1979), utiliza-se de uma regra similar à utilizada por TYAGI (1968), porém na primeira e na segunda etapa há a inclusão de um algoritmo de melhoramento de soluções.

Na segunda etapa dos métodos de duas fases constrói-se uma rota para cada um dos grupos de clientes criados na primeira etapa. A construção da rota se dá por um algoritmo adequado ao problema, por exemplo, algoritmos para o Problema do Caixeiro Viajante (LENSTRA *et al.*, 1986).

## 2.4 Métodos de Otimização Incompleta

Estes métodos consistem de algoritmos exatos modificados onde o processo de resolução é finalizado em uma solução viável sem garantia de otimalidade. Segundo PUREZA & FRANÇA (1990), “alguns métodos de otimização incompleta poderiam ser classificados como métodos de duas fases. Entretanto, em função de sua forte conexão com formulações de programação inteira, destaca-se como uma categoria separada”.

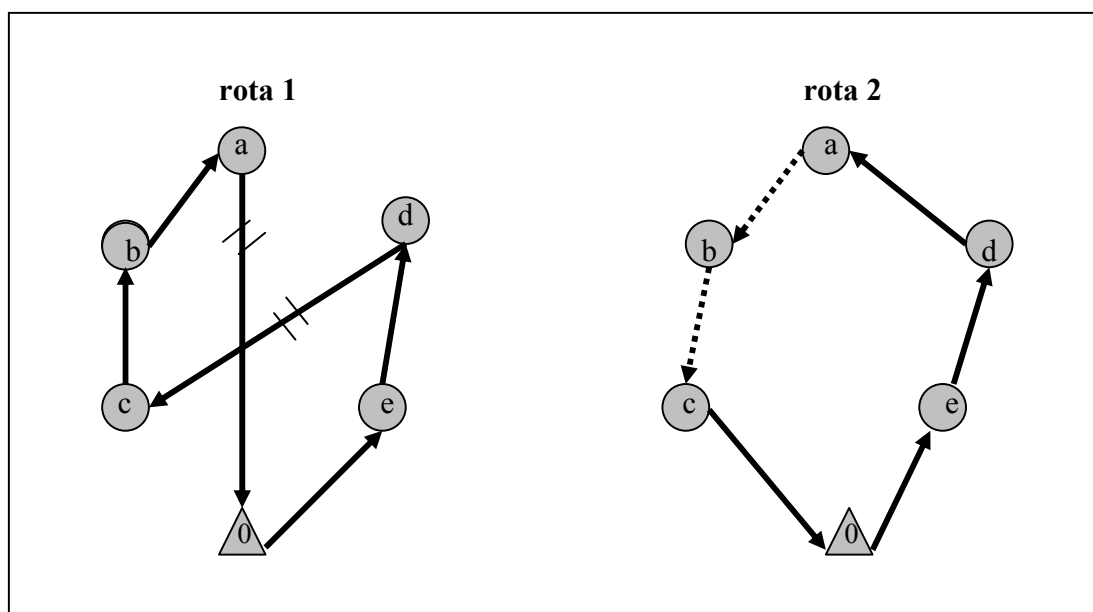
Exemplos de aplicação do método de otimização incompleta para o PRV e suas variantes pode ser encontrados em CHRISTOFIDES *et al.* (1979).

## 2.5 Métodos de Melhoria da Rota

Estes métodos heurísticos partem de uma rota viável, obtida por algum método heurístico descrito anteriormente, e tentam obter melhorias adicionais nesta rota inicial através de uma seqüência de alterações na estrutura da solução (*movimentos*) que mantêm a viabilidade das soluções. Tais técnicas de otimização geralmente visam à melhoria da função objetivo até que se atinja um *mínimo local*, ou seja, até que não haja movimentos a partir da solução corrente que resultem em ganhos na qualidade da solução. Neste caso, estes métodos são chamados de heurísticas de *busca local*, discutidos com mais detalhes no próximo capítulo.

Com este objetivo, vários mecanismos de melhoria de rotas têm sido propostos e dentre eles, o mais aplicado é a heurística de *troca de arestas r-opt* (LIN, 1965), onde  $r$  é o número de arestas da rota a serem trocadas. Em tais procedimentos são realizadas operações de substituição de  $r$  arestas, ou seja,  $r$  ligações definidas entre pares de nós, que fazem parte da solução corrente. Para cada rota gerada na etapa anterior, são computadas as aplicações viáveis destas operações, definindo assim os grupos de arestas eliminadas e adicionadas, e posteriormente escolhida aquela troca que provoca o maior ganho na função objetivo e factibilidade desta.

Este procedimento é repetido até que não haja nenhuma troca viável que melhore a solução corrente, com isso a solução resultante é chamada de *r-ótima*. Em geral, quanto maior o valor de *r*, maior é a qualidade da solução gerada. Entretanto, o esforço computacional dispendido na geração e avaliação de todas as possibilidades de *r* trocas cresce rapidamente com o número de nós. Isto limita a utilização de valores de *r* a 2 e 3. Em função desta limitação, CHRISTOFIDES & EILON (1969) desenvolveram um algoritmo onde operações 2-opt e 3-opt são realizadas alternadamente. A FIGURA 2.3 ilustra um movimento 2-opt, nesta figura é possível observar que o movimento 2-opt inverte o sentido de algumas arestas da rota assim pressupõe-se simetria de distâncias. (SOLOMON, 1987)

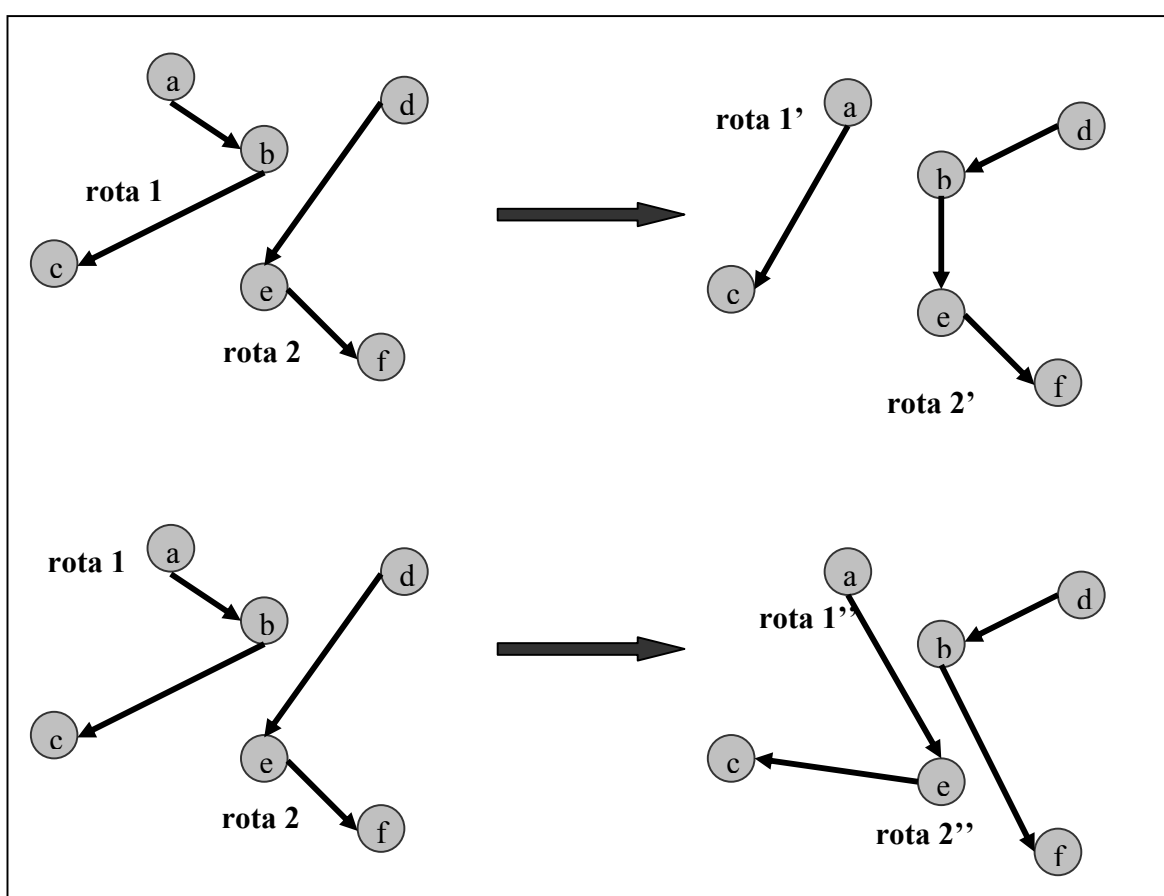


**FIGURA 2.3 - Aplicação de movimento 2-opt à rota 1. Rota 2 é resultante da eliminação das arestas  $a-0$  e  $c-d$  e adição das arestas  $d-a$  e  $c-0$ .**

Em contraste à heurística de troca de arestas *r-opt* limitadas a sistemas de uma *única* rota, DROR & LEVY (1986) desenvolveram um método que propõe operações de *troca e inserção de nós* a sistemas com *m* rotas. A FIGURA 2.4 ilustra os procedimentos de troca e inserção de nós em uma rota viável.



O critério adotado para a escolha de qual operação (troca ou inserção) a ser aplicada à rota é a maximização da economia em distância resultante da operação, bem como a factibilidade da rota em relação ao seu conjunto de restrições. Tomando o exemplo acima, a economia de inserir o nó  $b$  entre os nós  $d$  e  $e$  da rota 2 é dada por  $d(a, b) + d(b, c) + d(d, e) - (d(a, c) + d(d, b) + d(b, e))$ . Na troca do nó  $b$  pelo nó  $e$  entre as rotas 1 e 2, a economia resultante é  $d(a, b) + d(b, c) + d(d, e) + d(e, f) - (d(a, e) + d(e, c) + d(d, b) + d(b, f))$ .



**FIGURA 2.4 - Aplicação de movimentos de troca e inserção de nós entre as rotas**

Procedimentos de troca e inserção de nós foram posteriormente estendidos para o conceito mais amplo de  $\lambda$ -troca (OSMAN & CHRISTOFIDES, 1989; 1994), onde são considerados movimentos de um número, a princípio, arbitrário de nós que podem ser inseridos ou trocados entre quaisquer pares de rotas. Através de um par  $(R_p, R_q)$  se define a

ordem dos clientes a serem trocados. Por exemplo, quando  $\lambda = 2$ , o operador (0,1) indica a inserção de um cliente da rota  $q$  na rota  $p$ ; os operadores (1,0), (2,0) e (0,2) indicam a inserção de um cliente da rota  $q$  na rota  $p$ , dois clientes da rota  $q$  na rota  $p$ , e dois clientes da rota  $p$  na rota  $q$  ou mais clientes de uma rota para outra. De forma similar, os operadores (1,1), (1,2), (2,1) e (2,2) indicam a troca de um ou mais clientes entre as rotas  $p$  e  $q$ . Em THANGIAH *et al.* (1994), esta estrutura de vizinhança é aplicada no tratamento do PRVJT.

Note que apesar de mais poderoso que o proposto por DROR & LEVY (1986), o número de movimentos  $\lambda$ -troca a serem investigados pode crescer muito e tornar o tempo computacional nesta operação impraticável para a finalidade da aplicação.

Movimentos de troca e inserção de nós são geralmente aplicados até que um ótimo local seja alcançado, ou seja, quando todas as possíveis inserções ou trocas de nós (ou uma amostragem do conjunto total de movimentos) não resultarem em uma economia positiva. A grande limitação destes métodos é que em problemas combinatórios como o PRV, o ótimo local gerado não é geralmente o ótimo global.

Uma das maneiras de lidar com esta limitação no tratamento do PRV e suas variantes é a composição de heurísticas de construção e melhoria isto pode ser visto nos trabalhos de GILLET & MILLER (1974), RUSSEL (1995) e CORDONE & WOLFERCALVO (1997). Outra maneira de se lidar com esta limitação é a re-aplicação do procedimento a partir de outras soluções iniciais (*multiple starts*). O ótimo local com maior qualidade é então selecionado como solução resultante porém a maneira mais empregada atualmente é as chamadas *Meta-Heurísticas*.

As meta-heurísticas são metodologias desenvolvidas nos últimos 30 anos as quais, segundo PUREZA (1996), superpõem-se a métodos heurísticos, guiando a busca de soluções com vistas à superação da otimalidade local e à obtenção de soluções de qualidade superior. Dentre as mais promissoras, destacam-se *Simulated Annealing* (KIRKPATRICK, 1984), *Algoritmos Genéticos* (HOLLAND, 1975), *Greedy Randomized Adaptive Search Procedures - GRASP* (FEO & RESENDE, 1995) e *Busca Tabu* (GLOVER, 1989; GLOVER & LAGUNA, 1997).

Neste projeto será utilizada a meta-heurística *Busca Tabu*, pois de acordo com CORDEAU *et al.* (2002b) esta meta-heurística se destaca como a mais bem sucedida

no tratamento tanto do Problema de Roteirização de Veículos clássico como para o Problema de Roteirização com Janelas de Tempo. TAILLARD (1993) e ROCHAT & TAILLARD (1995) aplicam os princípios de busca tabu para o PRV e PRVJT obtendo, respectivamente, 85% e 48% das melhores soluções conhecidas atualmente.

## CAPÍTULO 3 META - HEURÍSTICA BUSCA TABU - CONCEITOS BÁSICOS

A meta-heurística Busca Tabu (BT) originou-se dos trabalhos de Fred Glover (1986), atingindo sua forma atual em 1987. BT possui como filosofia, desenvolver e explorar um conjunto de princípios de buscas inteligentes para a solução de um problema, utilizando a *memória* flexível como elemento básico, a qual cria estruturas que permitem explorar eficientemente a *história* de cada solução intermediária.

Os conjuntos de princípios que a compõem são: a modificação estrutural que provoca o movimento de uma solução a outra (*estrutura de vizinhança*), os elementos do movimento que estarão sujeitos a restrições (*atributos tabu*), o grau de restrição, e o período em que tais elementos estarão sujeitos às restrições (*período tabu*). Há situações onde o *status* tabu conferido a uma transição pode ser ignorado (*critérios de aspiração*).

Para se discutir a meta-heurística busca tabu é necessário, primeiramente, definir mais formalmente a técnica heurística de busca local ou busca em vizinhança. Esta técnica consiste de procedimentos iterativos que exploram um conjunto de soluções do problema pela repetida realização de movimentos  $\sigma$  de uma solução  $x$  para outra  $x'$ , localizada em uma amostragem total ou parcial da vizinhança de  $x$ . Assim, a vizinhança  $N(x, \sigma)$  de uma solução  $x$  é um conjunto de soluções que podem ser encontradas a partir de  $x$  por uma única operação  $\sigma$  ou movimento. Se não houver solução  $x'$  em  $N(x, \sigma)$  melhor do que  $x$ , então  $x$  é um ótimo local em relação ao tipo de movimento ( $\sigma$ ) adotado. Os passos do procedimento geral de busca local para um problema de minimização são apresentados a seguir.

### PROCEDIMENTO GERAL DE MÉTODOS DE BUSCA LOCAL

1. Gere uma solução inicial viável  $x$ . Faça  $x^* = x$ .
2. Faça  $fim = \text{falso}$ .  
Enquanto  $fim = \text{falso}$  repita a seqüência de passos:
3. Gere a vizinhança (total ou parcial)  $N$  da solução  $x^*$ , definida por movimentos do tipo  $\sigma$ .
4. Obtenha  $x'$  em  $N(x^*, \sigma)$  tal que  $f(x') < f(x^*)$ .
5. Se houver  $x'$ , faça  $x^* = x'$ . Caso contrário, faça  $fim = \text{verdadeiro}$  e retorne  $x^*$ .

**FIGURA 3.1 - Procedimento geral de métodos de busca local**

A busca local possui como inconveniente o fato do processo ser interrompido no primeiro ótimo local encontrado, o qual geralmente está ainda distante do ótimo global. Por este motivo, é interessante aceitar a deterioração do valor da função objetivo. Uma consequência da decisão de se aceitar uma solução de pior qualidade é a possibilidade de formação de ciclos, isto é, o retorno a um ótimo local já visitado anteriormente.

Uma das alternativas para se evitar a ciclagem é compor uma lista de soluções já visitadas. Isso implica em requerimentos de memória elevados, e é raramente praticado. A alternativa mais comum é a de incorporar uma estrutura de memória que armazena características ou atributos de movimentos que representem a reversão de movimentos já realizados. Movimentos candidatos à efetivação que possuem estes atributos estão sujeitos à proibição durante um número pré-definido de iterações.

Segundo LAGUNA (1995), “BT pode ser caracterizada como um método de vizinhança dinâmica. Isto significa que a vizinhança da solução corrente  $x$  não é um conjunto estático, mas um conjunto com elementos que são modificados de acordo com a história da busca. A utilização de formas de memória flexível que controlam o processo de busca é o tema central da metodologia de busca tabu. Uma importante distinção entre as formas de memória flexível é feita entre memória de curto prazo e memória de longo prazo. Estes conceitos são apresentados a seguir”.

### 3.1 Memória de Curto Prazo

A memória de curto prazo procura eliminar ou reduzir o risco de ciclagem pela penalização de movimentos que provocariam o retorno a soluções já encontradas. O tipo de memória de curto prazo utilizada é chamado de *memória baseada em recentidade* a qual armazena atributos de soluções que foram alterados no passado recente.

Atributos podem ser arestas que são adicionadas ou eliminadas de um movimento como no *Problema de Roteirização de Veículos*, ou elementos que trocam de posições como no caso do *Problema de Sequenciamento de Tarefas*, ou itens que são transferidos de um conjunto a outro como no *Problema da Mochila*.

De acordo com LAGUNA (1995), “para a exploração desta memória, selecionam-se atributos que ocorreram nas melhores soluções recentemente visitadas classificando-os como tabu-ativos. Os movimentos que contêm elementos tabu-ativos, em particular, combinações de atributos tabu-ativos, são denominados tabu. Isto previne que certas soluções encontradas anteriormente sejam incorporadas à vizinhança  $N(x)^*$  e conseqüentemente visitadas novamente”.

A eficácia da memória de curto prazo depende diretamente da seleção de atributos, da regra que designa elementos como tabu-ativos e do tempo que um elemento se mantém tabu-ativo. Além destes fatores, é importante a incorporação de estratégias que permitam a realização de movimentos tabu sob condições especiais, já que a classificação tabu-ativa pode abranger soluções boas não visitadas. Esta ferramenta é denominada *critérios de aspiração*.

Cada regra que classifica os atributos como tabu-ativos possui um grau de restritividade associado, assim dependendo de como ela é definida, é permitido que um número maior ou menor de movimentos esteja disponível para escolha. Considere um problema de roteirização de veículos. Uma dada solução pode ser representada por rotas de clientes (nós) ligadas por arestas. Os movimentos podem consistir de arestas adicionadas e eliminadas, assim uma possibilidade para a regra de ativação tabu do movimento seria a fixação de um número máximo de arestas tabu-ativos (adicionadas e/ou eliminadas)

toleradas no movimento corrente. Neste caso, quanto menor o valor fixado de arestas tabu-ativas toleradas no movimento corrente mais restritiva será a regra de ativação tabu.

A duração que um atributo permanece tabu-ativo, ou seja, o número de iterações, é chamado de período tabu. O período tabu mais adequado pode variar para diferentes tipos ou combinações de atributos e também para diferentes intervalos do tempo ou estágios da busca.

Segundo PUREZA (1996), “uma forma de identificar um período tabu eficiente para uma dada classe de problema e para uma regra de ativação tabu definida é observar a ocorrência de ciclagem; quando se tem um período tabu excessivamente curto ou longo pode haver deterioração na qualidade da solução. A deterioração é geralmente causada pela proibição extremada dos movimentos. De fato, um período tabu longo demais pode resultar no término precoce da busca. O período tabu mais adequado está em uma faixa intermediária entre os dois extremos (curto e longo)”.

Períodos tabu curtos permitem a exploração de soluções “próximas” a um ótimo local, enquanto períodos longos são preferíveis quando é necessário guiar a busca para longe das imediações de ótimos locais. A variação do período tabu durante a busca é uma maneira de induzir um equilíbrio entre o exame mais apurado de uma região e a transferência do processo para regiões diferentes do espaço de solução. Existem várias maneiras de se implementar um período tabu. Estas implementações podem ser classificadas como períodos tabu dinâmicos sistemáticos ou aleatórios e períodos tabu estáticos.

Há duas alternativas de períodos tabu aleatórios; ambas utilizam uma faixa definida, previamente, pelos parâmetros  $t_{\min}$  e  $t_{\max}$ . O período tabu  $t$  é selecionado aleatoriamente dentro desta faixa, geralmente seguindo uma distribuição uniforme. Na primeira alternativa, o período escolhido é mantido por  $\alpha\tau_{\max}$  iterações, quando então um novo período é selecionado. Na segunda alternativa, seleciona um novo  $t$  para cada atributo que se torna tabu em uma dada iteração. Ambas as alternativas se beneficiam de uma estrutura de memória que armazena o número da iteração em que o atributo perde seu *status* tabu.

O período tabu sistemático consiste em criar uma seqüência de valores de períodos definidos por  $t_{\min}$  e  $t_{\max}$  que são então utilizados para designar o valor de período tabu corrente. Já o período tabu estático é determinado por um único valor previamente definido.

Como a classificação de um movimento como tabu pode impedir a seleção de boas soluções não visitadas, critérios de aspiração definem situações em que as regras de ativação tabu são ignoradas. Ou seja, são critérios que determinam a liberação do *status* tabu do movimento.

De acordo com PUREZA (1996), o tipo padrão de critério de aspiração consiste na revogação da classificação tabu de um movimento candidato sempre que todos os movimentos disponíveis estão tabu-ativos e não se tornam admissíveis por outros critérios de aspiração. Um critério de aspiração sempre adotado em implementações tabu revoga a classificação tabu quando o movimento candidato resulta em uma solução melhor que a melhor obtida até o momento (*critério de objetivo global*). Um critério de aspiração sofisticado incorpora o conceito de *influência*. O grau de influência de um movimento consiste em uma medida de quão extensivamente a estrutura ou viabilidade da solução foi alterada com a realização do movimento.

Implementações tabu que incorporam apenas memória de recentidade são chamadas de algoritmos de busca tabu simples. Os passos do procedimento geral de busca tabu simples são apresentados a seguir.



### PROCEDIMENTO GERAL DE BUSCA TABU SIMPLES

1. Gere uma solução inicial  $x$ . Faça  $x^* = x$ .
2. Faça  $it$  (número da iteração corrente) = 0.
3. Enquanto algum critério de parada não seja satisfeito, repita:
  - 3.1. Faça  $it = it + 1$ .
  - 3.2. Gere a vizinhança (total ou parcial)  $N$  da solução  $x$ , definida por movimentos do tipo  $\sigma$ .
  - 3.3. Obtenha  $x'$  em  $N(x, \sigma)$  com melhor avaliação tal que o movimento associado seja não tabu ou satisfaça algum dos critérios de aspiração adotados.
  - 3.4. Faça  $x = x'$ . Se  $f(x) < f(x^*)$ , faça  $x^* = x$ .
4. Retorne  $x^*$ .

**FIGURA 3.2** Procedimento geral de busca tabu simples

Dentre os critérios de parada mais utilizados, incluem-se:

- ✓ Número máximo de iterações.
- ✓ Número máximo de iterações sem melhoria.
- ✓ Inexistência de movimentos disponíveis.

## 3.2 Memória de Longo Prazo

Em muitas aplicações, a memória de curto prazo é suficiente para assegurar soluções de qualidade, conforme relatado em PUREZA & MORABITO (2005). Entretanto, a fim de obter ganhos adicionais no desempenho da busca, é utilizada a memória de longo prazo e suas estratégias associadas. Em tais estratégias, a vizinhança pode conter soluções não presentes originalmente, geralmente consistindo de *soluções de elite* (ótimos locais de alta qualidade) encontradas em vários pontos do processo de busca e utilizadas como elementos de estratégias de *intensificação* da busca em regiões promissoras ou como

elementos de estratégias de *diversificação* da busca quando possibilidades de melhoria são mínimas.

A memória de longo prazo é comumente baseada na *freqüência* de um certo atributo no movimento. Este tipo de memória complementa a informação fornecida pela memória baseada em recentidade. Tais medidas de freqüências consistem de razões cujos numeradores representam o número de ocorrências de dois tipos de eventos:

- ✓ Medida de residência: número de vezes que um atributo reside em um conjunto de soluções, por exemplo, no problema de roteirização de veículos esta medida representa o número de vezes que uma aresta  $i$  pertenceu a solução;
- ✓ Medida de transição: número de vezes que um atributo é alterado, para o problema de roteirização de veículos representa o número de vezes que uma aresta  $i$  foi trocada com uma aresta  $j$ ;

Dentre as possibilidades de denominadores incluem-se:

- ✓ Número total de ocorrências de todos os eventos representados pelos numeradores;
- ✓ Soma dos numeradores;
- ✓ Valor máximo do numerador;
- ✓ Valor médio do numerador.

As razões descritas acima resultam em dois tipos de freqüência: (1) a *freqüência de transição*, que mantém o histórico de quão freqüentemente um atributo é alterado e, (2) a *freqüência de residência* que armazena a duração relativa da ocorrência dos atributos nas soluções geradas.

Se a freqüência de residência for alta, ou seja, um atributo aparece várias vezes durante o processo de busca, isto pode indicar que este atributo é altamente atraente, ou pouco atraente, caso contrário. Por outro lado, se a freqüência de residência é baixa para um conjunto de soluções contendo tanto soluções de alta como de baixa qualidade, então é possível que a busca esteja aprisionada em uma dada região, indicando a necessidade de eliminar ou incorporar algum atributo para que se possa permitir a diversificação da busca.

Se a frequência de transição for alta, é possível que o atributo associado entre e saia constantemente das soluções, funcionando como um fator de ajuste fino.

### **3.3 Estratégias de Intensificação da Busca**

Estratégias de intensificação são normalmente baseadas na modificação de regras de escolha de um movimento com o objetivo de encorajar combinações de características de soluções historicamente boas. Esta estratégia pode iniciar um retorno a regiões atraentes a fim de examiná-la com maior detalhe.

Há duas alternativas para a estratégia de seleção de soluções de elite que apresentam excelente desempenho. Uma introduz uma medida de diversificação a fim de assegurar que as soluções armazenadas em todas as iterações diferem uma das outras por um grau específico, e a partir da melhor solução encontrada se exclui toda a memória de curto prazo antes de continuar a busca novamente.

Uma alternativa é manter uma lista de tamanho limitado e adicionar uma nova solução se esta é melhor que qualquer outra já encontrada. O último membro corrente da lista é sempre o escolhido como uma base para continuidade da busca. A memória de curto prazo que acompanha esta solução é mantida e o primeiro movimento também proíbe o movimento anteriormente feito a partir desta solução, de maneira que um novo caminho será percorrido. Esta segunda alternativa está relacionada com uma estratégia que prossegue com a busca a partir dos vizinhos não visitados das soluções previamente geradas. Tal estratégia armazena informações acerca da qualidade destes vizinhos e seleciona um conjunto de elite, no qual a atenção se restringe aos tipos específicos de soluções. Estas soluções podem incluir os vizinhos de ótimos locais ou vizinhos de soluções visitadas em passos imediatamente anteriores à obtenção de ótimos locais.

### 3.4 Estratégias de Diversificação da Busca

A incorporação do conceito de memória de longo prazo em estratégias de diversificação visa orientar o processo de busca a explorar regiões ainda não exploradas. Geralmente, a exploração de novas regiões está baseada na modificação de regras de escolha dos movimentos; com isso é possível se incorporar atributos raramente utilizados nas soluções. Alternativamente, estas estratégias podem introduzir tais atributos com o reinício parcial ou total do processo de busca. Estratégias de diversificação são particularmente úteis quando a obtenção de melhores soluções exige a transposição de barreiras na topologia do espaço de solução.

As regras de escolha dos movimentos são alteradas para impedir ou penalizar a realização de movimentos cujos atributos têm alta frequência de residência nas soluções visitadas. Isto é feito com a utilização de uma variável não negativa *penalidade*, por sua vez, função de medidas de frequência, e um parâmetro de diversificação ajustável *d* cujo valor indica o grau de diversificação pretendido. O valor do movimento passa a ser dado por:

$$\text{Valor\_do\_movimento}' = \text{Valor\_do\_movimento} + d * \text{Penalidade}$$

Como visto no capítulo anterior, em problemas de roteirização de veículos é usual utilizar movimentos de inserção ou troca dos nós para explorar a vizinhança total. As trocas implicam em trocar nós de diferentes rotas. Inserções tomam um nó de uma rota e o coloca em uma rota distinta. Como as trocas não modificam a cardinalidade da rota, considera-se que tenham menor influência que movimentos de inserção.

Portanto, além da regra tradicional de selecionar o melhor movimento não-tabu a cada iteração, pode-se obter diversificação alterando-se o valor do movimento com a proibição da seleção de trocas de nós que não melhoram a solução após um período de estabilização da busca. Neste caso, *penalidade* pode assumir o valor 0 (movimentos de inserção) ou  $\infty$  (movimentos de troca).

Note que a modificação do valor do movimento pode ser também utilizada em estratégias de intensificação. São favorecidos movimentos cujos atributos têm alta frequência de residência em soluções historicamente de alta qualidade. Neste caso, a variável *penalidade* é substituída pela variável *incentivo*. O valor do movimento passa a ser dado por:

$$\text{Valor\_do\_movimento}' = \text{Valor\_do\_movimento} - f * \text{Incentivo}$$

onde  $f$  é o parâmetro de intensificação ajustável.

Para alguns tipos de problemas, o ajuste entre a variável  $d$  e a variável *penalidade* pode resultar em uma estratégia chamada *oscilação estratégica* em torno da fronteira de viabilidade. A oscilação estratégica é uma estratégia vinculada às origens da busca tabu, pois a partir dela se atinge um intercâmbio efetivo entre as estratégias de intensificação e diversificação. A estratégia de oscilação orienta a busca em direção a um nível crítico. Este nível crítico representa um ponto onde o método normalmente pararia. Entretanto, ao invés de parar quando esta fronteira é atingida, as regras de seleção de movimentos são alteradas, de forma a permitir que a região definida pelo nível crítico seja atravessada. A estratégia então prossegue por uma profundidade especificada além da fronteira de oscilação, para então retornar depois de atingida a profundidade especificada. A fronteira de oscilação é novamente atingida e atravessada, desta vez na direção contrária e o método prossegue para um novo ponto de retorno.

Em algumas situações, implementações que utilizam a estratégia de oscilação estratégica apresentam maior efetividade com a incorporação do *princípio da otimalidade próxima*. O propósito deste princípio é de que boas soluções em um certo nível têm boas chances de estarem próximas a boas soluções em um nível adjacente. O termo nível tanto pode ser referir a um estágio de um processo construtivo ou destrutivo, como a uma medida de distância de uma fronteira já especificada.

O *princípio da otimalidade próxima* é utilizado com o intuito de manter a busca em cada nível por um número especificado de iterações. Um movimento para o próximo nível é então iniciado a partir da melhor solução obtida. A entrada no nível adjacente pode ser feita forçando os valores de uma função a se manterem em uma dada

faixa. O processo pode ser mantido em um nível especificado, utilizando-se uma vizinhança cujos movimentos tenham seus valores funcionais dentro dos limites estabelecidos.

Estratégias de intensificação e diversificação podem ser integradas através da abordagem de religação de caminho (*path relinking*), que consiste em explorar trajetórias que conectam soluções de elite. Assim, espera-se obter novas soluções de qualidade, traçando-se trajetórias que se iniciam em uma solução de elite e seguem em direção a outras soluções de elite através do favorecimento de movimentos que introduzem seus atributos e penalizando-se a inclusão de atributos que foram eliminados durante a trajetória. Todas as outras considerações tornam-se subordinadas ao objetivo de criar uma boa composição de atributos na solução corrente.

A composição de atributos a cada passo é determinada pela escolha do melhor movimento a partir do conjunto restrito de movimentos que incorporam um número máximo (ou um valor ponderado máximo) dos atributos das soluções que devem ser atingidas. Apenas as formas mais vigorosas de critérios de aspiração (como, por exemplo, o critério do objetivo global) podem revogar este tipo de regra de escolha.

## **CAPÍTULO 4 ALGORITMOS DE BUSCA TABU PARA O PROBLEMA DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS**

O relato de implementações de busca tabu para o Problema de Roteirização de Veículos clássico teve início com o trabalho de WILLARD (1989) onde neste trabalho, o PRV é inicialmente transformado em um Problema Caixeiro Viajante pela replicação do depósito e a busca heurística compreende soluções vizinhas que possam ser alcançadas por meio de trocas de arestas 2-opt e 3-opt.

A partir de então, outros algoritmos de busca tabu foram desenvolvidos para o PRV e suas extensões, mas para relevância do corrente projeto, apenas nove destes são aqui ressaltados. Estes algoritmos<sup>2</sup> foram selecionados em função da qualidade das soluções geradas para o conjunto de 14 instâncias clássicas de 50 a 199 nós, propostas em CHRISTOFIDES *et al.* (1979) e para o conjunto de 56 instâncias clássicas de 100 nós, propostas por SOLOMON, 1987. Ambos os conjuntos são descritos na seção 5.2.1.1 e 5.3.1 do capítulo 5.

### **4.1 Algoritmo de OSMAN (1993)**

O algoritmo de Osman utiliza a heurística de construção de CLARKE & WRIGHT (1964) para geração da solução inicial. É empregado, então, o procedimento de  $\lambda$ -troca (mencionado na seção 2.5 do capítulo 2) para a definição da vizinhança de busca.

Nesta implementação, considera-se  $\lambda = 1$ , o qual compreende dois tipos de processos para a geração de soluções vizinhas idênticos aos propostos por DROR & LEVY

---

2- Para melhor entendimento da abordagem HTSA e dos algoritmos descritos neste capítulo a autora não ordenou estes cronologicamente.

(1986). O primeiro processo é o *deslocamento ou inserção de um nó* de sua rota original  $R_p$  para uma rota alternativa  $R_q$ . Este processo permite reduzir o número de rotas quando a rota original do nó deslocado é constituída apenas por ele. O segundo processo compreende a *troca* de um nó de uma rota  $R_p$  por um nó de uma rota  $R_q$ .

São empregadas duas estratégias para a escolha de soluções na vizinhança: (i) a estratégia *melhor-admissível* (*best-admissible* - BA) que seleciona o movimento que resulta na melhor solução não-tabu dentre todas as opções, e (ii) A estratégia *primeiro-melhor-admissível* (*first best admissible* - FBA) que aceita o primeiro movimento não tabu que resulta em melhoria. Caso não haja movimentos de melhoria, é selecionado o que resulta em menor deterioração da função objetivo.

Para a estratégia FBA, foi utilizada uma estrutura de dados especial, permitindo uma única avaliação completa dos movimentos na primeira iteração. Deste ponto em diante, são feitas apenas atualizações do custo do melhor movimento das rotas envolvidas na iteração anterior. Isso permitiu uma grande economia de tempo computacional, sem sacrifício da qualidade da solução.

O período tabu é obtido a partir da geração de um número  $t_s$ , por sua vez calculado por uma função que incorpora quatro características do problema (número de clientes, número de veículos, razão entre as demandas dos clientes, capacidades dos veículos) e a estratégia de seleção (FBA ou BA). Esta função foi elaborada com a regressão dos melhores tamanhos de listas obtidos em exemplos de pequeno tamanho. A partir deste valor, são determinados três valores de período tabu ( $t_s, 0.9t_s, 1.1t_s$ ). Cada valor opera por um número específico de iterações. Uma vez que todos os três valores tenham sido utilizados, uma ordem aleatória de utilização destes valores é obtida, dando-se continuidade ao processo de busca. Estas estimativas do período tabu e o esquema de variação foram usados apenas para os exemplos de maior tamanho.

O critério de parada é um número máximo de iterações sem melhoria. Este algoritmo foi aplicado as 14 instâncias-teste de Christofides resultando em 5 soluções de custos iguais aos das melhores soluções reportadas para este 5 casos. Em média, os custos apresentaram um desvio percentual em relação às melhores soluções conhecidas de 0,82%.



## 4.2 Algoritmo de TAILLARD (1993)

O algoritmo de Taillard utiliza dois *métodos de decomposição* do problema inicial em subproblemas, fazendo com que estes possam ser resolvidos independentemente. O algoritmo é paralelizado de maneira que cada subproblema gerado é tratado por um processador. A busca tabu é então aplicada aos problemas resultantes da decomposição.

A solução inicial consiste na designação de um único veículo para cada cliente. Dois movimentos são realizados: a *troca* de nós pertencentes a rotas distintas e a *inserção* de um nó em uma rota distinta. Movimentos que correspondem à reversão de um movimento realizado nas últimas  $t$  iterações são classificados como tabu-ativo, a não ser que gerem uma solução melhor do que a melhor até o momento (critério de aspiração global).

O autor observa que se a implementação tabu utilizar apenas esta estrutura de memória, as cidades próximas à garagem são mais freqüentemente movidas que as demais. Para suprimir este comportamento, o algoritmo inclui uma estratégia de diversificação que modifica o valor dos movimentos, a fim de que movimentos mais freqüentes sejam penalizados (veja seção 3.4 do capítulo 3). Caso o movimento resulte em uma solução de maior qualidade que a melhor até o momento, esta penalização é revogada.

Para se avaliar o custo de um movimento de inserção ou troca, dois problemas do Caixeiro Viajante precisam ser resolvidos, um para cada rota modificada. Para este fim, foi utilizado o mais simples método heurístico de inserção: se a cidade  $c$  é movida da rota 1 para a rota 2, vai-se diretamente, na rota 1, da cidade que precedia  $c$  à cidade que sucedia  $c$ . Na rota 2,  $c$  é inserido na posição que resulta no menor aumento do comprimento de 2 (no caso de movimentos de trocas, ambas as cidades são inicialmente removidas e depois recolocadas).

Como este procedimento não produz rotas ótimas em geral, é aplicado um método que resolve otimamente cada PCV a cada 200 iterações, ou quando é encontrada uma solução a menos de 0,1% do custo da melhor solução obtida até então.

Conforme citado anteriormente, o algoritmo de Taillard utiliza dois métodos de decomposição do problema inicial. O primeiro método consiste em gerar partições a partir das *coordenadas polares dos clientes*, onde é designado aproximadamente o mesmo número de clientes a cada região. Cada região é então tratada como um PRV independente, o qual é resolvido pelo processo de busca tabu. Entretanto, as soluções geralmente resultam em um número muito grande de veículos. Por esta razão, avalia-se um número total de veículos necessário para satisfazer a demanda de cada cliente e este total é distribuído entre as regiões. Cada subproblema pode construir um número de rotas até o número de veículos designado para esta região. Esta estratégia gera soluções ainda pouco satisfatórias, pois é comum se verificar veículos com capacidades subutilizadas e clientes com demandas insatisfeitas.

Em função disso, após ter sido realizado um número específico de iterações para cada subproblema, as rotas, os clientes e veículos são agrupados de outra forma gerando assim novos subproblemas. Estes novos subproblemas são gerados usando o segundo método de decomposição, similar ao primeiro, mas que utiliza o *centro de gravidade das rotas* ao invés das *coordenadas polares dos clientes*. O número de iterações antes de uma nova divisão é proporcional ao número de decomposições já realizadas, ao número total de clientes e inversamente proporcional ao número de regiões.

A aplicação deste algoritmo resultou em 11 soluções com custo iguais aos das melhores soluções reportadas até o momento. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 0,06%.

### **4.3 Algoritmo de ROCHAT & TAILLARD (1995)**

O algoritmo de Rochat e Taillard utiliza uma técnica que primeiro permite diversificar a busca visando à exploração de soluções diferentes entre si, a seguir intensifica a busca para que se possa identificar soluções de qualidade, e finalizando na paralelização de uma busca local. Este algoritmo foi inicialmente aplicado ao PRV clássico sendo

posteriormente estendido para o tratamento de problemas de roteirização de veículos com Janela de Tempo.

A busca tabu utilizada baseia-se no algoritmo TAILLARD (1993), onde são incorporadas algumas melhorias no procedimento de decomposição do problema inicial e no procedimento de otimização das rotas.

No algoritmo de ROCHAT & TAILLARD (1995) é desenvolvido o procedimento de *memória adaptativa*, uma das idéias mais úteis na área de meta-heurísticas dos últimos anos. Este procedimento de memória adaptativa mantém um conjunto de boas soluções produzidas por uma heurística, o qual é dinamicamente melhorado com o acréscimo de novas soluções de alta qualidade e pela remoção de soluções de menor qualidade. Novos elementos são gerados pela recombinação de boas soluções armazenadas.

Em sua fase inicial, a busca é diversificada ao gerar com a busca local,  $I$  soluções diferentes entre si. Ao se gerar soluções iniciais com a busca local espera-se obter todas as informações necessárias para se criar soluções de alta qualidade. A geração de  $I$  soluções iniciais cria um conjunto  $T$  de rotas. Em uma segunda fase, o objetivo é extrair boas rotas de  $T$  e otimizá-las.

O procedimento considera que se uma solução  $s$  contém rotas que pertencem a uma boa solução então o valor da função objetivo de  $s$  é provavelmente melhor do que o de uma solução que não contém rotas de uma boa solução. Assim, na implementação desta fase, cada rota é inicialmente rotulada com o valor da função objetivo da solução a qual pertence. O conjunto  $T$  é ordenado de forma crescente pelo valor dos rótulos e as rotas com somente um cliente são removidas do conjunto  $T$  desde que eles não contenham informações interessantes.

Do conjunto  $T$  são escolhidas rotas de forma probabilística, ou seja, através das informações geradas ao longo do processo de busca juntamente com uma medida de atratividade que, neste caso, é dada priorizando-se a escolha de rotas com rótulos de menor valor e ignorando-se rotas que incluem clientes pertencentes a rotas já selecionadas. Esta escolha é repetida até que não haja mais rotas possíveis para serem selecionadas do conjunto  $T$ . As rotas escolhidas formam o conjunto  $S$  que pode não conter todos os clientes do problema. Neste caso,  $S$  é uma solução parcial. Para se gerar uma solução viável a partir

de  $S$ , considera-se que os clientes não pertencentes às rotas de  $S$  como sendo pertencentes a PRVs independentes que podem ser resolvidos por uma busca local. Estas rotas independentes são adicionadas ao conjunto  $S$ , criando assim uma solução viável utilizada como solução inicial da busca local para o PRV.

Este algoritmo foi aplicado às 14 instâncias-teste de CHRISTOFIDES (1979), as 2 instâncias proposta por FISHER (1994) com 71 e 134 clientes, a 1 instância proposta por TAILLARD (1993) com 385 clientes e ao conjunto de 56 instâncias do PRVJT de 100 clientes proposto por SOLOMON (1987), por sua vez, descrito na seção 5.3.1 do capítulo 5. A aplicação do algoritmo às instâncias de Christofides resultou na obtenção de soluções com custos iguais em 13 casos quando comparados as melhores soluções reportadas até o momento. Em média, os custos apresentaram um desvio percentual em relação às melhores soluções conhecidas de 0,01%. Para as instâncias propostas por Fisher o algoritmo obteve uma melhor solução para a instância de 134 clientes dentre as reportadas na literatura e para a instância de Taillard com 385 clientes o algoritmo também obteve uma nova solução dentre as reportadas na literatura.

Para os problemas propostos por Solomon, o algoritmo padrão resultou em 19 soluções com custos menores em relação às então melhores soluções reportadas. Em média, os custos apresentaram um desvio percentual em relação às melhores soluções conhecidas de 1,85%.

#### **4.4 Algoritmo TABUROUTE (GENDREAU *et al.*, 1994)**

A principal diferença entre este algoritmo e os demais algoritmos descritos neste capítulo é permitir a geração de soluções inviáveis. A cada solução (viável ou inviável) é associado um valor de função objetivo que inclui termos de penalidades tanto para a capacidade do veículo como para a duração de rotas que sejam excedidas.

TABUROUTE possui como procedimento central *Search* (P) o qual busca melhorias a partir de uma dada solução. O movimento básico deste procedimento consiste em deslocar um nó da rota corrente para uma rota alternativa utilizando duas heurísticas (GENI e US) desenvolvidas pelos autores para o PCV (GENDREAU & HERTZ &

LAPORTE, 1992). GENI é uma rotina de inserção generalizada, mais poderosa que outros procedimentos de inserção padrão, dado que um vértice só pode ser inserido em rotas que contenham algum dos seus vizinhos mais próximos. Cada inserção é executada simultaneamente com uma re-otimização local da rota corrente. US é um procedimento de pós-otimização que sucessivamente remove e re-insere cada vértice, usando GENI.

A solução inicial da heurística TABUROUTE é gerada construindo-se uma rota que passa por todos os nós através das heurísticas GENI e US (GENIUS). A rota é então particionada em um número máximo de  $M$  rotas, de maneira que há a possibilidade da última rota ser inviável. Inicialmente,  $\lambda$  tentativas de soluções iniciais são geradas e SEARCH é aplicada a cada uma delas por um número limitado de iterações. A solução mais promissora é selecionada como ponto de partida de TABUROUTE e o procedimento SEARCH é chamado.

Para que o movimento seja efetivamente considerado para inserção, ele deve preencher certos requisitos. Se o movimento é tabu e  $S'$  (melhor solução da vizinhança) é viável (inviável) ele só é aceito como candidato potencial se o custo de  $S'$  for menor que o custo da melhor solução viável (inviável) obtida até então. Se o movimento é não tabu, a ele é associado um custo igual ao seu custo real se o movimento resultar em um custo inferior ao da solução  $S$ , caso contrário, adiciona-se um termo de penalidade ao custo real da solução  $S'$  proporcional à frequência que o movimento foi realizado. O candidato  $\bar{S}$  com o menor custo associado é então identificado.

Ao logo do algoritmo o procedimento SEARCH(P) é utilizado três vezes, e em cada ocasião são utilizados valores selecionados de parâmetros com propósitos definidos. A primeira chamada ( $P_1$ ) utiliza a vizinhança total  $W$  (conjunto total de cidades), geralmente resultando nas melhorias mais significativas em relação à solução inicial. A segunda chamada ( $P_2$ ) também utiliza esta vizinhança, porém procura intensificar a busca localmente, concentrando-se em subconjuntos específicos de cidades das melhores soluções viáveis obtidas. Se nenhuma solução viável tiver sido gerada, a busca se concentra nas melhores soluções inviáveis. Em ambas as etapas,  $q$  (o número de nós do conjunto  $W$ ) é escolhido de forma a assegurar uma alta probabilidade de selecionar pelo menos um vértice de cada rota. A terceira chamada ( $P_3$ ) utiliza metade da vizinhança  $W$ . Os

nós selecionados são aqueles com a maior frequência, e como estes nós foram movidos frequentemente, há chances de melhorias se forem utilizados. Nesta fase, o valor de  $q$  é igual a  $W$ , ou seja, todos os nós permitidos para inserção são candidatos.

Ao se aplicar à versão padrão de TABUROUTE às 14 instâncias de Christofides, foram obtidas soluções com custos iguais aos das melhores soluções reportadas em 4 casos. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 0,82%. Execuções com variações de valores de parâmetros resultaram um desvio percentual de 0,21%.

#### 4.5 Algoritmo UTSA (CORDEAU *et al.*, 2001)

O algoritmo UTSA (*Unified Tabu Search Algorithm*) foi inicialmente desenvolvido para tratar o problema de roteirização de veículos periódico e o problema de roteirização de veículos multi-depósito (CORDEAU *et al.*, 1997), sendo posteriormente modificado e estendido para os problemas: PRV clássico, PRV com multi-depósito e PRV periódico acrescidos, em todos os problemas, da restrição de janela de tempo. (CORDEAU *et al.*, 2001)

UTSA incorpora alguns aspectos do algoritmo TABUROUTE (GENDREAU *et al.*, 1994), incluindo a mesma estrutura de vizinhança, o procedimento GENI de inserção e a exploração de soluções inviável através do uso de penalização na função objetivo. A função objetivo é penalizada através da utilização de parâmetros ( $\alpha$ ,  $\beta$  e  $\gamma$ ) auto-ajustáveis. Contudo, UTSA não utiliza a estratégia de intensificação de TABUROUTE e adota períodos tabu fixos.

O algoritmo parte de uma única solução inicial  $s$ , sendo a vizinhança  $N(s)$  definida pela aplicação de um operador simples que remove um cliente  $i$  servido pelo veículo  $k$  (atributo  $(i, k)$ ) e o designa a um veículo  $k'$  onde  $k' \neq k$  (atributo  $(i, k')$ ). Quando o cliente  $i$  é removido da rota servida pelo veículo  $k$ , a rota é simplesmente re-conectada pela ligação do sucessor e predecessor do cliente  $i$ . A inserção do cliente  $i$  na rota servida

pelo veículo  $k'$  é feita entre dois clientes consecutivos, visando à minimização do valor da função objetivo. O atributo  $(i, k)$  é declarado tabu por um número fixo de iterações.

Para diversificação da busca, cada solução  $\bar{s} \in N(s)$  tal que  $f(\bar{s}) \geq f(s)$  é penalizada por um fator proporcional à frequência de seus atributos e a um fator de escala. Seja  $p_{ik}$  o número de vezes que um atributo  $(i, k)$  tem sido encontrado em soluções durante o processo de busca, então a penalidade  $p(\bar{s}) = \lambda c(\bar{s}) \sqrt{nm} \sum_{(i,k) \in B(\bar{s})} p_{ik}$  é adicionada a função  $f(\bar{s})$ . Se  $f(\bar{s}) < f(s)$  então  $p(\bar{s}) = 0$ . O fator de escala  $c(\bar{s}) \sqrt{nm}$  introduz uma correção para o ajuste das penalidades com respeito ao custo total da solução e o tamanho do problema medido pelo número de atributos possíveis. O parâmetro  $\lambda$  é usado como controle da intensidade da diversificação. Esta penalidade tem o efeito de dirigir a busca para regiões menos exploradas do espaço de busca

UTSA gera soluções de alta qualidade para todas as variantes de problemas de roteirização consideradas. Para o PRV clássico, em 4 das 14 instâncias de Christofides foram obtidas soluções com custos iguais aos das melhores soluções reportadas até o momento. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 0,67%. Para o PRVJT, em 19 das 56 instâncias de Solomon foram obtidas soluções com custos iguais aos das melhores soluções reportadas e dentre as 56 instancias se obteve uma solução de custo inferior à reportada na literatura até o momento. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 1,30%.

#### 4.6 Algoritmo de TAN *et al.*, 2001

Este trabalho é caracterizado pela utilização da vizinhança  $\lambda$ -troca, descrita na seção 2.5 do capítulo 2, e pela aplicação de um método de busca local e de três implementações de meta-heurísticas. Os quatro algoritmos resultantes foram utilizados no tratamento do PRVJT.

A solução inicial é obtida através das heurísticas de inserção *push forward* (SOLOMON, 1987). O procedimento de troca e inserção de nós  $\lambda$ -troca, (OSMAN & CHRISTOFIDES, 1989; 1994) utiliza  $\lambda = 2$ , sendo as trocas e inserções de nós aplicadas segundo a seguinte ordem de oito operadores 2-troca: (0,1), (1,0), (1,1), (0,2), (2,0), (2,1), (1,2), (2,2), sugeridas em THANGIAH *et al.* (1994). A estratégia *global-best* foi empregada para selecionar a solução da vizinhança, ou seja, investiga-se todo o espaço da vizinhança 2-troca antes de selecionar a melhor solução.

Definido o espaço da vizinhança, implementações das meta-heurísticas *simulated annealing* (SA), busca tabu (BT) e algoritmos genéticos (AG) são executadas separadamente. O algoritmo de SA é baseado no trabalho de THANGIAH *et al.* (1994). A implementação de busca tabu (GLOVER, 1989) emprega além da memória de recência, ferramentas de memória de frequência de forma a combinar estratégias de intensificação e diversificação. Na implementação do algoritmo genético são incorporados operadores de cruzamento sofisticados e outras técnicas tal como uma abordagem de mutação adaptativa.

Estes quatro algoritmos distintos foram aplicados às 56 instâncias de SOLOMON (1987). O algoritmo de busca local e as implementações de meta-heurísticas SA, BT e GA encontram, respectivamente, soluções com um desvio percentual médio em relação as melhores soluções reportadas de 28,51%, 24,85%, 6,13%, 7,7%.

#### **4.7 Algoritmo LC03 (BOUTHILLIER & CRAINIC, 2005)**

Este algoritmo apresenta um método de multi-busca com cooperação paralela (*parallel cooperative multi-search method*) para o problema de roteirização com janelas de tempo. Métodos de resolução paralela oferecem a possibilidade de aceleração computacional, contudo abordagens de paralelização clássicas (baseadas em decomposição funcional ou de dados) não modificam significativamente a trajetória de busca das meta-heurísticas. Desta forma, o melhoramento na qualidade das soluções é grandemente limitado. Por esta razão, métodos denominados multi-busca (*multi-search* ou *multi-thread*)



de meta-heurísticas começaram a ser empregados na resolução de problemas combinatórios difíceis, obtendo significativa melhora na qualidade das soluções.

O método de multi-busca com cooperação paralela proposto no trabalho está baseado na estratégia de soluções de elite (descrita na seção 3.2 do capítulo 3). Esta estratégia é o núcleo do mecanismo de cooperação, pois por meio dela se mantêm armazenadas as soluções factíveis de alta qualidade que serão requeridas nos vários estágios do processo de busca.

A solução inicial deste algoritmo é obtida através da implementação de várias heurísticas clássicas de construção que promovem a diversificação do conjunto de soluções. A otimização destas soluções é feita através do princípio de cadeia de ejeção (*ejection chain*) e das heurísticas clássicas de 2-opt, 3-opt e or-opt. O princípio cadeia de ejeção é utilizado em busca tabu para explorar vizinhanças grandes e complexas – nós são retirados inseridos em novas rotas ainda que violem restrições do problema, e outros nós pertencentes a ela, são “ejetados” para novas rotas com vistas a factibilização da solução. No trabalho de Bouthillier e Crainic se utiliza o tipo de cadeia de ejeção denominado processo de troca multi-nó (*multi-node exchange process*), no qual troca dos nós é feita envolvendo duas rotas. Após as trocas, são aplicadas à solução, as heurísticas 2-opt, 3-opt e or-opt.

Após a otimização das soluções, inicia-se a estratégia de cooperação entre as implementações de busca tabu e algoritmos evolutivos. Dada a sua eficácia, foram utilizados dois algoritmos baseados em busca tabu - TABURROUTE (GENDREAU *et al.* 1994) e Unified Tabu Search (CORDEAU *et al.*, 2001), descritos na seção 4.4 e 4.5, respectivamente - e dois algoritmos evolutivos com diferentes mecanismos de cruzamento, desenvolvidos pelos autores. A cooperação entre os algoritmos é realizada através da troca de informação armazenada no conjunto de soluções de elite.

Os experimentos utilizaram cinco processadores (dois dos quais dedicados aos algoritmos de busca tabu, dois dedicados aos algoritmos evolutivos, e um processador para o processamento de soluções de elite, método de pós-otimização e métodos de construção das soluções iniciais). Este método foi aplicado às 56 instâncias de SOLOMON

(1987) com 100 nós e para as 300 instâncias de HOMBEGER & GEHRING (1999) de 200 a 1000 nós.

A aplicação deste trabalho às 56 instâncias clássicas de Solomon gerou soluções de alta qualidade, obtendo resultados melhores aos reportados na literatura em cinco exemplos. O desvio percentual médio das soluções (distância e número de veículos) deste método em relação às melhores soluções reportadas é, respectivamente, 0,24% e 0,49%.

#### **4.8 Algoritmo PF (PUREZA & FRANÇA, 1990)**

O algoritmo PF (PUREZA & FRANÇA, 1990) foi elaborado com o intuito de se aplicar estratégias de busca tabu ao Problema de roteirização de Veículos com e sem janela de tempo. Como este algoritmo é uma das primeiras implementações de busca tabu ao PRV, somente os elementos mais simples da metodologia são utilizados. O algoritmo é constituído de duas etapas:

1. Construção da rota inicial através da heurística (i) *Push-forward* de SOLOMON (1987), descrita na seção 2.2 do capítulo 2;
2. Procedimento de melhoria da rota inicial baseado no algoritmo de DROR & LEVY (1986), descrito na seção 2.5 do capítulo 2. O procedimento consiste na aplicação de uma seqüência de movimentos inter-rotas (troca e inserções de nós entre rotas distintas) e intra-rotas (2-opt) até que um mínimo local seja obtido. A partir deste ponto, os movimentos são aplicados com o emprego de mecanismos restritivos de busca tabu.

Como atributo do movimento foram utilizadas as arestas eliminadas e adicionadas em qualquer movimento, uma vez que tanto trocas e inserções de nós e operações 2-opt são movimentos de eliminação e adição de arestas. O algoritmo considera 2 listas tabu circulares (período tabu fixo); na primeira lista são armazenadas as arestas adicionadas, e na segunda lista são armazenadas as arestas eliminadas nos movimentos. Os atributos armazenados em ambas listas são chamados tabu-ativos. Seja  $E(s)$  o conjunto

composto das arestas eliminadas e adicionadas em um dado movimento candidato  $s$ . Seja também  $E_t(s)$  um subconjunto de  $E(s)$  tal que seus elementos são tabu-ativos. Diz-se que o movimento  $s$  é tabu (regra de ativação tabu) se e somente se o número de elementos de  $E_t(s)$  for maior que  $L$ , onde  $L$  é um inteiro menor ou igual a soma das arestas eliminadas e adicionadas.

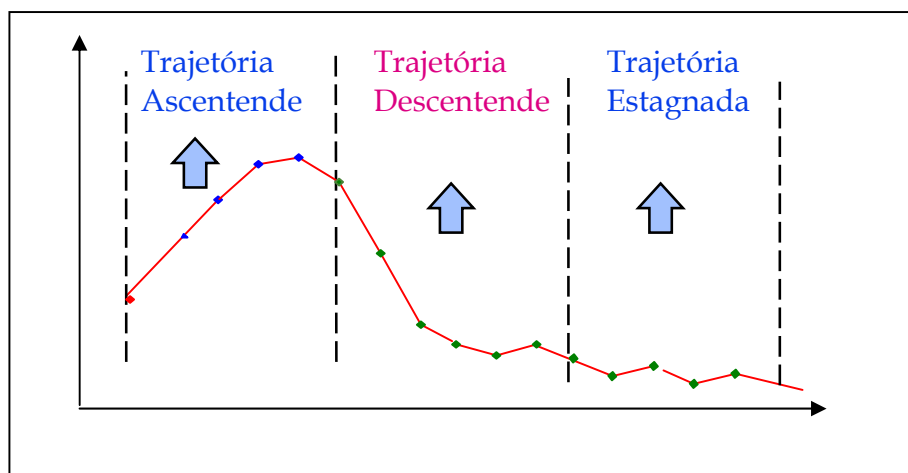
Este algoritmo foi aplicado a vinte problemas incluindo as 14 instâncias-teste de Christofides; em 2 casos se obteve soluções com custos iguais aos das melhores soluções reportadas até o momento. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 1,04%.

#### **4.9 Algoritmo HTA (PUREZA & FRANÇA, 2001)**

A abordagem adaptativa de busca tabu HTA (**H**orizon**T**e **A**daptativo) foi desenvolvida com o propósito de integrar estratégias de intensificação e diversificação. Algumas destas técnicas de integração caracterizam-se pela variação sistemática do período tabu e da configuração da lista tabu (veja HUBSCHER & GLOVER, 1992, e CHAKRAPANI & SKORIN-KAPOV, 1993), sendo, portanto uma extensão do conceito de lista tabu dinâmica.

A abordagem HTA, por sua vez, provoca perturbações na regra de ativação tabu, ou seja, nos critérios que prescrevem explicitamente quando um movimento deve ou não ser proibido. Além disso, tais perturbações não são ativadas somente por fases sem melhoria como nos métodos adaptativos, mas sim pela análise dos padrões de trajetória da busca recentemente traçada.

Um padrão de trajetória pode ser entendido como uma resposta a um dado nível de restritividade imposto à busca. O método identifica e reage a três padrões de trajetória relevantes ao desempenho da meta-heurística: estagnação da busca, trajetórias ascendentes e trajetórias descendentes. Estes padrões de trajetória podem ser observados na FIGURA 4.1:



**FIGURA 4.1 - Descrição dos padrões de trajetória**

Segundo PUREZA & FRANÇA (2001), a estagnação da busca é resultado de restrições tabu fracas ou de uma curta duração. Este padrão de trajetória é claramente uma condição adversa à efetividade de qualquer processo de busca, e sua superação requer o aumento dos níveis de restritividade.

A trajetória ascendente é uma condição caracterizada pelo aumento dos custos de solução quando o nível de restritividade é severo ou longo demais. Este padrão de trajetória é geralmente resultado da aplicação de diversificação da busca após uma fase de estagnação. O método responde a trajetórias ascendentes com a relaxação das restrições tabu. Estas relaxações podem assumir vários graus, dependendo dos níveis de restritividade alcançados. O propósito desta fase é permitir que novas regiões promissoras do espaço de busca sejam atingidas. Os níveis de restritividade devem impedir o retorno da busca a vales já visitados, mas ao mesmo tempo, permitir o aproveitamento das possibilidades de melhoria da nova região.

A trajetória descendente, caracterizada pela diminuição dos custos das soluções, estabelece níveis moderados de restritividade visando obter melhorias adicionais. Este padrão de trajetória pode resultar da fase de relaxação de restrições tabu.

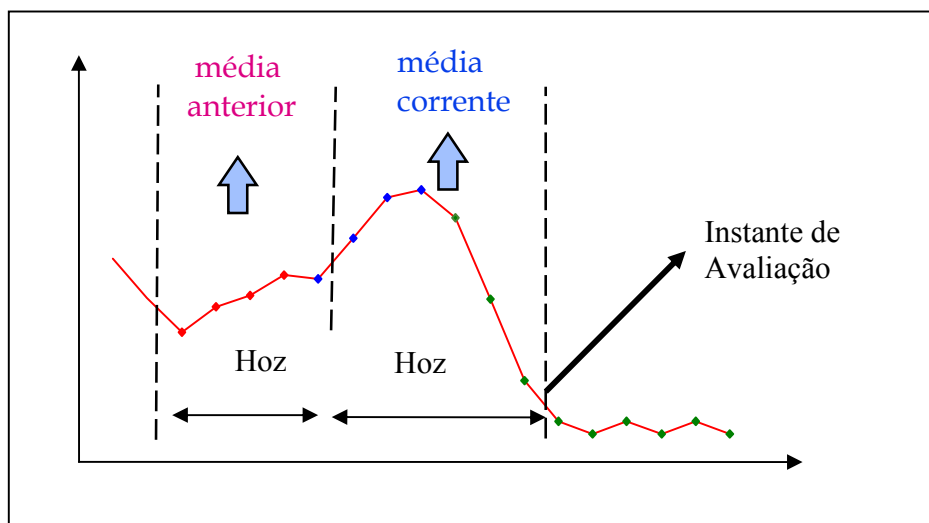
A análise e identificação dos padrões de trajetória são realizadas ao longo de todo o processo de busca, após um número relativamente pequeno de iterações. A análise

dos padrões de trajetória não é feita apenas em relação à diminuição e aumento dos níveis de restritividade, mas também em relação ao grau e a duração da perturbação.

A identificação se dá através da análise comparativa dos valores da média de duas amostras de custos da solução gerados nos últimos dois estágios de busca. As médias destes dois conjuntos de custos são chamadas de média anterior e média corrente. A FIGURA 4.2 ilustra estas médias. Caso as médias sejam aproximadamente iguais, a trajetória é considerada estagnada. Caso a média corrente seja maior que a média anterior, a trajetória é declarada ascendente. Caso a média corrente seja menor que a média anterior, obteve-se uma trajetória descendente.

O comprimento de cada estágio de análise da busca é definido por uma variável inteira chamada *horizonte de avaliação - hoz*. O horizonte de avaliação é um parâmetro exógeno (gerado aleatoriamente em uma faixa de valores pré-estabelecida) que juntamente com um parâmetro de ajuste, define o período durante o qual a perturbação prescrita na avaliação anterior deverá ser aplicada. Ao final deste período (iteração corrente + *hoz*) é feita a identificação dos padrões de trajetória da busca (veja FIGURA 4.2).

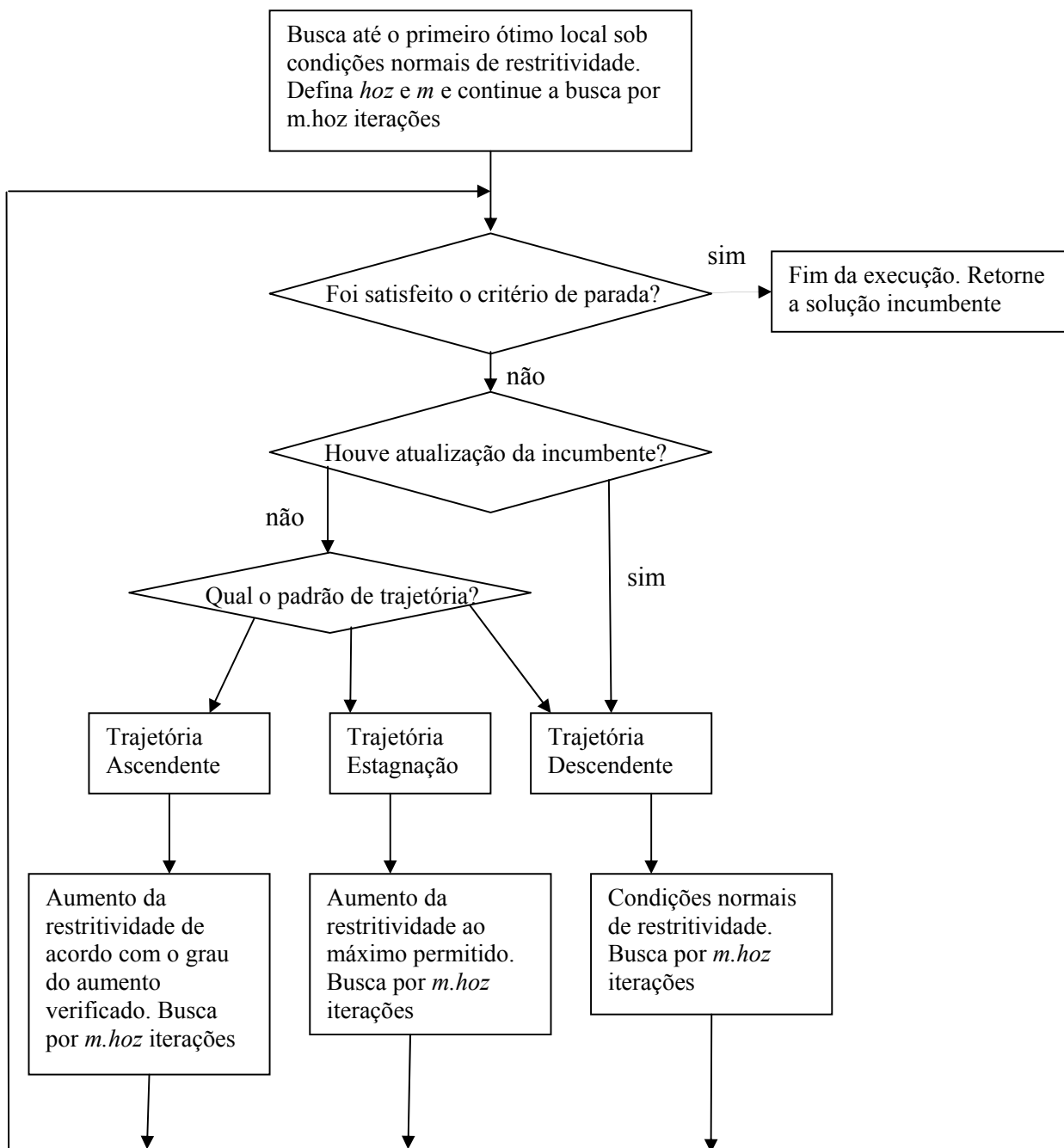
O algoritmo de PUREZA & FRANÇA (1990), descrito na seção 4.6, foi modificado e adaptado para a aplicação da abordagem HTA. Períodos tabu fixos foram substituídos por períodos tabu dinâmicos, e como critério de parada adotou-se um número máximo de 4000 iterações sem melhoria. Foram consideradas faixas de valores de  $L$  (número máximo permitido de arestas tabu-ativas no movimento corrente) diferenciadas para cada um dos 3 tipos de movimentos, uma vez que cada movimento envolve um número diferente de arestas eliminadas e adicionadas. Estas faixas de valores possíveis são denominadas *tolerâncias*. Por exemplo, em um movimento de troca de nós são eliminadas 4 arestas da solução corrente e adicionadas 4 outras arestas não presentes na solução, totalizando 8 arestas. Portanto, valores possíveis de tolerância para este tipo de movimento são 0, 1, 2, ..., 8.



**FIGURA 4.2 - Descrição das médias anterior e corrente e da variável hoz**

Dependendo do padrão de trajetória, os valores de tolerância são modificados para provocar o aumento da restritividade (redução dos valores de tolerância), diminuição da restritividade (aumento dos valores de tolerância) ou manutenção do nível de restritividade corrente. O valor da tolerância associada aos movimentos intra-rotas foi fixado em 4, pois apenas movimentos intra-rotas que provoquem melhorias locais na solução são permitidos. Assim, a possibilidade de ciclagem é excluída. A FIGURA 4.3 descreve o funcionamento da abordagem HTA após as modificações.

O algoritmo foi testado com as 14 instâncias-teste de Christofides, obtendo em relação às melhores soluções reportadas até o momento, solução com custo menor em 1 caso, e soluções com custos iguais em 8 casos. Em média, os custos apresentaram um desvio percentual em relação à melhor solução conhecida de 0,29%.



**FIGURA 4.3 – Fluxograma da Abordagem Adaptativa HTA**

## **CAPÍTULO 5 ABORDAGEM DE BUSCA TABU HTSA E SUA APLICAÇÃO AO PRV E AO PRVJT**

Neste capítulo é apresentado o objeto metodológico deste projeto - a abordagem adaptativa de busca tabu HTSA (HorizonTe Semi-Adaptativo). Proposta por PUREZA (1996), HTSA utiliza boa parte dos princípios da abordagem HTA (descrita na seção 4.9 do capítulo 4), pois visa também à integração de estratégias de intensificação e diversificação.

De acordo com PUREZA & FRANÇA (2001), os princípios adaptativos propostos na abordagem HTA poderiam ser refinados com o desenvolvimento de novas abordagens nas quais o conceito de horizonte de avaliação seja eliminado. A determinação de valores mais adequados de horizonte pode resultar em soluções de qualidade, porém o esforço computacional pode limitar a atratividade do método. Mesmo que os valores do horizonte sejam obtidos rapidamente, uma outra limitação da abordagem HTA é o fato de que análises de trajetórias de busca e decisões resultantes são feitas em intervalos pré-determinados, ignorando-se alterações que possam surgir durante aquela etapa da busca.

Para problemas que contemplam várias restrições, métodos de busca em espaços viável apresentam uma dificuldade maior de encontrar, a partir de certo ponto, novos caminhos que resultem em soluções de maior qualidade. Esta dificuldade é mantida ao se aplicar à abordagem HTA. Um dos argumentos da autora é que a qualidade das soluções obtidas pela aplicação de HTA a exemplos clássicos do Problema do Caixeiro Viajante (onde existem apenas restrições de fechamento de rotas) é muito menos dependente da faixa de período tabu do que quando aplicado ao PRV. Enquanto no PCV algumas soluções ótimas foram obtidas com faixas de período tabu tão distintas quanto 5-10 e 360-370, as faixas adequadas para o PRV se apresentaram bastante limitadas.

Esses resultados sugerem que HTA é mais adequada a problemas menos restritos, pois mesmo com períodos tabu a princípio inadequados e grandes alterações dos



valores de tolerância, existe uma quantidade maior de caminhos viável que resultam em soluções de alta qualidade. Suponha que em um dado processo de busca,  $R_1$  seja a região do espaço de soluções correntemente explorada. Suponha também que em  $R_1$  exista uma solução  $x$  de alta qualidade a qual não pode ser encontrada com os presentes parâmetros de busca. Um aumento drástico da restritividade (prescrito por HTA) direcionará a busca a uma nova região  $R_2$  do espaço. Em função do menor número de caminhos viável entre  $R_1$  e  $R_2$ ,  $x$  terá menores chances de ser encontrada. Parece interessante, portanto, a elaboração de métodos adaptativos para o PRV (mais restrito) onde mudanças nos valores dos parâmetros sejam feitas de forma mais *parcimoniosa* de modo a explorar ao máximo as possibilidades de melhoria de uma dada região.

Por estes motivos foi proposta por PUREZA (1996) a abordagem HTSA, sendo uma primeira experiência no sentido de gerar meta-heurísticas adaptativas desprovidas do conceito de horizonte de avaliação e voltadas ao tratamento de problemas mais restritos. Seus princípios são apresentados na próxima seção. Na seção 5.1 e 5.2 relata-se a forma original do algoritmo HTSA juntamente com resultados obtidos pela calibragem de seus parâmetros, as seções 5.2.1.2 e 5.3 apresentam as modificações incorporadas, na forma original, da abordagem HTSA e resultados decorrentes de sua aplicação.

## 5.1 Princípios da Abordagem HTSA

A forma original de HTSA, proposta por PUREZA (1996), compreende três etapas, onde se alternam sistematicamente o *aumento dos níveis de restritividade*, o *relaxamento dos níveis de restritividade* e o *alcance da estagnação*. As duas primeiras etapas ainda têm seu fim determinado pelo instante de avaliação e, portanto, arbitrariamente. Ao contrário de HTA, a abordagem HTSA aguarda de forma deliberada à estagnação da busca.

O processo de busca inicia-se com valores de tolerância correspondentes a níveis normais de restritividade, pois se considera estas condições devam ser mantidas

sempre que possível. Atingido o primeiro ótimo local, inicia-se a fase de **estagnação**. Durante esta etapa e *a cada nova iteração*, verifica-se se a trajetória de busca apresenta um padrão de estagnação. Assim como no algoritmo HTA, a determinação de padrões da trajetória é feita pela análise comparativa entre as médias anterior e corrente dos custos das soluções geradas. No caso da trajetória estagnada, estas médias têm valores relativamente próximos.

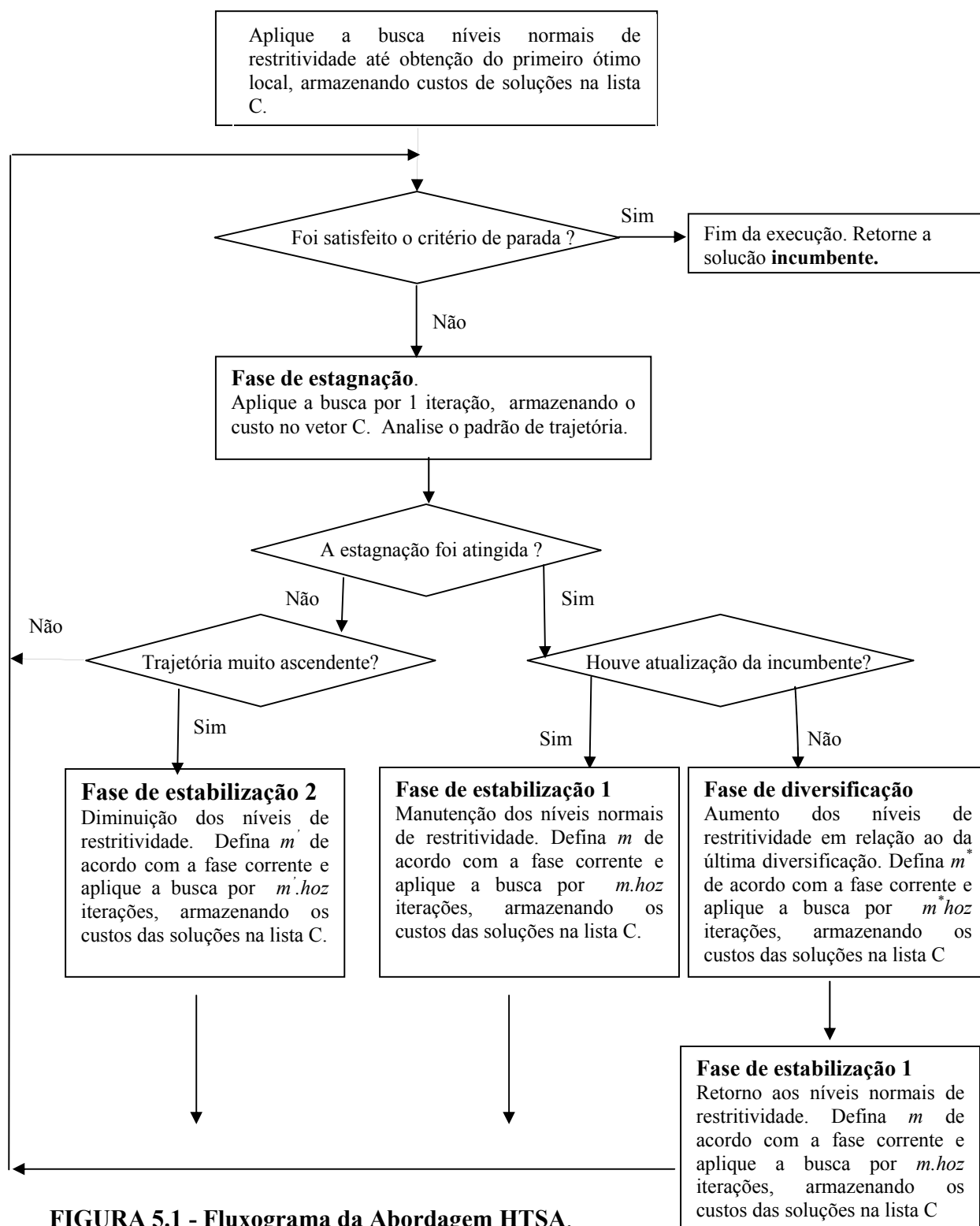
Quando a trajetória da busca se torna estagnada, a fase de estagnação é finalizada. Verifica-se então se houve melhoria da melhor solução obtida durante a fase. Neste caso, inicia-se uma **fase de estabilização 1** por um número pré-determinado de iterações e utilizando-se níveis normais de restritividade. Caso contrário, o nível de restritividade é aumentado em relação ao nível da *última fase de diversificação*, por sua vez chamado de nível referência. No início da execução, o nível referência assume os valores do nível normal de restritividade. A fase de diversificação é então mantida por um número de iterações pré-estabelecido. Note que caso não se obtenha melhorias após algumas seqüências das fases, o nível referência pode atingir valores máximos de restritividade. Neste caso, ele volta a assumir os níveis normais de restritividade.

Após a fase de diversificação, inicia-se novamente a **fase de estabilização 1** com o retorno aos níveis normais de restritividade estabelecidos. Admite-se que este período de estabilização é o mais propenso para a obtenção de melhorias. Após a finalização da fase, retorna-se à fase de estagnação sob as mesmas condições de operação.

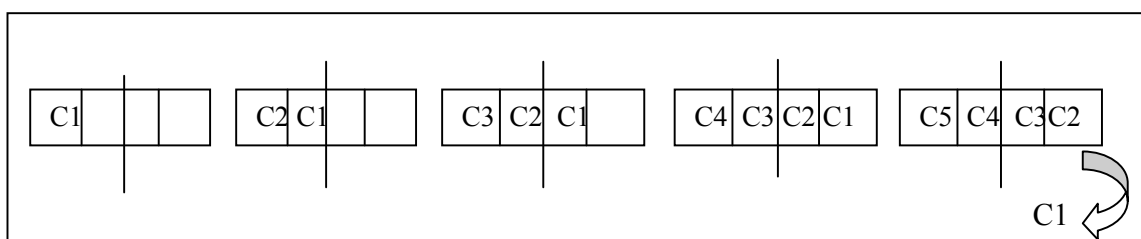
A preocupação em restringir a deterioração da função objetivo resultou em um passo adicional no algoritmo. Sempre que se verificam trajetórias muito ascendentes (em alguns casos, devido a períodos tabu longos), aplica-se a **fase de estabilização 2** durante a qual a restritividade da busca é diminuída para níveis abaixo dos normais e mantida por um número de iterações pré-estabelecido. A determinação destes padrões da trajetória é mais uma vez feita pela análise comparativa entre as médias anterior e corrente dos custos das soluções geradas. No caso da trajetória muito ascendente, a média corrente tem um valor bem superior ao da média anterior.

Conforme descrito acima, HTSA responde a dois únicos padrões de trajetória – a estagnação e trajetória muito ascendente. Exceto pelas fases de estabilização e

de trajetória muito ascendente, perturbações restringem-se ao aumento da restritividade e são ativadas pela ocorrência de um número de iterações sem melhoria. O aumento da restritividade é feito de forma sistemática e os níveis impostos são determinados em função do nível da última diversificação, guardando semelhanças ao conceito de *moving gaps* (CHAKRAPANI & SKORIN-KAPOV, 1993). A FIGURA 5.1 apresenta um fluxograma simplificado da abordagem.



A armazenagem dos custos de soluções para cálculo das *médias anterior e corrente* e análise do padrão de trajetória apresenta uma mudança estrutural em relação àquela utilizada na abordagem HTA. A fim de detectar a estagnação da busca foi utilizada uma lista C de tamanho igual a  $2 * \left\lfloor \frac{1}{2} (h_{\min} + h_{\max}) \right\rfloor$ , sendo  $h_{\min}$  e  $h_{\max}$  parâmetros exógenos. A armazenagem dos custos segue a estratégia *first in - first out (FIFO)* – o custo da solução mais recente é encontrado na primeira posição da lista enquanto o custo da solução mais antiga se encontra na última posição preenchida da lista. Conforme ilustrado na FIGURA 5.2, a inclusão de um novo custo na lista provoca o “deslocamento” de custos já armazenados para posições posteriores da lista (na prática, é utilizada uma lista circular e ponteiros que se deslocam para marcar as posições iniciais e finais de cada metade da lista). Quando a lista está totalmente preenchida, a inclusão de um novo custo implica na exclusão do custo mais antigo. Note, portanto, que está sendo utilizado o conceito de *médias móveis*. Os custos da primeira metade da lista são utilizados para o cálculo da média corrente e com os custos da segunda metade da lista é calculada a média anterior.



**FIGURA 5.2 - Lista de armazenagem de custos de soluções.**

## 5.2 Aplicação da Abordagem HTSA ao Problema de Roteirização de Veículos

A abordagem HTSA, na forma original, utiliza procedimentos dos algoritmos: PF (PUREZA & FRANÇA, 1990) e HTA (PUREZA & FRANÇA, 2001), descritos nas seções 4.8 e 4.9 do capítulo 4. Foram mantidos os atributos do movimento

(arestas eliminadas e adicionadas), a regra de ativação tabu (número de arestas eliminadas e adicionadas tabu-ativas excedendo valores de tolerância) e os critérios de aspiração (o movimento candidato resulta em uma solução melhor que a melhor obtida na execução ou todos os movimentos são tabu). A implementação da abordagem HTSA também considera as mesmas faixas de valores de tolerâncias diferenciadas para cada um dos 3 tipos de movimentos (2-opt, inserção e troca) e utilizadas na aplicação da abordagem HTA (veja seção 4.7 do capítulo 4), assim como períodos tabu e horizontes de avaliação (hoz) gerados dinamicamente em faixas  $[t_{\min}, t_{\max}]$  e  $[h_{\min}, h_{\max}]$ , respectivamente.

Como em fases de diversificação, os valores de  $TT$  e  $TI$  utilizados na última vez que se realizou esta fase (nível referência) devem ser diminuídos para aumentar a restritividade, foi utilizado um vetor  $\tau$  para armazená-los. Os passos do algoritmo são dados a seguir.

### 5.2.1 Passos do Algoritmo PF + HTSA

**ENTRADA:** Dados da instância (número de clientes, demanda dos clientes, distância euclidiana entre nós).

Parâmetros para construção da solução inicial:  $[\sigma, \delta, \alpha_1, \alpha_2]$ .

Faixa de horizonte de avaliação  $[h_{\min}, h_{\max}]$  e período tabu  $[t_{\min}, t_{\max}]$ .

Condição de estagnação da trajetória:  $0 \leq CE < 1$ .

Condição de trajetória ascendente:  $TA > 1$ .

Fator de ajuste para aplicação da perturbação:  $m > 0$ .

1. Inicialização: Gere uma solução inicial  $S_0$  através da heurística de construção de rotas e inicialize o contador  $it$  de iterações. Seja  $S$  a solução atual e  $S^*$  a melhor solução obtida até o momento. Faça  $S=S_0$ ,  $S^*=S$ ,  $TO=4$ ,  $T=(TT, IT)=(4,3)$  (níveis normais de restritividade) e armazene os valores de  $TT$  e  $IT$  em um vetor  $\tau$  (ou seja,  $\tau=T$ ).
2. Faça movimentos de melhoria inter-rotas e intra-rotas até que o primeiro ótimo local  $S^*$  seja obtido, armazenando os custos de todas as soluções geradas em uma lista  $C$  com  $hozp = 2 \lfloor \frac{1}{2}(h_{\min} + h_{\max}) \rfloor$  posições. Faça fase = **estagnação**.
3. Se algum critério de parada tiver sido satisfeito, pare. Caso contrário:
4. Selecione o melhor movimento não tabu, ou tabu que satisfaça algum critério de aspiração. Efetive o movimento, obtendo  $S'$  e faça  $S=S'$ . Se  $\text{custo}(S) < \text{custo}(S^*)$ , faça  $S^*=S$ . Armazene o custo  $S$  na lista  $C$ . Faça *média anterior* igual à média dos

custos nas primeiras  $hoz/2$  posições e *média atual* igual à média dos custos nas últimas  $hoz/2$  posições. Calcule  $diff = \left( \frac{\text{média anterior} - \text{média atual}}{\text{média anterior}} \right)$

5. Se  $\text{média atual} \geq TA * \text{média anterior}$  (ou seja, se a trajetória de busca está muito ascendente) faça fase = **estabilização 2**. Faça  $T=(6,5)$ , obtenha  $hoz$  na faixa  $[h_{min}, h_{max}]$  e mantenha a busca sob estas condições operacionais por  $hoz$  iterações ( $prox\_aval = it + hoz$ ), armazenando os custos das soluções geradas  $S$  em  $C$ , e atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ . Quando  $it = prox\_aval$  vá para o passo 3.

4.1 Caso contrário, se  $|diff| \leq CE$ , a estagnação da trajetória foi atingida. Neste caso:

4.1.1 Se houve atualização da melhor solução desde a fase anterior de estagnação, faça fase = **estabilização 1** e  $T = (4,3)$ . Obtenha  $hoz$  na faixa  $[h_{min}, h_{max}]$  e mantenha a busca sob estas condições por  $2*hoz$  iterações ( $prox\_aval = it + 2*hoz$ ), armazenando os custos das soluções  $S$  em  $C$ , e atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ . Quando  $it = prox\_aval$  vá para o passo 3.

4.1.2 Se não houve atualização da melhor solução desde a fase anterior de estagnação:

4.1.2.1 Faça fase = **diversificação**. Se  $\tau(i) = -1$  ( $\forall i = \{1, 2\}$ ), ou seja, se os níveis mínimos de tolerância na fase de diversificação foram excedidos, reinicialize o nível referência, fazendo  $\tau = T$ . Faça  $T = \tau$  e obtenha  $hoz$  na faixa especificada e decremente os valores de tolerância em  $\tau$  em uma unidade para serem usados na próxima fase de diversificação. Mantenha a busca sob estas condições por  $\lfloor hoz/2 \rfloor$  iterações ( $prox\_aval = it + \lfloor hoz/2 \rfloor$ ), armazenando os custos das soluções  $S$  em  $C$ , e atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ .

4.1.2.2 Faça fase = **estabilização 1**. Obtenha  $hoz$  na faixa especificada e mantenha a busca sob condições normais de restritividade ( $T = (4,3)$ ) por  $2*hoz$  iterações ( $prox\_aval = it + 2*hoz$ ), armazenando os custos das soluções  $S$  em  $C$ , atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ . Quando  $it = prox\_aval$  vá para o passo 3.

4.2 Caso contrário, a estagnação da trajetória não foi atingida. Faça fase = **estagnação** e vá para o passo 3.

**SAÍDA:**  $S^*$ .

As seções seguintes apresentam os experimentos para calibragem de parâmetros da forma original de HTSA e resultados computacionais obtidos. Este algoritmo

foi implementado na linguagem Delphi 5 e para todos os experimentos foi utilizado um microcomputador AMD Athlon(TM) XP 2600<sup>+</sup>, 1.92 Ghz, 1,00 GB de RAM.

### 5.2.1.1 Experimentos Computacionais para Calibragem de Parâmetros

Apesar de proposta em 1996, a implementação da abordagem HTSA não foi submetida a experimentos mais rigorosos para calibragem de parâmetros de busca. Por esta razão, o primeiro conjunto de experimentos deste projeto explorou as seguintes variações paramétricas:

1. Condição de estagnação (*CE*): 0,0001; 0,00025; 0,0005; 0,001; 0,0025; 0,01; 0,1.
2. Faixas de período tabu: 5 faixas com largura igual a 5 ou 10, sendo a faixa [0,5] ou [0,10] a menos restritiva e [55,60] ou [50,60] a mais restritiva.
3. Aumento da média corrente em relação à média anterior que define uma trajetória muito ascendente (*TA*): 2%, 10% e 20%.
4. Valores do fator de ajuste  $m$  que quando multiplicado por *hoz* determina o período em que a fase de estabilização (passos 4.2.1 e 4.2.2.2 do algoritmo) deve ser aplicada à busca: 2 e 3.
5. Faixas de valores  $[h_{min}, h_{max}]$  para o horizonte de avaliação (*hoz*): [0,5],[10,20], [15, 20], [15,25].
6. Parâmetros  $[\mu, \delta, \alpha_1, \alpha_2]$  para construção da solução inicial: [1,1,1,0], [1,1,0,1], [1,2,1,0] e [1,2,0,1].

Em cada variação paramétrica foi adotado o número máximo de 6000 iterações sem melhoria (critério de parada). Os testes contemplaram as 14 instâncias-teste descritas por Christofides, divididas em 4 classes (veja TABELA 5.1.) de acordo com suas características de estrutura, capacidade, tempo de serviço e tempo máximo de rota.



TABELA 5.1 - Instâncias em CHRISTOFIDES *et al.* (1979)

Classe	Instância	$n$	Estrutura	Capacidade	Tempo de serviço	Tempo máximo de rota
A	P 1	50	Uniforme	160	0	$\infty$
	P 2	75	Uniforme	140	0	$\infty$
	P3	100	Uniforme	200	0	$\infty$
	P4	150	Uniforme	200	0	$\infty$
	P5	199	Uniforme	200	0	$\infty$
B	P 6	50	Uniforme	160	10	200
	P 7	75	Uniforme	140	10	160
	P8	100	Uniforme	200	10	230
	P9	150	Uniforme	200	10	200
	P10	199	Uniforme	200	10	200
A	P11	120	Agrupado	200	0	$\infty$
	P12	100	Agrupado	200	0	$\infty$
B	P13	120	Agrupado	200	50	720
	P14	100	Agrupado	200	90	1040

Onde:

$n$ : número de nós; A: apenas restrição de capacidade.

B: restrições de capacidade, tempo de serviço e tempo máximo de rota.

Com base nos resultados obtidos, o conjunto de parâmetros que apresentou o menor desvio percentual médio em relação à melhor solução conhecida tem os seguintes valores:

- ✓ Condição de Estagnação ( $CE$ ) = 0,0025;
- ✓ Trajetória muito ascendente ( $TA$ ) = 10 %.
- ✓ Fator de ajuste  $m = 2$ ;
- ✓ Horizonte de avaliação [ $h_{min}, h_{max}$ ] = [10, 20];
- ✓ Parâmetros [ $\mu, \delta, \alpha_1, \alpha_2$ ] para construção da solução inicial = [1,1,0,1];

O conjunto de valores de parâmetros descrito acima foi selecionado como o mais adequado para todas as instâncias sendo adotado como a versão padrão para a forma

original do algoritmo PF + HTSA, mas uma variação deste conjunto foi obtida para cada instância teste.

Na TABELA 5.2 são apresentados os custos de solução e os tempos computacionais (entre parêntesis e em minutos) obtidos pelo conjunto padrão de parâmetros e suas variações. Ao se padronizar o conjunto de parâmetros há uma perda esperada na qualidade das soluções quando estas são comparadas com as soluções obtidas pela variação no conjunto de parâmetros. Note que para a instância P3 foi obtida uma solução que supera a melhor solução já reportada. Os resultados reportados na tabela correspondem à melhor execução de um conjunto de 5 faixas de período tabu investigadas.

**TABELA 5.2 - Resultados de PF + HTSA para as 14 instâncias de Christofides *et al.*1979 (PRV).**

Instância	nº de nós	PF + HTSA	
		(conjunto padrão)	(variação do conjunto padrão)
P1	50	524,8 (0,04)	524,61 (0,05)
P2	75	837,32 (0,12)	835,26 (0,15)
P3	100	828,91 (1,4)	825,45 <sup>†</sup> (0,52)
P4	150	1048,83 (2,65)	1029,71 (2,02)
P5	199	1353,64 (1,4)	1314,06 (1,44)
P6	50	560,38 (0,09)	555,43 (0,10)
P7	75	923,65 (0,57)	909,68 (0,46)
P8	100	865,51 (1,2)	865,51 (0,56)
P9	150	1168,89 (1,52)	1165,67 (2,59)
P10	199	1431,05 (1,42)	1428,39 (1,09)
P11	120	1043,14 (0,32)	1042,12 (0,52)
P12	100	825,64 (0,35)	819,56 (0,30)
P13	120	1546,59 (0,71)	1544,03 (0,60)
P14	100	866,37 (0,5)	866,37 (0,60)

Na TABELA 5.3 é feita uma comparação entre os melhores resultados obtidos pelo algoritmo PF + HTSA e os melhores resultados (em negrito) encontrados por algoritmos de alto desempenho descritos no capítulo 4. Tempos computacionais (em minutos) para obtenção da melhor solução são apresentados entre parêntesis quando disponíveis.

**TABELA 5.3 – Resultados de PF + HTSA e 5 algoritmos competitivos para as 14 instâncias de Christofides *et al.* 1979 (PRV).**

Instância	nºde nós	Taillard (1993)	TABUROUTE <sup>1</sup> (versão padrão) (1994)	Rochat & Taillard <sup>2</sup> (1995)	HTA <sup>3</sup> (2001)	UTSA <sup>4</sup> (2002)	PF + HTSA <sup>5</sup> (conjunto padrão)
P1	50	<b>524,61</b>	<b>524,61</b> (6,00)	<b>524,61</b>	<b>524,61</b> (0,15)	<b>524,61</b> (4,57)	524,8 (0,04)
P2	75	<b>835,26</b>	835,32 (53,8)	<b>835,26</b>	<b>835,26</b> (0,80)	835,45 (7,27)	837,32 (0,12)
P3	100	826,14	826,14 (18,4)	826,14	826,14 (0,70)	826,44 (11,23)	828,91 (1,4)
P4	150	<b>1028,42</b>	1031,07 (58,8)	<b>1028,42</b>	1032,53 (51,5)	1038,44 (18,72)	1048,83 (2,65)
P5	199	1298,79	1311,35 (90,9)	<b>1291,45</b>	1327,17 (26,6)	1308,87 (28,10)	1353,64 (1,4)
P6	50	<b>555,43</b>	<b>555,43</b> (13,5)	<b>555,43</b>	<b>555,43</b> (1,50)	<b>555,43</b> (4,61)	560,38 (0,09)
P7	75	<b>909,68</b>	931,23 (54,6)	<b>909,68</b>	<b>909,68</b> (5,70)	<b>909,68</b> (7,55)	923,65 (0,57)
P8	100	865,9	865,94 (25,6)	865,9	<b>865,51 (1,90)</b>	866,68 (11,17)	<b>865,51</b> (1,2)
P9	150	1162,89	1177,76 (71,0)	1162,89	<b>1161,93 (82,7)</b>	1171,81 (19,17)	1168,89 (1,52)
P10	199	1397,94	1418,51 (99,5)	<b>1395,85</b>	1407,63 (145,2)	1415,40 (29,74)	1431,05 (1,42)
P11	120	<b>1042,11</b>	1073,47 (22,2)	<b>1042,11</b>	<b>1042,11</b> (1,80)	1074,13 (14,15)	1043,14 (0,32)
P12	100	<b>819,56</b>	<b>819,56</b> (16,0)	<b>819,56</b>	<b>819,56</b> (0,80)	<b>819,56</b> (10,99)	825,64 (0,35)
P13	120	<b>1541,14</b>	1547,93 (59,2)	<b>1541,14</b>	1542,97 (47,0)	1568,91 (14,53)	1546,59 (0,71)
P14	100	<b>866,37</b>	<b>866,37</b> (65,7)	<b>866,37</b>	<b>866,37</b> (0,20)	<b>866,37</b> (10,65)	<b>866,37</b> (0,5)
DPM%		0,06	0,89	0,01	0,29	0,67	1,01

onde

DPM: desvio percentual médio das melhores soluções obtidas para as 14 instâncias;

melhor solução conhecida em negrito; 1- Silicon Graphics Workstation (36 MHz, 5.7 Mflops), 2 - S.G. 100mhz, 1 run, 3- Sun Ultra 1 wokrstation, 4 - Sun Ultrasparc 10 (440 MHz), 5 - AMD Athlon 2600+ MHz

Observa-se que ao se executar a abordagem HTSA com o conjunto de parâmetros padrão, encontra em dois casos, soluções com custos iguais aos obtidos pela abordagem HTA, apresentando um desvio percentual médio em relação às melhores soluções reportadas de 1,01 %. Caso não haja a padronização dos parâmetros da abordagem HTSA é possível obter para sete casos soluções com custos iguais aos obtidos pela abordagem HTA e para três casos soluções com qualidade superior as soluções da abordagem HTA, apresentando um desvio percentual médio em relação às melhores soluções reportadas de 0,33 %.

A comparação, entre as soluções dos algoritmos, quanto aos tempos computacionais não pode ser feita com precisão, pois os experimentos para a obtenção das soluções foram realizados em máquinas distintas, além do fato de nem todos os tempos computacionais terem sido disponibilizados.

Cabe ressaltar na comparação com os algoritmos de TAILLARD (1993), TABUROUTE (1994); ROCHAT & TAILLARD (1995) e UTSA (CORDEAU *et al.*, 2002) que estes empregam estratégias de busca tabu mais sofisticadas do que as utilizadas na abordagem HTSA, em particular, paralelização e exploração de regiões inactíveis. Portanto, o desempenho da abordagem HTSA é considerável ao se notar a simplicidade dos recursos da abordagem frente às heurísticas que a superam e os ganhos significativos na qualidade de solução quando não se padroniza o conjunto de parâmetros.

### **5.2.1.2 Combinação da Abordagem HTSA com Diversificação Baseada em Frequência**

Conforme visto na seção 3.2 do capítulo 3, estratégias de intensificação e diversificação comumente utilizadas em implementações tabu são baseadas na frequência observada de atributos dos movimentos realizados ao longo do processo de busca. Computando-se esta frequência, é possível tanto penalizar movimentos com atributos muito frequentes (diversificação) como favorecê-los (intensificação). A penalização modifica a regra de escolha dos movimentos, permitindo a incorporação de atributos raramente

explorados nos movimentos anteriores, e provocando a exploração de novas regiões do espaço de busca. O favorecimento de atributos freqüentes, por sua vez, provoca um melhor exame de regiões consideradas promissoras.

No segundo conjunto de experimentos deste trabalho adicionou-se, à forma origem, da abordagem HTSA a diversificação baseada na freqüência dos atributos com o intuito de melhorar o desempenho do algoritmo original de PUREZA (1996). A idéia por trás desta combinação de estratégias é de que ao permitir uma escolha de atributos pouco freqüentes durante a fase de diversificação de HTSA, estimular-se-ia o direcionamento da busca para regiões que a história da busca sugere pouco exploradas. Experimentos com estratégias de intensificação baseadas em freqüência também foram realizados, porém não são aqui reportados por terem gerado soluções substancialmente piores.

Para o uso da memória de longo prazo baseada na freqüência em uma estratégia de diversificação, foi criada uma matriz que atualiza e armazena o número de vezes que cada aresta aparece nas soluções geradas. Após a satisfação de um critério pré-definido, os valores destas matrizes são divididos pelo número de iterações realizadas obtendo-se assim a freqüência das arestas até o momento. O valor calculado é então comparado a um limitante (*lim*) que estipula quais arestas possuem freqüência alta.

Este procedimento é ativado sempre que algum dos elementos do vetor  $\tau$  (onde se armazena o nível de restritividade referência) for igual zero, ou seja, quando já foram aplicados todos os aumentos de níveis de restritividade (seguidos de estabilização) e não houve melhoria da melhor solução obtida.

A modificação da regra de escolha dos movimentos nesta fase se dá com a adição de um termo envolvendo os parâmetros *pen* e *d* ao valor da economia real gerada pelo movimento. *Pen* é a soma das freqüências das arestas com valor maior ou igual a *lim* que seriam adicionadas e a subtração das freqüências com valor maior ou igual a *lim* das arestas que são eliminadas com o movimento. O parâmetro *d* é um número maior que zero que ajusta *pen* à magnitude dos custos de soluções. Quanto maior *d*, maior é a diversificação atingida. Assim, o valor de um dado movimento candidato passa a ser calculado pela seguinte fórmula:

$$\text{Custo da solução resultante do movimento} = \text{Custo da solução corrente} - \text{economia do movimento} + d * \text{pen}$$

A modificação na regra de escolha do atributo é feita por  $m * \text{hoz}$  iterações

O algoritmo PF + HTSA com memória de frequência (PF + HTSA + MF)

difere do algoritmo original nos seguintes pontos:

- ✓ No passo 1 (inicialização), a matriz  $MO$  de ocorrência de atributos (arestas) é inicializada, fazendo o contador de ocorrência de cada aresta igual a zero.
- ✓ Após cada efetivação de movimento, o contador de ocorrência de cada aresta presente na solução gerada e armazenada em  $MO$  é incrementada em uma unidade.
- ✓ No passo 4.2.2.1 do algoritmo, se  $\tau(i) = 0$  ( $\forall i = \{1, 2\}$ ), calcula-se a frequência de qualquer aresta a partir dos valores armazenados em  $MO$ , identificando-se as arestas com frequência  $\geq \text{lim}$ . Os valores de movimentos candidatos com arestas com frequência  $\geq \text{lim}$  são então modificados adicionando-se o termo  $d * \text{pen}$  por  $\lfloor \text{hoz}/2 \rfloor$  iterações. Os custos *reais* das soluções geradas são armazenados em  $C$ . Após  $\lfloor \text{hoz}/2 \rfloor$  iterações, a escolha do movimento volta a ser somente função da economia real do movimento.

### 5.2.1.2.1 Passos do Algoritmo PF + HTSA + MF

**ENTRADA:** Dados da instância (número de clientes, demanda dos clientes, distância euclidiana entre nós).

Parâmetros para construção da solução inicial:  $[\sigma, \delta, \alpha_1, \alpha_2]$ .

Faixa de horizonte de avaliação  $[h_{min}, h_{max}]$  e período tabu  $[t_{min}, t_{max}]$ .

Condição de estagnação da trajetória:  $0 \leq CE < 1$ .

Condição de trajetória ascendente:  $TA > 1$ .

Fator de ajuste para aplicação da perturbação:  $m > 0$ .

Parâmetro de ajuste na penalização de custos de solução:  $d > 0$ .

Limitante inferior de frequência alta de arestas:  $0 \leq \text{lim} \leq 1$ .

- 1 Inicialização: Gere uma solução inicial  $S_0$  através da heurística de construção de rotas e inicialize o contador  $it$  de iterações. Seja  $S$  a solução atual e  $S^*$  a melhor solução obtida até o momento. Faça  $S = S_0$ ,  $S^* = S$ ,  $TO = 4$ ,  $T = (TT, IT) = (4, 3)$  (níveis

normais de restritividade) e armazene os valores de  $TT$  e  $IT$  em um vetor  $\tau$  (ou seja,  $\tau = T$ ). Inicialize a matriz de ocorrência de arestas  $MO$ .

- 2 Faça movimentos de melhoria inter-rotas e intra-rotas até que o primeiro ótimo local  $S^*$  seja obtido, armazenando os custos de todas as soluções geradas em uma lista  $C$  com  $hozp = 2 \lfloor \frac{1}{2}(h_{min} + h_{max}) \rfloor$  posições, e incrementando o valor armazenado na matriz de ocorrência  $MO$  das posições correspondentes às arestas que pertencem às soluções. Faça fase = **estagnação**.
- 3 Se algum critério de parada tiver sido satisfeito, pare. Caso contrário:
- 4 Selecione o melhor movimento não tabu, ou tabu que satisfaça algum critério de aspiração. Efetive o movimento, obtendo  $S'$  e faça  $S=S'$ . Se custo ( $S$ ) < custo ( $S^*$ ), faça  $S^*=S$ . Armazene o custo  $S$  na lista  $C$  e incremente o valor armazenado na matriz de ocorrência  $MO$  nas posições correspondentes às arestas que pertencem à solução. Faça *média\_anterior* igual à média dos custos nas primeiras  $hozp/2$  posições e *média\_atual* igual à média dos custos nas últimas  $hozp/2$  posições.

$$\text{Calcule } diff = \left( \frac{\text{média anterior} - \text{média atual}}{\text{média anterior}} \right).$$

- 4.1 Se  $\text{média atual} \geq TA * \text{média anterior}$  (ou seja, se a trajetória de busca está muito ascendente) faça fase = **estabilização 2**. Faça  $T=(6,5)$ , obtenha  $hoz$  na faixa  $[h_{min}, h_{max}]$  e mantenha a busca sob estas condições operacionais por  $hoz$  iterações ( $prox\_aval = it + hoz$ ), armazenando os custos das soluções  $S$  em  $C$ , incrementando o valor armazenado na matriz de ocorrência  $MO$  das posições correspondentes às arestas que pertencem às soluções, e atualizando  $S^*$  quando custo ( $S$ ) < custo ( $S^*$ ). Quando  $it = prox\_aval$  vá para o passo 3.

- 4.2 Caso contrário, se  $|diff| \leq CE$ , a estagnação da trajetória foi atingida. Neste caso:

- 4.2.1 Se houve atualização da melhor solução desde a fase anterior de estagnação, faça fase = **estabilização 1** e  $T = (4,3)$ . Obtenha  $hoz$  na faixa  $[h_{min}, h_{max}]$  e mantenha a busca sob estas condições por  $2*hoz$  iterações ( $prox\_aval = it + 2*hoz$ ), armazenando os custos das soluções em  $C$ , incrementando o valor armazenado na matriz de ocorrência  $MO$  das posições correspondentes às arestas que pertencem às soluções, e atualizando  $S^*$  sempre que custo ( $S$ ) < custo ( $S^*$ ). Quando  $it = prox\_aval$  vá para o passo 3.

- 4.2.2 Se não houve atualização da melhor solução desde a fase anterior de estagnação:

- 4.2.2.1 Faça fase = **diversificação**. Se  $\tau(i) = -1$  ( $\forall i = \{1, 2\}$ ), ou seja, se os níveis mínimos de tolerância na fase de diversificação foram excedidos, reinicialize o nível referência, fazendo  $\tau = T$ . Faça  $T = \tau$  e obtenha  $hoz$  na faixa especificada e decmente os valores de tolerância em  $\tau$  em uma unidade para serem usados na próxima fase de diversificação. Mantenha a busca sob estas condições por  $\lfloor hoz/2 \rfloor$  iterações ( $prox\_aval = it + \lfloor hoz/2 \rfloor$ ), armazenando os custos das soluções  $S$  em  $C$ , incrementando o valor

armazenado na matriz de ocorrência  $MO$  das posições correspondentes às arestas que pertencem às soluções, e atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ .

Quando  $\tau(i) = 0$  ( $\forall i = \{1, 2\}$ ), modifique o custo das soluções geradas por  $\lfloor \text{hoz}/2 \rfloor$  iterações da seguinte forma. Calcule a frequência das arestas a partir dos valores em  $MO$ . Para cada aresta  $j-k$  da solução gerada, se  $\text{frequência}(j-k) < \text{lim}$  faça  $\text{frequência}(j-k) = 0$ , caso contrário  $\text{frequência}(j-k) = 1$ . Faça  $\text{pen}$  igual à somatória das frequências das arestas da solução e some o termo  $d^* \text{Pen}$  ao custo da solução. Armazene os custos *reais* de  $S$  em  $C$  e atualize  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ .

**4.2.2.2** Faça fase = **estabilização 1**. Obtenha  $\text{hoz}$  na faixa especificada e mantenha a busca sob condições normais de restritividade ( $T = (4,3)$ ) por  $2^* \text{hoz}$  iterações ( $\text{prox\_aval} = \text{it} + 2^* \text{hoz}$ ), armazenando os custos das soluções  $S$  em  $C$ , atualizando  $S^*$  sempre que  $\text{custo}(S) < \text{custo}(S^*)$ . Quando  $\text{it} = \text{prox\_aval}$  vá para o passo 3.

**4.3** Caso contrário, a estagnação da trajetória não foi atingida. Faça fase = **estagnação** e vá para o passo 3.

**SAÍDA:**  $S^*$ .

### 5.2.1.2.2 Resultados Computacionais

O algoritmo PF + HTSA + MF foi aplicado às 14 instâncias de Christofides. Para cada instância foram utilizados os valores do conjunto de parâmetros padrão da abordagem HTSA descritos na seção 5.2.1.1. Para os parâmetros  $\text{lim}$  e  $d$  foram consideradas as seguintes variações:

✓  $\text{lim}$ : 0; 0,01; 0,05; 0,3; 0,5; 0,8; 1.

✓  $d$ : 0,01; 0,1; 0,5; 1; 5; 10; 30; 80; 100.

Com base nos resultados obtidos, os valores dos parâmetros admitidos na versão padrão foram  $\text{lim} = 0,3$  e  $d = 80$ , uma vez que este conjunto apresentou o menor desvio percentual médio em relação às melhores soluções conhecidas. Para instâncias até 100 nós, foram realizadas 6000 iterações sem melhoria. Para instâncias acima de 100 nós, o número de iterações sem melhoria foi aumentado para 24000.



A TABELA 5.4 apresenta os resultados obtidos com PF + HTSA + MF. As colunas 4 e 5 indicam respectivamente, as melhores soluções com o conjunto de parâmetros padrão e as melhores soluções obtidas com uma variação do conjunto de parâmetros padrão. Os tempos computacionais em minutos estão entre parêntesis. A última linha (SOMA) indica a somatória dos custos e tempos computacionais para a melhor solução obtida para as 14 instâncias.

**TABELA 5.4 - Resultados de PF + HTSA + MF para as 14 instâncias de Christofides *et al.*,1979 (PRV)**

Instância	nº de nós	Conjunto padrão	Varição do conjunto padrão
P1	50	524,62 (0,24)	524,61 (0,10)
P2	75	838,75 (0,83)	835,26 (0,20)
P3	100	827,59 (3,96)	826,64 (0,71)
P4	150	1041,08 (4,2)	1029,85 (6,75)
P5	199	1334,42 (3,65)	1313,03 (2,90)
P6	50	555,43 (0,39)	555,43 (0,10)
P7	75	913,34 (0,6)	915,4 (0,20)
P8	100	865,51 (1,59)	865,51 (0,70)
P9	150	1163,83 (6,21)	1164,85 (3,56)
P10	199	1412,53 (15,97)	1408,56 (10,1)
P11	120	1043,38 (2,09)	1042,20 (4,74)
P12	100	819,55 (6,1)	819,55 (1,04)
P13	120	1562,77 (2,99)	1543,85 (2,97)
P14	100	866,36 (4,99)	866,36 (0,60)
SOMA		13769,16 (53,81)	13711,1 (34,67)

A TABELA 5.5 a seguir, apresenta uma comparação entre PF + HTSA e PF + HTSA + MF.

**TABELA 5.5 – Comparação entre os resultados dos algoritmos PF + HTSA e PF + HTSA + MF para as 14 instâncias de Cristofides *et al.*, 1979 (PRV)**

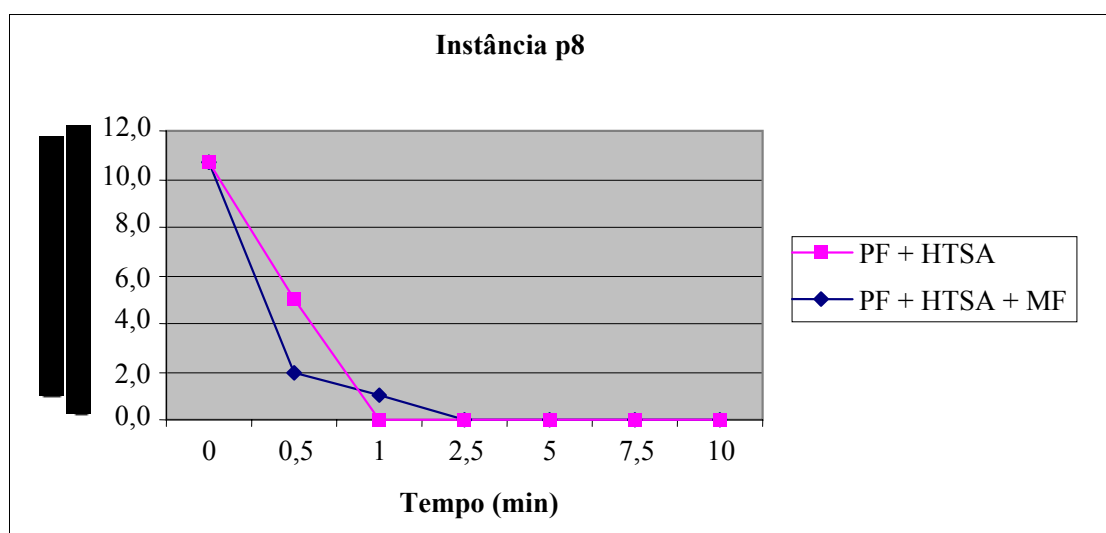
Instâncias	Conjunto de parâmetros padrão		Variação do conjunto de parâmetros padrão	
	PF + HTSA	PF + HTSA + MF	PF + HTSA	PF + HTSA + MF
P1	524,80	524,62	524,61	524,61
P2	837,32	838,75	835,26	835,26
P3	828,91	827,59	825,45	826,64
P4	1048,83	1041,08	1029,71	1029,85
P5	1353,64	1334,42	1314,06	1313,03
P6	560,38	555,43	555,43	555,43
P7	923,65	913,34	909,68	915,4
P8	865,51	865,51	865,51	865,51
P9	1168,89	1163,83	1165,67	1164,85
P10	1431,05	1412,53	1428,39	1408,56
P11	1043,14	1043,38	1042,12	1042,20
P12	825,64	819,56	819,56	819,56
P13	1546,59	1562,77	1544,03	1543,85
P14	866,37	866,37	866,37	866,37
DPM%	1,01%	0,60%	0,33%	0,27%

onde:

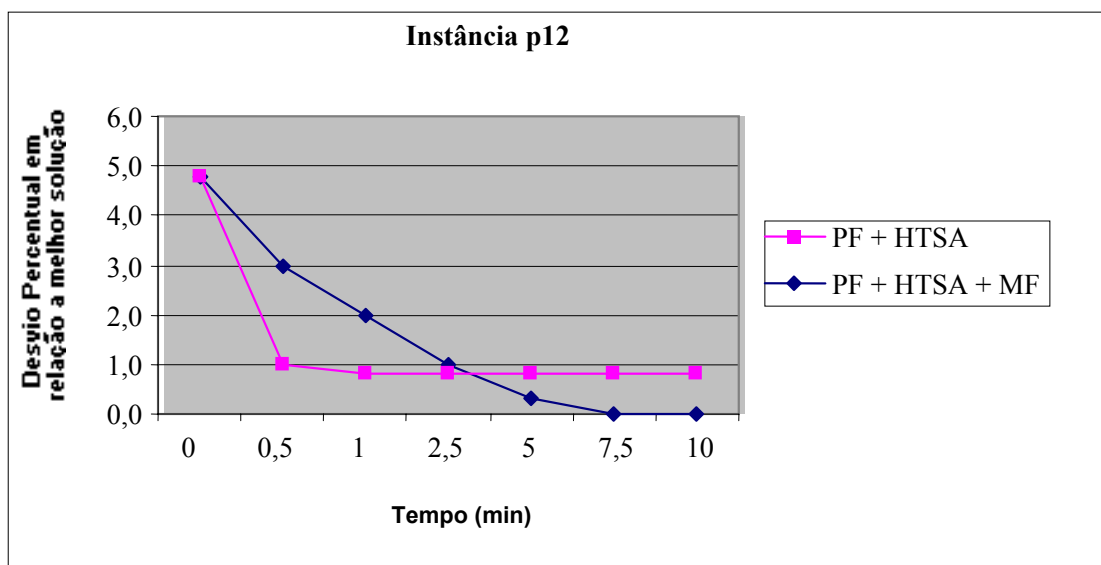
DPM: desvio percentual médio em relação as melhores soluções reportadas na literatura

Os resultados da TABELA 5.5 indicam que o algoritmo PF + HTSA + MF apresenta soluções com qualidade superior às obtidas por PF + HTSA. Além disso, a incorporação de memória de frequência tornou o algoritmo consideravelmente mais robusto à variação dos parâmetros. Entretanto, a obtenção destas soluções de qualidade superior requereu um aumento no esforço computacional em média de 337% (conjunto de parâmetros padrão) e 215% (outras conjuntos) maior que o utilizado por PF + HTSA.

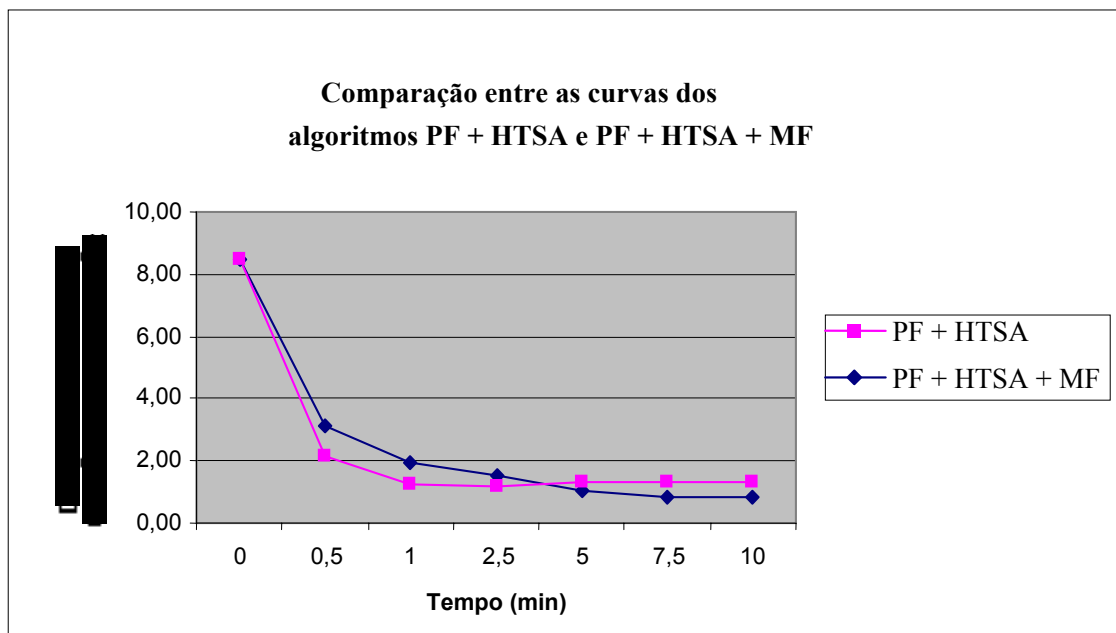
Ao se executar aos algoritmos PF + HTSA e PF + HTSA + MF no mesmo tempo computacional, é possível verificar que PF + HTSA + MF exibe uma tendência a uma convergência mais lenta, justificando assim o aumento no esforço computacional. Os gráficos 5.1 e 5.2 exibem este comportamento, tomando como exemplo as instâncias P8 e P12. O gráfico 5.3 exibe a convergência média dos dois algoritmos para todas as 14 instâncias. A análise do último gráfico sugere que a escolha dentre os dois algoritmos depende fundamentalmente do tempo computacional disponível – mantido o ambiente computacional e até 5 minutos de execução, sugere-se o uso de PF + HTSA. Caso tempos maiores sejam aceitáveis, PF + HTSA + MF parece ser uma melhor escolha.



**FIGURA 5.3 - Gráfico 5.1 de convergência dos algoritmos PF + HTSA e PF + HTSA + MF aplicados à instância P8**



**FIGURA 5.4 - Gráfico 5.2 de convergência dos algoritmos PF + HTSA e PF + HTSA + MF aplicados à instância P12**



**FIGURA 5.5 – Gráfico 5.3 de convergência média dos algoritmos PF + HTSA e PF + HTSA + MF para as 14 instâncias de Christofides et al.,1979**

### 5.3 Aplicação da Abordagem HTSA ao Problema de Roteirização de Veículos com Janelas de Tempo (PRVJT)

Nesta segunda etapa, o algoritmo PF + HTSA foi modificado para o tratamento de problemas de roteirização em que existem restrições de janela de tempo para clientes. Obedecendo a prática comum na literatura, o algoritmo tem como objetivo primário, a minimização do número de veículos que suprem a demanda dos clientes, e como objetivo secundário, a minimização da distância total percorrida.

Além da hierarquização dos objetivos e da verificação da viabilidade das janelas de tempo, outra alteração foi implementada, desta vez em relação à estrutura de vizinhança. Dada a maior restritividade do PRVJT, a heurística de inserção e troca de nós utilizada (DROR & LEVY, 1986) apresenta maior dificuldade de obter caminhos factíveis a regiões com soluções de maior qualidade; por esta razão, além de experimentos utilizando a heurística de troca e inserção foi implementada uma simplificação da estrutura de vizinhança  $\lambda$ -troca (OSMAN & CHRISTOFIDES, 1989, 1994), citada neste trabalho nas seções 2.5 do capítulo 2, com o objetivo de alargar o alcance de novas regiões factíveis.

Conforme já apresentado, o mecanismo de  $\lambda$ -troca possibilita que através de um par  $(R_p, R_q)$  se defina a ordem de até  $\lambda$  clientes de duas diferentes rotas a serem trocados. Tomando  $\lambda = 2$ , o operador  $(0,1)$  indica a inserção de um cliente da rota  $q$  na rota  $p$ ; os operadores  $(1,0)$ ,  $(2,0)$  e  $(0,2)$  indicam a inserção de um cliente da rota  $q$  na rota  $p$ , dois clientes da rota  $q$  na rota  $p$ , e dois clientes da rota  $p$  na rota  $q$  ou mais clientes de uma rota para outra. De forma similar, os operadores  $(1,1)$ ,  $(1,2)$ ,  $(2,1)$  e  $(2,2)$  indicam a troca de um ou mais clientes entre as rotas  $p$  e  $q$ . Como a vizinhança 2-troca é consideravelmente maior que aquela então adotada e requer tempos computacionais bastante elevados, adotou-se a estratégia *primeiro melhor* (*first-best*) que seleciona sempre a primeira solução que melhora a solução corrente (THANGIAH *et al.*, 1994). Além disso, movimentos que implicam na retirada de 2 clientes  $k$  e  $l$  da rota  $i$  para inserção na rota  $j$  (operadores  $(2,0)$ ,  $(0,2)$ ,  $(1,2)$ ,  $(2,1)$  e  $(2,2)$ ) se restringem às situações em que  $k$  e  $l$  são adjacentes quanto à ordem de visita tanto na rota  $i$  como na rota  $j$ .

### 5.3.1 Experimentos e Resultados Computacionais

O algoritmo PF + HTSA aplicado a instâncias com janela de tempo (PFJT + HTSA), com a vizinhança original e com vizinhança alargada, foi aplicado a exemplos clássicos do PRVJT, propostos por SOLOMON (1987). Estes exemplos são divididos em seis conjuntos (R1, R2, C1, C2, RC1 e RC2) diferindo entre si pelo modo como os dados geográficos (coordenadas  $x$  e  $y$  do nó  $i$ , demanda, janela de tempo inicial e final e tempo de serviço em cada cliente) são gerados. Dentre as características consideradas na geração destes exemplos, inclui-se a estrutura da instância (coordenadas dos nós geradas aleatoriamente segundo uma distribuição uniforme ou agrupados em áreas ou um híbrido entre os anteriores), posição e largura das janelas de tempo (mais ou menos apertadas) e tempos de serviço (pequenos ou grandes). Cada instância de um mesmo conjunto possui as mesmas coordenadas  $x$ ,  $y$ , demandas e tempo de serviço; já o restante dos dados geográfica é particular de cada instância.

Para cada instância foi utilizado o conjunto de parâmetros padrão já definidos em HTSA para o PRV. Para todas as instâncias foram realizadas 24000 iterações sem melhoria (critério de parada).

A TABELA 5.6 apresenta a média dos custos das soluções, de cada um dos seis conjuntos, obtidos pelos algoritmos PFJT + HTSA com vizinhança original e acrescido do mecanismo de 2-troca simplificado (PFJT + HTSA + 2-troca\*). Os resultados médios são apresentados em termos de número de veículos, custo total associado (distância) e tempo computacional para obtenção da melhor solução (em minutos). A última linha da tabela exhibe a média total dos seis conjuntos para cada algoritmo. Os resultados na íntegra são apresentados nos ANEXO 1.

**TABELA 5.6 - Resultados computacionais PFJT + HTSA e PFJT + HTSA + 2-troca\* para os 6 conjuntos das instâncias de Solomon, 1987 (PRVJT)**

Conjunto	PFJT + HTSA			PFJT + HTSA + 2-troca*		
	Número de Veículos	Custo	Tempo (min)	Número de Veículos	Custo	Tempo (min)
R1	12,83	1211,22	5,45	12,92	1219,80	5,5
C1	10,00	829,68	2,17	10,00	831,15	2,36
R2	3,09	972,34	9,82	3,18	967,20	13,05
C2	3,00	592,80	4,34	3,00	606,16	3,32
RC1	12,63	1370,98	5,16	12,88	1384,91	5,04
RC2	3,50	1176,46	9,46	3,88	1126,93	9,29
Média total	7,5	1025,58	6,06	7,64	1022,69	6,42

A análise da TABELA 5.6 indica que ao se incluir a vizinhança 2-troca\*, obteve-se soluções cujas distâncias são, em média, menores do que aquelas obtidas com a vizinhança anterior. Isto se deve ao fato desta estrutura de vizinhança guiar a busca a explorar novas regiões factíveis e promissoras. Entretanto, a inclusão desta vizinhança provoca também um aumento no número de veículos requeridos de 1,83% e do esforço computacional de 5,6%.

A TABELA 5.7 apresenta uma comparação, para os seis conjuntos de instâncias, entre os resultados médios (em termos de número de veículos e distância) obtidas pelos algoritmos: PFJT + HTSA, PFJT + HTSA + 2-troca\*, RT ( ROCHAT & TAILLARD, 1995), TAN (TAN *et al.*, 2001), LC03 (BOUTHILLIER & CRAINIC, 2005), USTA (CORDEAU *et al.*,2001) e as melhores soluções reportadas na literatura.

**TABELA 5.7 - Comparação entre os resultados computacionais de 5 algoritmos e as melhores soluções reportadas na literatura para os 6 conjuntos de instâncias de Solomon**

Algoritmos	Conjunto						Média Total
	R1	C1	R2	C2	RC1	RC2	
PFJT + HTSA <sup>1</sup>	12,83	10,00	3,09	3,00	12,63	3,50	7,5
	1211,22	829,68	972,34	592,80	1370,98	1176,46	1025,58
PFJT + HTSA +2-troca* <sup>1</sup>	12,92	10,00	3,18	3,00	12,88	3,88	7,64
	1219,80	831,15	967,20	606,16	1384,91	1126,93	1022,69
TAN <sup>2</sup>	13,83	10,00	3,81	3,25	13,63	4,25	8,12
	1266,37	870,86	1080,23	634,88	1458,16	1293,38	1100,65
RT <sup>3</sup>	12,25	10,00	2,91	3,00	11,88	3,38	<b>7,23</b>
	1208,5	828,38	961,72	589,86	1377,39	1117,44	<b>827,64</b>
UTSA <sup>4</sup>	12,5	10,00	11,50	2,72	3,00	3,25	7,16
	1257,86	828,38	1389,77	969,57	589,85	1134,51	1028,33
LC03 <sup>5</sup>	12,08	10,00	2,73	3,00	11,50	3,25	<b>7,09</b>
	1209,19	828,38	963,62	589,86	1389,22	1143,70	<b>1020,66</b>
Melhores soluções reportadas	11,92	10,00	11,50	2,73	3,00	3,25	<b>7,07</b>
	1209,89	828,38	1384,16	951,91	589,86	1119,35	<b>1013,92</b>

onde:

1- AMD Athlon 2600<sup>+</sup> MHz, 2- P II 266 MHz, 3 - S.G. 100 MHz 4 - Sun Ultrasparc 10 440 MHz, 5 – 5xP 850 MHz

Observa-se na TABELA 5.6 que a aplicação de PFJT + HTSA e PFJT + HTSA + 2-troca\* produziu soluções, em média, de qualidade comparável a vários algoritmos da literatura. Cabe ressaltar que o número elevado de veículos de ambos os algoritmos se deve ao fato de não ter sido empregada nenhuma heurística específica para redução de rotas. A hierarquização dos objetivos, aqui adotada, não é suficiente para redução do número de veículos uma vez que a redução de uma única rota implica em uma série de passos que designem todos os clientes desta rota para outras rotas. Com a hierarquização dos objetivos, a redução de rotas ocorre somente se houver rotas com no



máximo dois clientes, cuja inserção conjunta em outra rota não gere infactibilidade. A grande dificuldade de um procedimento deste tipo consiste na existência de janelas de tempo.

Observa-se também que algoritmos com resultam em menores distâncias também estão associados a um número maior de veículos. Esta constatação é corroborada em vista dos resultados obtidos pelo algoritmo RT (ROCHAT & TAILLARD, 1995) e LC03 (BOUTHILLIER & CRAINIC, 2005), descritos nas seções 4.3 e 4.9 do capítulo 4. O primeiro não utiliza mecanismos de redução de rotas e aplica o conceito de memória adaptativa para recombinação de boas soluções em termos distância, gerando soluções com menores distâncias e com um número maior de veículos. O segundo utiliza cadeias de ejeção que permitem a redução do número de veículos e heurísticas clássicas de busca, gerando soluções com maiores distâncias e número menor de veículos.

## CAPÍTULO 6 CONCLUSÕES E PERSPECTIVAS DE PESQUISA

Neste trabalho foram apresentados os resultados de experimentos realizados nos exemplos clássicos do problema de roteirização de veículos com e sem janela de tempo. Estes resultados foram obtidos através do refinamento da abordagem adaptativa de busca tabu HTSA (PUREZA, 1996), ou seja, realizou-se uma ampla análise paramétrica em seus parâmetros exógenos e com o objetivo de se explorar novas regiões factíveis e promissoras houve a inclusão de mecanismos mais sofisticados de estrutura de vizinhança e da meta-heurística Busca Tabu.

Esta abordagem HTSA tem como base os princípios da abordagem adaptativa HTA (PUREZA & FRANÇA, 2001), visando à integração de estratégias de intensificação e diversificação através da variação sistemática de valores de parâmetros tabu selecionados de acordo com padrões da trajetória da busca.

Para os experimentos com o PRV, a abordagem HTSA obteve soluções comparáveis às da abordagem HTA e uma nova marca em termos de qualidade de solução para um conjunto de instâncias clássicas de 50 a 199 nós. Na incorporação da estratégia de diversificação baseada na frequência de atributos do movimento, se obteve um algoritmo (PF + HTSA + MF) menos dependente dos valores de parâmetros adotados, com soluções, em média, ligeiramente superiores às obtidas pelo algoritmo PF + HTSA, tanto para o conjunto de parâmetros padrão quanto para suas variações. Entretanto, a obtenção destas soluções requereu um aumento no esforço computacional consideravelmente maior, possivelmente, justificável pelo fato de haver uma tendência a convergência mais lenta (FIGURA 5.5).

Para o tratamento do PRVJT, a abordagem JT + HTSA obteve soluções comparáveis a vários algoritmos da literatura para 56 instâncias clássicas de 100 nós. Devido o PRVJT ser altamente restritivo os experimentos envolveram a utilização da vizinhança empregada no algoritmo original assim como a inclusão da vizinhança 2-troca

simplificada. Os resultados obtidos mostram a superioridade em termos de distância das soluções geradas com a inclusão do mecanismo 2-troca\* isto se deve ao fato desta estrutura de vizinhança guiar a busca a novas regiões factíveis e promissoras, contudo houve um aumento de 1,83% no número de veículos requeridos e de 5,6% no esforço computacional.

A análise final dos resultados obtidos pela abordagem HTSA ao PRV e ao PRVJT mostra que o desempenho da abordagem HTSA é considerável ao se notar a simplicidade dos recursos da abordagem frente às heurísticas que a superam. A variação sistemática de valores de parâmetros tabu selecionados visando o padrão de estagnação se mostrou eficiente, porém esta variação não é capaz de guiar plenamente a busca a regiões promissoras não exploradas anteriormente. Por esta razão, adicionou-se estratégias de diversificação baseada na frequência dos atributos que se mostrou eficaz, porém houve um aumento significativo em relação ao tempo computacional requerido assim, acredita-se que novos experimentos para o desenvolvimento de estratégias de diversificação baseada na frequência dos atributos devam ser explorados.

Outras sugestões para pesquisas futuras com HTSA, em especial para o tratamento do PRVJT, incluem a utilização ou desenvolvimento de uma heurística de redução do número de veículos, a busca no espaço de soluções infactível, e a utilização de estruturas de vizinhança mais adequadas para a roteirização com restrições de janela de tempo. Outra promissora sugestão para o estudo do PRVJT são as cadeias de ejeção (BOUTHILLIER & CRAINIC, 2005).

## REFERÊNCIAS BIBLIOGRÁFICAS

- BALLOU, R. H. **Logística empresarial: Transporte, Administração de Materiais e Distribuição Física**. São Paulo: Atlas, 1993.
- BARBAROSOGLU, G.; OZGUR. A Tabu Search Algorithm for the Vehicle Routing Problem. **Computers & Operations Research**, 26, 255-270, 1999.
- BELLMORE, M; NEMHAUSER, G. L. The Traveling Salesman Problem: a survey. **Operations Research**, 16, 538-558, 1968.
- BODIN, L.D.; GOLDEN, B.; BALL, M. Routing Scheduling of vehicles and crews: the state of art. **Computers & Operations Research**, 9, 63-212, 1983.
- BOUTHILLIER, A.L.; CRAINIC, T.G. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows, **Computers & Operations Research**, 32, p. 1685-1708, 2005.
- CARLTON, W.B. **A tabu search approach to the general vehicle routing problem**. Ph.D. Thesis, The University of Texas at Austin, Austin, TX, 1995.
- CHAKRAPANI, J.; SKORIN-KAPOV, J. Massively parallel Tabu Search for the Quadratic Assignment Problem, **Annals of Operations Research**, 41, 327-41, 1993.
- CHRISTOFIDES, N.; ELION, S. An Algorithm for the Vehicle Dispatching Problem. **Operations Research Quarterly**, 20, 309-318, 1969.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. The Vehicle Routing Problem, **Combinatorial Optimization**, Wiley, Chichester, 315 –338, 1979.
- CLARKE, G.; WRIGHT, W. J. Scheduling of Vehicle from a central Depot to a Number of Delivery Points, **Operations Research**, 12, 568-81, 1964.
- COUNCIL OF SUPPLY CHAIN MANAGEMENT PROFESSIONALS - CSCMP., Apresenta artigos, publicações, grupos de estudo e etc. sobre logística. Disponível em < <http://www.cscmp.org> >. Acesso em 23 de janeiro de 2004.
- CORDEAU, J.F.; LAPORTE, G. Tabu Search Heuristics for the Vehicle Routing Problem. **Cahiers du GERAD G-2002-15**, École des Hautes Études Commerciales, Montreal, 2002.

- CORDEAU, J.F.; DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M.M.; SOUMIS F. The VRp with times windows. In: Toth P., Vigo D. editors. The vehicle routing problem, SIAM, **Monographs on Discrete Mathematics and Applications**, Philadelphia, PA: SIAM; 157-93, 2002a.
- CORDEAU, J.F.; GENDREAU, M.; LAPORTE, G.; POTVIN, J-Y.; SEMET, F. A guide to vehicle routing heuristics. **Journal of the Operational Research Society** 53, 512 - 522, 2002b.
- CORDEAU, J-F; LAPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows, **Journal of Operational Research Society** 52, 928-936, 2001.
- CORDEAU, J-F; GENDREAU, M.; LAPORTE, G. A tabu search heuristic for periodic and multi- depot vehicle routing problems. **Networks**, 30(2), 105-119, 1997.
- CORDONE, R.; WOLFER-CALVO, R. ; A Heuristic for the Vehicle Routing Problem with Time Windows, Internal Report. Departament of Eletronics and Information Polytechnic of Milan, Milan, Italy, **Journal of Heuristics**.
- CULLEN, F.; JAVIS, J., RATLIFF, H. D. Set-Partitioning Based Heuristics for Interactive Routing, **Networks**, 11, 125 -143, 1981.
- CUNHA, C. B. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. **Transportes**, Rio de Janeiro, 8(2), 51-74, 2000.
- CROES, G.A. A method for solving traveling-salesman problem, **Operations Research**, 6, 791-812, 1958.
- DROR, M.; LEVY, L. A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementation for Inventory Routing. **Computers & Operations Research**, 13(1), 33-45, 1986.
- EILON, S; EATSON-GANDY, C.; CHRISTOFIDES, N. **Distribution Management: Mathematical Modeling and Practical Analysis**. New York: Hafner, 1971.
- FEO, T.; RESENDE, M. G. C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, 6, 109-133, 1995.
- FERNANDES, A. J., **Avaliação da Implantação de um Software Roteirizador em um Sistema de Distribuição de Bebidas: estudo de caso**. 2001. Dissertação (Mestrado em Engenharia de Produção), Departamento de Engenharia de Produção, Universidade Federal de São Carlos.
- FISHER, M. L.; JAIKUMAR, R. A Generalized Assignment Heuristic for Vehicle Routing. **Networks**, 11, 109-124, 1981.

- FISHER, M.L., Optimal solution of vehicle routing problems using minimum K-trees, **Operations Research**, 42, 626-642, 1994.
- GAREY, M.D.; JOHNSON, D.S. **Computers and Intractability – A Guide to the Theory of NP-Completeness**, Freeman, San Francisco, 1979.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G., New Insertion and Post optimization Procedures for the Traveling Salesman Problem, **Operations Research**, 40(6), 1992.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. A Tabu Search Heuristic for the Vehicle Routing Problem. **Management Science**, 40(10), 1994.
- GILLET, B.E; MILLER, L.R. A heuristic algorithm for the vehicle-dispatch problem. **Operations Research**, 22(2), 340-349, 1974.
- GLOVER, F. Future Paths for Integer Programming and Links to Artificial Intelligence, **Computers & Operations Research**, 13, 533-549, 1986.
- GLOVER, F. Tabu Search. Part I, **ORSA Journal on Computing**, 1(3), 19-206, 1989.
- GLOVER, F., Tabu Search. Part II, **ORSA Journal on Computing**, 2(1), 1989.
- GLOVER, F.; LAGUNA, M. **Tabu Search**. Klumer, Boston, 1997
- HALL, R. W.; PARTYKA, J. G. Vehicle Routing Software Survey. In: OR/MS Today, disponible em: <[www.lionhrtpub.com/orms/surveys/Vehicle\\_Routing/vrss.html](http://www.lionhrtpub.com/orms/surveys/Vehicle_Routing/vrss.html)>.
- HOMBERGER, J; GEHRING, H. Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows, **INFOR**, 37, 297-318, 1999.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**, Ann Harbor, University of Michigan Press, 1975.
- HUBSCHER, R.; GLOVER, F. Applying Tabu Search with Influential Diversification to Multi-processor Scheduling. **Computers & Operations Research**, 21 (8), p. 877-884, 1994.
- KIRKPATRICK; S.; Optimization by Simulated Annealing: Quantitative Studies, **J. Stat. Physics**, 34, 976-986, 1984.
- LAPORTE, G.; SEMET F. **Classical heuristics for the capacitated VRP**. In: Toth P e Vigo D, editors. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM Publishing: Philadelphia, PA, 109-128, 2002.
- LAGUNA, M., Tabu Search Tutorial. **II Escola de Verano Latino-Americana de Investigación Operativa**, Rio de Janeiro, Brazil, 1995.

- LAWLER, E.,L.; LENSTRA, K.J.; KAN, A.H.G.R., **The Traveling Salesman Problem – A Guided Tour of Combinatorial Optimization**, John Wiley & Sons, New York, 1986.
- LIN,S. Computer Solutions of the TSP. **Bell System Tech. J.**, 44, 2245-70, 1965.
- LOPES, I. Na rota da Eficiência e Economia. **Revista Tecnológica**, 2(11), 20, 1996.
- MELO, A. C. S.; **Avaliação do Uso de Sistemas de Roteirização de Veículos**. Dissertação de Msc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- NOVAES, A. G. **Logística e Gerenciamento da Cadeia de Distribuição: Estratégia, Operação e Avaliação**. Rio de Janeiro, Editora Campus, 2001.
- OSMAN, I. H; CHRISTOFIDES, N. Capacitated clustering problems by hybrid simulated annealing and tabu search. **International Transactions in Operations Research** 1, 317-336, 1994.
- OSMAN, I. H. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. **Annals of Operations Research**, 41, 1993.
- OSMAN, I. H; CHRISTOFIDES, N. Simulated annealing and descent algorithms for capacitated clustering problems. Imperial College, **The Management School**, University of London, 1989.
- PELIZARO, C. **Avaliação de Desempenho de Algoritmo de um Programa Comercial para Roteirização de Veículos**, 2000. Dissertação (Mestrado em Engenharia Civil – Transportes)- Escola de Engenharia, Departamento de Engenharia Civil, Universidade de São Paulo, São Carlos.
- POTVIN, Y.J.; KERVAHUT, T.; GARCIA, B.L.; ROUSSEAU, J.M. The vehicle routing problem routing with time Windows – Part I: Tabu Search, **INFORMS Journal on Computing**, 8, 158-164, 1996.
- PUREZA, V.; MORABITO, R. . Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers & Operations Research**, 2005.
- PUREZA, V.; FRANÇA, P. Uma abordagem Adaptativa de Busca Tabu aplicada ao Problema de Roteamento de Veículos, **Transportes**, 9(2), 28-47, 2001.
- PUREZA, V. **Abordagens Adaptativas de Metaheurística Tabu**, Tese de Doutorado, 1996, Faculdade de Engenharia Elétrica, DENNIS, UNICAMP.
- PUREZA V, FRANÇA P. Vehicle Routing Problems via Tabu Search Metaheuristic. **Centre de recherche sur les transports**, Université de Montréal, *cahier CRT 747*, 1991.

- PUREZA, V.; FRANÇA, P. *Problemas de Roteamento de Veículos via Metaheurística Tabu*, 1990. Tese de Mestrado, Faculdade de Engenharia Elétrica, DENISIS, UNICAMP.
- REGO, C. (1998). A subpath ejection method for the vehicle routing problem, *Management Science*, 44,1447-1459, 1998.
- REGO, C.; ROUCAIROL, C. A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In: Osman I.H. e Kelly J.P.(eds). **Meta-heuristics: Theory and Applications**. *Kluwer*: Boston, 661-675, 1996.
- ROCHAT, Y.; TAILLARD, E. Probabilistic diversification and intensification in local search for vehicle routing, **Journal of Heuristics**, 1, 147-167, 1995.
- RUSSEL, R. A.; Hybrid Heuristics for the Vehicle Routing Problem with Time Windows, **Transportation Science**, 29, 156-166, 1995
- SALIBY, E. Lidando com Sazonalidade no Processo Logístico. **Revista Tecnológica**. 4(49), 20, 1999.
- SOLOMON, M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Windons Constrains. **Operations Research**, 35(2), 254-265, 1987.
- TAILLARD, E.; BADEAU, P.; GENDREAU, M.; GEURTIN, F.; POTVIN, J.P. A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows, **Transportation Science**, 31, 170-186, 1997.
- TAILLARD, E. Parallel Iterative Search Methods for Vehicle Routing Problems, **Networks**, 23(8), 1993.
- TAN, K. C.; LEE, L.H.; ZHU, Q.L.; OU, K. Heuristic methods for vehicle routing problem with time windows, **Artificial Intelligence in Engineering** 15, p. 281-295, 2001.
- TOTH, P.; VIGO, D. The granular tabu search (and its application to the vehicle routing problem). **Technical report OR/98/9**, DEIS, Università di Bologna: Italy, 1998.
- THANGIAH, S.R.; OSMAN, I.H.; SUN, T. Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows, working paper UKC/OR 1994/4, Institute of Mathematics and Statistics, University of Kent, Canterbury.
- TYAGI, M. S. A Pratical Method for Truck Dispatching Problem. **Journal of Operational Research Society Japan**, 10, 76-92, 1968.
- XU, J.; KELLY, J.P. A Network Flow-Based Tabu Search Heuristics for the Vehicle Routing Problem. **Transportation Science**, 30, 379-393, 1996.



WILLARD, J. A. G., **The Vehicle Routing Using r-Optimal Tabu Search**, 1989. Tese de Mestrado, The Management School, Imperial College, Inglaterra.

## APÊNDICE A

Os resultados, na íntegra, obtidos pelos algoritmos: PFJT + HTSA com a vizinhança original e acrescido do mecanismo de 2-troca simplificado (PFJT + HTSA + 2-troca\*), descritos na seção 5.3 do capítulo 5, para as 56 instancias de SOLOMON 1987, descritas na seção 5.3.1 do capítulo 5, seguem, respectivamente, na TABELA A.1 e TABELA A.2. Estes resultados estão em termos de número de veículos (nv), custo total (distância) e tempo computacional para obtenção da melhor solução (em minutos).

**TABELA A.1 Resultados computacionais obtidos pelo algoritmo PFJT + HTSA para as 56 instancias de instâncias de Solomon, 1987 (PRVJT).**

Instancias	nv	Distância	Tempo/min	Instancias	nv	Distância	Tempo/min
r101	19	1662,60	2,90	r201	4	1288,79	6,64
r102	18	1507,06	4,66	r202	4	1159,44	11,10
r103	14	1273,20	6,24	r203	3	1016,01	16,94
r104	10	1020,90	5,91	r204	3	808,94	9,88
r105	15	1392,66	5,72	r205	3	1052,16	7,21
r106	13	1272,05	4,92	r206	3	983,08	6,59
r107	11	1099,86	7,24	r207	3	853,78	8,43
r108	10	964,98	3,55	r208	2	746,77	15,62
r109	12	1177,76	8,27	r209	3	961,03	8,00
r110	11	1107,66	3,66	r210	3	1011,13	7,11
r111	11	1083,43	3,50	r211	3	814,61	10,49
r112	10	972,52	8,85				
c101	10	828,93	1,46	c201	3	591,55	2,69
c102	10	828,93	2,35	c202	3	591,55	3,60
c103	10	828,93	2,51	c203	3	599,62	4,03
c104	10	835,66	4,21	c204	3	605,33	11,53
c105	10	828,93	1,60	c205	3	588,87	3,15
c106	10	828,93	1,60	c206	3	588,49	2,99
c107	10	828,93	1,68	c207	3	588,49	3,76
c108	10	828,93	1,80	c208	3	588,49	3,00
c109	10	828,93	2,30				
rc101	15	1649,75	3,35	rc201	4	1519,37	5,98
rc102	14	1491,58	3,25	rc202	4	1296,6	13,33
rc103	11	1323,56	7,00	rc203	3	1138,05	15,70
rc104	10	1175,62	3,55	rc204	3	886,51	9,87
rc105	15	1554,95	6,04	rc205	4	1384,16	4,68
rc106	13	1395,42	5,63	rc206	3	1191,95	5,32
rc107	12	1237,67	8,07	rc207	4	1092,87	10,68
rc108	11	1139,27	4,40	rc208	3	902,18	10,11

**TABELA A.2 Resultados computacionais obtidos pelo algoritmo PFJT + HTSA + 2-troca\* para as 56 instancias de instâncias de Solomon, 1987 (PRVJT).**

Instancias	nv	solução	Tempo/min	Instancias	nv	solução	Tempo/min
r101	19	1655,26	7,20	r201	4	1294,49	6,63
r102	18	1492,94	8,53	r202	4	1159,30	5,42
r103	15	1255,77	8,26	r203	3	1012,81	7,77
r104	10	1021,10	1,37	r204	3	801,50	15,70
r105	15	1380,86	5,30	r205	3	1060,53	14,01
r106	13	1264,41	5,94	r206	3	944,46	19,42
r107	11	1129,74	5,27	r207	3	873,67	11,43
r108	10	958,58	10,16	r208	2	740,58	15,71
r109	12	1178,70	8,35	r209	3	950,50	20,54
r110	11	1115,67	5,44	r210	4	973,99	10,21
r111	11	1106,82	1,17	r211	3	827,40	16,71
r112	10	1077,71	1,11				
c101	10	828,93	1,59	c201	3	591,55	1,2
c102	10	828,93	2,16	c202	3	591,55	4,46
c103	10	828,75	3,78	c203	3	681,44	2,56
c104	10	848,66	4,03	c204	3	618,82	11,8
c105	10	828,93	1,83	c205	3	600,45	1,10
c106	10	828,93	1,80	c206	3	588,49	1,94
c107	10	828,93	1,98	c207	3	588,49	4,33
c108	10	828,93	1,91	c208	3	588,49	4,82
c109	10	829,38	3,98				
rc101	16	1658,89	7,58	rc201	5	1420,7	4,4
rc102	14	1497,51	9,61	rc202	4	1227,94	12,9
rc103	11	1330,07	5,59	rc203	4	1076,22	6,15
rc104	11	1178,12	4,59	rc204	3	854,58	16,57
rc105	15	1558,14	5,14	rc205	5	1308,22	5,22
rc106	13	1412,46	4,06	rc206	4	1142,56	6,81
rc107	12	1277,06	1,15	rc207	3	1111,29	12,16
rc108	11	1167,02	2,60	rc208	3	873,95	10,09

