

Universidade Federal de São Carlos
Centro de Ciências Exatas e Tecnologia
Departamento de Engenharia de Produção
Programa de Pós-Graduação em Engenharia de Produção

**Desenvolvimento e aplicação de algoritmos
adaptativos de busca tabu para a resolução de
Problemas de Roteamento de Veículos Periódico
(PRVP)**

Renato Hallal

Orientadora: Profa. Dra. Vitória M. Pureza

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Engenharia
de Produção como requisito para obtenção
do título de Mestre em Engenharia de
Produção.

Dezembro de 2004
São Carlos - SP, Brasil.

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

H174da

Hallal, Renato.

Desenvolvimento e aplicação de algoritmos adaptativos de busca tabu para a resolução de Problemas de Roteamento de Veículos Periódico (PRVP) / Renato Hallal. - São Carlos : UFSCar, 2005.
108 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2004.

1. Distribuição física de bens. 2. Busca – tabu. 3. Otimização combinatória. 4. Problema de roteamento de veículos periódico. 5. Logística. I. Título.

CDD: 658.788 (20^a)



cada um de nós compete uma tarefa específica, na difusão do bem. Erga-se, para trabalhar, porque as tarefas são muitas e importantes, e poucos são os que têm consciência delas. Ajude o mundo, para que o mundo possa ajudá-lo. Estenda seus braços eficientes no cultivo do bem, para que, quando os recolher, os traga cheios dos frutos abençoados da felicidade e do amor.

Aos Meus Pais, Clive Hallal e Lourdes T. C. Hallal

Agradecimentos

Agradeço a Deus e a minha família, Clive Hallal, Lourdes T. C. Hallal, Rogério Hallal, Teresa C. Hallal de Oliveira e Emerson de Oliveira pela ajuda e incentivo em todos os momentos.

Especialmente aos amigos que me ajudaram durante o desenvolvimento desse trabalho, ao mestre José Laércio Doricio, ao mestre Isaías Torres e ao doutor Luís Gustavo Nonato.

Às pessoas por quem tenho muito carinho, Fabiana Simões, Andreza Dias, Juliana Rangel, Fabio Azeka e Robson T. S. O. Junior.

Aos companheiros Marco e Alessandro, da secretaria do departamento de Engenharia de Produção.

À orientadora doutora Vitória M. Pureza.

Aos demais professores e amigos da USP e da UFSCar.

Sumário

Agradecimentos	iii
Sumário	iv
Lista de Figuras	vii
Lista de Tabelas	ix
Resumo	x
<i>Abstract</i>	xi
1 Introdução	1
1.1 Objetivo	6
1.2 Organização da dissertação	7
2 A meta-heurística busca tabu	8
2.1 Introdução	8
2.2 A busca tabu	9
2.2.1 Memória de curto prazo	10
2.2.2 Memória de longo prazo	14
2.2.2.1 Intensificação	16
2.2.2.2 Diversificação	17
2.2.2.3 Integração de intensificação e diversificação	18
2.3 Aplicações de busca tabu	19

3	PRV, PRVP e técnicas de resolução	21
3.1	Introdução	21
3.2	O problema de roteamento de veículos clássico (PRV)	23
3.3	Limitação dos métodos exatos	25
3.4	O problema de roteamento de veículos periódico (PRVP)	27
3.5	Descrição do problema	29
3.6	Heurísticas para o problema de roteamento de veículos periódico	32
3.6.1	Heurística de RUSSELL & IGO (1979)	33
3.6.2	Heurísticas de CHRISTOFIDES & BEASLEY (1984)	35
3.6.3	Heurística de TAN & BEASLEY (1985)	35
3.6.4	Heurística de RUSSELL & GRIBBIN (1991)	38
3.6.5	Heurística de CHAO, GOLDEN & WASIL (1995)	40
3.6.6	Heurística de CORDEAU, GENDREAU & LAPORTE (1997)	43
3.6.6.1	Construção da solução inicial	44
3.6.6.2	Funcionamento geral do algoritmo	44
4	Aplicação da abordagem adaptativa HTA ao PRV	47
4.1	Heurística de construção de rotas	47
4.2	Heurística de melhoria de rotas	49
4.3	A abordagem adaptativa HTA	52
4.3.1	Padrões da trajetória da busca	52
4.3.2	Identificação dos padrões da trajetória	54
4.4	Passos do algoritmo HTA para o PRV	56
4.5	Resultados do algoritmo HTA	58
5	Aplicação da abordagem HTA ao PRVP	61
5.1	Introdução	61
5.2	Designação dos clientes aos dias	62
5.2.1	Construção inicial 1 (CI-1)	62
5.2.2	Construção inicial 2 (CI-2)	69
5.2.3	Construção inicial 3 (CI-3)	72

5.3	Movimentos de melhorias	74
5.3.1	Troca e inserção	74
5.3.1.1	Definindo a família U de subconjuntos de clientes	75
5.3.2	Movimento <i>clean up</i>	81
5.4	Buscando melhores soluções	82
5.4.1	Analisando as soluções	82
5.4.2	Instante de avaliação das soluções globais (Av)	87
6	Resultados computacionais	90
6.1	Comparação dos resultados com heurísticas da literatura	94
7	Conclusões e perspectivas futuras	99
	Referências Bibliográficas	100
	Anexo A: Rotas geradas pelo programa Versão 1	i
	Anexo B: Rotas geradas pelo programa Versão 2	xvi
	Anexo C: Dados para os problemas com 50, 75 e 100 clientes	xxx

Lista de Figuras

3.1	Roteamento de quinze nós (clientes)	25
3.2	Roteamento dos clientes no horizonte de planejamento.	29
3.3	Roteamento utilizando o algoritmo de CLARKE & WRIGHT.	34
3.4	Representação de um modelo de rede de trabalho.	39
4.1	Movimento de Troca.	49
4.2	Movimento de Inserção	49
4.3	Movimento 2-OPT.	50
4.4	Representação da trajetória estagnada.	53
4.5	Representação da trajetória ascendente.	53
4.6	Representação da trajetória descendente.	54
4.7	Avaliação das médias nos dois últimos estágios na abordagem adaptativa HTA.	55
5.1	Escolhendo os centros.	63
5.2	Atribuindo os clientes aos dias.	67
5.3	Clientes atribuídos aos dias.	68
5.4	Solução inicial S1.	69
5.5	Lista ordenada e atribuição dos clientes aos dias.	70
5.6	Solução inicial S2.	71
5.7	Roteamento dos clientes.	72
5.8	Clientes atribuídos aos dias.	73
5.9	Solução inicial S3.	74
5.10	Escolhendo M1, α e β	77

5.11	Cálculo da menor distância ($d_{j,\alpha} + d_{j,\beta}$).	78
5.12	Movimentos admissíveis para os clientes 3 e 5. a) Solução inicial, b) Inserção do cliente 5 no dia 1, c) Inserção do cliente 3 no dia 2 e inserção do cliente 5 no dia 1 (troca) e d) Inserção do cliente 3 no dia 2.	79
5.13	Movimentos admissíveis de combinações de visita para os clientes 7 e 10. a) Solução inicial, b) Inserção do cliente 10 no dia 1, c) Inserção do cliente 10 no dia 1 e inserção do cliente 7 no dia 3 (troca) e d) Inserção do cliente 7 no dia 3.	81
5.14	Ilustração da aproximação dos custos guardados na lista LC pelo método dos mínimos quadrados. a) Trajetória ascendente, b) Trajetória descendente. c) Trajetória estagnada.	88

Lista de Tabelas

1.1	Transporte de carga em alguns países - 2000.	2
1.2	Índice Nacional da Variação do Custo do Transporte Rodoviário (INCTr). . .	3
2.1	Medidas de residência e de transição.	15
2.2	Aplicações de busca tabu.	20
3.1	Caracterização dos problemas de roteamento e programação.	22
3.2	Especificação dos dias de visitas para os clientes ($T = 2$).	28
4.1	Regras para trajetórias ascendentes.	58
4.2	Resultados computacionais de PF_D_HTA com os outros algoritmos de busca tabu.	58
5.1	Matriz $M_{j,t}$ ($\text{centros}(j) \times \text{dias}(t)$).	66
5.2	Possíveis pares de combinações de visitas.	84
6.1	Problemas tratados.	91
6.2	Resultados computacionais da VERSÃO 1.	92
6.3	Comparação dos resultados computacionais.	95
6.4	Comparando a melhor solução com as demais heurísticas.	97
6.5	Comparando os tempos da VERSÃO 2 com os da VERSÃO 1.	98
1	Dados para os problemas com 50 clientes.	xxx
2	Dados para os problemas com 75 clientes.	xxxii
3	Dados para os problemas com 100 clientes.	xxxv

Resumo

Esta pesquisa consiste no desenvolvimento de algoritmos para resolver o Problema de Roteamento de Veículos Periódico (PRVP), o qual tem sido pouco abordado na literatura de Pesquisa Operacional. O objetivo do PRVP é elaborar um conjunto de rotas para atender à demanda de clientes ao longo de um horizonte de planejamento. Cada cliente requer que as visitas aconteçam em uma combinação predefinida de dias. Foram desenvolvidas duas heurísticas para o PRVP, chamadas de VERSÃO 1 e VERSÃO 2. Na VERSÃO 1 são utilizados três tipos de construções iniciais para atribuir os clientes aos dias. Em seguida, são realizadas mudanças de combinações de dias de visitas na tentativa de melhorar a solução. O processo de busca por soluções é controlado por uma heurística tabu adaptativa da literatura que determina as ações de intensificação e diversificação, aplicado a cada dia do período. A VERSÃO 2 incorpora uma abordagem similar para o período como um todo. Resultados computacionais indicam que esta abordagem leva a soluções de boa qualidade.

Palavras chave: Problema de Roteamento de Veículos Periódico, Busca Tabu, Otimização Combinatória.

Abstract

This research consists of the development of algorithms to solve the Periodic Vehicle Routing Problem (PVRP), which has not received a great deal of attention in the O. R. literature. The objective of the PVRP is to elaborate a set of routes to attend to the customers' demand along a planning horizon. Each customer requests that the visits occur in a combination predefined of days. Two heuristics were developed for the PVRP. In the first heuristic, three types of initial solution construction are used to attribute the customers to days. After that, visiting day combinations are changed in order to improve the solution. The search process is controlled by an adaptative tabu heuristic from the literature which determines intensification and diversification actions, applied for each day in the period. The second heuristic incorporates a similar approach for the period as a whole. Computational results show that this approach leads to good solution.

Keywords: Periodic Vehicle Routing Problem, Tabu Search, Combinatorial Optimization.

1

Introdução

Nos últimos anos, o nível de competição entre empresas aumentou bastante. Os competidores, antes regionais, agora se encontram dispersos em mercados espalhados globalmente. O ambiente de competição empresarial passou a estar sujeito a incertezas crescentes, fazendo com que as empresas buscassem novas formas de sobrevivência, procurando tecnologias e técnicas capazes de mantê-las produzindo e obtendo lucros. Surge então a necessidade de uma reformulação empresarial e da busca de soluções que otimizem as várias funções de sistemas produtivos. Em especial, os consumidores de mercados competitivos esperam por produtos e serviços que satisfaçam suas expectativas de qualidade, tempo de entrega e custo (SIPPER & BULFIN, 1997). O consumidor passa, portanto, a desempenhar um papel de maior importância para as empresas, chegando a participar de projetos de novos produtos. A garantia de um nível de serviço diferenciado é hoje o objetivo de empresas que desejam permanecer no mercado e, diante disso, os empresários têm que revisar os setores das suas empresas para identificar novas oportunidades de reduzir custos e manter a competitividade. Dentre os vários setores, a área de logística recebeu atenção considerável.

A logística desempenha um papel cada vez mais importante nesse cenário de mercados dispersos, tornando-se uma área vital para diversas empresas. Conforme BALLOU (1993), o conceito de logística não engloba apenas a distribuição física de produtos finais, mas também a integração de todas as atividades relacionadas ao processamento de

pedidos, à manutenção de estoques, à movimentação, à armazenagem e manuseio de produtos e materiais, ao transporte e às informações a eles associados e à outras atividades de apoio. Assim, o conceito atual de logística considera todas as movimentações e armazenagens, que facilitam o fluxo de produtos desde o ponto de aquisição da matéria prima até o ponto de consumo final, assim como todos os fluxos de informações que colocam os produtos em movimento, com o propósito de providenciar níveis de serviços adequados aos clientes a um custo razoável (ANTT, 2004).

Todas as áreas operacionais de uma cadeia de suprimentos oferecem uma variedade de desafios para sua administração eficiente. O elo final da cadeia, o transporte a partir dos centros de distribuição até os consumidores (chamados de transporte local), usualmente é o mais caro da cadeia de distribuição (GALVÃO, 2001). Para que a empresa tenha êxito na realização desse transporte, ela deve planejar e executar as atividades de distribuição de uma maneira racional, levando em consideração todos os fatores econômicos envolvidos. Uma pequena discussão sobre a magnitude dos custos envolvidos, apresentada a seguir, destaca a importância desta etapa da distribuição.

Atualmente no Brasil, a maioria das empresas (em especial, as dos setores de soja, leite, refrigerantes, entre outros) utiliza o setor rodoviário para o transporte de seus produtos. A Tabela 1.1 apresenta a participação dos principais modais no Brasil e em outros países.

Tabela 1.1: Transporte de carga em alguns países - 2000.

País	Rodovia %	Hidrovia %	Ferrovias %
Alemanha	61,21	16,51	22,28
Bélgica	65,31	13,69	21,00
Brasil	64,56	22,72	12,72
Estados Unidos	32,41	20,37	47,22
França	72,44	3,33	24,23
Holanda	75,49	20,98	3,53
Inglaterra	66,60	25,67	7,73

Fonte: Ministério dos transportes / World Road Statistics.

Tabela 1.1: (continuação)

País	Rodovia %	Hidrovia %	Ferrovia %
Itália	88,95	0,70	10,35
Japão	50,25	44,77	4,98
Polônia	42,65	0,64	56,71

Fonte: Ministério dos transportes / World Road Statistics.

Apesar do relativo aumento do transporte de carga no setor ferroviário e hidroviário, observa-se que o Brasil, além de utilizar principalmente o modal rodoviário, o utiliza muito mais que, por exemplo, os Estados Unidos. Assim, verifica-se que o principal tipo de transporte utilizado pelos Estados Unidos é o ferroviário, cujo custo é bem menor que o rodoviário (PEREIRA, 2003). No Brasil, soma-se ao custo do modal rodoviário, a alta dos fretes e dos insumos. O frete (o que se paga ao transportador pelo transporte) aparece em várias etapas do processo, na forma de transporte (rodoviário, ferroviário, hidroviário, aéreo, e/ou dutoviário), deslocando insumos (matérias-primas, e componentes para manufatura), levando produtos acabados para os centros distribuidores (depósitos regionais e locais) e destes para os pontos de vendas, ou lojas de varejos e, em muitos casos, entregando mercadorias diretamente aos consumidores.

Segundo o Sindicato das Empresas de Transporte de Cargas de São Paulo e Região (SETCESP), o Índice Nacional da Variação do Custo do Transporte Rodoviário (INCTr), que mede a evolução de todos os custos de carga fracionada, para distâncias médias, foi de 18,14% entre outubro de 2002 e setembro de 2003 (veja Tabela 1.2).

Tabela 1.2: Índice Nacional da Variação do Custo do Transporte Rodoviário (INCTr).

Distâncias	Km	Custo R\$/Ton.	Variação 36 meses	Variação 24 meses	Variação 12 meses	Variação em 2003	Variação mensal
Muito Curtas	50	195,69	30,49%	23,51%	15,75%	14,60%	0,07%
Curtas	400	245,35	34,15%	25,88%	17,13%	13,66%	0,07%

Tabela 1.2: (continuação)

Distâncias	Km	Custo R\$/Ton.	Variação 36 meses	Variação 24 meses	Variação 12 meses	Variação em 2003	Variação mensal
Médias	800	302,11	36,97%	27,66%	18,14%	12,96%	0,18%
Longas	2400	529,14	42,47%	31,08%	20,06%	11,69%	0,38%
Muito Longas	6000	1039,96	42,27%	33,39%	21,31%	10,86%	0,51%

Fonte: (FIPE - USP) / ANTC, (setembro de 2003).

Contribuíram para este aumento nestes 12 meses, o aumento médio de 27,17% do óleo diesel. Outras variações significativas em insumos importantes foram: pneus (38,98%), veículos (33,72%), carroçeria (28,32%), câmara (29,56%) e recapagem (26,61%). O gasto relacionado ao transporte rodoviário tem, portanto, se tornado muito aparente devido ao aumento contínuo dos custos do combustível, alto custo do capital para substituição dos veículos, manutenção dos veículos, aumento do salário do trabalhador, entre outros fatores.

A avaliação dos custos de transporte apresentada tem como objetivo ressaltar que, de fato, as empresas podem ser beneficiadas se adotarem práticas operacionais eficientes em atividades de distribuição. Uma pequena porcentagem na economia (tal como 2% a 5%) destas despesas pode resultar em ganhos substanciais no decorrer dos anos, e permitir uma maior flexibilização dos preços finais praticados. Como decorrência, maior competitividade é alcançada.

A busca da minimização dos custos logísticos nas atividades de distribuição passou a ser um objeto de intensa pesquisa e de grande interesse pelas empresas. Como o Brasil apresenta-se primordialmente dependente do modal rodoviário para o transporte de suas cargas, ressalta-se ainda mais a necessidade de que as empresas distribuidoras optem por otimizar seus gastos neste setor.

Na área da Pesquisa Operacional, torna-se importante, portanto, a proposta de algoritmos eficientes, e sempre que possível, voltados para o atendimento da real necessidade das empresas. De fato, novas formulações e técnicas de resolução vêm sendo

propostas, enquanto, simultaneamente, identifica-se uma extensa lista de aplicações bem sucedidas, dentre as quais incluem-se o roteamento de ônibus urbanos, sistemas de coleta de lixo, entrega de periódicos, entre outros (BODIN, 1983).

No caso específico do Brasil, observa-se, embora de forma tímida, um interesse crescente das empresas pelo emprego de implementações de sistemas computacionais com o objetivo de racionalizar e reduzir os custos da distribuição de seus produtos. Dentre as empresas brasileiras destacam-se as indústrias de refrigerantes, mais precisamente a Companhia de Bebidas Ipiranga que implantou um software de roteirização (ROUTESHOW) para fazer a distribuição de suas bebidas (FERNANDES, 2001). Outros exemplos de implantação destes sistemas em empresas reportam economia de 5% em uma grande distribuidora de roupas, móveis e eletrodomésticos (BORGES, 2002), 5% em uma fábrica de panetone (Revista Distribuição, 2002), 20% em uma distribuidora de produtos químicos, 10% em uma indústria de cerveja e refrigerantes, entre vários outros (veja GHISI et al., 2004).

Neste sentido, o tema central da presente dissertação são os problemas de roteamento de veículos. Nesta classe de problemas, busca-se determinar rotas de baixo custo a serem operadas com uma frota de veículos que, partindo de uma ou mais garagens ou depósitos centrais, devem atender demandas de um conjunto de clientes. De forma geral, cada rota é atendida por um veículo cuja capacidade de carga não pode ser excedida.

Em especial, neste projeto é tratada uma subclasse dos problemas de roteamento, o chamado problema de roteamento de veículos periódico (PRVP). O PRVP é um problema multiperíodico, ou seja, as decisões de roteamento são tomadas em um horizonte de entregas de T dias. Especificamente, o objetivo do PRVP é elaborar um conjunto de rotas para cada dia t de um dado período de T dias. Cada cliente pode requisitar várias visitas para o veículo durante o período. Se um cliente requisitar k visitas durante o período, então estas visitas podem somente ocorrer em uma dada combinação permissível de visitas para o cliente. Por exemplo, se o cliente requisitar duas visitas durante o período (suponha um período de 5 dias na semana) e as combinações de visitas permissíveis para o cliente forem segunda-quinta, segunda-sexta ou terça-quinta, apenas uma destas combinações seria aceitável. O objetivo é minimizar a distância total percorrida sobre todo o

período.

Conforme GOLDEN & WASIL (1987), os problemas de roteamento de veículos periódico são naturalmente encontrados em empresas de distribuição de refrigerante, leite, gás industrial, coleta de lixo, entre outras. Uma das motivações para a escolha deste objeto é o fato de que apesar de sua evidente relevância prática, o PRVP tem sido relativamente pouco estudado, o que se reflete no reduzido número de trabalhos publicados nas últimas décadas.

O PRVP é um problema de otimização combinatória e assim como a grande maioria dos problemas de roteamento de veículos pertence à classe *NP-hard* (CORDEAU et al., 1997). Desta forma, a resolução ótima de exemplos a partir de um grande número de clientes torna-se inviável, o que demanda a utilização de métodos heurísticos. Os métodos heurísticos não garantem a otimalidade das soluções, porém permitem a obtenção de soluções subótimas de alta qualidade. Em particular, este trabalho aborda a utilização da meta-heurística tabu. A meta-heurística tabu é uma ferramenta heurística poderosa, caracterizada pelo uso de conceitos selecionados de inteligência artificial. Ela vem sendo aplicada com sucesso em uma série de problemas combinatórios, em particular e de forma marcante, em problemas de roteamento de veículos.

SEÇÃO 1.1

Objetivo

Esta pesquisa consiste em elaborar uma aplicação do algoritmo de busca tabu HTA de PUREZA & FRANÇA (2001), ao problema de roteamento de veículos periódico. Uma das motivações para o uso do algoritmo HTA, que apesar de sua inerente simplicidade, é que apresenta ganhos não desprezíveis quanto a qualidade das soluções.

Organização da dissertação

O texto da dissertação foi estruturado como segue:

- No capítulo 2 são apresentados os fundamentos básicos de busca tabu.
- No capítulo 3 são apresentados a definição e formulação matemática do PRV clássico, a complexidade computacional, as limitações dos métodos exatos, definição e formulação matemática do PRVP assim como as heurísticas existentes.
- No capítulo 4 é apresentado a aplicação da abordagem HTA ao PRV, constituindo o algoritmo HTA de PUREZA & FRANÇA, 2001.
- O capítulo 5 apresenta uma aplicação similar da abordagem HTA para a resolução do PRVP.
- O capítulo 6 apresenta os resultados computacionais.
- No capítulo 7 são apresentadas conclusões e perspectivas futuras deste trabalho.

2

A meta-heurística busca tabu

SEÇÃO 2.1

Introdução

Métodos heurísticos de melhoria, apesar de não garantirem a obtenção de uma solução ótima, fornecem soluções de boa qualidade em tempos computacionais razoáveis. Entretanto, o processo de busca é invariavelmente finalizado ao se atingir um ótimo local. Para permitir a continuidade das explorações para além dos ótimos locais, o processo de busca por soluções de boa qualidade pode ser guiado ou orientado pelas chamadas *meta-heurísticas*. Dentre estas, *simulated annealing* (KIRKPATRICK et al., 1983), algoritmos genéticos (HOLLAND, 1975), *GRASP* (FEO & REZENDE, 1995) e busca tabu (GLOVER & LAGUNA, 1997) são as mais amplamente empregadas.

Enquanto *simulated annealing* é baseada em um processo físico de recozimento, algoritmos genéticos procuram imitar o processo da reprodução evolutiva, *GRASP* realiza amostragens inteligentes do espaço de solução, a busca tabu pode ser considerada uma técnica que incorpora conceitos selecionados de inteligência artificial, cujo objetivo é o de imitar usos inteligentes de *memória*, promovendo a exploração do espaço de busca de forma estratégica e responsiva. As próximas seções apresentam, em maiores detalhes, os

fundamentos e os componentes básicos da metodologia de busca tabu, empregada neste projeto.

SEÇÃO 2.2

A busca tabu

Um problema genérico de otimização combinatória pode ser escrito através da fórmula $Min \{f(x) \mid x \in X\}$, onde X representa o conjunto finito de soluções factíveis do problema.

Os métodos heurísticos de busca local exploram o conjunto de soluções do problema, denotado por X , através da repetida realização de movimentos de uma solução x para outra solução x' , localizadas em uma amostragem V^* de uma vizinhança $N(x)$. A vizinhança $N(x)$ consiste de todas as soluções que podem ser obtidas pela realização de um movimento em x . Os movimentos são realizados até que seja atingida uma solução tão boa ou melhor que todas as soluções da amostragem, segundo a avaliação da função objetivo $f(x)$. Os passos de um método de busca local consistem de:

ENTRADA: Solução inicial x em X .

- Faça $x^* = x$ e $fim = falso$.
- Enquanto $fim = falso$, repita:
 - Gera-se uma amostragem V^* de soluções em $N(x^*)$.
 - Encontra-se o melhor x' em V^* , ou seja, tal que $f(x') \leq f(\bar{x}), \forall \bar{x} \in V^*$.
 - Se $f(x') < f(x^*)$ faça $x^* = x'$. Caso contrário, faça $fim = verdadeiro$.

SAIDA: Solução x^* .

Uma falha evidente dos métodos de busca local é que, na maioria dos casos, o ótimo local x^* obtido não é também um ótimo global, isto é, não minimiza a $f(x)$ sobre

todo $x \in X$. Este é o primeiro tipo de dificuldade que pode surgir com tais técnicas heurísticas e que provocam a necessidade de procedimentos que buscam encontrar melhores soluções pela continuidade da busca. Neste sentido, a busca tabu procura restringir o problema de otimização local fazendo uso sistemático da *memória* para explorar um conhecimento além do que está contido na função $f(x)$ e da vizinhança $N(x)$. A cada passo, a busca tabu usa um conhecimento heurístico para realizar a transição de uma solução para a próxima, guiando a busca em X .

A utilização de certas formas de memória flexível para controlar o processo de busca é, portanto, o tema central de busca tabu. A memória é classificada em dois tipos: *memória de curto prazo* e *memória de longo prazo*. Cada tipo de memória é acompanhado por suas próprias estratégias, mas o efeito de sua utilização pode ser visto como o da modificação da vizinhança $N(x)$ da solução corrente x . A vizinhança modificada é o resultado de se manter um histórico seletivo H dos estados encontrados durante a busca, provocando a alteração da vizinhança $N(x)$ para $N(H, x)$. Tais informações são, então, utilizadas para modificar a avaliação original da função objetivo (RAYWARD-SMITH et al., 1996) ao longo do processo de busca.

SEÇÃO 2.2.1

Memória de curto prazo

O principal objetivo da memória de curto prazo é evitar o aprisionamento da busca em ótimos locais. Isto pode ser realizado, proibindo-se que soluções já obtidas sejam novamente visitadas. Suponha, então, que as T soluções mais recentes sejam armazenadas de alguma forma. A proibição de movimentos que resultem em alguma destas soluções, pode ser utilizada para impedir que ciclos de comprimento menor ou igual a $|T|$ ocorram na trajetória da busca. Esta abordagem é a base da memória de curto prazo chamada de *memória de recentidade*.

Na prática, existe pouco interesse em se armazenar soluções completas, dados os excessivos requerimentos de memória e tempo computacional necessários. Ao invés

disso, são armazenadas características ou atributos de movimentos ou soluções recentemente visitadas, e a partir desses, proíbe-se a reversão de certos movimentos realizados no passado recente, isto é, tais movimentos tornam-se *tabu*, de forma a impedir que tais soluções sejam revisitadas.

Exemplos de atributos consistem de itens retirados e adicionados a um conjunto com capacidade restrita (Problema da Mochila) e tarefas movidas de posição em uma sequência de processamento (Problemas de Programação de Tarefas) uma vez que métodos de busca local para estes problemas muitas vezes geram novas soluções através de ações de movimentação, inclusão ou eliminação destas entidades.

Atributos que ocorrem em movimentos recentemente realizados são classificados como *tabu-ativos*. Movimentos que contém elementos tabu-ativos, são aqueles que podem vir a se tornar tabu. Entretanto, a determinação do *status* do movimento propriamente dito é ditado pela *regra de ativação tabu*. Isso significa que um movimento pode conter atributos tabu e não ser classificado como movimento tabu.

Um exemplo de regra de ativação para o problema da mochila, utilizando movimentos que trocam um item de dentro da mochila por outro de fora dela, consiste em classificar como tabu os movimentos nos quais o item incluído na mochila ou o item retirado da mesma são tabus. Se a regra prescrever como tabu os movimentos nos quais ambos os itens são tabus, tem-se uma regra mais restritiva.

Como nos problemas de roteamento, os métodos de busca local geralmente envolvem trocas de arestas, um exemplo natural de atributo do movimento consiste nas arestas eliminadas e adicionadas no processo de busca. Neste caso, uma possibilidade para regra de ativação é a fixação de um número máximo de arestas tabu tolerável, a partir do qual o movimento é considerado tabu. Note que neste caso, quanto menor o valor da tolerância, mais restritiva se torna a regra.

Caso um movimento proibido gere uma solução melhor do que qualquer uma obtida até o momento, o *status* tabu do movimento é desconsiderado e o movimento é aceito. Tal situação exemplifica a aplicação de *critérios de aspiração*, os quais prescrevem condições em que o *status* tabu de um movimento pode ser ignorado. Um outro exemplo de critério de aspiração é aquele que permite a realização de um movimento tabu (com

atributos tabu-ativos mais antigos) se todos os movimentos são tabus.

Cada regra possui portanto, um grau de restritividade associado, de maneira a permitir que um número maior ou menor de movimentos esteja disponível para escolha.

O processo de busca prossegue segundo esses princípios até que se atinja algum critério de parada. Os critérios de parada mais simples são:

- Uma solução ótima é encontrada.
- A vizinhança é vazia, ou seja, não há movimentos disponíveis.
- Foi atingido o número máximo de iterações.
- Foi atingido um número máximo de iterações sem melhorias.

A aplicação da memória de recentidade é implementada pela armazenagem de atributos em uma estrutura de dados T , tradicionalmente chamada de *lista tabu*, ainda que não se trate necessariamente de uma estrutura do tipo lista. Soluções ou movimentos tabu ativos estão sujeitos a receber o *status* tabu, e outras soluções além das já visitadas estarão também sujeitas à proibição. Não é interessante, entretanto, que o *status* tabu dos movimentos seja mantido indefinidamente, uma vez que isso resultaria na proibição total de movimentos em um prazo relativamente curto. Por essa razão, define-se um *período tabu*, ou seja, um número de iterações em que o *status* tabu de um dado movimento é mantido. Idealmente, o período tabu deve ter uma duração que garanta que a solução não seja revisitada.

O valor do período tabu pode variar para diferentes tipos ou combinações de atributos e para diferentes intervalos do tempo ou estágios da busca. A escolha dos tipos apropriados depende do contexto; entretanto, algumas orientações podem ser formuladas. Empiricamente, tem-se demonstrado que períodos tabu dependem do tamanho do problema. Entretanto, não se conhece nenhuma regra que determine um tamanho eficaz para todas as classes de problemas. Isto se explica parcialmente pela dependência do período tabu com o grau de restritividade da regra de ativação tabu. Em geral, regras mais restritivas são mais eficientes quando utilizadas com período tabu mais curto.

Períodos tabu curtos permitem a exploração de soluções ‘próximas’ a um ótimo local, enquanto períodos longos são preferíveis quando é necessário guiar a busca

2. A meta-heurística busca tabu

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 soluções.

Pode-se utilizar períodos tabu fixos ou dinâmicos, ou seja, onde os valores são variáveis. Dentre estes últimos é comum utilizar uma faixa definida por um limitante inferior e superior, e selecionar aleatoriamente, para cada atributo, um período tabu t dentro desta faixa, geralmente seguindo uma distribuição uniforme.

Em contraste aos métodos de busca local, os passos gerais de um algoritmo de busca tabu com memória de curto prazo para problemas de minimização consistem de:

ENTRADA: Solução inicial x em X .

- Faça $x^* = x$, $iteração_corrente = 0$, $condição_de_parada = falso$ e $T = \{\emptyset\}$.
- Enquanto $condição_de_parada = falso$, repita:
 - Faça $iteração_corrente = iteração_corrente + 1$.
 - Gere uma amostragem V^* de soluções em $N(x)$.
 - Selecione o melhor vizinho x' em V^* . Se o movimento s de geração de x' é não tabu, aceite o movimento. Caso contrário, se s é tabu e $f(x') < f(x^*)$ aceite s (critério de aspiração). Caso contrário, se todos os movimentos são tabu, libere os atributos, a partir dos mais antigos, até que um movimento s resultando em uma solução x' perca o *status* tabu e seja aceito (critério de aspiração).
 - Faça $x = x'$.
 - Adicione os atributos de s a T .
 - Se $f(x') < f(x^*)$, faça $x^* = x'$.

SAIDA: Solução x^* .

Memória de longo prazo

A utilização de memória de longo prazo e estratégias a ela associada confere ganhos, muitas vezes significativos, ao desempenho de implementações de busca tabu. Geralmente, a vizinhança modificada passa a incluir soluções de elite (ótimos locais de alta qualidade) encontrados em vários pontos do processo de busca. Um dos principais atrativos do uso de memória de mais longo prazo é que não se exigem experimentos preliminares demorados para que seus benefícios se tornem aparentes.

Assim como memórias de curto prazo estão associadas ao conceito de recentidade, memórias de longo prazo são geralmente baseadas em *freqüência*. A memória baseada em freqüência provê o tipo de informação que complementa a informação fornecida pela memória baseada em recentidade. 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.
- *Medida de transição*: número de vezes que um atributo é alterado.

Exemplos destas medidas são apresentadas na Tabela 2.1. Em medidas como estas, existem várias possibilidades para os denominadores:

- O número total de ocorrências de todos os eventos representados pelos numeradores.
- A soma dos denominadores.
- O valor máximo do numerador.
- O valor médio do numerador.

Tabela 2.1: Medidas de residência e de transição.

Problema	Medida de Residência	Medida de Transição
Sequenciamento de tarefas	Número de vezes que uma tarefa i ocupou a posição $\pi(i)$	Número de vezes que uma tarefa i trocou de posição com uma tarefa j
Mochila	Número de vezes que um item i fez parte da mochila	Número de vezes que um item i saiu da mochila para a entrada de outro item
Roteamento de veículos	Número de vezes que uma aresta i pertenceu a solução	Número de vezes que uma aresta i foi trocada com uma aresta j .

Estas razões resultam em *freqüências de transição* (que mantêm a informação de quantas vezes um atributo é alterado) e *freqüências de residência* (que armazenam a duração relativa da ocorrência dos atributos nas soluções geradas). Freqüências de residência altas em soluções de alta qualidade podem indicar que o atributo é muito atraente, enquanto que em soluções de baixa qualidade sugerem que o atributo é pouco atraente. Se o conjunto de soluções contém tanto soluções de alta como de baixa qualidade e a freqüência de residência é alta, então isso pode indicar um atributo ‘aprisionado’ e que necessita ser eliminado para permitir maior diversidade. Se, por outro lado, a freqüência de residência é baixa, então isso pode indicar um atributo excluído e que necessita ser incorporado para permitir também maior diversidade.

Uma freqüência de transição alta, por sua vez, pode indicar que o atributo associado entra e sai constantemente das soluções. Independentemente de qual freqüência será utilizada, o uso mais produtivo deste tipo de memória consiste em incorporá-la a uma estratégia de mais longo prazo.

SEÇÃO 2.2.2.1

Intensificação

Estratégias de intensificação são baseadas na modificação de regras de escolha com o objetivo de encorajar combinações de características de soluções historicamente boas. Elas também podem iniciar um retorno a regiões atraentes a fim de examiná-las em maior detalhe. Um exemplo de aplicação deste tipo de estratégia é o seguinte:

- Aplica-se memória de curto prazo ao processo de busca.
- Aplica-se uma estratégia de seleção de elite e elabora-se uma lista de candidatos.
- Enquanto $iteração_corrente \leq número_máximo_intensificação$ e a lista não estiver vazia, realiza-se:
 - Escolhe-se uma das soluções elite.
 - Prossegue-se com o processo de busca a partir da solução escolhida utilizando memória de curto prazo.
 - Adiciona-se novas soluções da lista de elite, se aplicável.

Duas variantes para a estratégia de seleção de soluções de elite têm apresentado excelente desempenho:

- A primeira procura assegurar que as soluções armazenadas diferem uma da outra por um grau especificado (medida de diversidade) e então apaga toda a memória de curto prazo antes de continuar a partir da melhor das soluções. Uma medida de diversidade pode estar associada ao número de movimentos necessários para transformar uma solução em outra.

- A outra variante mantém uma lista de tamanho limitado e adiciona uma nova solução se esta é melhor que qualquer outra já visitada. O último membro corrente da lista é sempre o escolhido (e removido) como uma base para a continuidade da busca. Entretanto, a memória 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 seja percorrido.

SEÇÃO 2.2.2.2

Diversificação

Na busca tabu, a inclusão de funções de memória de longo prazo é utilizada também para incorporar diversificação. Estratégias de diversificação, como o nome sugere, são elaboradas para dirigir a busca para novas regiões. Geralmente, elas são baseadas na modificação de regras de escolha de movimentos, ou seja, são geralmente alteradas para impedir ou penalizar a realização de certos movimentos, como parte de alguma estratégia adequada ao problema (veja LAGUNA, 1995).

- Modificação de regras de escolha:

Regras de escolha de movimentos são geralmente alteradas para impedir ou penalizar a realização de certos movimentos, como parte de alguma estratégia adequada ao problema. Por exemplo, em problemas de particionamento é comum utilizar movimentos de inserção ou troca para exploração da vizinhança local. Trocas implicam em intercambiar um elemento de dois conjuntos diferentes da partição. Inserções tomam um elemento de um conjunto e o colocam em um conjunto distinto. Como as trocas não modificam a cardinalidade da partição, 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, a seleção de trocas que não melhoram a solução obtida é proibida após um período de estabilização da busca. Isto é feito incluindo-se um valor de *penalidade* ao valor original do movimento:

$$V' = V + dP .$$

onde V é o valor do movimento, P é a penalidade e d é um parâmetro de diversificação ajustável. O valor da penalidade é geralmente uma função de medida de frequência. Quanto maior d , maior é a diversificação atingida.

- Re-inicialização:

Existem várias formas de utilizar informações de frequência para elaborar procedimentos de re-inicialização. Em problemas de sequenciamento de tarefas, por exemplo, a frequência total de tarefas que ocupam certas posições pode ser usada para construir novas soluções de partida. Problemas de sequenciamento de tarefas normalmente utilizam uma regra para determinar o índice de prioridade ou ordem de alocação das tarefas, o que resulta na construção de uma única construção de partida. Em implementações tabu, novas soluções de partida podem ser obtidas através da modificação do índice de prioridade de cada tarefa de acordo com a frequência total apresentada. O índice modificado pode ser representado por:

$$I' = I + dF .$$

onde I é o índice de prioridade, F é a medida de frequência e d é um parâmetro de diversificação ajustável. Este é um procedimento geral que pode ser incorporado em muitas aplicações, onde a informação de frequência é facilmente armazenada.

SEÇÃO 2.2.2.3

Integração de intensificação e diversificação

A alteração das listas tabu é um mecanismo que vem sendo pouco utilizado para promover a integração das estratégias de intensificação e diversificação. HÜBSHER & GLOVER (1992), propuseram uma forma elaborada de variação sistemática do período

tabu dinâmico e da configuração da lista, ao introduzir o conceito de *moving gaps* em problemas de programação de multiprocessadores. A lista tabu consiste de uma parte dinâmica e de uma parte estática; a configuração da parte dinâmica é alterada de tal maneira que uma fase de intensificação é seguida de uma fase de diversificação e vice-versa. Seis configurações de lista tabu foram utilizadas e a transição entre uma configuração e a próxima é ativada se nenhuma melhoria for encontrada dentro de um número especificado de iterações.

CHAKRAPANI & SKORIN-KAPOV (1993), aplicaram um conceito similar ao problema de designação quadrática com a variação dinâmica do tamanho da lista tabu através de oito configurações fixas. Quando uma fase sem melhorias é verificada, o número de atributos tabu-ativos é sistematicamente diminuído, definindo novas configurações de lista. A lista diminui até que esteja, de fato, inoperante. O objetivo desta lista é quebrar a ciclagem e produzir maior diversificação. No fim desta fase, a configuração inicial da lista é restabelecida, resultando em um processo de busca na qual a lista tabu cicla através de diferentes configurações.

SEÇÃO 2.3

Aplicações de busca tabu

Desde 1989, a busca tabu vem sendo aplicada com bastante sucesso num extenso número de áreas de conhecimento, o que justifica a continuidade nas pesquisas com vistas ao desenvolvimento de novas ferramentas e aplicações a problemas novos. Algumas destas aplicações bem sucedidas são apresentadas na Tabela 2.2.

Tabela 2.2: Aplicações de busca tabu.

Área	Descrição	Referência
<i>Programação</i>	Sequenciamento de tarefas Programação <i>job-shop</i> Programação de máquinas	WOODRUFF & SPERMAN (1993) WIDMER (1991) DELL'AMICO & TRUBIAN (1993) LAGUNA & GLOVER (1993) LAGUNA et al. (1993)
<i>Transporte</i>	Problema de roteamento de veículo Problema do caixeiro viajante	TAILLARD (1993) GENDREAU et al. (1994) PUREZA & FRANÇA (2001) MONTANÉ & GALVÃO (2004) GLOVER (1991) FIETCHER (1994)
<i>Telecomunicações</i>	Designação de caminhos Projeto de redes	OLIVEIRA & STROUD (1989) ANDERSON et al. (1993) LAGUNA (1994)
<i>Grafos</i>	Problema de clique máximo Agrupamento Coloração de grafos	GENDREAU et al. (1993) GLOVER et al. (1985) HANSEN et al. (1992) HERTZ & WERRA (1987) HERTZ et al. (1992)
<i>Logística e inteligência artificial</i>	Problema de satisfabilidade máxima Logística probabilística	HANSEN (1990) JAUMARD (1991)
<i>Otimização combinatória geral</i>	Problema da mochila 0-1 com múltiplas restrições Programação inteira mista 0-1	DAMMEYER & VOB (1993) LOKKETANGEN et al. (1994)

3

PRV, PRVP e técnicas de resolução

SEÇÃO 3.1

Introdução

O objetivo em problemas de roteamento de veículos é principalmente minimizar o custo variável da operação, por sua vez expresso como o tempo ou a distância total percorrida por todos os veículos. Se os clientes a serem servidos impõem restrições temporais (horário de atendimento), o problema passa a ser também de programação. Problemas de roteamento e programação podem ser classificados de acordo com diversas outras características, conforme mostrado na Tabela 3.1.

Portanto, existem inúmeras situações que caracterizam problemas de roteamento (e programação) de veículos, cada qual caracterizada por um conjunto de restrições e de objetivos. Dentre as mais comuns, incluem-se:

- *Capacidade máxima do veículo*: a soma das demandas dos clientes de qualquer rota não deve exceder a capacidade do veículo;
- *Número de clientes*: o número máximo de clientes por rota é limitado por um valor máximo;

3. PRV, PRVP e técnicas de resolução

- *Tempo ou distância máxima permitido para cada rota:* o tempo ou distância da rota depende do tempo de viagem entre clientes e tempo gasto em cada ponto da demanda, portanto, limita-se o tempo ou a distância de cada rota de forma a respeitar a jornada de trabalho diária do motorista;
- *Janelas de tempo dos clientes:* os clientes podem exigir que as visitas ocorram em períodos específicos do dia;
- *Janelas de tempo dos motoristas:* o tempo dispendido na viagem e execução da entrega deve excluir paradas para almoço e descanso programadas.

Tabela 3.1: Caracterização dos problemas de roteamento e programação.

Características	Alternativas Possíveis
Tamanho da frota	Um veículo / Diversos veículos
Tipo de frota disponível	Homogênea / Heterogênea
Número de depósitos	Único / Vários
Natureza da demanda	Determinística / Estocástica / Atendimento parcial é permitido
Rede associada	Orientada / Não orientada / Mista
Localização da demanda	Nos nós da rede / Nos arcos da rede / Em arcos e nós da rede
Capacidade dos veículos	Imposta e única / Imposta e múltipla / Não imposta
Tempo máximo de viagem	Imposto e único / Imposto e diversificado / não imposto
Tipo de operação	Coleta / Entrega / Entrega e coleta
Objetivo	Minimizar distância total / Minimizar tamanho da frota

Fonte: (BODIN et al., 1983).

Nos dois últimos casos, têm-se o problema de roteamento associado ao problema de programação de entregas. Ou seja, para cada veículo é preparado um roteiro e uma programação dos horários das atividades. O roteiro especifica a seqüência dos pontos de demandas que serão atendidos e a programação indica os instantes em que serão realizadas as atividades nesses locais.

Caso a frota consista de um único veículo não capacitado, sendo ignoradas as restrições temporais, obtém-se o problema do caixeiro viajante (PCV), um dos problemas mais estudados de otimização combinatória. O roteamento pode incluir entrega e/ou coleta de itens, ou nenhum dos dois (neste caso, tem-se demanda por serviços).

SEÇÃO 3.2

**O problema de roteamento de veículos clássico
(PRV)**

A classe mais simples de problemas de roteamento com múltiplos veículos é o chamado problema de roteamento de veículos (PRV) (BODIN et al., 1983). No PRV, admite-se a existência de um único depósito onde todas as rotas começam e terminam. Considera-se a frota homogênea, ou seja, todos os veículos são idênticos tanto em tipo como em capacidade. Admite-se que informações relativas aos clientes, isto é, suas posições geográficas e demandas, sejam conhecidas (problema determinístico). Cada veículo estabelece uma única rota e cada cliente tem sua demanda atendida por um único veículo, cuja capacidade total não pode ser excedida. Pode-se estabelecer intervalos de tempo em que os veículos devem atender os clientes, definindo-se assim a duração máxima permitida para cada rota (a jornada de trabalho diário do entregador). Podem também ser considerados tempos de serviço em cada ponto de demanda (geralmente, para coleta ou entrega dos produtos). O objetivo é minimizar a distância ou o tempo total percorrido sobre todo o roteamento.

O PRV pode ser formulado matematicamente por diversos modelos de programação inteira. A seguinte modelagem de programação 0-1 apresentada foi proposta por BODIN et al. (1983).

$$Min \left(\sum_{i,j} c_{i,j} \sum_k x_{i,j,k} \right), \quad (3.1)$$

sujeito às seguintes restrições:

$$\sum_k y_{i,k} = \begin{cases} 1 & i = 2 \dots n \\ m & i = 1 \end{cases} \quad (3.2)$$

$$\sum_i q_i y_{i,k} \leq Q_k \quad k = 1 \dots m \quad (3.3)$$

$$\sum_j x_{i,j,k} = \sum_j x_{j,i,k} = y_{i,k} \quad i = 1 \dots n, \quad k = 1 \dots m \quad (3.4)$$

$$\sum_{i,j \in S} x_{i,j,k} \leq |S| - 1 \quad \forall S \subseteq [2, n], \quad k = 1 \dots m \quad (3.5)$$

$$y_{i,k} \in [0, 1] \quad i = 1 \dots n, \quad k = 1 \dots m \quad (3.6)$$

$$x_{i,j,k} \in [0, 1] \quad i, j = 1 \dots n, \quad k = 1 \dots m \quad (3.7)$$

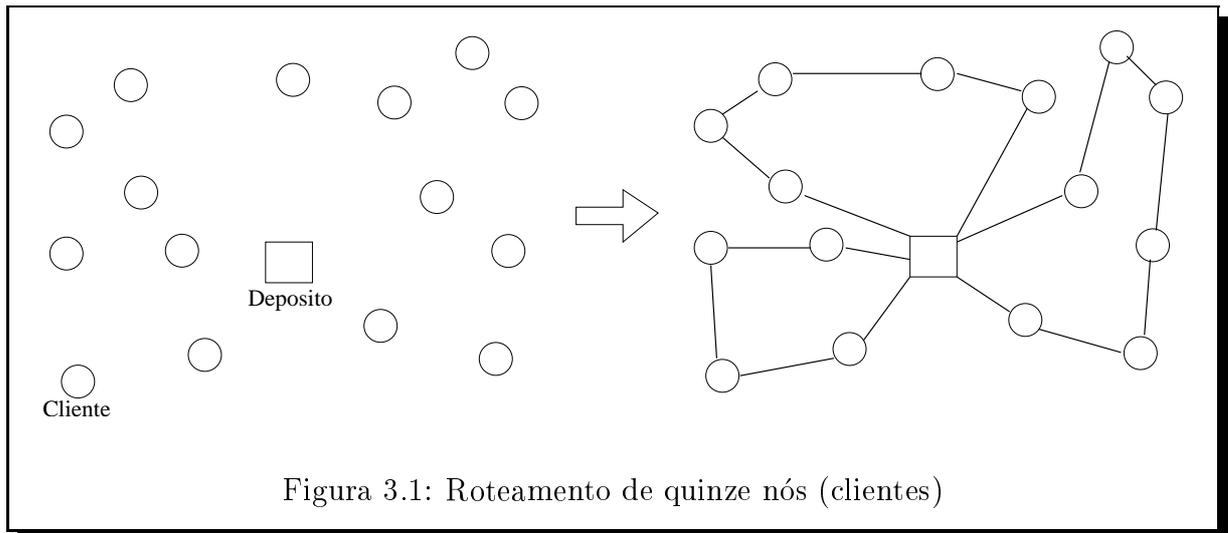
onde:

- n : número de clientes.
- m : número máximo de veículos.
- $c_{i,j}$: distância do cliente i ao cliente j .
- q_i : demanda do cliente i .
- Q_k : capacidade máxima do veículo k .
- $|S|$: cardinalidade de qualquer subconjunto de clientes.
- $x_{i,j,k} = 1$ se o veículo k visita o cliente j imediatamente após visitar o cliente i e $x_{i,j,k} = 0$ caso contrário.
- $y_{i,k} = 1$ se o cliente i é visitado pelo veículo k e $y_{i,k} = 0$ caso contrário.

O objetivo do PRV é dado pela equação (3.1), a função objetivo, que visa minimizar a distância total percorrida. As restrições (3.2) asseguram que cada cliente i seja servido por um único veículo e que o depósito deva ser visitado por todos os veículos, já que todos os veículos partem dele. A restrição (3.3) garante que a capacidade total do veículo não seja violada. A restrição (3.4) assegura que um dado veículo que visita o

cliente i deve partir de i . A restrição (3.5) impede a formação de sub-rotas. As restrições (3.6) e (3.7) impõem valores binários às variáveis.

Com as equações (3.1) a (3.7) minimiza-se e garante-se as condições necessárias e suficientes para obtenção de uma solução factível e ótima para o PRV. A Figura 3.1 ilustra um exemplo de roteamento de um conjunto de quinze nós (clientes).



SEÇÃO 3.3

Limitação dos métodos exatos

Segundo GAREY & JOHNSON (1979), um algoritmo polinomialmente limitado para um problema é um procedimento cuja carga computacional cresce apenas polinomialmente em função de uma medida do tamanho do problema. A classe de problemas para os quais são conhecidos algoritmos polinomialmente limitados é chamada de P, os quais podem ser resolvidos a otimalidade de forma eficiente. Por outro lado, problemas para os quais não se conhecem algoritmos polinomiais que os resolvam são classificados como NP (problemas polinomiais não determinísticos). Nestes casos, a complexidade do melhor algoritmo conhecido cresce exponencialmente (ou pelo menos superpolinomialmente com o tamanho do problema). Por esta razão, tais problemas são considerados

complexos e de difícil tratamento. Em especial, a maioria dos problemas de roteamento e programação de veículos pertence a classe NP.

Um número substancial de métodos exatos foi elaborado com vistas à resolução de problemas de roteamento. Levantamentos detalhados a este respeito podem ser encontrados em CHRISTOFIDES et al. (1979), WATSON-GANDY & FOULDS (1981), BODIN et al. (1983), CHRISTOFIDES (1985), GOLDEN & ASSAD (1986), BOTT & BALLOU (1986), LAPORTE & NORBERT (1987), GOLDEN & WATTS (1988), BODIN (1990), LAPORTE (1992). Dentre as técnicas exatas para resolução deste tipo de problema, podemos citar *branch-and-bound*. Os algoritmos exatos apesar da qualidade da solução apresentada, não conseguem, entretanto, tratar PRV de interesse prático devido à evolução exponencial do tempo computacional requerido em relação ao aumento do tamanho do problema (número de nós), uma vez que se trata de um problema NP. Em outras palavras, tais métodos são aplicáveis apenas para problemas pequenos.

Embora o desenvolvimento de técnicas exatas para resolver o problema de roteamento de veículos tenha mostrado um avanço considerável, verifica-se que as *técnicas heurísticas* continuam a representar um papel fundamental nas aplicações práticas, especialmente em problemas de grande porte. Heurísticas são técnicas de resolução que não garantem a otimalidade da solução, porém permitem a obtenção de soluções de alta qualidade em um tempo computacional tolerável. Segundo PIDD (1996), existem duas razões para a predominância dos métodos heurísticos sobre os exatos:

1. permitem o tratamento de problemas computacionalmente complexos, apresentando um retorno prático, e
2. permitem que modelos mentais aplicados na prática possam ser implementados computacionalmente.

Tendo em vista a limitação dos métodos exatos em aplicações práticas, destacam-se alguns dos procedimentos heurísticos 'mais conhecidos' para a resolução do PRV: heurística de construção de CLARKE & WRIGHT (1964), heurística de melhoria de LIN & KERNINGHAN (1973), heurística de duas fases de GILLET & MILLER (1974), busca em árvore incompleta de CHRISTOFIDES et al.(1979). Outras heurísticas podem

serem encontradas em GOLDEN & ASSAD (1988), LAPORTE (1992), LAPORTE & OSMAN (1995), FISHER (1995), GENDREAU et al. (1997) e outros.

SEÇÃO 3.4

O problema de roteamento de veículos periódico (PRVP)

Em algumas aplicações, tais como coleta de lixo, distribuição de combustível e distribuição de refrigerantes, o problema de roteamento de veículos possui uma característica que o diferencia do problema clássico. Nestas situações, os clientes não são geralmente atendidos em uma base diária, mas com uma dada frequência em um horizonte ou período de planejamento determinado. Neste período, os clientes não só especificam quantas vezes devem ser visitados, mas em geral estipulam as possíveis combinações de dias de visitas. Estes problemas são chamados de *problemas de roteamento de veículos periódico* (PRVP). O problema de roteamento de veículos periódico é, portanto, uma generalização do clássico problema de roteamento de veículos onde o período de planejamento é estendido de um único dia para T dias.

Sejam n clientes requisitando visitas em um período de T dias. Dada a demanda diária de cada cliente e suas possíveis combinações de visitas, o problema consiste em projetar um conjunto de rotas para cada dia no período de planejamento (considerando a capacidade máxima do veículo, tempo máximo da rota, um único depósito, ou seja, mantendo as demais características do PRV) tal que todos os clientes sejam servidos pelo número requerido de vezes e que a distância total viajada por toda a frota no período seja minimizada. Além das distâncias, os dados de entrada no PRVP geralmente consistem do número de visitas F_i (frequência) no horizonte de T dias que cada cliente i requer, e as possíveis combinações de dias para estas visitas, denotada por C_i . Por exemplo, supondo um período de $T = 5$ dias e que o cliente i tenha $F_i = 2$ e $C_i = \{(1, 3), (2, 4), (3, 5)\}$. Então, este cliente deve ser visitado 2 vezes no período de 5 dias e estas visitas devem

acontecer nos dias 1 e 3, ou nos dias 2 e 4 ou nos dias 3 e 5. O problema consiste em selecionar a combinação de dias de visitas para cada cliente e a ordem com que cada visita em cada dia deve ser efetuada, de forma a minimizar os custos totais do roteamento no horizonte de planejamento.

A Tabela 3.2 apresenta um exemplo de frequências e combinações de dias de visitas para 8 clientes. Nota-se que o cliente a pode ser atendido no dia 1 ou no dia 2 (frequência = 1), enquanto o cliente f precisa ser atendido no dia 1 e no dia 2 (frequência = 2).

Tabela 3.2: Especificação dos dias de visitas para os clientes ($T = 2$).

Clientes	Frequência (F_i)	Combinações de visitas (C_i)
a	1	(dia 1) ou (dia 2)
b	1	(dia 1) ou (dia 2)
c	1	(dia 1) ou (dia 2)
d	1	(dia 1) ou (dia 2)
e	2	(dia 1) e (dia 2)
f	2	(dia 1) e (dia 2)
g	2	(dia 1) e (dia 2)
h	2	(dia 1) e (dia 2)

A Figura 3.2 ilustra uma possível solução para o problema da Tabela 3.2 (considerando restrições de capacidade e tempo máximo de rota) no horizonte de planejamento de 2 dias. Nota-se que os clientes a e d são atendidos no dia 1, assim como os clientes e , f , g e h . Para estes últimos, o atendimento no dia 1 é obrigatório devido às especificações dos dias de visita. No dia 2, os clientes c , b , e , f , g e h são visitados.

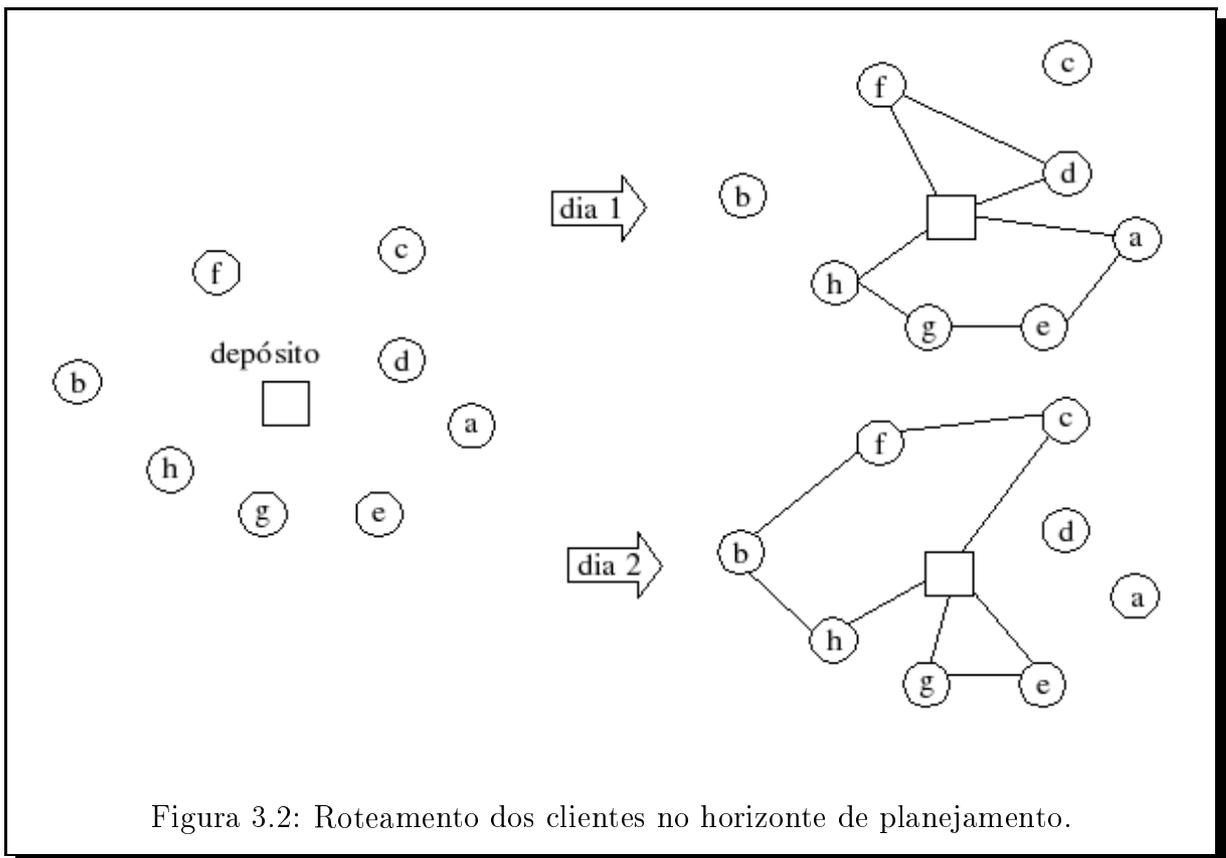


Figura 3.2: Roteamento dos clientes no horizonte de planejamento.

Nestes problemas, caso a variável T seja substituída por D (referentes a depósitos), estaria-se trabalhando com um Problema de Roteamento de Multi Depósitos (monoperiódico), veja CORDEAU et al (1997), para maiores detalhes.

SEÇÃO 3.5

Descrição do problema

Para descrever formalmente o problema de roteamento de veículos periódico, são consideradas as seguintes premissas:

1. um conjunto de clientes, cada um com uma dada demanda a cada visita, especifica o número de vezes que devem ser atendidos em um período de planejamento, considerando um conjunto de possíveis combinações de visitas (somente uma combinação de visita possível é selecionada para cada cliente).

2. em cada dia do período as rotas devem começar e terminar no único depósito.
3. há um número de veículos (homogêneos) disponíveis a cada dia do período.
4. a capacidade máxima de cada veículo disponível da frota não deve ser ultrapassada.
5. o tempo máximo de cada rota deve ser respeitado.
6. podem existir tempos de serviço em cada cliente i .

O objetivo é minimizar o tempo (ou distância) total de viagem percorrido por todos os veículos no período. CHRISTOFIDES & BEASLEY (1984) apresentam a seguinte formulação para o problema de roteamento de veículos periódico:

n : número total de clientes.

N : conjunto de clientes ($i | i = 1, 2, \dots, n$).

T : número de dias no período de planejamento.

C_i : conjunto de combinações permitidas para o cliente i .

q_i : demanda do cliente i em cada visita ($i = 1, 2, \dots, n$).

$c_{i,j}$: tempo (ou distância) de viagem do cliente i até o cliente j (i e $j = 1, 2, \dots, n$).

R_t : conjunto de veículos disponíveis no dia t ($t = 1, 2, \dots, T$).

Q_r : capacidade do veículo r .

D_r : tempo de viagem total do veículo r .

$V_{i,t} = 1$ se o cliente i é visitado no dia t ; 0 caso contrário ($i = 1, 2, \dots, n$; $t = 1, 2, \dots, T$).

O depósito é representado pelo nó 0 e $V_{0,t} = 1$ ($t = 1, 2, \dots, T$).

$a_{k,t} = 1$ se o dia t pertence à combinação k , 0 caso contrário.

$x_{i,k} = 1$ se a k -ésima combinação é escolhida para o cliente i , caso contrário $x_{i,k} = 0$.

$u_{i,j,t,r} = 1$ se o veículo $r \in R_t$ vai de i para j no dia t ; 0 caso contrário.

O problema de roteamento de veículos periódico pode ser definido como:

$$\text{Min} \left(\sum_{t=1}^T \sum_{i=0}^n \sum_{j=0}^n \sum_{r \in R_t} c_{i,j} u_{i,j,t,r} \right), \quad (3.8)$$

sujeito a:

$$\sum_{k \in C_i} x_{i,k} = 1, \quad \forall i (i \neq 0), \quad (3.9)$$

$$V_{i,t} = \sum_{k \in C_i} x_{i,k} a_{k,t}, \quad \forall t, \forall i (i \neq 0), \quad (3.10)$$

$$\sum_{r \in R_t} u_{i,j,t,r} \leq \frac{V_{i,t} + V_{j,t}}{2}, \quad \forall i, j, t (i \neq j), \quad (3.11)$$

$$\sum_{i=0}^n u_{i,p,t,r} = \sum_{j=0}^n u_{p,j,t,r}, \quad \forall p, t, r \in R_t, \quad (3.12)$$

$$\sum_{r \in R_t} \sum_{i=0}^n u_{i,j,t,r} = \begin{cases} V_{j,t} & \forall j, t (j \neq 0) \\ |R_t| & \forall t (j = 0) \end{cases}, \quad (3.13)$$

$$\sum_{i \in W} \sum_{j \in W} u_{i,j,t,r} \leq |W| - 1, \quad \forall t, r \in R_t, \forall W \subseteq N, \quad (3.14)$$

$$\sum_{j=1}^n u_{0,j,t,r} \leq 1, \quad \forall t, r \in R_t, \quad (3.15)$$

$$\sum_{i=1}^n q_i \left(\sum_{j=0}^n u_{i,j,t,r} \right) \leq Q_r, \quad \forall t, r \in R_t, \quad (3.16)$$

$$\sum_{i=0}^n \sum_{j=0}^n c_{i,j} u_{i,j,t,r} \leq D_r, \quad \forall t, r \in R_t, \quad (3.17)$$

$$x_{i,k} \in (0, 1), \quad \forall i, k \in C_i, \quad (3.18)$$

$$u_{i,j,t,r} \in (0, 1), \quad \forall i, j, t, r \in R_t. \quad (3.19)$$

A expressão (3.8) (função objetivo) visa a minimização da distância total percorrida por todos os veículos no período de planejamento. A restrição (3.9) garante que somente uma combinação de visita é escolhida para cada cliente. A restrição (3.10) impõe que um cliente seja visitado no dia t somente se a combinação da visita escolhida para este cliente pertencer ao dia t . A restrição (3.11) assegura que nenhum veículo atenda a dois clientes em seqüência em um dia t , a menos que ambos estejam programados para serem visitados neste dia. A restrição (3.12) garante que se um veículo visita um certo cliente, o veículo deve partir deste cliente. As restrições (3.13) asseguram que cada cliente seja visitado no dia em que foi programada a visita. A restrição (3.14) impede a formação de subrotas. A restrição (3.15) impõe que um veículo disponível no dia t , pode ser utilizado no

máximo uma vez. A restrição (3.16) assegura a não violação da capacidade dos veículos. A restrição (3.17) limita o tempo de viagem dos veículos. E finalmente, as restrições (3.18) e (3.19) restringe o domínio das variáveis de decisão a valores binários.

O modelo apresentado por CHRISTOFIDES & BEASLEY (1984) é uma formulação exata para o PRVP cuja aplicação limita-se a problemas trivialmente pequenos, uma vez que é um problema inteiro e envolve um enorme número de variáveis ($u_{i,j,t,r}$ em particular) e restrições. Como se trata de uma generalização do PRV, considera-se o PRVP tão difícil quanto o primeiro, sendo classificado como um problema de otimização combinatória pertencente a classe NP-*hard*. Por esta razão, a maioria dos trabalhos reportados sobre o PRVP utilizam métodos heurísticos para sua resolução, como será visto a seguir.

SEÇÃO 3.6

**Heurísticas para o problema de roteamento de
veículos periódico**

A primeira aplicação reportada para o PRVP é devida a BELTRAMI & BODIN (1974) que abordam o problema prático de coleta de lixo. Os autores propõem duas estratégias possíveis para sua resolução:

1. as rotas são desenvolvidas e então atribuídas aos dias da semana.
2. os clientes são aleatoriamente atribuídos a dias da semana e as rotas então desenvolvidas.

Suas análises são limitadas a freqüência de visitas igual a 3 ou 6 vezes por semana. Algoritmos mais elaborados foram posteriormente desenvolvidos por RUSSELL & IGO (1979), CHRISTOFIDES & BEASLEY (1984), TAN & BEASLEY (1984), RUSSELL & GRIBBIN (1991), GAUDIOSO & PALLETA (1992), CHAO et al. (1994), CHAO et al. (1995) e CORDEAU et al. (1997).

SEÇÃO 3.6.1

Heurística de RUSSELL & IGO (1979)

Os autores desenvolveram três algoritmos, também aplicados ao problema de coleta de lixo. Inicialmente, os clientes são atribuídos aos dias da semana e posteriormente roteados a cada dia separadamente. As heurísticas envolvem os seguintes passos:

- Inicialmente, seleciona-se para atribuição os clientes que requerem visitas em todos os cinco dias da semana ($T = 5$) ou em dias específicos, ou seja, o cliente 1 requer uma visita no dia 2, o cliente 2 requer uma visita no dia 5, e assim por diante. Esses clientes serão chamados de núcleos e servem de base para a alocação dos clientes não atribuídos.

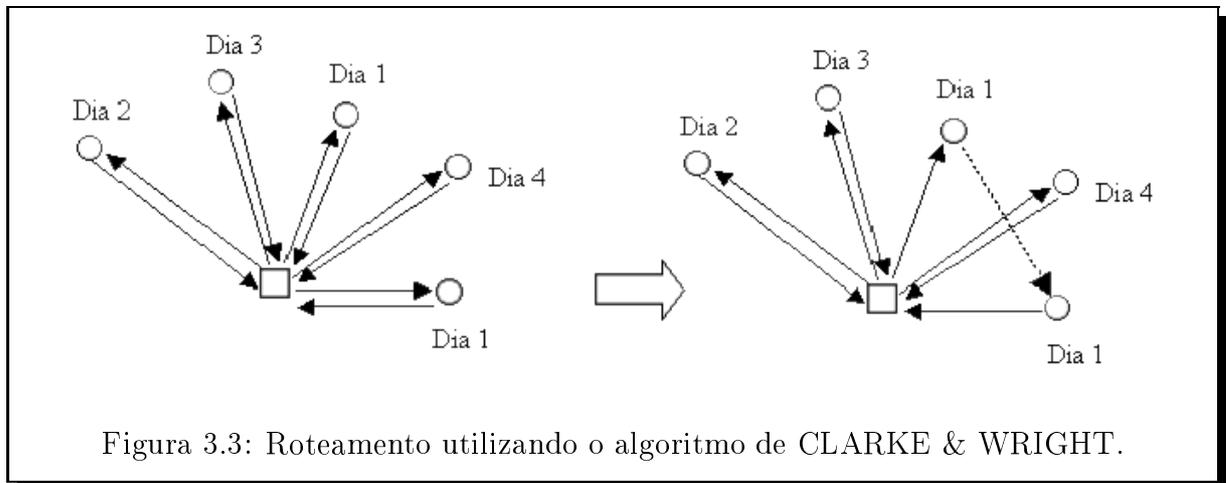
Os núcleos representam os dias no período de planejamento. Os clientes restantes são então atribuídos seqüencialmente em ordem de frequência de serviço (F_i), e sua atribuição deve respeitar todas as demais restrições do problema.

- **Heurística 1:** Suponha que um cliente i (ainda não atribuído) possua frequência de visita igual a 2 ($F_i = 2$), sendo o espaçamento entre uma visita e outra igual a 1. Logo as possíveis combinações de visitas para este cliente seria $C_i = \{(\text{dia 1 e dia 3}) \text{ ou } (\text{dia 2 e dia 4}) \text{ ou } (\text{dia 3 e dia 5})\}$. O cliente é, então, atribuído a uma determinada combinação, respeitando as seguintes regras:

1. se a distância média do cliente aos núcleos relacionados a tal combinação for 10% menor do que a distância média do cliente aos núcleos relacionados às outras combinações, ou
2. caso contrário, as duas combinações de dias tendo as menores médias são comparadas, baseadas em sua variância. Será escolhida a de menor variância, desde que seja menor ou igual a 40% da variância da outra combinação .
3. caso o critério acima não seja atendido, a combinação escolhida é aquela com a menor média da distância mais curta do cliente até os núcleos dos dias da combinação. O processo é repetido até que todos os clientes sejam atribuídos

aos dias. Aplica-se então um procedimento no qual os clientes podem ser redesignados a outras combinações e roteados de forma a gerar soluções factíveis.

- **Heurística 2:** Aplica-se à solução inicial, gerada pela primeira heurística, uma modificação do algoritmo de melhoria de rotas MTOUR desenvolvida por RUSSELL (1977) para o PRV. O algoritmo MTOUR é uma generalização do bem sucedido algoritmo de troca de arestas de LIN & KERNINGHAN (1973) para o PCV. A fim de tratar o PRVP, MTOUR foi modificado para considerar restrições de espaçamento entre dias de visita, capacidade dos veículos e tempos máximos de viagem.
- **Heurística 3:** A partir da atribuição de clientes a dias da semana realizada pela primeira heurística aplica-se o algoritmo de roteamento de CLARKE & WRIGHT (1964). O algoritmo admite inicialmente que todos os nós são conectados diretamente ao depósito e procura juntar rotas de um mesmo dia segundo o critério de maior economia, conforme a Figura 3.3.



A formulação difere da proposta de CHRISTOFIDES & BEASLEY (1984) no sentido de que não é especificado um conjunto de combinações de visitas (C_i) para cada cliente, mas sim, definido um número de entregas no período (F_i). O espaçamento em dias requisitado entre entregas sucessivas ao mesmo cliente é restrito a ser maior que um número mínimo de dias.

SEÇÃO 3.6.2

Heurísticas de CHRISTOFIDES & BEASLEY

(1984)

Os autores desenvolveram dois algoritmos heurísticos para a resolução do PRVP. Ambas as heurísticas baseiam-se na escolha inicial de dias de visitas para os clientes, seguidas de um procedimento de melhoria de rotas, no qual trocam-se clientes de um dia para o outro (inter-troca), desde que seja viável esta mudança. Uma vez que os clientes tenham sido atribuídos aos dias, um algoritmo de roteamento é aplicado a cada dia. Eles relaxam o PRV de cada dia do período, substituindo-o por (i) um problema da mediana (PM), ou (ii) um problema do caixeiro viajante (PCV), ambas simples de serem resolvidos. Desta forma, passa-se a ter (1) um problema da mediana periódica (PMP) e (2) um problema do caixeiro viajante periódico (PCVP).

Maiores detalhes sobre esta heurística, veja capítulo 5 seção 5.2.1 (escolha inicial de dias de visitas para os clientes) e seção 5.3.1 (procedimento de melhoria de rotas), uma vez que foram utilizados no desenvolvimento dos algoritmos propostos neste trabalho.

SEÇÃO 3.6.3

Heurística de TAN & BEASLEY (1985)

Neste trabalho, foi desenvolvida uma heurística baseada no algoritmo de roteamento de veículos monoperiódico de FISHER & JAIKUMAR (1981). Os autores definem uma matriz $(D_{i,k})$, gerando-se primeiramente k pontos sementes (um para cada veículo). $D_{i,k}$ é a distância extra viajada, quando o cliente i é inserido na rota servida pelo veículo k , partindo do depósito para o ponto semente e retornando novamente ao depósito. Posteriormente, a seqüência de visitas para cada veículo pode ser determinada resolvendo o PCV para os clientes atribuídos aos veículos.

Utilizando uma idéia similar, TAN & BEASLEY (1985), ao invés de atribuir os clientes aos veículos, atribuem clientes a centros (pontos sementes). Esses centros são clientes escolhidos e associados aos dias do período. Para associar estes centros aos dias foi desenvolvido um modelo de programação matemática. Inicialmente procede-se da seguinte forma:

Aplique o método de geração de semente de FISHER & JAIKUMAR (1981) consistindo de KT pontos sementes, onde K é o conjunto de veículos disponíveis a cada dia e T é o número total de dias. Cada veículo possui uma capacidade Q^* dado por:

$$Q^* = \frac{\sum_{k=1}^K Q_k}{K} .$$

Considere P o conjunto contendo os KT pontos sementes. Para associar um dia t para cada ponto semente $p(\in P)$ utiliza-se a matriz $V_{p,t}$. Esta matriz é formada como segue:

Seja B o conjunto de clientes tal que:

- (i) a demanda total dos clientes em B , não exceda a capacidade dos veículos.
- (ii) os clientes em B sejam os clientes mais próximos do ponto p .
- (iii) o número de clientes em B seja máximo.

Então define-se a matriz $V_{p,t}$ como:

$$V_{p,t} = \sum_{i \in B} \left[\frac{\sum_{r \in C_i} a_{r,t}}{|C_i|} \right] .$$

onde

C_i : conjunto de combinações de visitas permitidas para o cliente i .

$|C_i|$: é o número total de combinações do cliente i .

$a_{r,t} = 1$ se a combinação r envolve a visita no dia t ; 0 caso contrário.

Quanto maior é o valor de $V_{p,t}$ mais atraente é associar o dia t com a semente p . Considere $Y_{p,t} = 1$ se o ponto semente $p \in P$ for associado ao dia t ; 0 caso contrário. Então, o problema que associa os dias aos pontos sementes é dado por:

$$\text{Max} \left(\sum_{p \in P} \sum_{t=1}^T V_{p,t} Y_{p,t} \right) , \tag{3.20}$$

sujeito a:

$$\sum_{t=1}^T Y_{p,t} = 1, \quad \forall p \in P, \quad (3.21)$$

$$\sum_{p \in P} Y_{p,t} = K, \quad t = 1 \dots T, \quad (3.22)$$

$$Y_{p,t} \in (0, 1), \quad \forall p \in P, t = 1 \dots T. \quad (3.23)$$

A restrição (3.21) assegura que cada ponto semente está atribuído a exatamente um dia e a restrição (3.22) impõe que k pontos sementes estão atribuídos a cada dia. Note que esta abordagem atribui pontos sementes a dias, sendo baseada em uma abordagem similar a CHRISTOFIDES & BEASLEY (1984). Em seguida, os clientes são atribuídos aos centros através do seguinte modelo de programação matemática. Seja:

$a_{r,t} = 1$ se a combinação $r = 1, \dots, R$ envolve a visita no dia $t = 1, \dots, T$; 0 caso contrário.

$x_{i,r} = 1$ se o cliente i está atribuído à combinação de visita $r \in C_i$; 0 caso contrário.

$D_{i,t}$ = medida (deduzida de alguma maneira) da contribuição de distância do cliente i a alguma rota envolvendo o cliente i no dia t ($t = 1, \dots, T$).

C_i : conjunto de combinações de entrega permitidas para o cliente i , $C_i \subseteq [1, 2, \dots, R]$ ($i = 1, \dots, n$).

q_i : demanda do cliente i ($i = 1, \dots, n$) em cada entrega.

Q_k : capacidade do veículo k ($k = 1, \dots, K$).

O PRVP pode ser formulado como:

$$\text{Min} \left(\sum_{i=1}^n \sum_{t=1}^T \sum_{r \in C_i} D_{i,t} a_{r,t} x_{i,r} \right), \quad (3.24)$$

sujeito a:

$$\sum_{r \in C_i} x_{i,r} = 1, \quad i = 1 \dots n, \quad (3.25)$$

$$\sum_{i=1}^n \sum_{r \in C_i} q_i a_{r,t} x_{i,r} \leq \sum_{k=1}^K Q_k, \quad t = 1 \dots T, \quad (3.26)$$

$$x_{i,r} \in (0, 1), \quad \forall r \in C_i, i = 1 \dots n. \quad (3.27)$$

A expressão (3.24) representa a distância total viajada, a restrição (3.25) assegura que uma combinação de visita aceitável seja escolhida para cada cliente, a restrição (3.26) assegura que a entrega total em algum dia não exceda a capacidade total do veículo e (3.27) é a restrição de integralidade. A heurística consiste nos seguintes passos:

- **Passo 1:** Gera-se KT pontos sementes (centros), usando o método de geração de sementes de FISHER & JAIKUMAR (1981) para o problema monoperiódico.
- **Passo 2:** Associa-se um dia t a cada centro.
- **Passo 3:** Atribui-se os clientes aos centros que representam dias no período de planejamento.
- **Passo 4:** Utiliza-se o algoritmo de FISHER & JAIKUMAR (1981) para fazer o roteamento para cada dia.
- **Passo 5:** Como a escolha dos pontos semente (**passo 1**) é feita de forma arbitrária, o processo é repetido com as sementes ajustadas para lidar com as combinações de entrega escolhidas (maiores detalhes, ver TAN & BEASLEY, 1985).

SEÇÃO 3.6.4

Heurística de RUSSELL & GRIBBIN (1991)

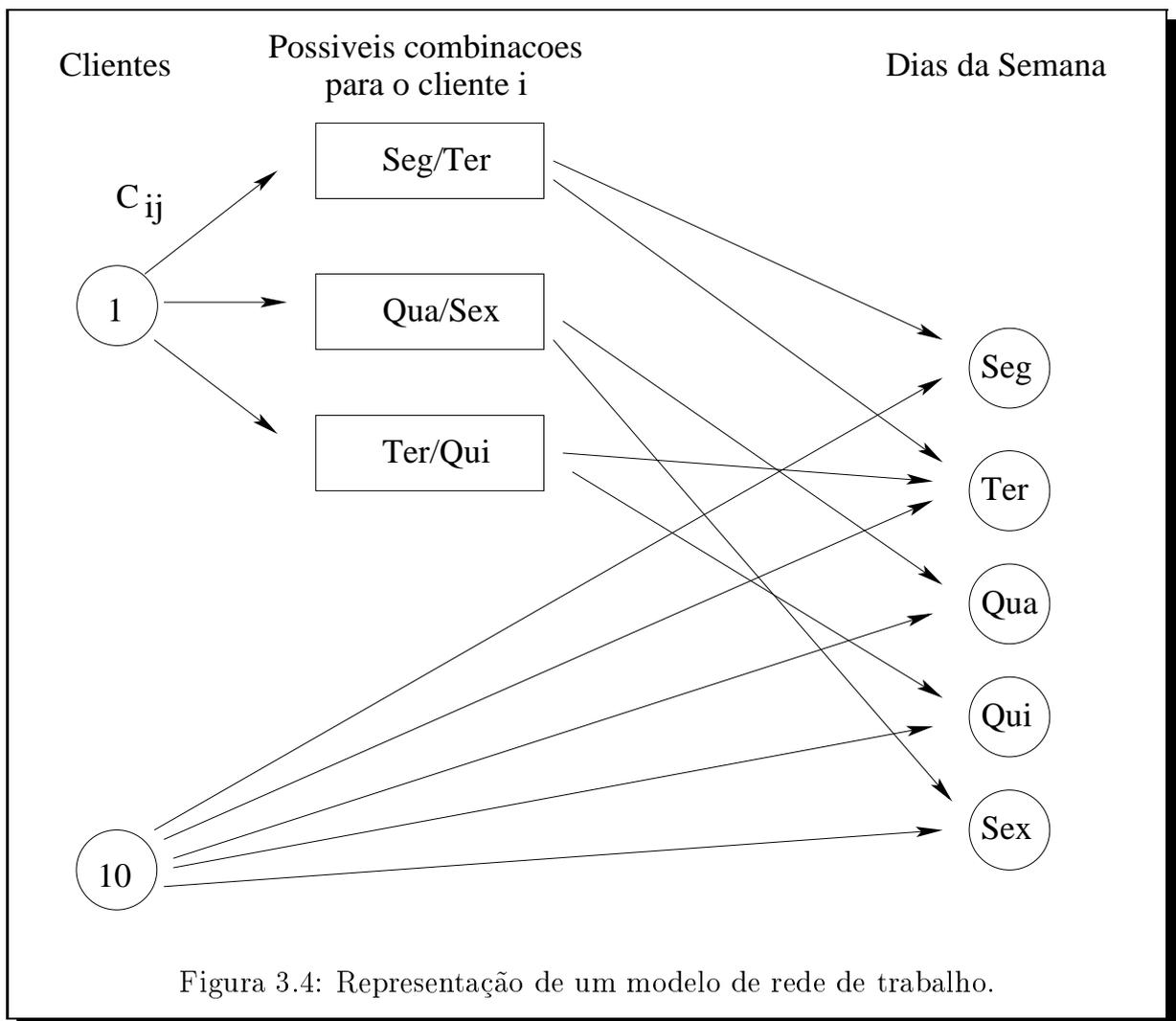
Os autores propõem um método heurístico de 4 fases para resolver o PRVP. Na primeira fase, uma estrutura de rede é gerada e utilizada para atribuir clientes a pontos sementes que representam dias do período de planejamento.

A rede gerada consiste em um grafo direcionado com um conjunto N de nós e um conjunto A de arcos. Pares ordenados de nós são representados como (i, j) , onde i denota o início do nó de onde o arco parte e j denota o nó onde o arco incide. Para cada arco existe um custo por unidade de fluxo $C_{i,j}$.

Na Figura 3.4, tem-se uma rede composta de 10 clientes, os quais estão atribuídos a suas combinações de dias de visitas, por sua vez atribuídas aos dias de visitas

na semana. Um modelo matemático para este problema de rede foi formulado, e soluções foram obtidas utilizando o software GNO (Generalized Network Optimizer).

Uma vez que os clientes tenham sido atribuídos a dias, a segunda fase consiste na aplicação de uma heurística de melhoramento na qual trocam-se clientes de um dia para o outro. Foi empregado o procedimento inter-troca de CHRISTOFIDES & BEASLEY (1984) tomando-se a relaxação PCVP.



Na terceira fase é aplicada uma heurística de melhoramento similar a da segunda fase, só que considera-se o roteamento de cada dia (sujeito a restrições como capacidade dos veículos, e outras) ao invés da aplicação do PCV.

De forma a melhorar os resultados obtidos pelos passos anteriores, na quarta

fase foi desenvolvido um modelo matemático inteiro 0-1, no qual considera-se apenas as trocas de combinações de dias de visita para os clientes i que não foram consideradas nas fases anteriores.

SEÇÃO 3.6.5

Heurística de CHAO, GOLDEN & WASIL (1995)

Todos os procedimentos propostos anteriormente procuram resolver o PRVP utilizando dois estágios principais: inicialmente, gera-se uma solução inicial e a seguir aplica-se um ou mais procedimentos para melhoria da solução inicial. Uma limitação básica destes métodos é que o procedimento de melhoria pode não ser capaz de evitar a geração de ótimos locais de qualidade inferior. A idéia da heurística proposta de CHAO et al. (1995) é a de produzir uma solução inicial rápida e fácil, e então usar um passo de melhoria que permita o escape de ótimos locais de pior qualidade. A heurística consiste em:

- **Passo 1:** Inicialmente, a restrição de capacidade dos veículos é relaxada de forma a permitir uma maior flexibilidade em movimentos de melhoria do **passo 4**. Seja Q_i a demanda total do cliente i ao longo do período, C_p a capacidade de cada veículo, V_t o número de veículos disponíveis no dia t , M o número de dias no horizonte de planejamento, e N número total de clientes. Computa-se a seguinte razão para ajudar a determinar quanto da capacidade de cada veículo deve ser liberada (admita-se que a frota é homogênea):

$$r_{capacidade} = \frac{\sum_{i=1}^N Q_i}{\sum_{t=1}^M V_t C_p}, \quad (3.28)$$

Se $r_{capacidade} \geq 90\%$, a capacidade de cada veículo é relaxada em 10%. Por exemplo, se a capacidade do veículo é 100 unidades e $r_{capacidade} = 0.93$, então permite-se que a carga máxima da rota seja de 110 unidades. Caso contrário ($r_{capacidade} < 90\%$),

então o veículo tem capacidade não utilizada suficiente, e a restrição de capacidade máxima não é relaxada.

- **Passo 2:** A solução inicial é produzida de forma a balancear a quantidade total da demanda nos dias do período. Esta solução inicial é obtida resolvendo-se o seguinte programa inteiro 0 - 1. Seja:

L : é a quantidade máxima total de demanda entregue em um único dia.

C_i : é o conjunto de combinação de visita permitidos para o cliente i .

$q_{i,k,t}$: é a demanda do cliente i no dia t da combinação k .

$a_{k,t} = 1$ se o dia t pertence a combinação k ; 0 caso contrário.

$x_{i,k} = 1$ se o cliente i é designado a combinação k ; 0 caso contrário.

Então, o modelo é dado por:

$$\text{Min } L , \tag{3.29}$$

sujeito a:

$$\sum_{k \in C_i} x_{i,k} = 1 , \quad i = 1 \dots N , \tag{3.30}$$

$$\sum_{i=1}^N \sum_{k \in C_i} a_{k,t} q_{i,k,t} x_{i,k} \leq L , \quad t = 1 \dots M , \tag{3.31}$$

$$x_{i,k} \in (0, 1) , \quad \forall k \in C_i, i = 1 \dots N . \tag{3.32}$$

A função objetivo (3.29) minimiza ao máximo a quantidade de demanda entregue em um único dia. As restrições (3.30) e (3.32) asseguram que exatamente uma única combinação de dia é designada a cada cliente. A restrição (3.31) força que a quantidade total de demanda entregue em um único dia seja menor ou igual a L .

Para resolver este programa é relaxada a restrição (3.32) ($x_{i,k} \in (0, 1)$), a qual é substituída por outra restrição ($0 \leq x_{i,k} \leq 1$). Os autores mostram que para o problema em questão, o número de variáveis $x_{i,k}$ fracionárias não é grande (para maiores detalhes, veja CHAO et al., 1995) e ao resolver o problema relaxado, a maior fração $x_{i,k}$ para um dado cliente i é arredondada para 1, de tal forma que qualquer cliente i seja atribuído para algum dia t .

- **Passo 3:** Uma modificação do algoritmo econômico de CLARKE & WRIGHT (1964) é então usada para resolver o problema de roteamento de veículos (PRV) em cada dia no período. As rotas resultantes ao se aplicar o algoritmo constituem a solução inicial. Note, entretanto, que ao resolver a relaxação do **passo 2**, não existe garantia de que haja capacidade suficiente para atender as demandas (solução infactível). O tratamento desta situação será descrito posteriormente.
- **Passo 4:** Neste passo são aplicados movimentos de melhoria. Os principais são: movimento de um ponto, movimento *clean up* e 2-OPT. Em movimentos de um ponto, elabora-se uma lista dos clientes em ordem decrescente do número de visitas. Para cada cliente da lista, investiga-se sua atribuição a outras rotas; isso pode ou não alterar seu padrão de visita corrente. Por exemplo, em um período de 5 dias, suponha que a solução inicial tenha atribuído o cliente i à seguinte combinação de visita: (dia 1, dia 3 e dia 5). Aplicando o movimento de um ponto, o cliente i é atribuído a uma outra combinação admissível para este cliente, por exemplo (dia 2, dia 4 e dia 5). Assim considera-se a inserção deste cliente na melhor posição factível (quanto à redução da distância total) das rotas do dia 2 e do dia 4, permanecendo na mesma posição na rota do dia 5. Alternativamente, supondo que cada dia do horizonte de planejamento possua duas rotas, o cliente pode ser simplesmente movido para a outra rota, sem ocorrer alteração da combinação de visita.

Se o movimento do cliente i reduzir a distância total, realiza-se o movimento imediatamente. Se a nova distância total excede a distância da melhor solução em no máximo 1%, move-se o cliente; caso contrário, o movimento é rejeitado.

Nota-se que o movimento de um ponto é realizado com a capacidade dos veículos relaxada (**passo 1**), o que implica em algumas rotas infactíveis. Para aproveitar a flexibilidade de tal situação, permite-se movimentos de melhoria infactíveis que mudam a solução infactível atual para uma nova solução infactível desde que esta apresente ganhos em distância. Espera-se com isso, que após a factibilização (veja abaixo) da nova solução infactível, tenha-se uma solução factível com um valor de função objetivo menor.

De forma a factibilizar a solução, elabora-se uma lista das rotas atuais em ordem decrescente da quantidade de demanda. Toma-se então a rota que mais viola a capacidade e tenta-se factibilizá-la, movendo-se um ou mais clientes desta rota a outras rotas. Ao considerar movimentos, preserva-se a factibilidade das outras rotas. Um cliente que requer mais do que uma visita no período está limitado a movimentos entre rotas no mesmo dia, enquanto um cliente que requer apenas uma visita pode ser movido para qualquer rota em qualquer dia. Entretanto, os autores reportam alguns casos em que a solução resultante mantém-se infactível, mesmo com a aplicação destes passos.

São então realizados movimentos *clean up*, onde tenta-se inicialmente melhorar cada rota de um dia específico, movendo um cliente por vez na mesma rota. A seguir, são considerados movimentos de clientes entre rotas no mesmo dia. Finalmente, tenta-se mover clientes que requerem apenas uma visita no período a outras rotas em outros dias. Apenas movimentos *clean up* que reduzem a distância total são aceitos.

Movimentos 2-OPT são então aplicados em cada rota, na tentativa de melhorar ainda mais a solução.

SEÇÃO 3.6.6

**Heurística de CORDEAU, GENDREAU &
LAPORTE (1997)**

No trabalho, os autores propõem uma heurística de busca tabu capaz de resolver três problemas de roteamento: o Problema do Caixeiro Viajante Periódico, o Problema de Roteamento de Veículos Multi Depósitos e o Problema de Roteamento de Veículos Periódico. Devido ao tema deste trabalho, será apresentado somente a heurística relacionada ao problema de roteamento de veículos periódico.

SEÇÃO 3.6.6.1

Construção da solução inicial

São apresentados os passos da heurística de construção de CORDEAU, GENDREAU & LAPORTE (1997):

1. Em uma lista, ordena-se os clientes em ordem crescente do ângulo que fazem com o depósito.
2. Escolhe-se aleatoriamente uma combinação de visitas para cada cliente i na lista.
3. Para cada dia l no período:
 - (a) Escolhe-se, dentre os clientes mais próximos ao depósito, um cliente j aleatório.
 - (b) Inicializa-se $k = 1$.
 - (c) Usando a sequência de clientes $j, j + 1, \dots, n, 1, \dots, j - 1$, realiza-se os seguintes passos para cada cliente i a ser atendido no dia l :
 - i. Se a inserção do cliente i na rota k do dia l , resultar na violação de alguma restrição do problema, atribui-se $k = \min\{k + 1, m\}$, onde m é o número máximo de veículos.
 - ii. Insere-se o cliente i dentro da rota k do dia l usando a heurística de inserção GENI (maiores detalhes conferir CORDEAU et al., 1997).

SEÇÃO 3.6.6.2

Funcionamento geral do algoritmo

A solução inicial produzida, não é necessariamente factível desde que o número de veículos avaliados em cada dia seja limitado. Como no algoritmo TABU-

ROUTE (1994), soluções infactíveis são permitidas durante a busca e a solução s é avaliada por meio da função custo $f(s) = c(s) + \alpha q(s) + \beta d(s)$, onde:

$$\begin{aligned}
 c(s) &= \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m \sum_{l=1}^t c_{i,j,k,l} x_{i,j,k,l} , \\
 q(s) &= \sum_{k=1}^m \sum_{l=1}^t \left[\left(\sum_{i=0}^n \sum_{j=0}^n q_i x_{i,j,k,l} \right) - Q_k \right]^+ , \\
 d(s) &= \sum_{k=1}^m \sum_{l=1}^t \left[\left(\sum_{i=0}^n \sum_{j=0}^n (c_{i,j,k,l} + d_i) x_{i,j,k,l} \right) - D_k \right]^+ .
 \end{aligned}$$

α e β são parâmetros positivos, e $[x]^+ = \max\{0, x\}$. Assim se s é positivo, o valor de $c(s)$ e $f(s)$ coincide. Caso contrário, inclui-se dois termos de penalidade proporcionais à quantidade de violação da capacidade e da duração das rotas.

Para cada solução s é associado um conjunto de atributos $B(s) = (i, k, l)$: o cliente i é visitado pelo veículo k no dia l . A transição da solução corrente s para a solução $s' \in N(s)$ pode ser espessa pela remoção e adição de atributos para o conjunto $B(s)$. De fato, a vizinhança $N(s)$ de uma solução s é composta por todas as soluções que podem ser obtidas executando-se uma das seguintes transformações:

1. Remove-se o cliente i da rota k no dia l e insere-se este cliente dentro de outra rota k' .
2. (a) Substitui-se a combinação r correntemente atribuída para o cliente i por outra combinação $r' \in C_i$.
 (b) Para $l = 1$ até t , faz-se
 - Se $a_{r,l} = 1$ e $a_{r',l} = 0$, remove-se o cliente i da rota no dia l
 - Se $a_{r,l} = 0$ e $a_{r',l} = 1$, insere-se o cliente i dentro da rota no dia l minimizando o aumento em $f(s)$.

Se o cliente i é removido da rota k no dia l , sua reinserção é proibida nas próximas θ iterações. O número da última iteração para o qual o atributo i, k, l é declarado

tabu é denotado por $\tau_{i,k,l}$. O *status* tabu de um atributo pode, contudo, ser eliminado se este resultar em uma solução factível de menor custo do que a melhor solução.

O nível de aspiração de um atributo (i, k, l) , denotado por $\sigma_{i,k,l}$ é inicialmente igual a $c(s)$ se (i, k, l) pertencer ao conjunto de atributos de uma solução inicial factível, e a infinito, caso contrário. Toda vez que uma solução factível s é identificada, o nível de aspiração de cada um de seus atributos (i, k, l) é atualizado para $\min\{\sigma_{i,k,l}, c(s)\}$.

Para cada iteração, o subconjunto $M(s) \subseteq N(s)$ consiste de todas as soluções $s' \in N(s)$ que não impliquem em risco de ciclagem. Estas soluções s' são tais que pelo menos um dos atributos que precisam ser adicionados a s para se obter s' , não são tabu ou $c(s')$ é menor que o nível de aspiração destes atributos.

Para diversificar a busca, a avaliação do custo de uma solução que resulta em um aumento no valor de $f(s)$ é modificada de forma a penalizar atributo mais frequentemente adicionado a solução.

Finalmente, a cada iteração, os valores de α e β são modificados por um fator $1 + \delta > 1$. Se a solução corrente é factível com respeito a quantidade (duração), o valor de α (β) é dividido por $1 + \delta$ (é multiplicado por $1 + \delta$).

4

Aplicação da abordagem adaptativa HTA ao PRV

Assim como as abordagens de HÜBSHER & GLOVER (1993) e CHAKRAPANI & SKORIN-KAPOV (1992) (veja seção 2.2.2.3 do capítulo 2), a abordagem HTA (**H**orizon**T**e **A**daptativo) (PUREZA & FRANÇA, 2001) também procura promover a estratégia de intensificação e diversificação. Esta abordagem foi originalmente aplicada ao problema de roteamento de veículos, o qual será explicado a seguir. A apresentação dos elementos que compõem o algoritmo é feita da seguinte forma: na seção 4.1 a heurística de construção de rotas (SOLOMON, 1987); na seção 4.2, a heurística de melhoria baseada no método de DROR & LEVI (1986) e na seção 4.3, a integração das estratégias de intensificação e diversificação por meio da abordagem HTA, que avalia padrões da trajetória da busca baseada em variações do custo da solução, alterando dinamicamente os parâmetros tabu.

SEÇÃO 4.1

Heurística de construção de rotas

O algoritmo de construção de rotas utilizado é creditado a SOLOMON (1987) e foi originalmente elaborado para resolução do PRVJT (Problema de Roteamento

de Veículos com Janela de Tempo). O algoritmo inicia com a escolha de um ponto semente, correspondente ao cliente situado mais distante do depósito, para iniciar a primeira rota. Para cada cliente não roteado u , é calculada sua melhor posição factível de inserção, segundo um critério que favorece a economia em distância, tempo ou uma combinação de ambos. O critério é definido por:

$$c_1(i(u), u, j(u)) = \text{Min}_{1 \leq p \leq m} \{c'_1(i_{p-1}, u, i_p)\} .$$

sendo m o número máximo de posições de alocação do cliente u na rota.

A função $c'_1(i, u, j)$ é dada por:

$$c'_1(i, u, j) = \alpha_1 c_{1,1}(i, u, j) + \alpha_2 c_{1,2}(i, u, j) ,$$

onde

$$\alpha_1 + \alpha_2 = 1; \alpha_1 \geq 0, \alpha_2 \geq 0 ,$$

$$c_{1,1}(i, u, j) = d_{i,u} + d_{u,j} - \eta d_{i,j}, \eta \geq 0 ,$$

$$c_{1,2}(i, u, j) = b_{u,j} - b_j ,$$

onde, $b_{u,j}$ é o instante de início de atendimento do cliente j após a inserção do cliente u .

Em seguida, outro critério baseado no primeiro selecionará o melhor entre os clientes ainda não alocados para fazer parte da rota atual, definida por:

$$c_2(i(u^*), u^*, j(u^*)) = \text{Max}_u \{c'_2(i(u), u, j(u))\} ,$$

onde

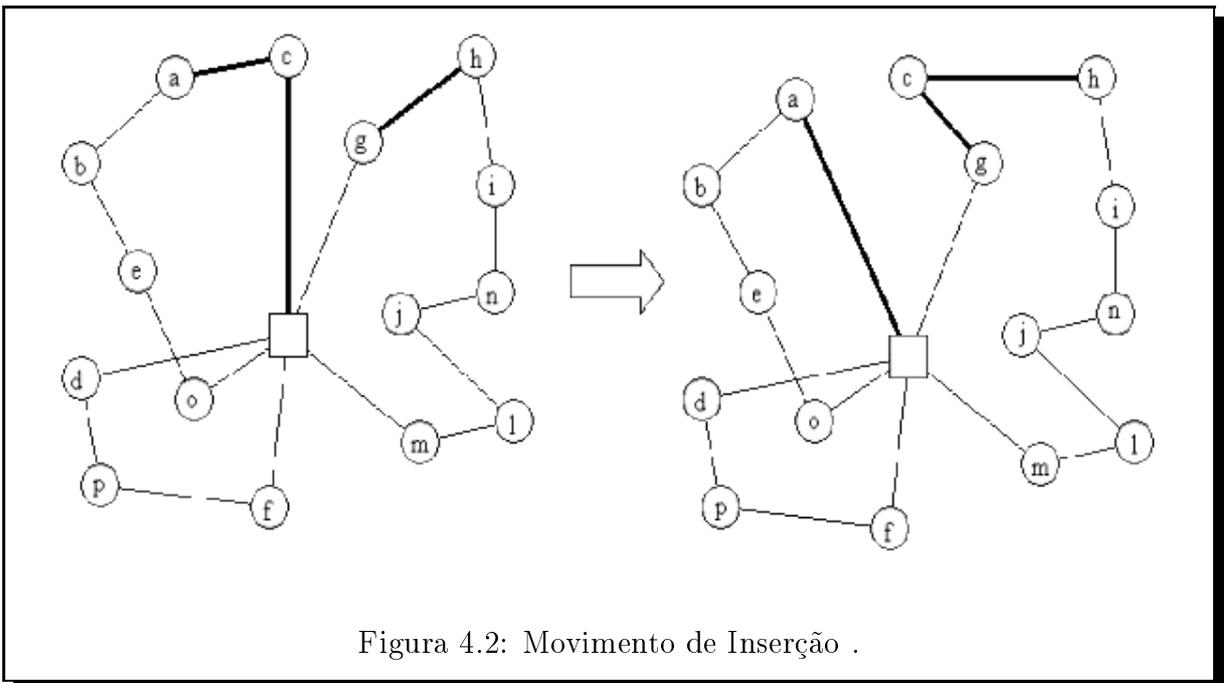
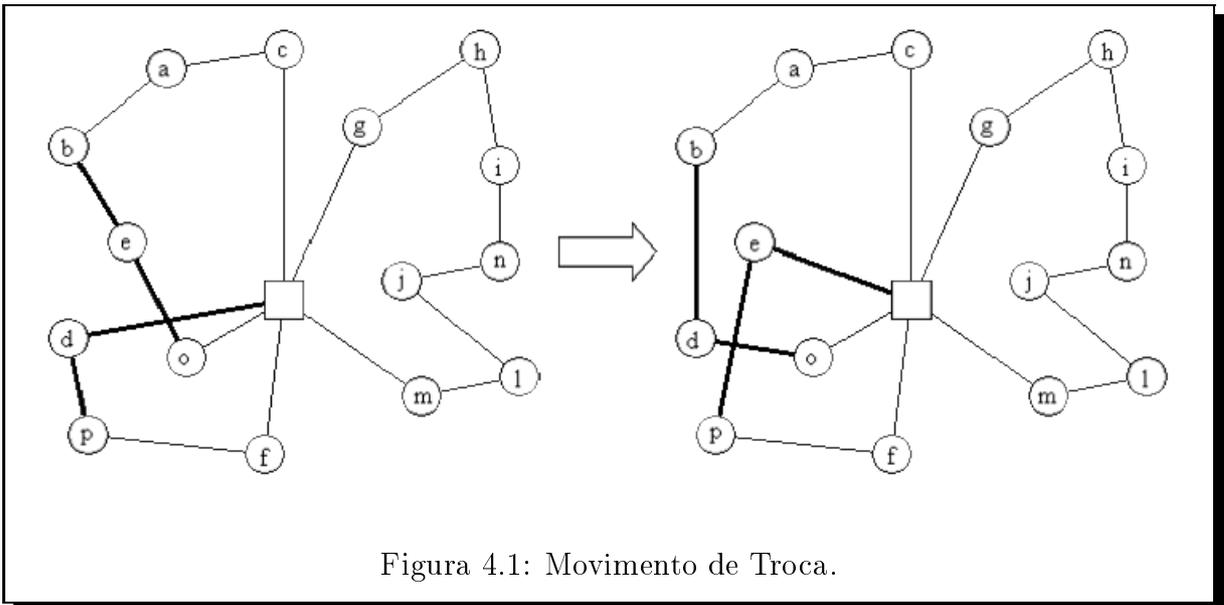
$$c'_2(i, u, j) = \lambda d_{0,u} - c_1(i, u, j), \lambda \geq 0 .$$

O cliente u^* é inserido na rota entre $i(u^*)$ e $j(u^*)$. Quando a inserção de um novo cliente resultar na infactibilidade da rota corrente, uma nova rota é iniciada, escolhendo-se uma nova semente, a menos que todos os clientes já estejam alocados. Na heurística de construção de Solomon existem três parâmetros exógenos α_1 , α_2 e η cujos valores impõem diferentes impactos sobre a avaliação dos candidatos. Dependendo dos valores adotados, pode-se considerar a maximização da economia em distância, em tempo, ou uma combinação de ambos. O parâmetro λ impõe graus diferentes de penalização às rotas compostas de um único cliente. O cliente (relativo a posição) com a maior avaliação é então selecionado para inserção.

SEÇÃO 4.2

Heurística de melhoria de rotas

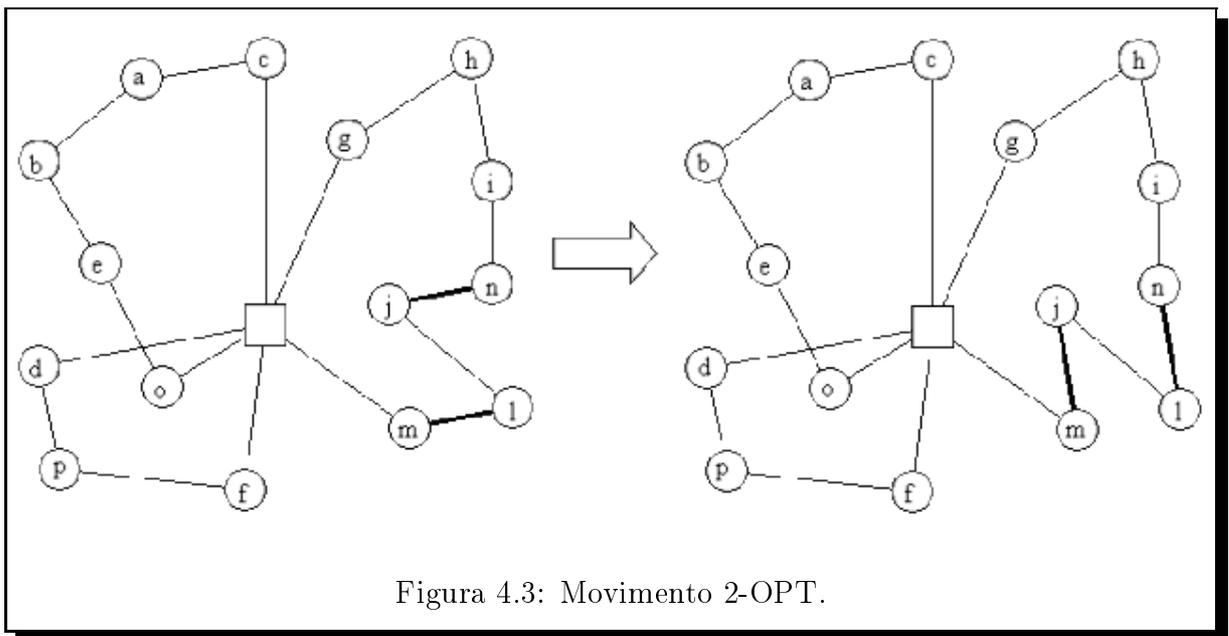
A heurística de melhoria em que foram incorporados mecanismos de Busca Tabu é baseado no método de DROR & LEVY (1986). A heurística realiza dois tipos de movimentos: inserção e troca de nós, apresentados respectivamente nas Figuras 4.1 e 4.2.



Nota-se na Figura 4.1 que o movimento realiza uma operação de troca simples entre dois nós que pertençam a rotas distintas, enquanto que na Figura 4.2, é realizada uma operação de inserção de um único nó da rota corrente a uma rota alternativa. Estas operações são chamadas de *movimentos inter-rotas*.

A partir da solução inicial gerada pelo algoritmo de construção, o algoritmo de PUREZA & FRANÇA (2001) aplica repetidamente movimentos inter-rotas até que as possibilidades de melhoria tenham sido esgotadas.

A cada iteração, o melhor movimento é selecionado. Ao se atingir um ótimo local em relação a estas vizinhanças, a heurística de busca local de troca de arestas 2-OPT é aplicada a cada rota individualmente (conforme Figura 4.3). Estas operações são chamadas de *movimentos intra-rota*.



A partir da solução final da aplicação da heurística 2-OPT, movimentos inter-rotas são novamente aplicados, desta vez, sujeitos a restrições tabu. Como atributos são utilizadas as arestas adicionadas e eliminadas, em cada movimento. Isto é válido tanto para os movimentos intra-rotas quanto para os movimentos inter-rotas.

São criadas duas estruturas (matrizes) tabu *TABUADD* e *TABUDEL*, onde as linhas e colunas referem-se aos identificadores dos nós do problema. Isso significa

que $TABUADD(i, j)$ indica a aresta que liga os nós i e j . Quando um movimento é efetivado, para cada aresta eliminada, gera-se aleatoriamente, com distribuição uniforme, um número em um intervalo entre $[t_{min}, t_{max}]$. Este número é somado ao valor da iteração corrente e o resultado é armazenado na posição $TABUDEL(i, j)$. Similarmente, para cada aresta adicionada (i, j) armazena-se o número gerado pelo mesmo processo em $TABUADD(i, j)$.

Em um dado movimento candidato na iteração $iter$, as arestas que devem ser eliminadas (adicionadas) da rota tal que $TABUDEL(i, j) \leq iter$ ($TABUADD(i, j) \leq iter$), são classificadas como tabu-ativas. Note que são adotados períodos tabu dinâmicos e que as arestas envolvidas em um dado movimento podem perder seu *status* tabu-ativo em iterações diferentes.

O *status* de movimentos candidatos que compreendam uma ou mais arestas tabu-ativas é determinado pelo parâmetro tolerância Tol a qual define o número máximo aceitável de arestas tabu-ativas presentes no movimento.

Como o algoritmo realiza três tipos de movimentos (inserção de nós, troca de nós e troca de arestas intra-rotas), para cada tipo de movimento, Tol pode teoricamente assumir diferentes valores inteiros dependendo do número de arestas afetadas. Os valores 0 a 8 correspondem a movimentos de troca de dois nós inter-rotas pois este movimento envolve 8 arestas, 4 arestas eliminadas e 4 arestas adicionadas (conforme Figura 4.1). Os valores de 0 a 6 correspondem a movimentos de inserção de um nó inter-rotas pois 3 arestas são eliminadas e 3 arestas são adicionadas (conforme Figura 4.2). Finalmente, os valores de 0 a 4 correspondem a movimentos de troca de arestas intra-rotas, uma vez que 2 arestas são eliminadas e 2 arestas são adicionadas (conforme Figura 4.3).

Por esta razão, introduziu-se o conceito de 3 tipos de tolerância: TT (tolerância em movimentos de troca de nós), TI (tolerância em movimentos de inserção de nós) e TO (tolerância em movimentos de trocas de arestas). Como critério de parada, o algoritmo utiliza o número de iterações sem melhoria. Foi também incluído um critério de aspiração que permite a efetivação de movimentos tabu que resultem em soluções melhores do que a melhor até o momento.

SEÇÃO 4.3

A abordagem adaptativa HTA

Como visto no capítulo 2, variações na configuração da lista e, em particular, do período tabu ao longo do processo de busca são abordagens adaptativas que integram as estratégias de intensificação e diversificação e provocam melhorias no desempenho da busca. Em regiões atraentes, ou seja, regiões próximas a ótimos locais recomenda-se a utilização de períodos tabu menores de forma a explorar detalhadamente a região (estratégia de intensificação). Períodos tabu maiores são usados quando se torna necessário explorar outras regiões de forma a escapar dos ótimos locais já visitados e com grande influência (estratégia de diversificação). A abordagem tabu adaptativa HTA também pode ser considerada uma estratégia de intensificação e diversificação uma vez que provoca perturbações em elementos tabus selecionados, modificando os níveis de restritividade. A principal diferença em relação às abordagens adaptativas anteriores é que estas perturbações são aplicadas depois de uma análise e identificação de *padrões da trajetória local da busca*. Tais análises são realizadas ao longo de todo o processo de busca em intervalos especificados, e a perturbação sugerida é aplicada por um certo número de iterações. A análise do padrão de trajetória determina a natureza, o grau, e a duração da perturbação. Uma outra diferença em relação às técnicas anteriormente empregadas é que as perturbações são aplicadas não ao período tabu, mas à regra de ativação tabu.

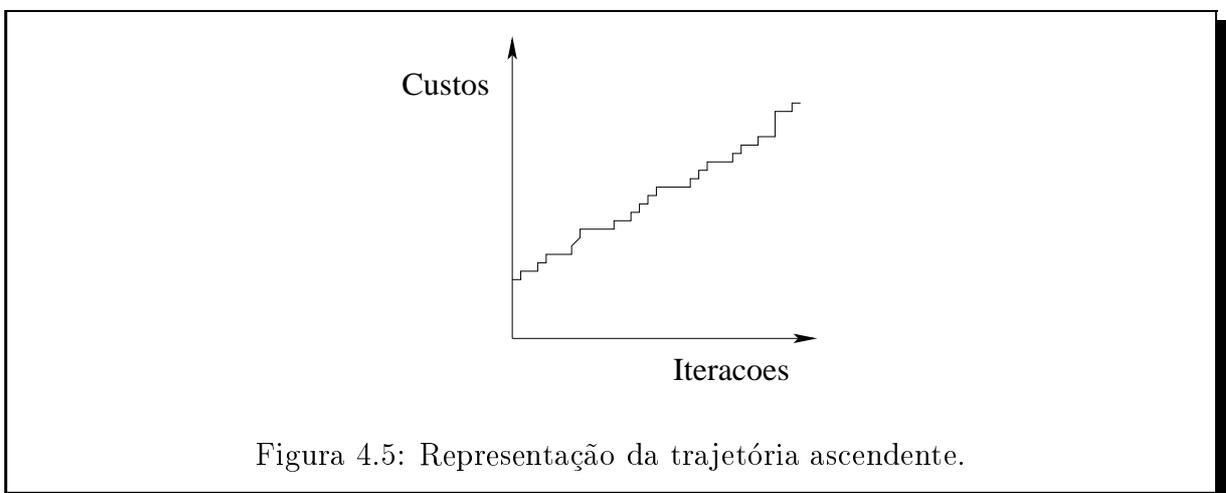
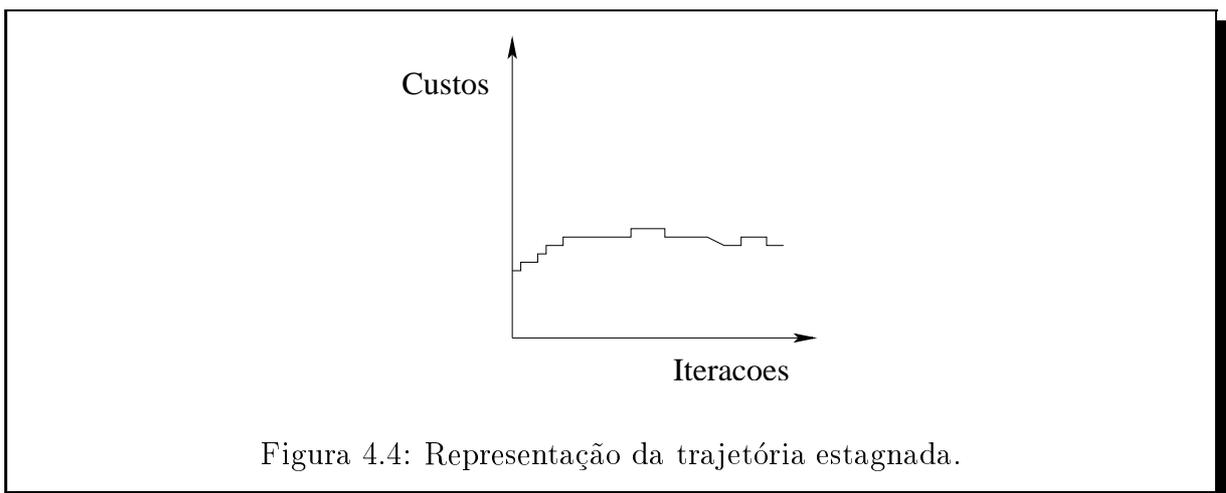
SEÇÃO 4.3.1

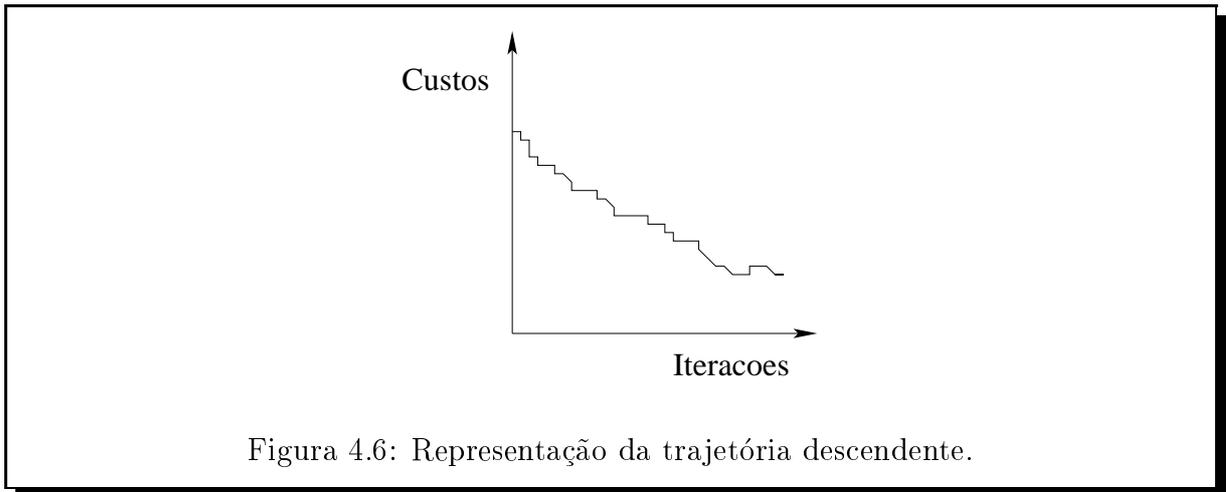
Padrões da trajetória da busca

A abordagem adaptativa HTA foi elaborada para identificar três padrões considerados relevantes ao desempenho da busca, reagindo diferentemente a cada um deles. Estes padrões são *estagnação da busca*, *trajetórias ascendentes* e *trajetórias descendente*. Quando baixa restritividade é imposta (restrições tabu fracas ou períodos de

curta duração) pode-se dizer que a trajetória torna-se estagnada. A trajetória estagnada é caracterizada pela geração de custos de soluções limitados a uma faixa estreita (Figura 4.4). Entende-se que a pequena variação nos custos resulta do fato da busca estar restrita a uma pequena porção do espaço onde as soluções obtidas compartilham várias arestas. No caso extremo, verifica-se ciclagem de soluções. A abordagem HTA responde a este tipo de trajetória, provocando o aumento dos níveis de restritividade (diversificação da busca).

Quando alta restritividade é imposta (movimentos recentes são restritos mais severamente ou são deixados tabu por períodos mais longos) podemos dizer que a trajetória torna-se ascendente. A trajetória ascendente é caracterizada pelo aumento dos custos das soluções (Figura 4.5). A abordagem HTA responde a este tipo de trajetória relaxando as restrições tabu, de forma que novas regiões sejam atingidas.





Quando a trajetória é descendente, ou seja, caracterizada pela diminuição dos custos de solução (Figura 4.6), a abordagem HTA estabelece níveis moderados de restritividade em busca de melhorias adicionais (intensificação da busca). Trajetórias descendentes podem resultar da fase de relaxação das restrições tabu prescrita pela abordagem adaptativa, ou seja, quando se verifica que a trajetória imediatamente anterior é ascendente.

SEÇÃO 4.3.2

Identificação dos padrões da trajetória

A identificação dos padrões da trajetória é feita comparando-se as médias dos custos de soluções no penúltimo e último estágio da busca, respectivamente denominadas de *média anterior* e *média corrente*. O comprimento de cada estágio de busca é definido pelo parâmetro *horizonte de avaliação*. Este parâmetro define o momento em que as perturbações nos elementos tabu devem ser aplicadas, o número de iterações a serem utilizados na próxima avaliação e os custos das soluções, também a serem utilizados na próxima avaliação. O comprimento de cada estágio de busca em uma dada avaliação é definido por uma variável inteira e positiva (*hoz*) gerado aleatoriamente segundo distri-

buição uniforme em um intervalo $[h_{min}, h_{max}]$. A identificação dos padrões da trajetória pode ser melhor visualizada analisando a Figura 4.7.

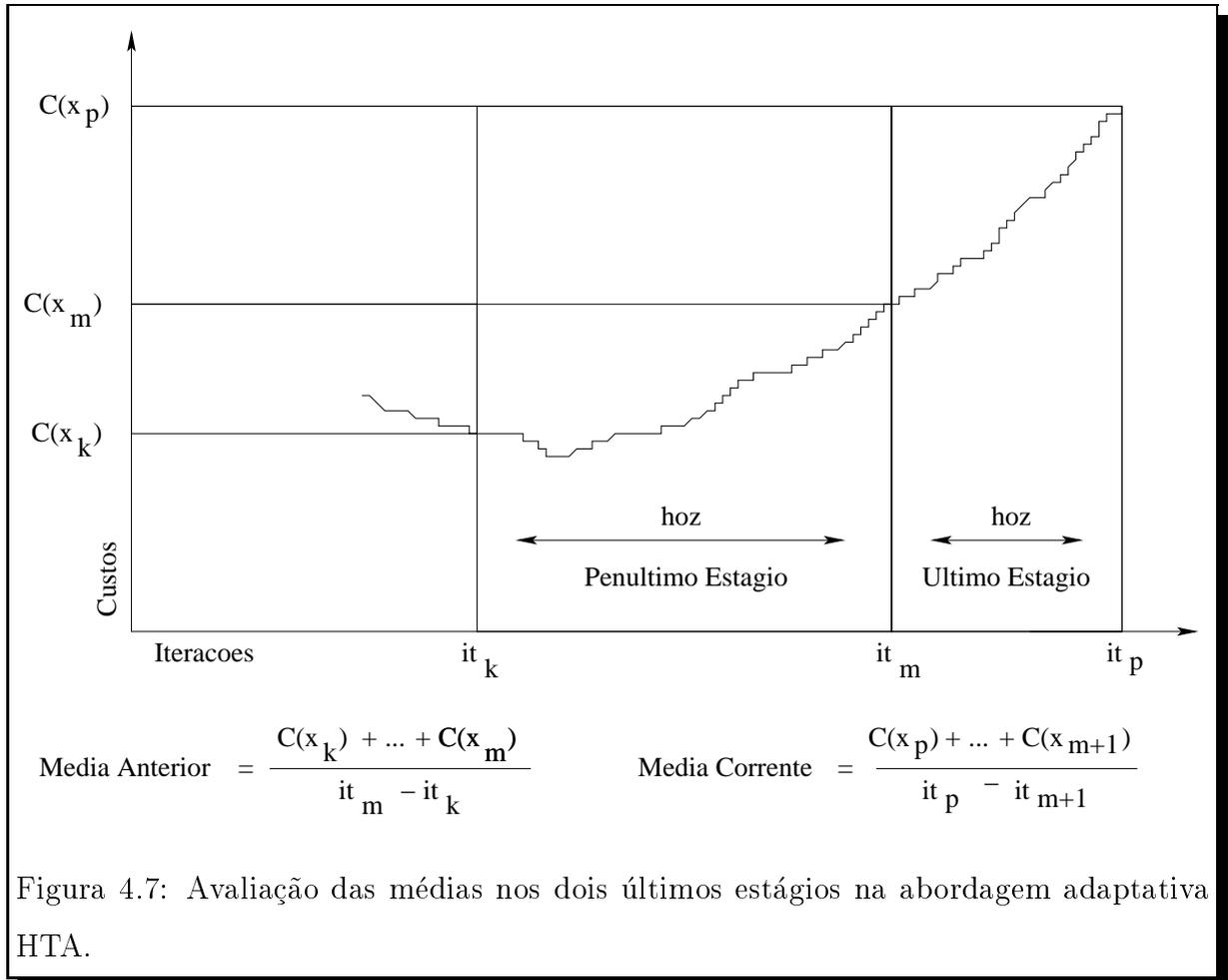


Figura 4.7: Avaliação das médias nos dois últimos estágios na abordagem adaptativa HTA.

A figura mostra os elementos que compõem a abordagem adaptativa HTA. A curva do gráfico representa a trajetória dos dois últimos estágios de busca realizados dados pelos custos das soluções ($c(x_i)$) versus iteração (it_i). O horizonte *hoz* define it_m como o fim do penúltimo estágio e it_p como o fim do último estágio. A iteração it_p corresponde à última iteração do processo até o momento e ao instante em que a análise dos padrões da trajetória é realizada. A análise utiliza os custos de soluções obtidos entre as iterações it_k e it_m para calcular a média anterior e it_{m+1} e it_p para calcular a média corrente.

Assim, se os valores das médias são aproximadamente iguais entende-se que está ocorrendo a estagnação; se a média corrente é maior que a anterior então a busca está

descrevendo uma trajetória ascendente; se a média corrente é menor que a anterior então a busca está descrevendo uma trajetória descendente. Para cada uma destas trajetórias a abordagem adaptativa HTA reage alterando o valor dos parâmetros de tolerância, ou seja, da regra de ativação tabu, de uma forma diferente. Para trajetórias estagnadas, diminui-se os valores de tolerância ao mínimo de forma a aumentar os níveis de restritividade. Para trajetórias ascendentes, os valores de tolerância são aumentados, podendo assumir diferentes valores dependendo da severidade dos graus de restritividade atuais. Para trajetórias descendentes, as tolerâncias mantêm níveis de restritvidades moderados.

Tais perturbações são realizadas em intervalos pré-determinados, de forma que a iteração da próxima avaliação é conhecida previamente. Em geral, a iteração da próxima avaliação é calculada adicionando-se a parte inteira de m *hoz* à iteração corrente ($m \geq 0$). O parâmetro m é um fator de ajuste do período de aplicação a cada tipo de perturbação.

Perturbações que envolvem a aplicação de altos níveis de restritividade requerem pequenos valores de m , geralmente menores que 1, de forma a limitar o aumento dos custos. Trajetórias descendentes sugerem valores de m maiores que 1 como forma de explorar as possibilidades de melhoria. Utilizou-se também um critério de aspiração que mantém a busca operando em valores de tolerância padrão (independente das características das trajetórias), caso ocorra a atualização da melhor solução no último estágio da busca. Neste caso, m é maior que 1.

Ao contrário de TT e TI , a tolerância TO (movimentos intra-rotas) é mantida fixa e igual a 4, uma vez que se trata de buscas locais para refinamento da solução.

SEÇÃO 4.4

Passos do algoritmo HTA para o PRV

1. Seja $it = 0$ (iteração corrente). A partir da solução inicial, seja $TT = TI = 3$ e $TO = 4$. Procede-se com a busca até que o primeiro ótimo local seja atingido,

armazenando-se os custos das soluções em um vetor C.

2. Calcule a média dos custos armazenados no vetor C (*média anterior*). Reinicialize o vetor C. Obtém-se um valor de horizonte de avaliação (*hoz*) entre $[h_{min}, h_{max}]$; assim a próxima avaliação (*prx_aval*) ocorrerá na iteração $it + hoz$. Neste meio tempo, armazena-se os custos das soluções no vetor C. Quando $it = prx_{aval}$, faça *média corrente* = a média dos custos em C. Compute um novo *hoz* e reinicialize o vetor C.
3. Se tiverem sido realizadas x ($x = 3000$) iterações sem melhoria, termina-se o algoritmo e retorna-se a incumbente. Caso contrário:
 - (a) Se houve atualização da solução incumbente, faça $TT = 6$ e $TI = 5$. Mantem-se a busca por $2hoz$ iterações, armazenando-se os custos no vetor C.
 - (b) Caso contrário, calcula-se:

$$diff = \frac{\text{média anterior} - \text{média corrente}}{\text{média anterior}}.$$

- i. Se $|diff| \leq 0,0025$, está ocorrendo *estagnação* da busca. Aplica-se máxima restritividade ($TT = 0$ e $TI = 0$) e mantêm-se a busca por $\frac{hoz}{2}$ iterações ($prx_{aval} = it + hoz$). Armazena-se os custos da solução no vetor C.
 - ii. Caso contrário, se $diff < 0$, então verifica-se uma trajetória ascendente. Os níveis de restritividade são alterados de acordo com a Tabela 4.1. Mantêm a busca por $2hoz$ iterações ($prx_{aval} = it + 2hoz$), armazenando-se os custos no vetor C.
 - iii. Caso contrário, se $diff > 0$, então verifica-se uma trajetória descendente. Se $TT = 7$ e $TI = 6$, então faça $TT = 6$ e $TI = 5$. Mantêm-se a busca por $2hoz$ iterações ($prx_{aval} = it + 2hoz$). Neste meio tempo, armazena-se os custos no vetor C.
4. Quando $it = prx_{aval}$, faça *média anterior* = *média corrente* e a *média corrente* = média dos custos armazenados no vetor C. Reinicializa-se o vetor C. Obtém-se um novo valor de horizonte de avaliação entre $[h_{min}, h_{max}]$ e volta-se ao passo 3.

Tabela 4.1: Regras para trajetórias ascendentes.

Se $lim_1 \leq diff < lim_2$ então:			
lim_1	lim_2	TT	TI
$-\infty$	-0,03	7	6
-0,03	-0,0025	6	5
-0,025	-0,02	5	4
-0,02	-0,015	4	3
-0,015	-0,005	3	3
-0,005	-0,0025	2	2

SEÇÃO 4.5

Resultados do algoritmo HTA

Nesta seção será apresentado o desempenho do algoritmo juntamente com o de outras heurísticas de busca tabu competitivas. O algoritmo foi testado com 14 exemplos clássicos descritos em CHRISTOFIDES et al. (1979).

Tabela 4.2: Resultados computacionais de PF_D_HTA com os outros algoritmos de busca tabu.

Pr.	T	O	TR	DETABA	PF_D_HTA				
p1	524,61	524,61	0,6	524,61	1,4	524.61	-	524,61	2,3
p2	835,26	844	0,4	835,32	-	836.71	-	835,77	8,0
p3	826,14	835	6,7	826,14	-	828.72	-	826,64	37,4
p4	1028,42	1044,35	22,9	1031,07	-	1043.89	-	1039,30	113,0
p5	1298,79	1334,35	28,4	1311,35	-	1306.16	-	1329,17	98,0
p6	555,43	555,43	1,1	555,43	7,8	-	-	555,43	3,1

Tabela 4.2: (continuação)

Pr.	T	O	TR	DETABA	PF_D_HTA
p7	909,68	911 17,6	909,68	- -	910,94 14,5
p8	865,94	866,75 13,7	865,94 5,9	- -	865,51 30,3
p9	1162,89	1184 14,9	1162,89	- -	1164,68 211,9
p10	1397,94	1422 28,8	1404,75	- -	1411,34 105,5
p11	1042,11	1042,11 13,4	1042,11	- 1051.18	- 1042,11 16,6
p12	819,56	819,56 2,1	819,56 1,7	819,56	- 819,56 16,7
p13	1541,15	1547 10,2	1545,93	- -	- 1542,97 60,8
p14	866,37	866,37 6,9	866,37 29,7	- -	- 866,37 1,2

Fonte: PUREZA & FRANÇA (2001).

Em negrito: melhor solução obtida.

Primeira coluna de resultados de cada algoritmo: custo da solução.

Segunda coluna de resultados de cada algoritmo: tempo de processamento para obter a melhor solução em minutos.

Na Tabela 4.2, os algoritmos são referenciados como:

- T : algoritmo de TAILLARD (1992) - computador Silicon Graphics 4D/35.
- O : algoritmo de OSMAN (1993) - computador VAX 8600.
- TabuRoute (TR) : algoritmo de GENDREAU, HERTZ & LAPORTE (1994) - computador Silicon Graphics 36 MHz, 5.7 Mflops.
- DETABA: algoritmo de BARBAROSOGLU et al. (1999) - computador Pentium 133 MHz, 32Mb RAM.
- PF_D_HTA: algoritmo de PUREZA & FRANÇA (2001) com abordagem HTA - Estação de trabalho Sun Ultra 1.

A primeira coluna da Tabela 4.2 apresenta os custos das melhores soluções

obtidas pelos algoritmos para cada um dos problemas, enquanto a segunda coluna indica o tempo de processamento para obtenção da melhor solução (em minutos), se disponível.

Os algoritmos T, O, TR e DETABA foram selecionados em função de sua alta competitividade em termos de qualidade de solução. Todos são implementações de busca tabu e resultam nos melhores resultados que se tem conhecimento para este conjunto de problemas. Em particular, T e TR utilizam estratégias sofisticadas tais como paralelização e busca em regiões inactíveis. Observa-se que os resultados do algoritmo PF_D_HTA apresentam um desempenho comparável a TabuRoute. Resultados equivalentes foram obtidos em cinco problemas e no problema p8 foi possível obter uma solução melhor que a melhor solução dos demais cinco algoritmos. Quanto aos tempos computacionais, existem dificuldades em se comparar os algoritmos pelo fato de terem sido testados em máquinas diferentes.

Em função de sua simplicidade e desempenho competitivo com este conjunto de problemas, o algoritmo PF_D_HTA foi selecionado para rotear os dias no Problema de Roteamento de Veículos Periódico a ser explicado no próximo capítulo.

5

Aplicação da abordagem HTA ao PRVP

SEÇÃO 5.1

Introdução

Neste capítulo é apresentada uma aplicação similar ao do algoritmo HTA para o tratamento do Problema de Roteamento de Veículos Periódico. Com o objetivo de obter melhores soluções para o problema, foram desenvolvidas duas heurísticas, chamadas de VERSÃO 1 e VERSÃO 2.

A VERSÃO 1 consiste de duas etapas principais:

- (1) designação dos clientes aos dias do período e posterior roteamento;
- (2) movimentos de trocas de clientes entre dias e posterior ajuste das rotas.

Na etapa (1) foram desenvolvidas três formas diferentes de designação dos clientes aos dias, descritas na seção 5.2. O roteamento foi obtido aplicando-se o algoritmo HTA a cada dia do período. Na etapa (2), com o objetivo de tentar minimizar os custos das soluções foram realizados movimentos de trocas de clientes entre dias, baseados na heurística de CHRISTOFIDES & BEASLEY (1984) descrita na seção 5.3.1. O ajuste das

rotas foi obtido aplicando-se o movimento *clean up*, proposta por CHAO et. al (1995) e descrita na seção 3.6.5 do capítulo 3.

A VERSÃO 2 consiste de todos os passos utilizados na VERSÃO 1, sendo simplesmente acrescentada uma lista *LC*, que armazena os custos das soluções obtidas nas *it* últimas iterações. Estes custos são analisados para identificar o padrão da trajetória, como *estagnada*, *ascendente* ou *descendente*. A partir de então, com base no padrão da trajetória identificada, realizam-se alterações nos parâmetros do algoritmo para permitir uma maior intensificação ou diversificação na busca.

Observa-se que na VERSÃO 2 foi utilizada uma idéia “semelhante” ao do algoritmo HTA. Pretende-se com esta idéia, acompanhar os custos da solução total, ou seja, de todo o horizonte de planejamento, e então intensificar ou diversificar a busca. O processo de intensificação e diversificação utilizado nesta versão é realizada de forma diferente ao do algoritmo HTA, pois não são utilizados mecanismos tabu para controlar o processo de busca. De forma simplificada, trajetórias estagnadas são tratadas permitindo apenas soluções que representem pioras nos custos, enquanto em trajetórias ascendentes e descendentes aceitam-se apenas soluções que melhoram os custos.

O algoritmo HTA neste trabalho, foi utilizado tanto na VERSÃO 1 quanto na VERSÃO 2, para rotear os dias no período. Este roteador é composto de mecanismos adaptativos de busca tabu que determinam as ações de intensificação e diversificação, conforme mencionado no capítulo 4.

SEÇÃO 5.2

Designação dos clientes aos dias

SEÇÃO 5.2.1

Construção inicial 1 (CI-1)

Utiliza-se o procedimento proposto por CHRISTOFIDES & BEASLEY (1984), para a designação dos clientes aos dias. Inicialmente geram-se T centros (um para cada dia do período). Posteriormente, os centros são associados aos dias, através da

utilização de uma matriz $M_{j,t}$, onde j representa os centros e t o dia do período. Cada posição desta matriz informa o quanto cada centro está sendo requisitado pelos clientes no dia t . O maior valor associado para cada centro entre os dias nesta matriz define o dia associado àquele centro. Finalmente, os clientes são atribuídos aos dias escolhendo-se a combinação de visita que resulte em uma menor distância entre o cliente e o centro. Detalhadamente, temos:

- Definindo os T centros:

Os centros correspondem a T clientes selecionados, onde T é o comprimento do período. Estes clientes (centros) são obtidos pela equação:

$$\text{Max}_{\substack{i=1, \dots, n \\ i \notin J}} \left[d_{0,i} \left(\prod_{j \in J} d_{i,j} \right) \right] \quad (5.1)$$

onde:

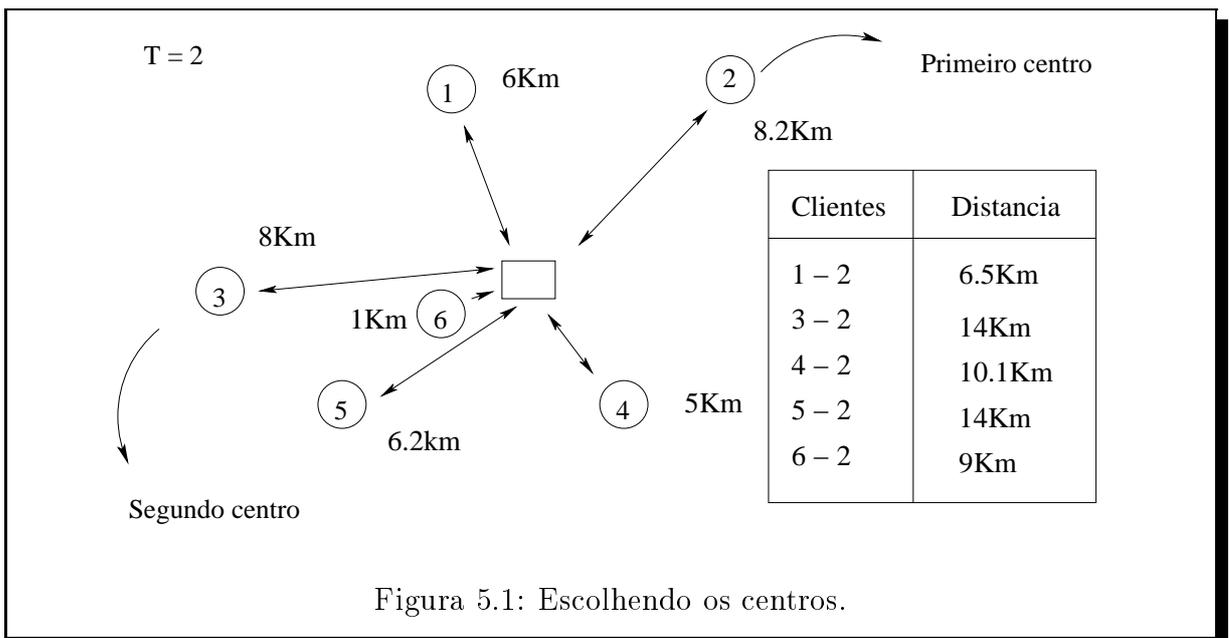
n : número de clientes ($i = 1, 2, \dots, n$).

j : clientes escolhidos como centros.

J : conjunto de clientes escolhidos como centros.

$d_{0,i}$: distância do depósito ao cliente i .

$d_{i,j}$: distância do cliente i ao centro j .



Pela equação 5.1, escolhe-se inicialmente o cliente mais distante do depósito como o primeiro centro, em seguida escolhe-se o cliente mais longe do depósito e mais longe do primeiro centro, como sendo o segundo centro e assim sucessivamente até obter-se T centros. Seja J o conjunto de centros. Inicialmente J é vazio ($J = \{\emptyset\}$), e na medida que os centros são escolhidos, são armazenados no conjunto J ($J = \{j_1, j_2, \dots, j_T\}$). A Figura 5.1 apresenta um exemplo do procedimento acima. Nela está representada a distribuição geográfica dos clientes. Supondo que trabalha-se com um período de dois dias ($T = 2$), utilizando a equação 5.1 obtém-se o cliente 2 como o primeiro centro (pois é o cliente mais longe do depósito) e o cliente 3 como o segundo centro (pois é o cliente mais longe do depósito e mais longe do primeiro centro). Então, armazena-se no conjunto J os dois centros escolhidos ($J = \{2, 3\}$).

- Associando os dias aos centros:
 - **Passo 1:** Para cada cliente i é armazenado em um vetor J_i os centros mais próximos a ele. O número de centros mais próximos que são armazenados no vetor J_i do cliente i corresponde à frequência de visitas do cliente. Ou seja, se o cliente i deseja ser visitado uma vez no período, armazena-se no vetor J_i o centro mais próximo dele, se o cliente i deseja ser visitado duas vezes no período, armazenam-se em seu vetor J_i os dois centros mais próximos dele, e assim sucessivamente.

Como exemplo, considera-se que todos os clientes têm frequência um, ou seja, serão visitados uma vez no período. Supõe-se as seguintes combinações de visitas C_i para os clientes representados na Figura 5.1:

$C_1 = \{dia1\}$	$F_1 = 1$
$C_2 = \{dia1\}$ ou $\{dia2\}$	$F_2 = 1$
$C_3 = \{dia1\}$ ou $\{dia2\}$	$F_3 = 1$
$C_4 = \{dia1\}$	$F_4 = 1$
$C_5 = \{dia1\}$ ou $\{dia2\}$	$F_5 = 1$
$C_6 = \{dia1\}$ ou $\{dia2\}$	$F_6 = 1$

Considerando que os clientes 2 e 3 são centros e aplicando o **passo 1** aos dados da Figura 5.1, obtém:

$J_1 = 2$ (o cliente 1 tem o centro 2, como o centro mais próximo)

$J_2 = 3$ (o cliente 2 tem o centro 3, como o centro mais próximo)

$J_3 = 2$ (o cliente 3 tem o centro 2, como o centro mais próximo)

$J_4 = 3$ (o cliente 4 tem o centro 3, como o centro mais próximo)

$J_5 = 3$ (o cliente 5 tem o centro 3, como o centro mais próximo)

$J_6 = 3$ (o cliente 6 tem o centro 3, como o centro mais próximo)

- **Passo 2:** Constrói-se uma matriz quadrada $M_{j,t}$, onde j representa os centros escolhidos ($j \in J$) e t representa os dias do período ($t = 1, \dots, T$). Cada posição da matriz é dada pela seguinte equação:

$$M_{j,t} = \sum_{\substack{i \\ j \in J_i}} \left[\frac{\sum_{k \in C_i} a_{k,t}}{|C_i|} \right] \quad (5.2)$$

onde

C_i : são as possíveis combinações de dias de visitas para o cliente i .

$|C_i|$: é o número total de combinações de visitas do cliente i . Ou seja, se o cliente i apresenta $C_i = (\{dia1\}$ ou $\{dia2\})$, então $|C_i| = 2$.

$a_{k,t} = 1$, se o dia t pertence à combinação k e 0 caso contrário.

Nesta etapa, verifica-se em cada posição da matriz $M_{j,t}$ quais são os clientes i que tem o centro j como um de seus centros mais próximos (já armazenados no vetor J_i) e se o cliente i pode ser visitado no dia t . O objetivo desta matriz é informar quantas vezes cada centro está sendo requisitado pelos clientes no dia t (veja o exemplo à partir da Tabela 5.1). Intuitivamente, quanto maior $M_{j,t}$ mais atraente será associar o dia t ao centro j . Exemplo:

Pelos passos anteriores, verifica-se que estamos trabalhando com um período

de dois dias ($T = 2$) e com seis clientes ($n = 6$). No entanto:

$J = \{2, 3\}$	$J_1 = 2$	$C_1 = \{dia1\}$	$ C_1 = 1$
	$J_2 = 3$	$C_2 = \{dia1\}$ ou $\{dia2\}$	$ C_2 = 2$
	$J_3 = 2$	$C_3 = \{dia1\}$ ou $\{dia2\}$	$ C_3 = 2$
	$J_4 = 3$	$C_4 = \{dia2\}$	$ C_4 = 1$
	$J_5 = 3$	$C_5 = \{dia1\}$ ou $\{dia2\}$	$ C_5 = 2$
	$J_6 = 3$	$C_6 = \{dia1\}$ ou $\{dia2\}$	$ C_6 = 2$

onde:

J : refere-se aos centros.

J_i : indica qual é o centro mais próximo do cliente i , $i = 1, 2, \dots, 6$

C_i : são as possíveis combinações de visitas para o cliente i , $i = 1, 2, \dots, 6$

$|C_i|$: é o número total de combinações de visitas do cliente i , $i = 1, 2, \dots, 6$

Tabela 5.1: Matriz $M_{j,t}$ (centros(j) \times dias(t)).

$M_{j,t}$	1	2
2	A	B
3	C	D

Utilizando a equação 5.2 obtém-se os seguintes valores para a matriz $M_{j,t}$:

Para $j = 2$ e $t = 1$, da equação 5.2 tem-se:

$$A = \left[\frac{(1)}{1} \right] + 0 + \left[\frac{(1+0)}{2} \right] + 0 + 0 + 0 = \frac{3}{2}$$

Para $j = 2$ e $t = 2$:

$$B = \left[\frac{(0)}{1} \right] + 0 + \left[\frac{(0+1)}{2} \right] + 0 + 0 + 0 = \frac{1}{2}$$

Para $j = 3$ e $t = 1$:

$$C = 0 + \left[\frac{(1+0)}{2} \right] + 0 + \left[\frac{(0)}{1} \right] + \left[\frac{(1+0)}{2} \right] + \left[\frac{(1+0)}{2} \right] = \frac{3}{2}$$

Para $j = 3$ e $t = 2$:

$$D = 0 + \left\lceil \frac{(0+1)}{2} \right\rceil + 0 + \left\lceil \frac{(1)}{1} \right\rceil + \left\lceil \frac{(0+1)}{2} \right\rceil + \left\lceil \frac{(0+1)}{2} \right\rceil = \frac{5}{2}$$

Observa-se que $\frac{5}{2}$ clientes têm o centro 3 como centro mais requisitado, podendo ser atendidos no dia 2, enquanto $\frac{3}{2}$ clientes têm o centro 2 como centro mais requisitado, podendo ser atendidos no dia 1 (veja Tabela 5.1). O maior valor na matriz representará o dia de cada centro, portanto o centro 2 é associado ao dia 1 e o centro 3 é associado ao dia 2.

- Atribuindo clientes aos dias:

Depois de associar os centros aos dias, o próximo passo é associar os clientes aos dias a partir dos centros. Para isto efetua-se o seguinte procedimento:

Seja $j(t)$, com $t = 1, 2$ o centro do dia (definido pelo **passo 2**), ou seja:

$j(1) = 2$, significa que o centro 2 refere-se ao dia 1.

$j(2) = 3$, significa que o centro 3 refere-se ao dia 2.

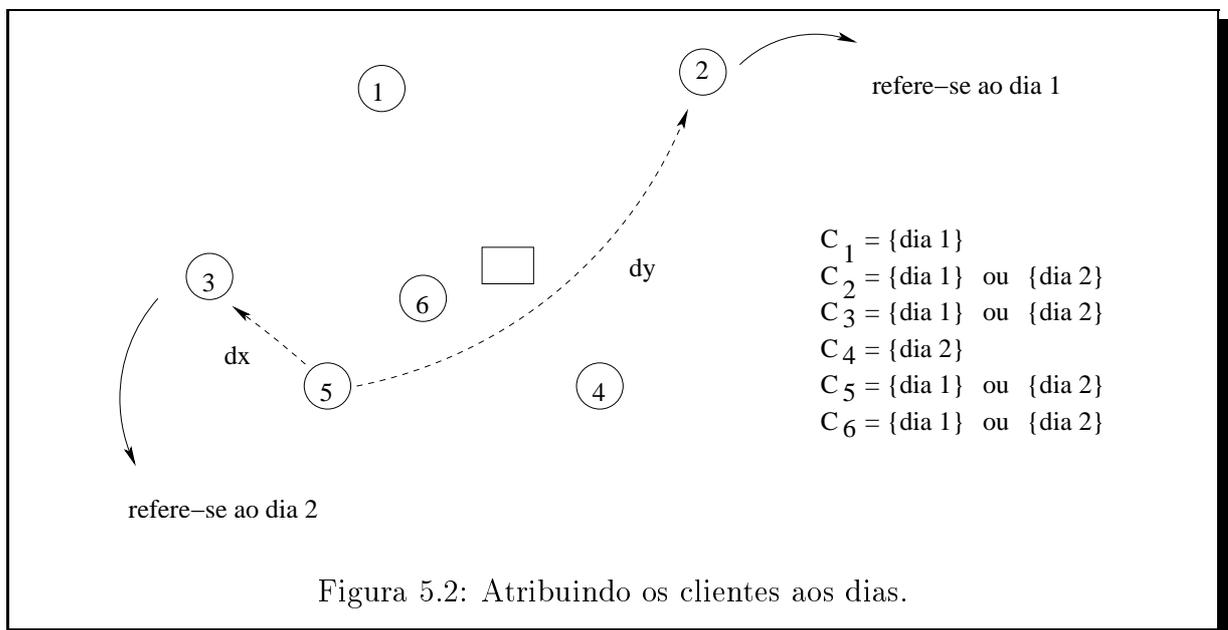
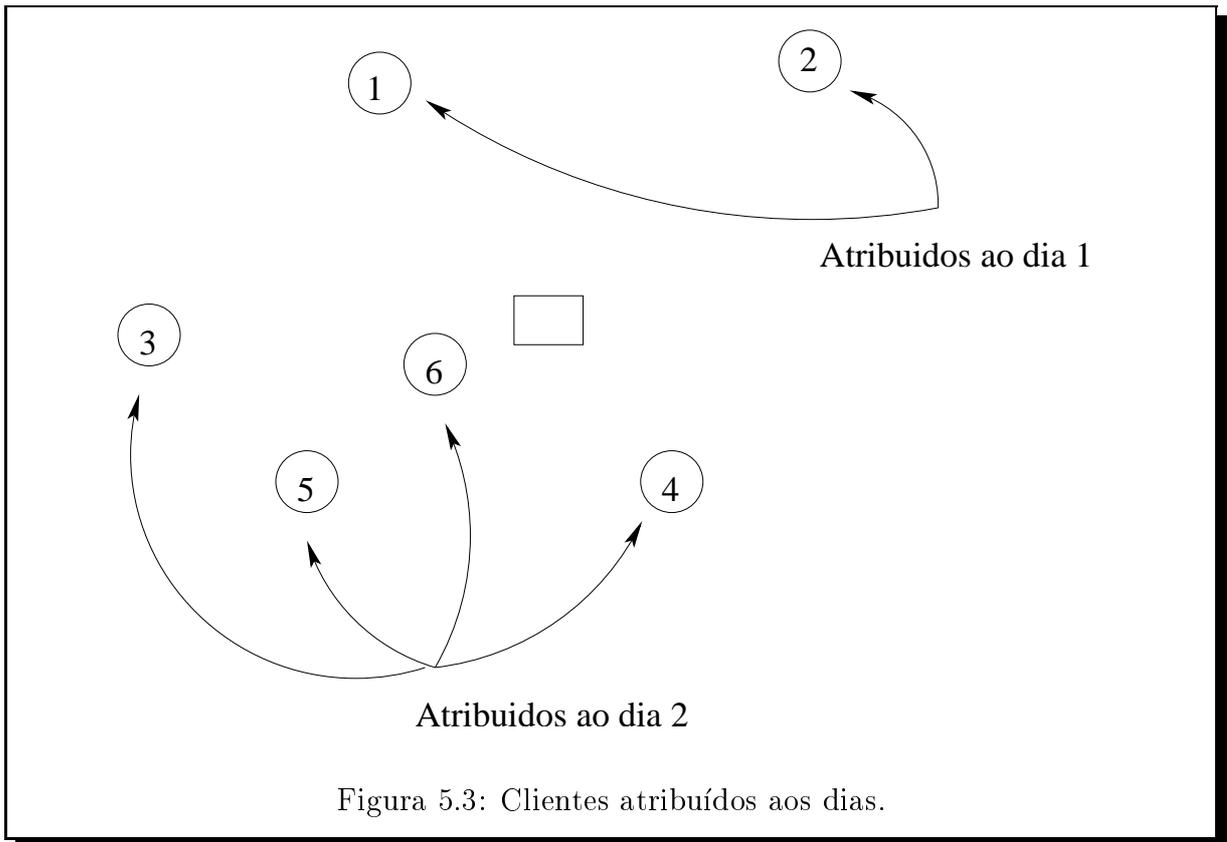


Figura 5.2: Atribuindo os clientes aos dias.

Acompanhando o esquema da Figura 5.2, percebe-se que obrigatoriamente o cliente 1 deve ser atendido no dia 1 (veja C_1) e o cliente 4 deve ser atendido no dia 2 (C_4). Já os clientes 2, 3, 5 e 6, podem ser atendidos tanto no dia 1 quanto no dia 2 (C_2 ,

C_3 , C_5 e C_6). A menor distância destes clientes aos centros 2 e 3, representará o dia a que eles serão atribuídos, conforme a equação 5.3. Por exemplo, observe que o cliente 5 pode ser atendido tanto no dia 1 quanto no dia 2 (C_5). Como a distância d_x é menor que a distância d_y atribui-se o cliente 5 ao dia 2, pois o centro 3 refere-se ao dia 2. Analogamente, é feita a designação do clientes 2, 3 e 6 aos dias (veja o resultado na Figura 5.3).



A determinação da distância (custo) é dada por:

$$\sum_{t=1}^T d_{i,j(t)} a_{k,t} , \quad \forall i, k \in C_i . \quad (5.3)$$

onde:

$d_{i,j(t)}$: distância do cliente i ao centro $j(t)$.

$a_{k,t} = 1$, se o dia t refere-se a combinação k e 0 caso contrário.

Nota-se que ao determinar o dia em que o cliente deve ser designado, verifica-se se

a capacidade do dia suporta tal designação, em caso afirmativo atribui-se o cliente ao dia, caso contrário, o cliente é designado aleatoriamente para outro dia.

- Roteando os clientes em cada dia:

Aplica-se o algoritmo HTA a cada dia t para gerar a solução inicial (S1). Por exemplo, considerando os clientes na Figura 5.4, observa-se que a rota do dia 1 é formada pelos clientes 1 e 2 (pois foram atribuídos ao dia 1, veja Figura 5.3) e a rota do dia 2 é formada pelos clientes 3, 4, 5 e 6, cujo o percurso também é mostrado na Figura 5.4.

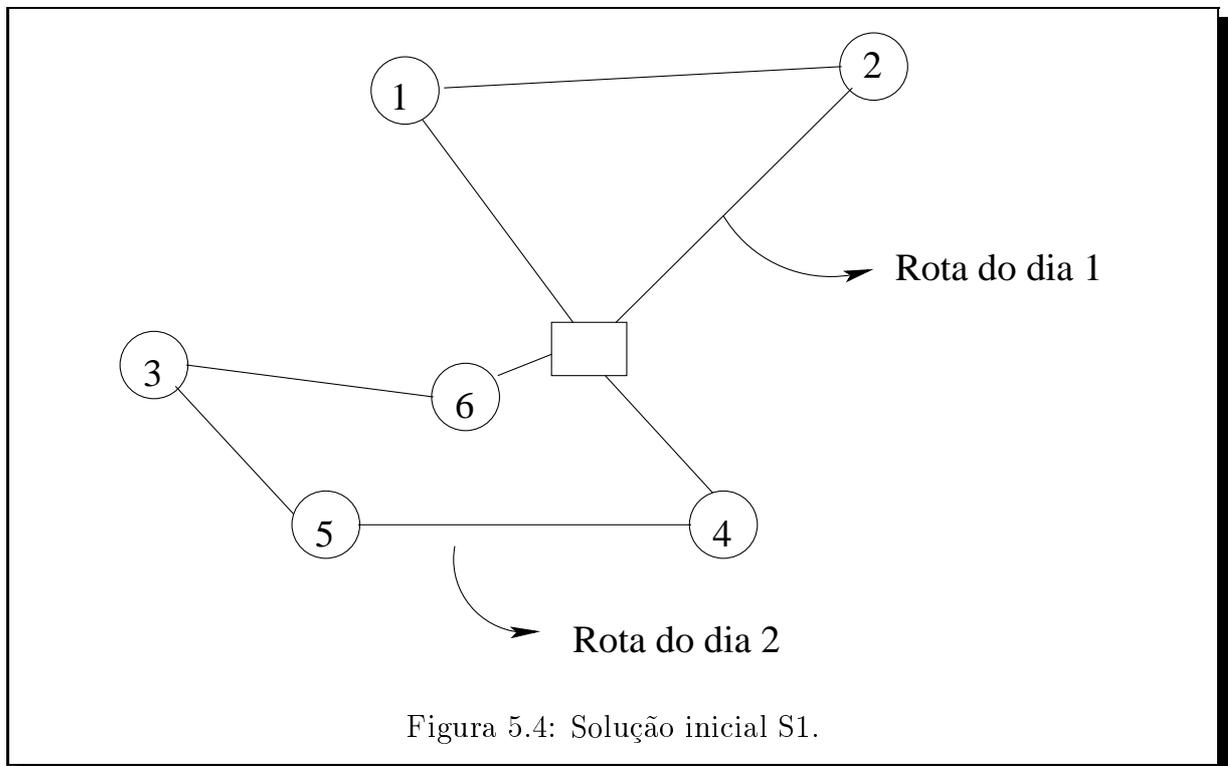


Figura 5.4: Solução inicial S1.

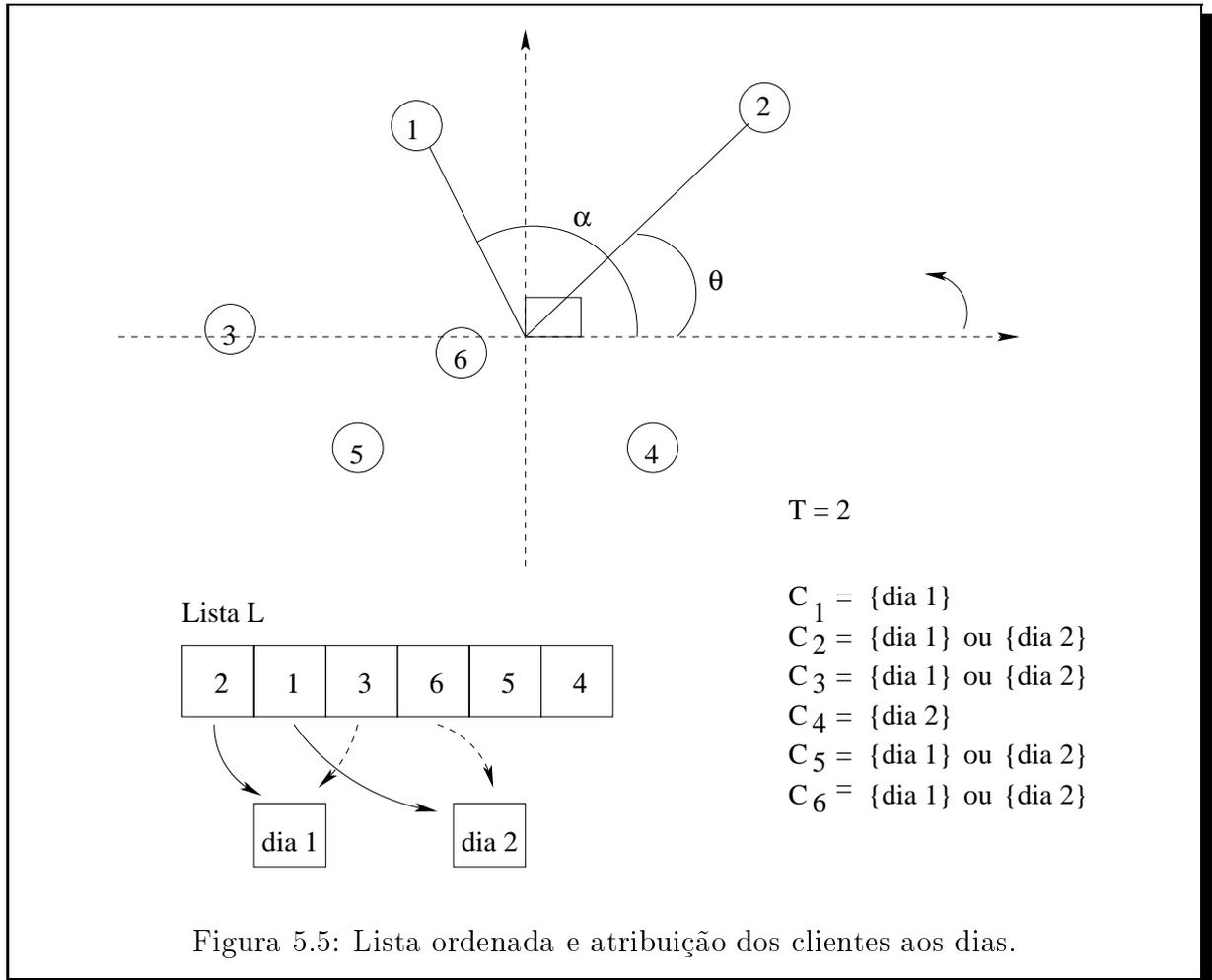
SEÇÃO 5.2.2

Construção inicial 2 (CI-2)

Para exemplificar este procedimento é utilizado o mesmo problema da seção anterior, ou seja: a mesma distribuição geográfica, o mesmo número de clientes, as mesmas

combinações de visitas e um período de dois dias.

Nessa construção os clientes são designados de acordo com uma adaptação do algoritmo de varredura de GILLET & MILLER (1974). Em uma lista L , os clientes são ordenados em ordem crescente de ângulo que formam com o depósito (ver Figura 5.5).

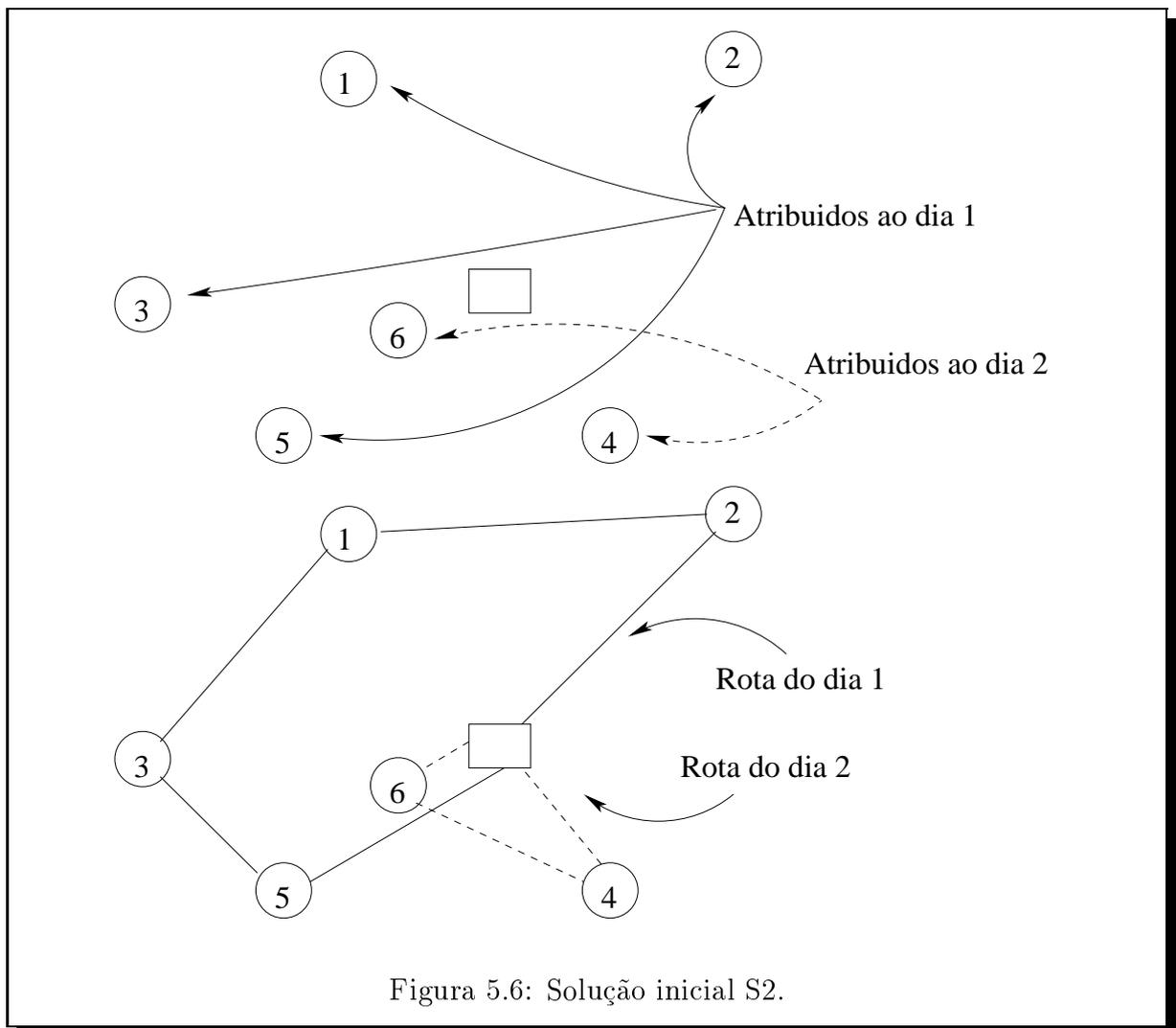


Para atribuí-los aos dias realiza-se os seguintes passos:

- Toma-se o primeiro cliente da lista ordenada L e, se possível, este é atribuído a uma combinação de visita que envolva o dia 1. Toma-se o segundo cliente da lista e se possível, este é atribuído a uma combinação de visita que envolva o dia 2, e assim sucessivamente, até o último dia (de acordo com a lista ordenada da Figura 5.5). Neste ponto, o próximo cliente da lista ordenada é atribuído ao dia 1 novamente, e assim sucessivamente. Caso o cliente i não possua a combinação de visita referente

ao dia, o atribuímos aleatoriamente a um dia permitido, até que todos os clientes tenham sido designados.

Note que a atribuição dos clientes é baseada num critério arbitrário. Outros critérios como a atratividades do cliente a um dia particular (conforme utilizado em CRISTOFIDES & BEASLEY, 1984), poderiam ter sido empregados e, provavelmente, teriam um impacto mais positivo nos resultados.



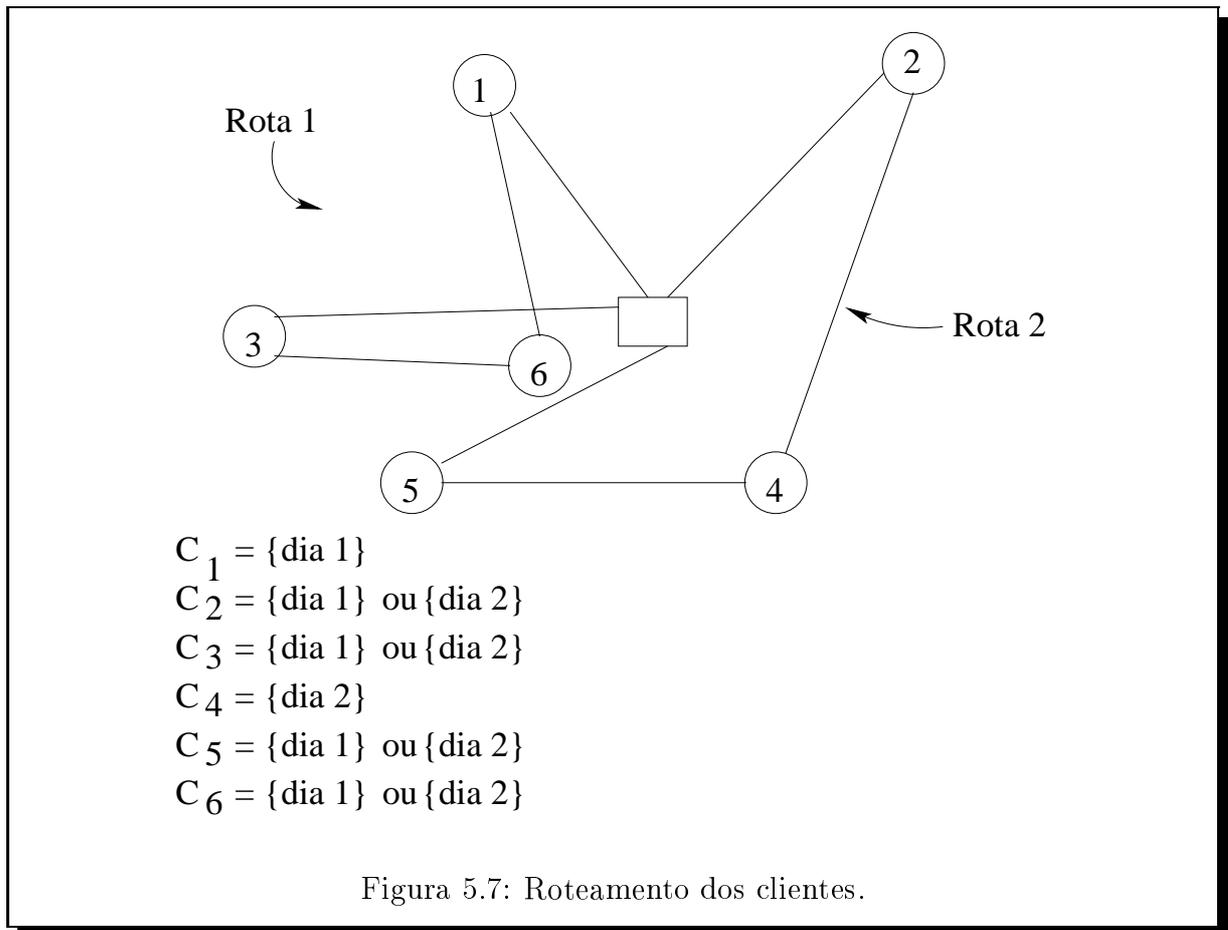
- Roteando os clientes em cada dia:

Depois de atribuir os clientes aos dias, aplica-se o algoritmo HTA a cada dia t , obtendo a solução inicial (S2) mostrada na Figura 5.6:

SEÇÃO 5.2.3

Construção inicial 3 (CI-3)

O exemplo apresentado nesta seção mantém a mesma distribuição geográfica dos clientes, o mesmo número de clientes, as mesmas combinações de visitas e um período de dois dias (conforme apresentado nas seções anteriores).



Foi elaborado o seguinte procedimento:

1. Inicialmente, atribui-se todos os clientes ao mesmo dia. Nesta etapa ainda não é especificada uma combinação de visita para cada cliente.
2. Aplica-se o algoritmo HTA, obtendo-se um conjunto de rotas como se todos os clientes fossem atendidos no mesmo dia.
3. Atribui-se, então, cada uma das rotas formadas aos T dias do período.

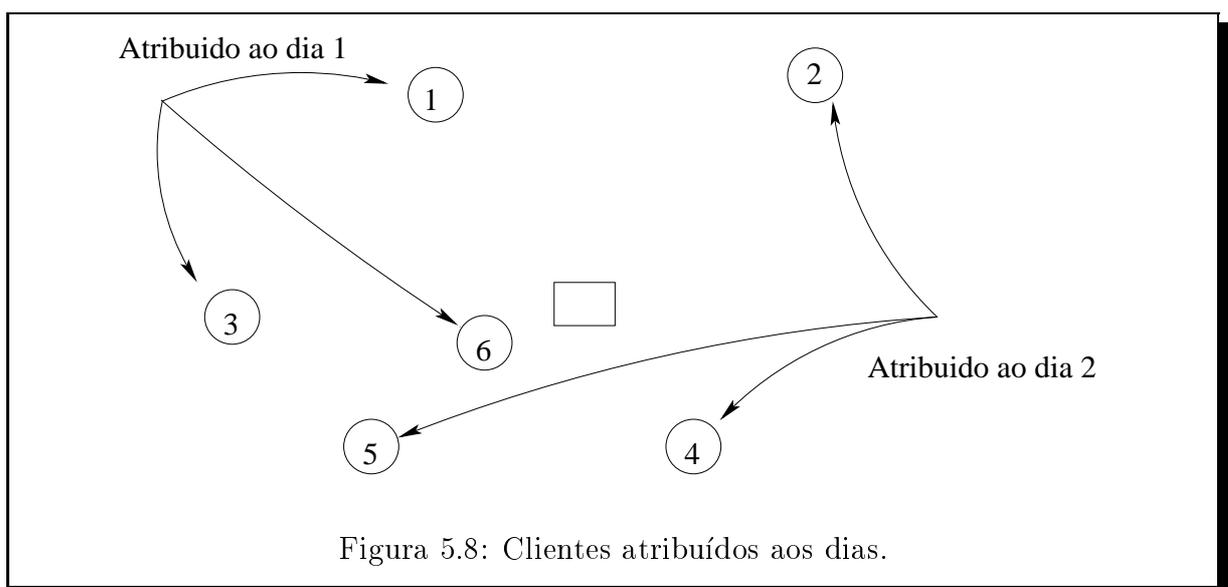
4. Aplica-se o algoritmo HTA a cada dia t para gerar a solução inicial.

- Para exemplificar a atribuição das rotas iniciais aos T dias do período (passo 3) considere as explicações a seguir. Depois da aplicação do HTA (passo 2), obtém-se um conjunto de rotas desconsiderando restrições de combinações de dias de visita e, portanto, monoperiódico. Para atribuir as rotas aos dias, atribui-se cada cliente da **rota 1** ao **dia 1** (se possível) e cada cliente da **rota 2** ao **dia 2** (se possível). Com isso, especifica-se o dia em que o cliente será visitado, atribuindo-lhe a respectiva combinação. Por exemplo:

(a) Analisa-se se os clientes 1, 3 e 6 da rota 1 (Figura 5.7) podem ser atendidos no dia 1.

- Verifica-se na combinação C_1 (Figura 5.7) se o cliente 1 pode ser atendido no dia 1. Se for possível escolhe-se a respectiva combinação, caso contrário outra combinação deve ser escolhida aleatoriamente. O mesmo é repetido para os demais clientes.

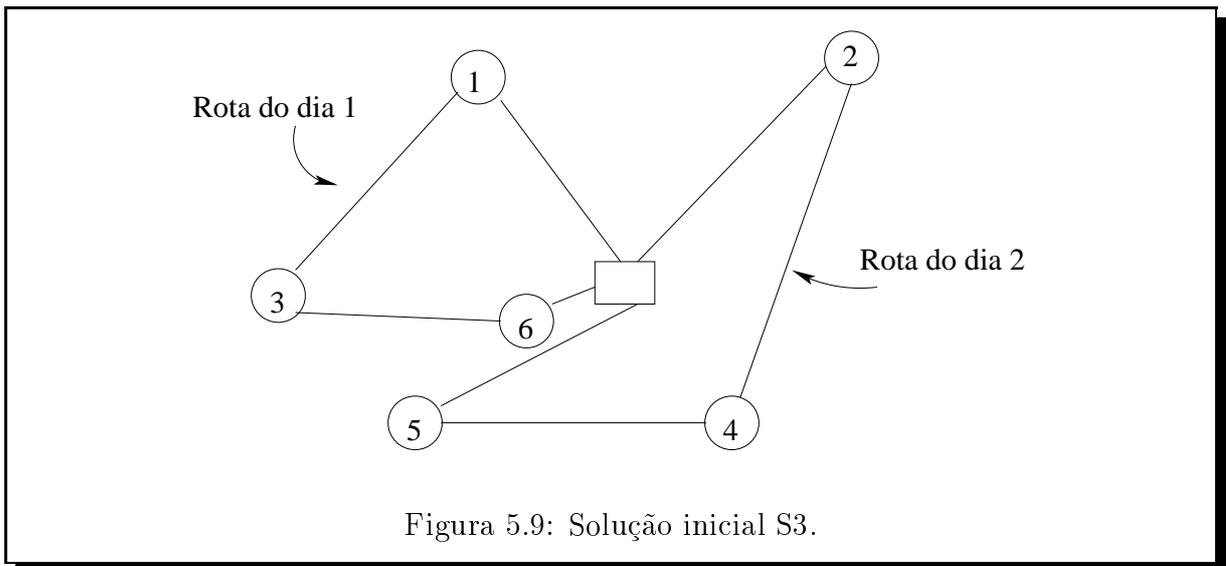
(b) Analisa-se de forma análoga se os clientes 2, 4 e 5 da rota 2, podem ser atendidos no dia 2. Caso o número de rotas geradas na Figura 5.7 for maior que o tamanho do período (no exemplo $T = 2$), atribua a rota excedente ao dia 1 e repita o processo do passo (a).



Ao final deste processo, têm-se os clientes 1, 3, e 6 atribuídos ao dia 1 e os clientes 2, 4 e 5 atribuídos ao dia 2 conforme apresentado na Figura 5.8.

- Roteando os clientes em cada dia:

Aplica-se o algoritmo HTA a cada dia t para gerar a solução inicial (S3), veja como exemplo a Figura 5.9.



SEÇÃO 5.3

Movimentos de melhorias

SEÇÃO 5.3.1

Troca e inserção

Para melhorar a solução inicial, realiza-se movimentos de troca de combinações de dias de visitas para os clientes, efetivando aquela que resulte em uma solução melhor. A heurística baseia-se em CHRISTOFIDES & BEASLEY (1984). Esta heurística foi utilizada porque realiza tanto movimentos de troca de clientes entre dias, quanto movimentos de inserção. Após a aplicação de cada movimento, o algoritmo HTA é aplicado para formar as rotas dos dias envolvidos.

SEÇÃO 5.3.1.1

Definindo a família U de subconjuntos de clientes

Na heurística de CHRISTOFIDES & BEASLEY (1984), pequenos subconjuntos de clientes são escolhidos e todas as possíveis combinações de dias de visitas para cada subconjunto são enumeradas. Forma-se assim uma família U de subconjuntos de clientes.

Cada subconjunto de clientes da família U é a união de dois outros conjuntos de clientes ($M1$ e $M2$). $M1$ é um conjunto contendo $T1$ clientes que aparecem consecutivamente na rota de algum dia t , e $M2$ é um conjunto contendo $T2$ clientes geograficamente próximos do conjunto $M1$.

- *Escolhendo o conjunto $M1$:*

Considera-se um cliente i de uma rota de algum dia t e os $L1$ clientes que sucedem i na rota. O cliente i e os $L1$ clientes constituem o conjunto $M1$. Logo, $T1 = (i + L1)$, compõe o conjunto $M1 = \{T1\}$.

- *Escolhendo o conjunto $M2$:*

Sejam α e β , respectivamente, o cliente que precede e o cliente que sucede o conjunto $M1$ na rota do dia t e sejam suas combinações atuais de visita $K\alpha^*$ e $K\beta^*$. Gera-se uma lista L de todos os clientes $j \notin M1$, ordenados em:

1. ordem crescente do número de dias de visitas de j (combinação atual de $j = k_j^*$) que coincida com o dia de visita de α e β , isto é, para cada j :

$$\sum_{t=1}^T a_{K_j t}^* \text{Min} \left\{ 1, a_{K_\alpha t}^* + a_{K_\beta t}^* \right\} \quad (5.4)$$

onde

$a_{K_j t}^* = 1$, se a combinação de visita atual k do cliente j pertence ao dia t e 0 caso contrário.

$a_{K_\alpha t}^* = 1$, se a combinação de visita atual k do cliente α pertence ao dia t e 0 caso contrário.

$a_{K\beta t}^* = 1$, se a combinação de visita atual k do cliente β pertence ao dia t e 0 caso contrário.

T : número de dias no período de planejamento.

2. no caso de empate (dos resultados obtidos pela utilização da equação 5.4), prioriza-se os clientes com menor distância em relação ao cliente α e ao cliente β :

$$d_{j,\alpha} + d_{j,\beta}$$

onde

j : clientes $j \notin M1$.

$d_{j,\alpha}$: distância do cliente j ao cliente α .

$d_{j,\beta}$: distância do cliente j ao cliente β .

Os primeiros $L2$ clientes desta lista L ordenada constituem o conjunto $M2$. A união do conjunto $M1$ com o conjunto $M2$ ($M1 \cup M2$) constitui o primeiro subconjunto de clientes da família U . A família toda é gerada considerando todos os dias t e todos os clientes i na rota para cada dia t .

Utilizando a solução inicial **S3**, ilustrar-se-á um exemplo conforme a Figura 5.10. Seja $T1 = 1$ e $T2 = 1$. Neste exemplo, será considerado apenas um cliente no conjunto $M1$ e apenas um cliente no conjunto $M2$. Logo:

- O cliente 3 (designado ao dia 1) pertencente ao conjunto $M1$. Logo o conjunto $M1 = \{3\}$.
- O cliente 1 (designado ao dia 1) é o cliente que precede o cliente 3. Logo $\alpha = 1$.
- O cliente 6 (designado ao dia 1) é o cliente que sucede o cliente 3. Logo $\beta = 6$.

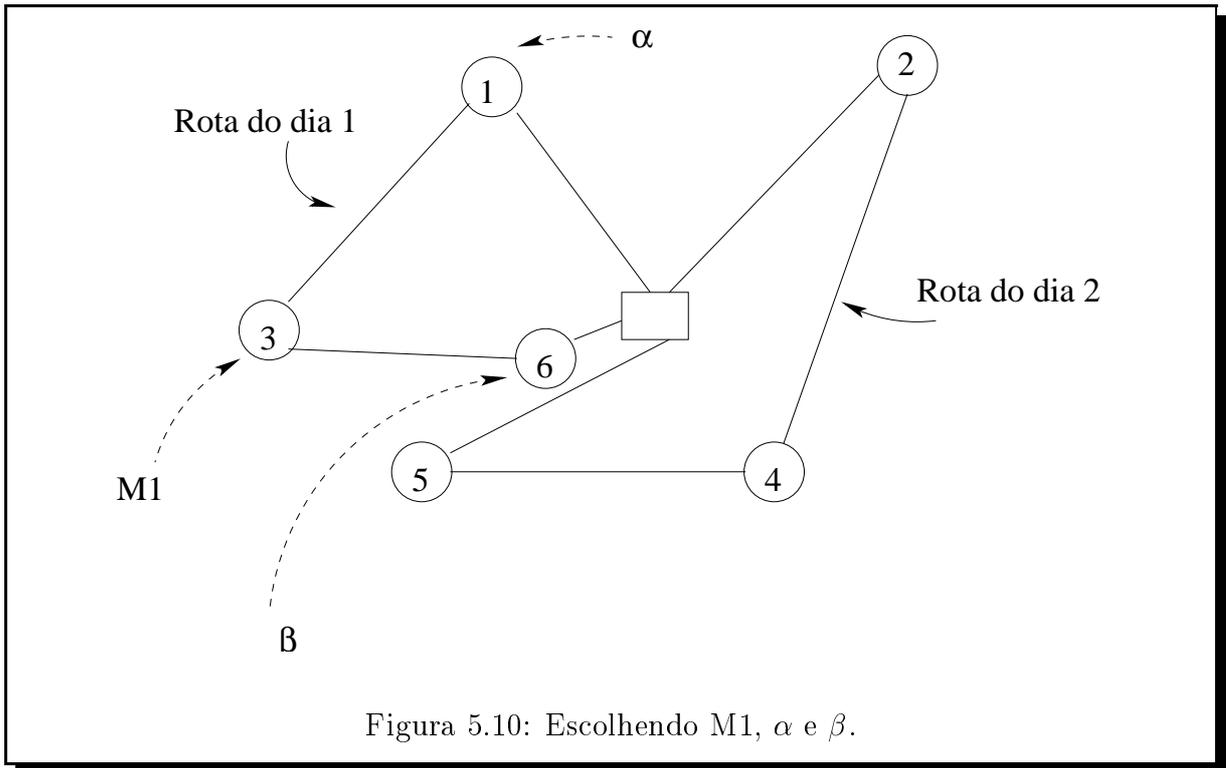


Figura 5.10: Escolhendo M1, α e β .

O objetivo neste exemplo é escolher qual cliente fará parte do conjunto $M2$, utilizando a equação 5.4. Assim, seguindo as explicações anteriormente dadas, primeiramente gera-se uma lista $L = \{j_1, j_2, \dots, j_n\}$ de todos os clientes $j \notin M1$. Como $M1 = \{3\}$, a lista de clientes $L = \{1, 2, 4, 5, 6\}$.

Pela equação 5.4 obtemos os seguintes resultados:

Para $t = 1, 2$ e $j_1 = 1$

$$(1 \times \text{Min}\{1, 1 + 1\}) + (0 \times \text{Min}\{1, 0 + 0\}) = 1$$

Para $t = 1, 2$ e $j_2 = 2$

$$(0 \times \text{Min}\{1, 1 + 1\}) + (1 \times \text{Min}\{1, 0 + 0\}) = 0$$

Para $t = 1, 2$ e $j_3 = 4$

$$(0 \times \text{Min}\{1, 1 + 1\}) + (1 \times \text{Min}\{1, 0 + 0\}) = 0$$

Para $t = 1, 2$ e $j_4 = 5$

$$(0 \times \text{Min} \{1, 1 + 1\}) + (1 \times \text{Min} \{1, 0 + 0\}) = 0$$

Para $t = 1, 2$ e $j_5 = 6$

$$(1 \times \text{Min} \{1, 1 + 1\}) + (0 \times \text{Min} \{1, 0 + 0\}) = 1$$

Ordena-se a lista L em ordem crescente, a partir dos resultados gerados pela equação 5.4. Para resolver o problema de empate entre os clientes 5, 4 e 2 (uma vez que os resultados são 0) e dos clientes 6 e 1 (uma vez que os resultados são 1), utiliza-se o critério de desempate pela menor distância ($d_{j,\alpha} + d_{j,\beta}$), conforme apresentado na Figura 5.11.

Analisando a distância dos clientes 5, 4 e 2 (ver Figura 5.11) em relação aos clientes α e β , têm-se $d_1 + d_2 < d_3 + d_4 < d_5 + d_6$. Assim, coloca-se o cliente 5 na primeira posição da lista L , seguido do cliente 4, seguido do cliente 2.

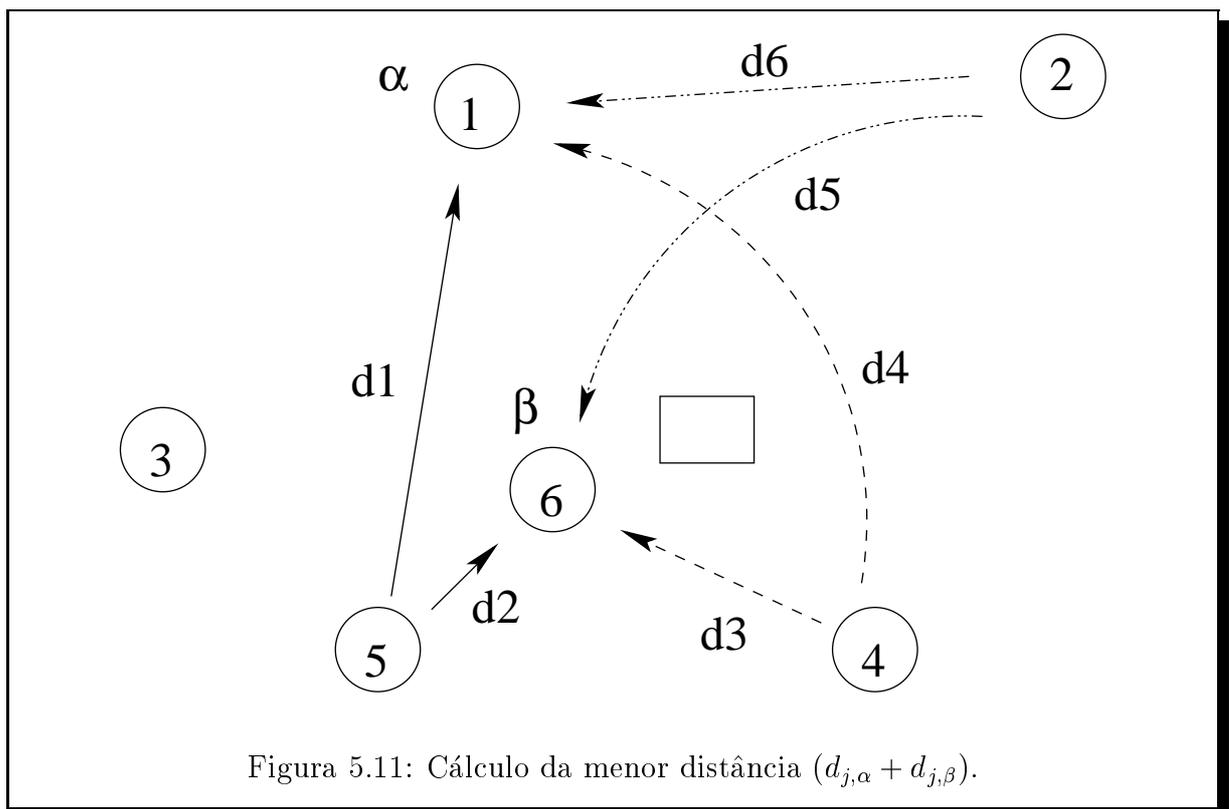


Figura 5.11: Cálculo da menor distância ($d_{j,\alpha} + d_{j,\beta}$).

Procede-se de forma análoga para o cliente 1 e 6 com relação aos clientes α e β . Caso haja empate na utilização do critério da menor distância ($d_{j,\alpha} + d_{j,\beta}$), os clientes

são escolhidos aleatoriamente para serem posicionados na lista L . Obtém-se a seguinte lista L ordenada de clientes:

$$L = \{5, 4, 2, 6, 1\}$$

$$\{0, 0, 0, 1, 1\} \rightarrow \text{Resultado da equação 5.4}$$

Como se deseja apenas um cliente no conjunto $M2$ ($T2 = 1$), toma-se o primeiro cliente da lista L ordenada. Logo $M2 = \{5\}$. Assim, o primeiro subconjunto da família U é $M1 \cup M2 = \{3, 5\}$. Para este subconjunto são enumeradas todas as possíveis combinações de visitas:

$C_3 = \{\text{dia1}\} \text{ ou } \{\text{dia2}\}$ (possíveis combinações de visitas para o cliente 3).

$C_5 = \{\text{dia1}\} \text{ ou } \{\text{dia2}\}$ (possíveis combinações de visitas para o cliente 5).

Todos os possíveis movimentos de combinações de dias de visitas para estes clientes são investigados com relação a solução inicial (ver Figuras 5.12).

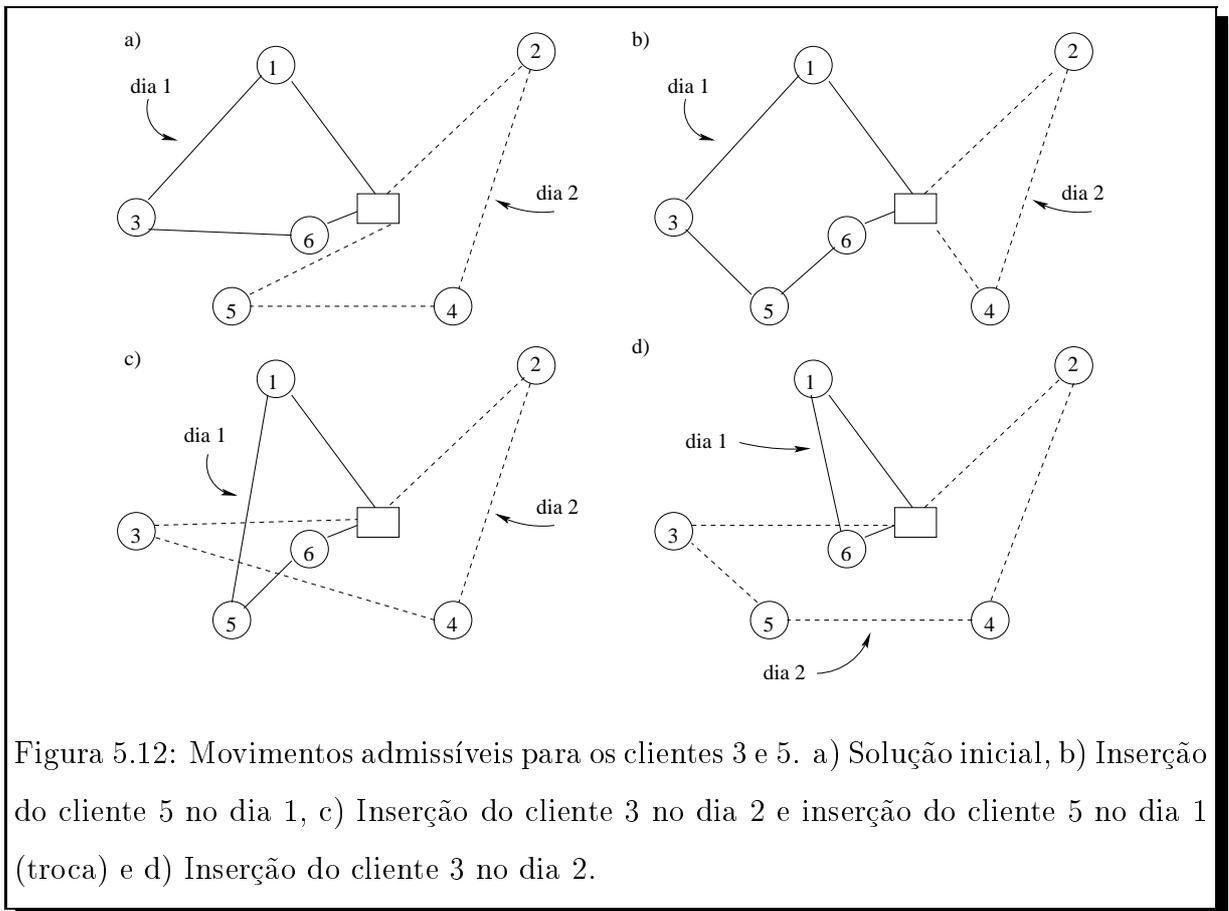


Figura 5.12: Movimentos admissíveis para os clientes 3 e 5. a) Solução inicial, b) Inserção do cliente 5 no dia 1, c) Inserção do cliente 3 no dia 2 e inserção do cliente 5 no dia 1 (troca) e d) Inserção do cliente 3 no dia 2.

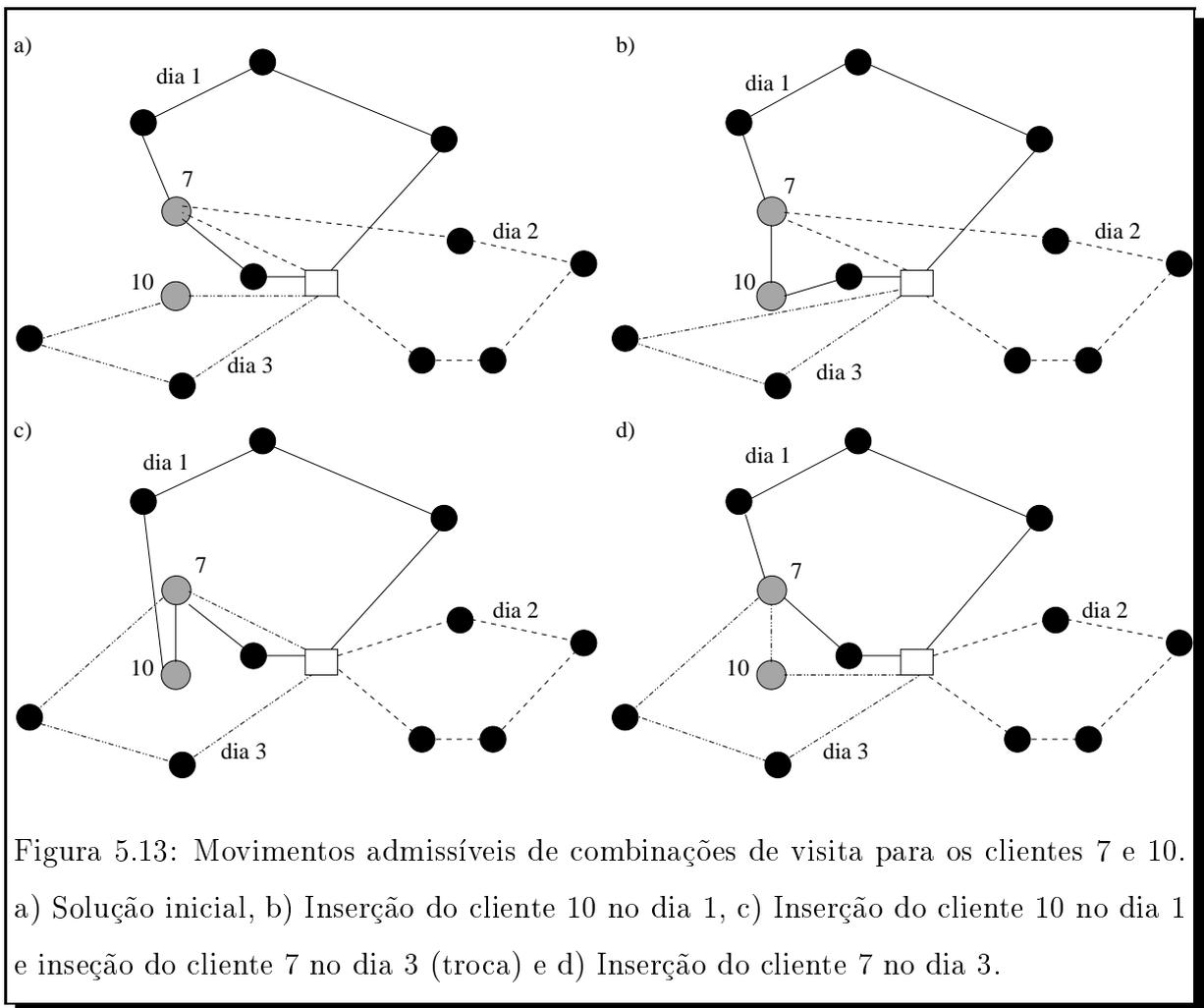
No caso que um destes movimentos resultar em uma solução melhor que a solução corrente, este é efetuado. Repete-se o procedimento para o próximo subconjunto da família U . Uma família U completa é gerada considerando todos os dias T e todos os clientes i da rota do dia t .

Generalizando o caso para clientes com maior frequência de atendimento:

O exemplo a seguir, ilustrará os *movimentos de melhoria* com clientes que apresentam frequências de atendimento maiores. Para isto, considera-se:

- período de planejamento de três dias ($T = 3$);
- o conjunto $M1$ possua apenas um cliente, no caso $M1 = \{7\}$;
- o conjunto $M2$ possua apenas um cliente, no caso $M2 = \{10\}$;
- que o subconjunto $M1 \cup M2 = \{7, 10\}$ constituem os clientes utilizados para a troca (cuja as combinações de visita atuais serão modificadas);
- as possíveis combinações de visitas para o cliente 7 são: $C_7 = (\{dia1, dia2\}$ ou $\{dia1, dia3\})$ (duas visitas no período);
- as possíveis combinações de visitas para o cliente 10 são: $C_{10} = (\{dia1\}$ ou $\{dia3\})$ (uma visita no período).

A Figura 5.13 mostra todos os possíveis movimentos de combinações de visitas entre os clientes 7 e 10 com relação a solução inicial.



SEÇÃO 5.3.2

Movimento *clean up*

Após todos os movimentos de troca e inserção, aplica-se o movimento *clean up* proposto por CHAO et al. (1995). Especificamente, *considera-se mudanças de combinação de visitas dos clientes que requerem somente uma visita no período* (ou seja, clientes com frequência de visita igual a um).

Primeiramente, estes clientes são armazenados em uma lista W . Toma-se o primeiro cliente de W e verifica-se se o custo de atribuí-lo ao dia d é menor que o

custo da solução corrente. Se o custo for menor, o cliente é atribuído para o dia d ; caso contrário este é mantido no mesmo dia. O procedimento é aplicado para todos os clientes da lista. Após a efetivação de um movimento, o algoritmo HTA é aplicado às rotas dos dias envolvidos.

SEÇÃO 5.4

Buscando melhores soluções

Até a seção 5.3 apresentou-se a VERSÃO 1 deste trabalho, com resultados relativamente promissores (ver Tabela 6.3). Na tentativa de melhorar a solução e o tempo computacional do algoritmo foi elaborada uma estratégia similar a HTA, que utiliza os custos das soluções globais (isto é, custo total do roteamento) para análise de trajetórias. O algoritmo resultante constitui a VERSÃO 2 deste trabalho.

A VERSÃO 2 utiliza uma lista LC , que armazena o histórico dos custos das últimas it iterações. Estes custos são analisados para determinar o padrão da trajetória em *estagnada*, *ascendente* ou *descendente*. Isso é feito aproximando os dados contidos na lista LC por uma reta e o ângulo desta reta identificará a trajetória. Para determinar essa reta foi utilizado o Método dos Mínimos Quadrados (RUGGIERO & LOPES, 1988). Maiores detalhes da abordagem serão vistos nas próximas seções.

SEÇÃO 5.4.1

Analisando as soluções

Cada subconjunto de clientes $M1 \cup M2$ (visto na seção 5.3.1.1), constitui uma iteração (it) neste algoritmo. Todas as possíveis combinações de visitas envolvendo os clientes de cada subconjunto $M1 \cup M2$ são geradas, enquanto os clientes restantes mantêm suas posições nas rotas. Para cada combinação i de visitas obtém-se um custo CS_i ,

correspondente ao valor da função objetivo, ou seja, o custo total do roteamento no período com a utilização da combinação i . Cada custo CS_i é comparada a um custo intermediário CS_{interm} que pode vir a ser atualizado. Esse custo intermediário (e suas atualizações) é utilizado para identificar o padrão da trajetória de busca. CS_{interm} inicialmente recebe o valor do custo do roteamento da solução inicial e o custo CS^* armazena o custo da solução factível que minimiza o roteamento.

Para cada tipo de trajetória identificada (após a análise da trajetória), atualiza-se o custo intermediário de acordo com as seguintes regras:

- *Se a trajetória for estagnada:*

O algoritmo permite apenas soluções de piora, como forma de diversificar a busca, desde que estas soluções estejam $x\%$ piores que o custo intermediário CS_{interm} . Ou seja, se um dos custos CS_i gerados por mudanças de combinações de visitas entre os clientes for $x\%$ pior que o custo intermediário CS_{interm} , atualiza-se o CS_{interm} . Portanto se $CS_i > CS_{interm}$, respeitando o limite de piora, $CS_{interm} = CS_i$. Caso alguns destes custos CS_i seja menor que a solução CS^* (melhor solução factível encontrada) e a solução associada a CS_i for factível, esta regra é desconsiderada aceitando-se a solução. Assim, caso $CS_i < CS^*$ então $CS_{interm} = CS_i$ e $CS^* = CS_{interm}$.

- *Se a trajetória for ascendente:*

Neste caso, o algoritmo permite apenas soluções melhores que a solução intermediária CS_{interm} . Portanto, se um dos custos CS_i gerados pelas mudanças de combinações de visitas entre os clientes for menor que o custo CS_{interm} , aceita-se a solução. Ou seja, se $CS_i < CS_{interm}$, então $CS_{interm} = CS_i$ e se $CS_{interm} < CS^*$ e a solução associada a CS_i for factível, então $CS^* = CS_{interm}$. Nesse caso objetiva-se intensificar a busca para mudar o padrão da trajetória para descendente.

- *Se a trajetória for descendente:*

Neste caso, como a busca está percorrendo uma região promissora, tenta-se explorar esta região ao máximo (intensificação), aceitando-se apenas soluções de boa qualidade. Portanto, se $CS_i < CS_{interm}$, então $CS_{interm} = CS_i$ e se $CS_{interm} < CS^*$ e a

solução associada a CS_i for factível, então $CS^* = CS_{interm}$. Note que o tratamento é o mesmo dado quando se verifica trajetória ascendente.

A cada iteração it armazena-se numa lista LC os custos intermediários encontrados, os quais são posteriormente analisados para determinação do padrão da trajetória dos custos.

Para exemplificar o processo acima, supõe-se que na iteração it o subconjunto de clientes escolhido seja $M1 \cup M2 = \{3, 5\}$. Supõe-se também que está trabalhando com um período de dois dias ($T = 2$) e que as combinações de visitas para os clientes 3 e 5 sejam dadas por:

$$C_3 = (\{dia1\} \text{ ou } \{dia2\})$$

$$C_5 = (\{dia1\} \text{ ou } \{dia2\})$$

Para cada par de elementos de C_3 e C_5 obtém-se um custo CS_i , conforme mostrado na Tabela 5.2

Tabela 5.2: Possíveis pares de combinações de visitas.

C_3	C_5	Custo CS_i
Dia 1	Dia 1	CS_1
Dia 2	Dia 1	CS_2
Dia 1	Dia 2	CS_3
Dia 2	Dia 2	CS_4

Observa-se que ao designar o cliente 3 ao dia 1 e o cliente 5 ao dia 1 obtém-se uma solução com custo CS_1 . Analogamente, ao designar o cliente 3 ao dia 2 e o cliente 5 ao dia 1 obtém-se uma solução com custo CS_2 e assim sucessivamente. Cada custo CS_i (no caso, $i = 1, 2, 3$ e 4) é comparado com o custo intermediário CS_{interm} atual. A medida que o processo de busca se desenvolve, o valor de CS_{interm} é atualizado da seguinte forma:

1. Se a *trajetória for estagnada*, o algoritmo permite apenas soluções de piora, com o objetivo de diversificar a busca. Soluções possíveis de escolha são aquelas com

custos até $x\%$ piores com relação ao custo CS_{interm} .

Para exemplificar, suponha que $CS_{interm} = 580$, $x = 2\%$, $CS^* = 570$, $CS_1 = 590$, $CS_2 = 600$, $CS_3 = 596$ e $CS_4 = 568$. Portanto, realiza-se as seguintes análises:

- Analisando o custo CS_1 :

Como $CS_1 > CS_{interm}$ e como $CS_1 < CS_{interm}(1+x) = 591.6$, então atualiza-se $CS_{interm} = CS_1 = 590$ (observe que o custo intermediário aumentou).

- Analisando o custo CS_2 :

No momento, $CS_{interm} = 590$. Como $CS_2 > CS_{interm}$ e como $CS_2 < CS_{interm}(1+x) = 601.8$ logo $CS_{interm} = CS_2 = 600$.

- Analisando o custo CS_3 :

No momento $CS_{interm} = 600$. Como $CS_3 < CS_{interm}$ e além disso $CS_3 > CS^*$, não se atualiza CS_{interm} .

- Analisando o custo CS_4 :

Como CS_{interm} não foi atualizado no passo anterior, $CS_{interm} = 600$. Como $CS_4 < CS_{interm}$ e $CS_4 < CS^*$, e supondo que a solução que gera CS_4 é factível, atualiza-se $CS_{interm} = CS_4 = 568$ e além disso, $CS^* = CS_{interm} = 568$. Observa-se que neste caso houve uma exceção à regra, permitindo-se a geração de uma solução de melhor qualidade.

Obseva-se que o valor do custo intermediário (quando o padrão da trajetória é estagnado) é atualizado somente com movimentos de piora com o objetivo de diversificar a busca, a não ser que o custo CS_i seja menor que o custo da melhor solução CS^* (custo CS_4).

2. Se a *trajetória for ascendente*, o algoritmo permite apenas soluções de melhoria. Por exemplo, suponha que $CS_{interm} = 520$, $CS^* = 512$, $CS_1 = 530$, $CS_2 = 525$, $CS_3 = 516$ e $CS_4 = 510$. Assim têm-se:

- Analisando o custo CS_1 :

Como $CS_1 > CS_{interm}$, não ocorre a atualização de CS_{interm} .

- Analisando o custo CS_2 :

No momento, $CS_{interm} = 520$. Como $CS_2 > CS_{interm}$, não ocorre a atualização de CS_{interm} .

- Analisando o custo CS_3 :

No momento, $CS_{interm} = 520$. Como $CS_3 < CS_{interm}$ então $CS_{interm} = CS_3 = 516$.

- Analisando o custo CS_4 :

No momento, $CS_{interm} = 516$. Supondo que o movimento que gera CS_4 é factível e como $CS_4 < CS_{interm}$ e $CS_4 < CS^*$ então $CS_{interm} = CS_4 = 510$ e $CS^* = CS_{interm} = 510$.

3. Se a trajetória for descendente, o algoritmo permite apenas soluções de melhoria, para intensificar a busca, de forma análoga ao exemplo anterior.

Os procedimentos descritos acima permitem soluções infactíveis, com o objetivo de permitir uma maior flexibilidade na busca. Essa infactibilidade corresponde a uma relaxação na carga máxima do dia, permitindo-se uma carga de até $\zeta\%$ acima da carga do dia. A carga máxima do dia é calculada multiplicando-se o número máximo de veículos por dia no período pela capacidade máxima do veículo.

A cada iteração it armazena-se o custo CS_{interm} em uma lista LC de tamanho Tm variável. Quando a lista estiver preenchida, o próximo custo CS_{interm} é guardado a partir do início da lista, de forma que os novos custos sobrescreverão os valores mais antigos armazenados (lista cíclica).

A análise da trajetória com os custos armazenados em LC é feito na iteração Av (instante de avaliação das soluções globais).

Em seguida, reinicializa-se $it = 0$ e continua-se com o processo de busca. Como critério de parada utiliza-se duas reinicializações, ou seja, aplica-se novamente os movimentos de melhoria sobre a melhor solução encontrada. Na próxima seção será descrito detalhadamente o instante de avaliação das soluções globais.

SEÇÃO 5.4.2

Instante de avaliação das soluções globais (Av)

Quando $it = Av$, deve-se realizar a identificação do padrão da trajetória. Para identificar o padrão da trajetória utiliza-se o método de interpolação linear conhecido como Método dos Mínimos Quadrados, que ajusta a reta, que mais se aproxima dos custos armazenados na lista LC , ver Figura 5.14. Nessa figura $f(x_k)$ corresponde ao custo C_{interm} que está armazenado na posição x_k da lista.

O valor do ângulo θ da reta ajustada identifica a trajetória como *ascendente*, *descendente* ou *estagnada*. Considera-se que a trajetória é ascendente se $\theta > \epsilon$, descendente se $\theta < -\epsilon$ e estagnada se $\theta \leq |\epsilon|$, onde ϵ é um parâmetro exógeno.

Por exemplo, suponha que os custos armazenados em LC sejam distribuídos conforme a Figura 5.14a. Nessa figura, o ângulo de inclinação é $\theta = 30^\circ$.

Nesse caso a trajetória é classificada como *ascendente* e portanto toma-se as seguintes providências:

- muda-se o tamanho da lista Tm , ou seja, o horizonte de avaliação.
- calcula-se o novo instante de avaliação de soluções globais Av .
- aplica-se o movimento *clean up* na tentativa de melhorar a solução.
- faz-se $x = 0\%$ (parâmetro de tolerância), ou seja, aceita-se apenas soluções de melhoria.
- faz-se $it = 0$, para que o algoritmo reinicie o processo de busca e percorra Av iterações .

Na Figura 5.14b, o padrão da trajetória é classificada como *descendente* e portanto tomam-se as seguintes providências:

- muda-se o tamanho da lista Tm .

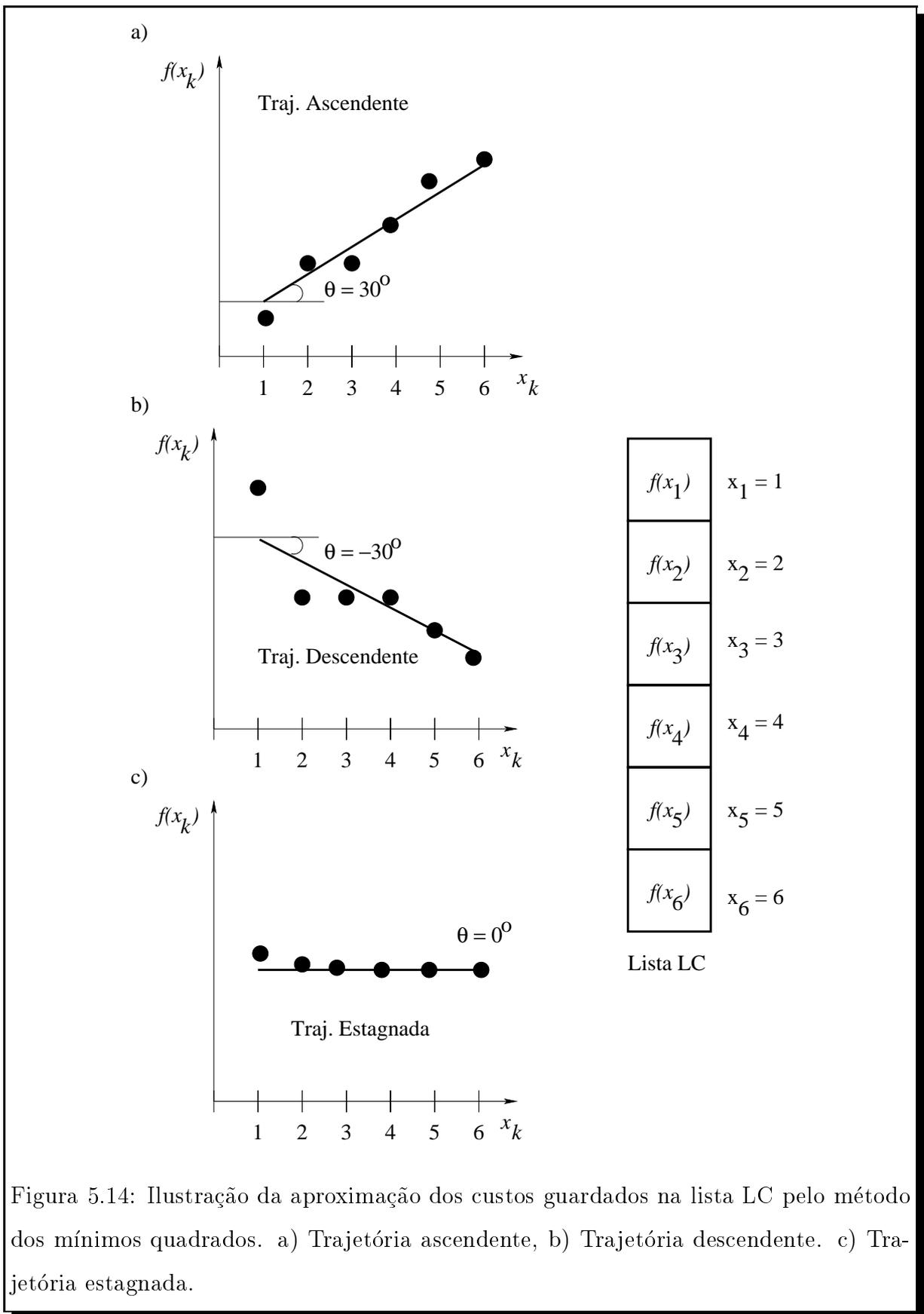


Figura 5.14: Ilustração da aproximação dos custos guardados na lista LC pelo método dos mínimos quadrados. a) Trajetória ascendente, b) Trajetória descendente. c) Trajetória estagnada.

- calcula-se o novo instante de avaliação Av .
- aplica-se o movimento *clean up* na tentativa de melhorar ainda mais a solução, constituindo uma etapa de intensificação.
- faz-se $x = 0\%$, ou seja, aceita-se apenas soluções de melhoria.
- faz-se $it = 0$.

Supondo que os custos armazenados em LC estejam distribuídos conforme a Figura 5.14c. Neste caso a trajetória é classificada como *estagnada*, logo:

- muda-se o tamanho da lista Tm .
- calcula-se o novo instante de avaliação Av .
- permite-se que nas próximas it iterações a nova solução intermediária CS_{interm} seja piorada em até $x\%$ em relação a solução intermediária anterior, constituindo uma etapa de diversificação.
- faz-se $it = 0$

Após o cálculo das trajetórias atualiza-se $it = it + 1$ e prossegue-se o processo de busca utilizando os novos parâmetros Tm , Av e x .

6

Resultados computacionais

Neste capítulo são apresentados os resultados computacionais obtidos com a VERSÃO 1 e a VERSÃO 2 propostas, sendo posteriormente comparados aos resultados de heurísticas da literatura. Especificamente, foram utilizados os resultados de CHRISTOFIDES & BEASLEY (1984) com relaxação PCVP, TAN & BEASLEY (1984), RUSSELL & GRIBBIN (1991), CHAO et al. (1995) e o de CORDEAU et al.(1997). Estas heurísticas foram selecionadas pela qualidade da solução e relevância ao desenvolvimento deste trabalho. O algoritmo foi implementado em Pascal, compilado em Free Pascal i386 v 1.0.6 e os experimentos realizados em um computador AMD Athlon (TM) XP 2200+.

Foram tratados 10 problemas descritos em CHRISTOFIDES & BEASLEY (1984), cujos tamanhos variam entre 50 e 100 nós (além do depósito), com períodos variando de 2 a 10 dias. Todos os problemas são derivados de problemas de PRV mono-periódico propostos por EILON et al. (1971) e artificialmente divididos em dias. Suas características são descritas na Tabela 6.1, onde n é o número de clientes, T o comprimento do período de planejamento e NMVPD é o número máximo de veículos por dia. A quarta coluna da Tabela 6.1 apresenta os detalhes das combinações dos dias de visitas para cada cliente. Observa-se que para alguns problemas, os clientes recebem suas combinações de visitas para algum dia de acordo com sua demanda. A expressão 10100, por exemplo, significa que o cliente é atendido nos dias 1 e 3 e não é atendido nos dias 2, 4 e 5. Maiores detalhes podem ser obtidos em CHRISTOFIDES & BEASLEY (1984).

Tabela 6.1: Problemas tratados.

Problema	n	T	NMVPD	Detalhes da combinação
50a	50	2	2/3	Uma visita no período
50b	50	5	3	Se demanda for ≤ 10 , uma visita no período. Se $11 \leq \text{demanda} \leq 25$ então (10100) ou (01010) ou (00101). Se demanda ≥ 26 , visita todos os dias.
50c	50	5	1	Uma visita no período
75a	75	2	5	Uma visita no período
75b	75	5	6	Se demanda for ≤ 15 , uma visita no período. Se $16 \leq \text{demanda} \leq 27$ então (10100) ou (01010) ou (00101). Se demanda ≥ 28 , visita todos os dias.
75c	75	10	1	Uma visita no período
100a	100	2	4	Uma visita no período
100b	100	5	5	Se demanda for ≤ 10 , uma visita no período. Se $11 \leq \text{demanda} \leq 25$ então (10100) ou (01010) ou (00101). Se demanda ≥ 26 , visita todos os dias.
100c	100	8	1	Uma visita no período
100d	100	5	4	Se demanda for ≤ 10 , uma visita no período. Se $11 \leq \text{demanda} \leq 25$ então (10100) ou (01010) ou (00101). Se demanda ≤ 26 então (10101) ou (01011) ou (11010).

Fonte: CHRISTOFIDES & BEASLEY (1984).

Os problemas possuem outras características particulares, apresentadas no anexo C em maiores detalhes.

Com relação ao algoritmo de roteamento (o algoritmo HTA) aplicado a cada dia do período, considerou-se em seus mecanismos adaptativos uma variação paramétrica de 10 faixas de período tabu e 5 faixas de horizonte de avaliação. As 10 faixas de período tabu têm comprimento igual a 4. Para todos os problemas utilizou-se como faixa menos restritiva $[t_{min}, t_{max}] = [0, 4]$ e a mais restritiva $[t_{min}, t_{max}] = [26, 30]$. Quanto aos horizontes de avaliação, todas as faixas têm também comprimento igual a 4. Como critério de parada para o algoritmo HTA utilizou-se x ($x = 3000$) iterações sem melhoria para todos os problemas.

Com relação a primeira heurística desenvolvida (VERSÃO 1), os testes foram realizados com 3 soluções iniciais diferentes, resultantes das construções iniciais CI-1, CI-2 e CI-3. A Tabela 6.2 apresenta os resultados obtidos para a VERSÃO 1. Nessa tabela, a primeira coluna indica os problemas tratados e os tipos de construções iniciais (CI-1, CI-2 e CI-3). Na segunda e terceira coluna são apresentadas as faixas de horizonte de avaliação e período tabu que resultaram na melhor solução. A quarta coluna apresenta o custo total do roteamento da melhor solução e a última coluna, o tempo de obtenção da solução em segundos.

Tabela 6.2: Resultados computacionais da VERSÃO 1.

Problema	[h_{min}, h_{max}]	[t_{min}, t_{max}]	Custo	Tempo
50a (CI-1)	[6, 10]	[4, 8]	532,9958	21 s
50a (CI-2)	[2, 10]	[12, 16]	549,6816	32 s
50a (CI-3)	[4, 8]	[22,26]	524,6111	6 s
50b (CI-1)	[6, 10]	[4, 8]	1376,3202	135 s
50b (CI-2)	[4, 8]	[8, 12]	1435,1753	207 s
50b (CI-3)	[4, 8]	[6, 10]	1379,7564	829 s
50c (CI-1)	[4, 8]	[0, 4]	589,6897	25 s
50c (CI-2)	[4, 8]	[12, 16]	589,9766	46 s

Tabela 6.2: (continuação)

Problema	$[h_{\min}, h_{\max}]$	$[t_{\min}, t_{\max}]$	Custo	Tempo
50c (CI-3)	[4, 8]	[22, 26]	524,8098	4 s
75a (CI-1)	[6, 10]	[6, 10]	850,5837	90 s
75a (CI-2)	[10, 15]	[10, 14]	868,9134	100 s
75a (CI-3)	[10, 15]	[6, 10]	835,3212	6 s
75b (CI-1)	[4,8]	[6, 10]	2053,9893	635 s
75b (CI-2)	[10, 15]	[4, 8]	2140,4137	833 s
75b (CI-3)	[4, 8]	[8, 12]	2084,2025	3378 s
75c (CI-1)	[4, 8]	[2, 6]	968,3075	190 s
75c (CI-2)	[10, 15]	[10, 14]	972,1158	258 s
75c (CI-3)	[10, 15]	[6, 10]	836,3663	180 s
100a (CI-1)	[6, 10]	[8, 12]	851,1557	521 s
100a (CI-2)	[4, 8]	[10, 14]	856,8093	734 s
100a (CI-3)	[3, 7]	[10, 14]	828,7893	104 s
100b (CI-1)	[6, 10]	[0, 4]	2095,9281	2113 s
100b (CI-2)	[10,15]	[4,8]	2136,9812	2870 s
100b (CI-3)	[6, 10]	[0, 4]	2164,4114	3075 s
100c (CI-1)	[4, 8]	[10, 14]	899,4017	42 s
100c (CI-2)	[10, 15]	[2, 6]	978,3539	85 s
100c (CI-3)	[10, 15]	[10, 14]	534,1210	76 s
100d (CI-1)	[4, 8]	[4, 8]	1674,2688	3555 s
100d (CI-2)	[10, 15]	[2, 6]	1751,6439	2731 s
100d (CI-3)	[7, 11]	[8, 12]	1676,2850	3900 s

Observa-se que em 60% dos problemas, a construção CI-3 obteve melhores soluções.

Com relação a segunda heurística desenvolvida (VERSÃO 2), utilizou-se

apenas a construção inicial 3 (CI-3) em todos os problemas, por ter sido considerada a mais bem sucedida em todos os experimentos com a VERSÃO 1. Com relação aos mecanismos adaptativos (referentes as faixas de período tabu e horizontes de avaliação) manteve-se a mesma variação paramétrica utilizada pela VERSÃO 1.

Em ambas as versões, verificou-se que períodos tabu pequenos (< 14) são suficientes para obtenção das melhores soluções.

Com relação aos movimentos de troca, utilizou-se nas duas versões, apenas um cliente no conjunto $M1$ e um cliente no conjunto $M2$. Outros testes foram realizados, aumentando e diminuindo o número de clientes no conjunto $M1$ e $M2$, mas não foram satisfatórios. Também estipulou-se um acréscimo na demanda do dia de $\zeta = 20\%$, permitindo uma maior variabilidade nas trocas.

O instante de avaliação global Av utilizado (na VERSÃO 2) no decorrer de todo o período é dependente da trajetória, que controla o valor da função objetivo, intensificando ou diversificando a busca. Após testes preliminares, foram estabelecidos os valores de $Av = 47$ e $Tm = 46$ para a trajetória ascendente e descendente e $Av = 5$ e $Tm = 4$ para a trajetória estagnada. Da mesma forma, permitiu-se uma piora na solução de $x = 2\%$, quando a trajetória estiver estagnada. Utilizou-se $\epsilon = 0.5^o$ como parâmetro para caracterizar o padrão da trajetória.

Como critério de parada em ambas as versões, utilizaram-se duas reinicializações, ou seja, aplicou-se novamente o movimento de melhoria sobre a melhor solução, sendo satisfatório para a obtenção das soluções. Na próxima seção é feita a comparação do desempenho deste trabalho com as heurísticas da literatura, tomando como base a construção CI-3.

SEÇÃO 6.1

**Comparação dos resultados com heurísticas da
literatura**

A Tabela 6.3 apresenta uma comparação dos resultados das versões propostas e os melhores resultados dentre cinco heurísticas da literatura. Em parênteses,

6. Resultados computacionais

têm-se o tempo computacional em segundos (quando disponibilizado) requerido para obtenção das soluções. Os computadores empregados são mencionados abaixo da tabela. Em função dos diferentes computadores utilizados e outros aspectos dos ambientes computacionais, não é possível fazer comparações em relação aos tempos de execução, exceto entre VERSÃO 1 e VERSÃO 2.

Quanto à qualidade das soluções (Tabela 6.3), os resultados indicam que VERSÃO 1 apresenta um desempenho superior aos algoritmos C/B, T/B e R/G, e iguala seu resultado ao da melhor solução reportada na literatura para o problema 50a. Em dois problemas 75a e 75c foi possível obter soluções com um custo ligeiramente inferiores ao das melhores soluções já reportadas. O anexo A apresenta as rotas geradas pela VERSÃO 1. A VERSÃO 2, quanto à qualidade da solução, apresentou melhores resultados que a VERSÃO 1 em 7 problemas, com empates nos problemas 50a, 75a e 75c. A VERSÃO 2 apresentou bons resultados em relação aos melhores resultados da literatura (C/G/L) nos problemas 50a, 50c, 75a, 75c, 100a e 100c. Além disso, apresentou resultados razoáveis para os demais problemas. O anexo B apresenta as rotas geradas pelo algoritmo proposto na VERSÃO 2.

Tabela 6.3: Comparação dos resultados computacionais.

Pr.	C/G/L	C/G/W	C/B	T/B	R/G	Versão 1	Versão 2
50a	524.61*	524.6	547.4(0.1)	-	537.3(6.0)	524.61*	524.61*
50b	1322.87*	1322.9	1443.1(1.7)	1481.3(2.2)	1355.4(34.8)	1379.75	1335.59
50c	524.61*	524.6	547.5(0.1)	-	-	524.8	524.61*
75a	835.43	840.2	843.9(0.3)	-	867.8(6.6)	835.32*	835.32*
75b	2027.99*	2046.2	2187.3(14.2)	2192.5(3.6)	2141.3(64.2)	2084.2	2068.2
75c	836.37	847.2	938.2(0.4)	-	-	836.36*	836.36*
100a	826.14*	831.1	839.2(0.2)	-	833.6(22.8)	828.78	826.63
100b	2034.15*	2042.0	2153.3(39.2)	2281.8(4.1)	2108.3(112.2)	2164.41	2061.25
100c	826.14*	828.3	875.0(4.2)	-	-	834.12	826.63
100d	1595.84*	1611.9	1674.0(3.7)	1833.7(4.2)	1638.5(121.2)	1676.28	1652.88

* : Melhor solução reportada.

onde:

- C/G/L : algoritmo de CORDEAU et al. (1997) - Estação SUN SparcStation 10.
- C/G/W : algoritmo de CHAO et al. (1995) - Estação SUN 4/370.
- C/B : algoritmo de CHISTOFIDES & BEASLEY (1984), com relaxação PCVP - Computador CDC 7600.
- T/B : algoritmo de TAN & BEASLEY (1984) - Computador CDC 7600.
- R/G : algoritmo de RUSSELL & GRIBBIN (1991) - Computador IBM 20 MHz.
- VERSÃO 1: Primeira versão proposta do algoritmo.
- VERSÃO 2: Segunda versão proposta, denotada por PVRP-CS
(Periodic Vehicles Routing Problem - Controlled Search).

A Tabela 6.4 apresenta uma comparação dos resultados em relação a melhor solução reportada, ou seja, foram calculados os desvios percentuais (dp) dos custos entre cada problema. Esses percentuais foram calculados através da seguinte fórmula:

$$dp = 100 \times \frac{\text{Custo da solução} - \text{Custo da melhor solução}}{\text{Custo da melhor solução}}$$

Nesta tabela também foram calculadas as médias dos desvios percentuais para posterior análise.

Portanto, quando $dp = 0$ significa que foi encontrado a melhor solução, quando $dp > 0$ indica o quanto a solução de cada heurística foi pior que a melhor solução encontrada para os respectivos problemas.

Observa-se nesta tabela que na VERSÃO 1 foram encontradas três ótimas soluções enquanto que na VERSÃO 2 foram encontradas quatro soluções ótimas. A VERSÃO 2 na média foi 250% melhor que a VERSÃO 1. Comparando com os demais problemas vemos que a VERSÃO 2 se coloca como uma das melhores heurísticas com uma média abaixo de 1% com relação a média das outras heurísticas..

Tabela 6.4: Comparando a melhor solução com as demais heurísticas.

Pr.	C/G/L(%)	C/G/W(%)	C/B(%)	T/B(%)	R/G(%)	Versão 1(%)	Versão 2(%)
50a	0	0	4.34	-	2.42	0	0
50b	0	0	9.0	11.97	2.45	4.29	0.96
50c	0	0	4.36	-	-	0.038	0
75a	0.01	0.58	1.0	-	3.9	0	0
75b	0	0.9	7.8	8.1	5.6	2.77	1.98
75c	0	1.3	12.17	-	-	0	0
100a	0	0.6	1.58	-	0.9	0.32	0.06
100b	0	0.38	5.86	12.17	3.64	6.4	1.3
100c	0	0.26	5.9	-	-	0.9	0.06
100d	0	1.0	4.9	17.9	2.67	5.0	3.57
média	0.01	0.502	5.69	11.78	2.94	1.97	0.79

A Tabela 6.5 mostra uma comparação entre os tempos computacionais obtidos na VERSÃO 1 com os tempos obtidos na VERSÃO 2, onde \overline{tmp} é o tempo médio para obtenção das melhores soluções para este conjunto de problemas.

Conforme pode ser observado na tabela, a VERSÃO 2 requisitou um tempo computacional médio 31% menor que a VERSÃO 1 encontrando soluções de melhor qualidade (Tabela 6.4). Entretanto, em termos absolutos, estes tempos podem ser considerados excessivos.

Observa-se também que nos problemas cuja a frequência é menor, requisita-se um tempo menor de processamento para se encontrar a melhor solução. Isso ocorre devido a complexidade do algoritmo que aumenta com a frequência.

6. Resultados computacionais

Tabela 6.5: Comparando os tempos da VERSÃO 2 com os da VERSÃO 1.

Problema	VERSÃO 1(tempo em segundos)	VERSÃO 2(tempo em segundos)
50a	6	4
50b	829	653
50c	4	4
75a	6	6
75b	3378	1190
75c	180	4
100a	104	147
100b	3075	2998
100c	76	126
100d	3900	2820
\overline{tmp}	1155,8	795,2

7

Conclusões e perspectivas futuras

Neste trabalho, foi proposto um algoritmo para a resolução do problema de roteamento de veículos periódico (PRVP), um problema pouco estudado. O PRVP é um problema multiperíódico, ou seja, as decisões de roteamento são tomadas em um horizonte de entrega de T dias. O objetivo é elaborar um conjunto de rotas para cada dia t , de forma a minimizar a distância total percorrida sobre todo o período.

Utilizou-se o algoritmo adaptativo de busca tabu HTA para o roteamento. Testes computacionais envolvendo dez problemas da literatura foram realizados e os resultados foram comparados com cinco heurísticas da literatura.

A heurística desenvolvida (Periodic Vehicles Routing Problem - Controlled Search), mostrou-se eficaz quanto a qualidade da solução em relação as heurísticas existentes, colocando-a entre as três melhores implementações para este conjunto de problemas. Os resultados indicam que o algoritmo com mecanismos adaptativos de controle dos padrões das trajetórias é capaz de produzir soluções de alta qualidade.

Para o desenvolvimento de novos trabalhos envolvendo roteamento de veículos periódico, sugere-se a utilização de uma meta-heurística tabu nos movimentos de trocas e inserções dos clientes entre os dias.

Referências Bibliográficas

- ANDERSON, C. A.; JONES, K. F.; PARKER, M.; RYAN, J.; *Parth assignment for call routing: an applications of tabu search*, Annals of operations research, vol. 41, pp. 301-312, 1993.
- ANTT: *Agência Nacional de Transporte Terrestre*, Encontra-se disponível na página: <http://www.antt.gov.br>, último acesso em 01/12/2004.
- BALLOU, R. H.; *Logística empresarial*, Atlas, São Paulo, 1993.
- BELTRAMI, E.; BODIN, L.; *Networks and vehicle routing for municipal waste collection*. Networks, 4, pp. 65-94, 1974.
- BERGER, J.; BARKAOVI, M.; BRAYSY, O.; *A parallel hybrid genetic algorithm for the vehicle routing problem with time windons*, Working paper, Defense Research Establishment Valcartier, Canadá, 2001.
- BODIN, L.; *Twenty years of routing and scheduling*, Operations Research, n^o 38, pp. 571-579, 1990.
- BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M.; *Routing and scheduling of vehicle and crews: The state of the art*, Comput. Oper. Res., Vol. 10, No 2, pp. 63-211, Pergamon Press, Oxford, 1983.
- BODIN, L.; GOLDEN, B.; *Classification in vehicle routing and scheduling*, Networks, v. 11(2), pp. 97-108, 1981.

REFERÊNCIAS BIBLIOGRÁFICAS

- BORGES, C.; *Pernambucanas querem chegar mais rapidamente ás lojas*, Revista Tecnológica, ano VI, número 75, 2002.
- BOTT, K.; BALLOU, R.; *Research perspectives in vehicle routing and scheduling*, Transportation Research, 20 A, pp. 239-243, 1996.
- BREEDAN, A.; *Improvement heuristics for the vehicle routing basead on simulated annealing*, Eur. J. Oper. Res., 86, pp. 480-490, 1995.
- CARRETO, C.; BAKER, B.; *A GRASP interactive approach to the vehicle routing problem with backharls*, In the Hansem and C. C. Ribeiro, editors, Eways and surveis on metaheuristics, Klumer Academic Publishers, 2001.
- CHRISTOFIDES, N.; EILON, S.; *An algorithm for the vehicle-despatching problem*, Operational research Quart., vol 20, pp. 309-318, 1969.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH P.; *The vehicle routing problem*, Combinatorial Optmization, pp. 315-338, 1979.
- CHRISTOFIDES, N.; *The vehicle routing in The Traveling salesman problem*, editado por LAWLER E. L., LENTRA J. K., RINNOY KAN A. H. G., pp 431-448, North-Holland, 1985.
- CHRISTOFIDES, N.; BEASLEY J. E.; *The Period Routing Problem*, Networks, 14, pp. 237-256, 1984.
- CHAO, M.; GOLDEN, L. B.; WASIL, A. E.; *A new heuristic for the period traveling salesman problem*, Computer Ops Res., vol. 22, pp. 553-565, 1994.
- CHAO, M.; GOLDEN, L. B.; WASIL, E.; *An improved heuristic for the period vehicle routing problem*, Networks, vol. 26, pp. 25-44, 1995.
- CLARKE, G.; WRIGHT, J.; *Scheduling of vehicles from a central depot to a number of delivery points*, Ops. Res. Vol. 12, pp. 568-581, 1964.

REFERÊNCIAS BIBLIOGRÁFICAS

- CORDEAU J. F.; GENDREAU M.; LAPORTE G.; *A tabu search heuristic for periodic and multi-depot vehicle routing problems*, Center for research on transportation, Technical Report, Networks, 30, pp. 105-119, Montreal, Canada, 1997.
- CZECH, Z. J.; CZARNAS, P.; *Parallel simulated annealing for the vehicle routing problem with time windows*, 10th Euramicro Workshop on Parallel, Distributed and Networks based Processing, Canary - Islands - Spain, pp. 376-383, January, 2002.
- DAMMEYER, F.; VOB, F.; *Dynamic tabu list management using the reverse elimination method*, Annals of operations research, vol. 41, pp. 31-36, 1993.
- DELL'AMICO, M.; TRUBIAN, M.; *Applying tabu search to the job shop scheduling problem*, Annals of operations research, vol. 41, pp. 231-252, 1993.
- DROR, M.; LEVY, L.; *A vehicle routing improvement algorithm comparison of a greedy and a matching implementation for inventory routing*, Computer & Ops. Res., Vol. 13, pp. 33-45, 1986.
- EILON, S.; WATSON-GANDY, C.D.T.; CHRISTOFIDES, N.; *Distribution management: mathematical modelling and practical analysis*, Griffin, London, 1971.
- FEO, T.; REZENDE, M. G. C.; *Greedy randomized adaptive search procedures*, Journal of Global Optimization 6, pp. 109-133, 1995.
- FERNANDES, J. A.; *Avaliação da Implantação de um software roteirizador em um sistema de distribuição de bebidas - estudo de casos*, Dissertação de Mestrado, UFSCAR, São Carlos, 2001.
- FIETCHER, C.; *A parallel tabu search algorithm for large traveling salesman problems*, Discrete applied mathematics, vol. 51, n. 3, pp. 243-267, 1994.
- FISHER, M.; GREENFIELD, A.; JAIKUMAR, R.; LESTER, J.; *A Computadorized Vehicle Routing Application*, Interface, 12, pp. 42-52, 1982.
- FISHER, M.; JAIKUMAR, R.; *A generalized assignment heuristic for vehicle routing*, Networks, 11(2), pp. 109-124, 1981.

REFERÊNCIAS BIBLIOGRÁFICAS

- FISHER, M.; *Vehicle routing*, In: Ball, M. O., Magnanti, T. L., Monma, C. L., Nemhauser, G. L., (Eds.), *Handbooks in OR & MS*, vol. 8, Elsevier, Amsterdam, pp. 1-33, 1995.
- GALVÃO, A. N.; *Logística e gerenciamento da cadeia de distribuição*, Rio de Janeiro: Campus, 2001.
- GAREY, M. R., JOHNSON, D. S.; *Computers and intractability - A guide to the theory of NP-Completeness*, Freeman, San Francisco, 1979.
- GASKELL, T. J.; *Bases for vehicle fleet scheduling*, *Opnl. Res. Quart.*, Vol. 18, pp. 281-295, 1967.
- GAUDIOSO, M.; PALETTA, G.; *A heuristic for the periodic vehicle routing problem*, *Transportation Science*, vol. 26, n^o 2, 1992.
- GENDREAU, M.; SORIANO, P.; SALVAIL, L.; *Solving the maximum clique problem using a tabu search approach*, *Annals of operations research*, vol. 41, pp. 385-402, 1993.
- GENDREAU, M., HERTZ, A., LAPORTE, G.; *A tabu search heuristic for the vehicle routing problem - (TABUROUTE)*, *Management Science*, vol. 40, n^o 10, 1994.
- GENDREAU, M.; BRAYSY, O.; *Genetic Algorithms for the vehicle routing problem with time windows*, Internal Report STF42 A01021, SINTEF Applied mathematics, department of optimization, Oslo, Norway, 2001.
- GENDREAU, M.; LAPORTE, G.; POTVIN, J. Y.; *Vehicle routing: modern heuristics*, In: Aarts, E. H. L., Lenstra, J. K., (Eds.), *Local search in combinatorial optimization*, Wiley, Chichester, pp. 311-336, 1997.
- GUISI, M. A.; CONSOLI, M. A.; MARCHETTO, R. M.; NEVES, M. F.; *Uso e benefícios de softwares de roteirização na gestão de transportes*, VII SEMEAD, Estudo de caso, 2004.
- GILLET, B. E.; MILLER, L. R.; *A heuristic algorithm for the vehicle-dispatch problem*, *Ops. Res.*, Vol. 22, pp. 340-349, 1974.

REFERÊNCIAS BIBLIOGRÁFICAS

- GOLDEN, B; ASSAD, A.; *Perspectives on vehicle routing: exciting new developments*, Operations Research, n^o 24, pp. 803-810, 1986.
- GOLDEN, B. L., ASSAD, A. A. (Eds): *Studies in management science and systems*, Elsevier science publ., Amsterdam, vol. 16, pp. 7-45, 1988.
- GOLDEN, B.; WATTS, E.; *Vehicle routing: methods and studies*, Elsevier Science/North Holland, 1988.
- GOLDEN, B.; ASSAD, A.; DAHL, R.; *Analysis of a Large Scale Vehicle Routing Problem with an Inventory Component*, Large Scale Syst, 7, pp. 181-190, 1984.
- GOLDEN, B. L.; WASIL, E. A.; *Computerized vehicle routing in the soft drink industry*, Operations Res., 35, pp. 6-17, 1987.
- GLOVER, F.; *Surrogate Constraints*, Operations Research, v. 16, n^o 4, 1987.
- GLOVER, F.; *Heuristics for integer programming using surrogate constraints*, Decision Sciences, vol. 8, 1977.
- GLOVER, F.; *Multilevel tabu search and embedded search neighborhoods for the traveling salesman problem*, To appear in ORSA journal on computing, 1991.
- GLOVER, F.; McMILLAN, C.; NOVICK, B.; *Interactive decision software and computer graphics for architectural and space planning*, Annals of operations research, vol. 5, pp. 557-573, 1985.
- HANSEN, P.; JAUMARD, B.; *Algorithms for the maximum satisfiability problem*, Computing, vol. 44, pp. 279-303, 1990.
- HANSEM, P.; *The steepest ascent mildest descent heuristic for combinatorial programming*, Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy, 1986.
- HANSEN, P.; JAUMARD, B.; SILVA, D.; *Average linkage divisive hierarchical clustering*, To appear in journal of classification, 1992.

- HERTZ, A.; WERRA, D.; *Using tabu search techniques for graph coloring*, Computing, vol. 29, pp. 345-351, 1987.
- HERTZ, A.; JAUMARD, B.; ARAGÃO, P. M.; *Topology of local optima for the k-coloring problem*, To appear in discrete applied mathematics, 1992.
- HOLLAND J. H.; *Adaptation in natural and artificial systems*, Ann Harbor, University of Michigan Press, 1975.
- HUBSHER, R.; GLOVER, F.; *Applying tabu search with influential diversification to multi-processos scheduling*, 1992.
- JAUMARD, B.; HANSEN, P.; ARAGÃO, P. M.; *Column generations methods for probabilistic logic*, ORSA journal on computing, vol. 3, pp. 135-148, 1991.
- LOKKETANGEN, A.; JORNSTEN, K.; STOROY, S.; *Tabu search within a pivot and complement framework*, International transactions in operational research, vol. 1, n. 3, pp. 305-316, 1994.
- KANTORAVDIS, G.; BARD, J. F.; *A grasp for the vehicle routing problem with time windows*, ORSA J. on Computing, v.7, pp. 10-23, 1995.
- KIRKPATRICK S.; GELLAT C. D.; VECCHI M. P.; *Optimizations by simulated annealing*, Science 220, pp. 671-680, 1983.
- LAGUNA, M.; *Clustering for the design of SONET rings in interoffice telecommunications*, Management science, vol. 40, n. 11, pp. 1533-1541, 1994.
- LAGUNA, M.; *Tabu Search Tutorial*, II Escola de Verano Latino-Americana de Investigación Operativa, Rio de Janeiro, Brazil, 1995.
- LAGUNA, M.; BARNES, J. W.; GLOVER, F.; *Tabu search methods for a single machine scheduling problem*, Journal of intelligenet manufacturing, vol. 2, pp. 63-74, 1993.
- LAGUNA, M.; GLOVER, F.; *Integrating target analysis and tabu search for improved scheduling systems*, Experts systems with applications, vol. 6, pp. 287-297, 1993.

REFERÊNCIAS BIBLIOGRÁFICAS

- LAPORTE, G.; *The vehicle routing problem: an overview of exact and approximate algorithms*, European Journal of Operations Research, n^o 59, pp. 345-358, 1992.
- LAPORTE, G.; NORBERT, Y.; *Surveys in combinatorial optimization*, North Holland, Amsterdam, pp. 147-187, 1987.
- LAPORTE, G.; OSMAN, I. H.; *Routing problems: a bibliography*, In: (Eds.) Laporte, G., Gendreau, M., Ann. Oper. Res., vol. 61, pp. 227-262, 1995.
- LIN, S.; *Computer solutions of the TSP*, Bell System Tech. J., Vol. 44, pp. 2245-2270, 1965.
- LIN, S.; KERNIGHAN, B. W.; *An effective heuristic algorithm for the TSP*, Ops. Res., Vol. 21, pp 498-516, 1973.
- METROPOLIS, N.; ROSENBLUTH, A. W.; TELLER, A. H.; TELLER, E.; *Equation of state calculation by fast computing machines*, J. of Chem. Phys, 21, pp. 1087-1091, 1953.
- MONTANÉ, F. A. T.; GALVÃO, R. D.; *A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service*, Aceito para publicação em computers and operations research, 2004.
- NUNES, L. F.; *Algoritmo genético aplicado na abordagem de um problema real de roteirização de veículos*, Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, 1998.
- OLIVEIRA, S.; STROUD, G.; *A parallel version of tabu search and the path assignment problem*, Heuristics for combinatorial optimizations, vol. 4, pp. 1-24, 1989.
- OSMAN, I. H.; *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Annals of Operations Research, 41, pp. 421-451, 1993.
- PEREIRA, G. R.; *O custo do transporte entre o centro de produção e o mercado*, Tirado de www.moraesjunior.edu.br, 2003.

REFERÊNCIAS BIBLIOGRÁFICAS

- PIDD, M.; *Tolls for thinking: modelling in management science*, John Wiley & Sons, Chichester, 1996.
- PUREZA, V.; *Problemas de roteamento de veículos via metaheurística tabu*, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 1991.
- PUREZA, V.; FRANÇA, P.; *Uma abordagem adaptativa de busca tabu aplicada ao problema de roteamento de veículos*, Transportes, Vol. 9, n. 2, pp. 28-47, R.J., Brasil, 2001.
- RAYWARD-SMITH, V. J.; SMITH, G. D.; OSMAN, I. H.; REEVES C. R.; *Modern Heuristic Search Methods*, John Wiley and Sons, New York, 1996.
- REBELLO, F. R.; HAMACHER, S.; *Uma proposta de algoritmo genético de duas fases para roteamento de veículos*, Pontifícia Universidade Católica do Rio de Janeiro, PUC-RIO, 1999.
- REVISTA DISTRIBUIÇÃO; Edição número 120 de novembro de 2002.
- RUGGIERO, M. A. G.; LOPES, V. L. R.; *Cálculo numérico: aspectos teóricos e computacionais*, São Paulo, McGraw-Hill, 1988.
- RUSSELL, R. A.; IGO, W.; *An assignment routing problem*, Networks, 9, pp. 1-17, 1979.
- RUSSELL, R. A.; GRIBBIN, D.; *A multiphase approach to the period routing problem*, Networks, vol. 21, pp. 747-765, 1991.
- RUSSELL, R. A.; *An effective heuristic for the m-tour traveling salesman problem with some side condition*, Ops. Res., 25, pp. 517-524, 1977.
- SIPPER, D.; BULFIN JR.; *Planning, control and integration*, McGraw-Hill International Edition, 1997.
- SOLOMON, M.; *Algorithms for the vehicle routing and scheduling problems with time windows constraints*, Operations Research, vol. 35, n^o 2, pp. 254-265, 1987.
- TAN, C. C.; BEASLEY, J. E.; *A heuristic algorithm for the period vehicle routing problem*, Omega, 12(5), pp. 497-504, 1984.

REFERÊNCIAS BIBLIOGRÁFICAS

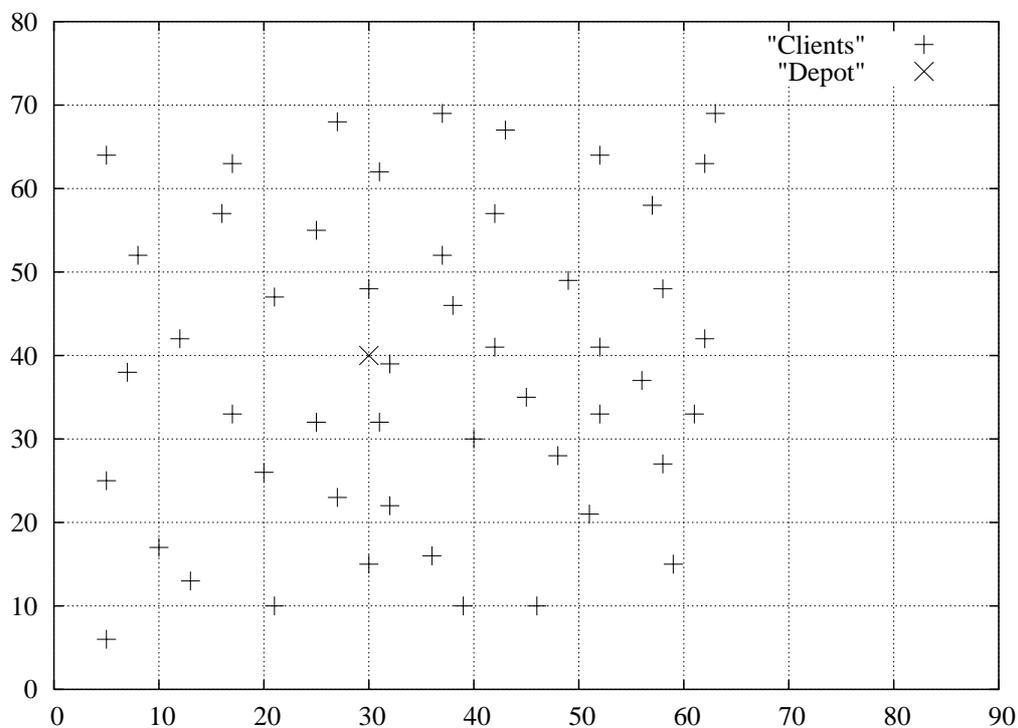
- TAILLARD, E.; *Parallel iterative search methods for vehicle routing problems*, Networks, vol. 23, n^o 8, 1993.
- TYAGI, M. S.; *A practical method for truck dispatching problem*, J. Ops. Soc. Japan, Vol. 10, pp. 76-92, 1968.
- WATSON-GANDY, C. D.; FOULDS, L. R.; *The vehicle scheduling problem: a survey*, New Zealand Ops. Res., Vol. 9, pp. 73-91, 1981.
- WIDMER, M.; *Job Shop scheduling with tooling constraints: a tabu search approach*, Journal of the operational research society, vol. 24, n. 1, pp. 75-82, 1991.
- WILLARD, J. A. G.; *The vehicle routing using r-optimal tabu search*, Tese de mestrado, The management School, Imperial College, Inglaterra, 1989.
- WOODRUFF, D. L.; SPEARMAN, M. L.; *Sequencing and batching for two classes of jobs with deadlines and setup times*, Production and operations management, vol. 1, n. 1, pp. 87-102, 1993.
- YELLOW, P. C.; *A computacional modification of the savings method of vehicle scheduling*, Opnl. Res. Quart., Vol. 21, pp. 281, 1979.

Anexo A: Rotas geradas pelo programa

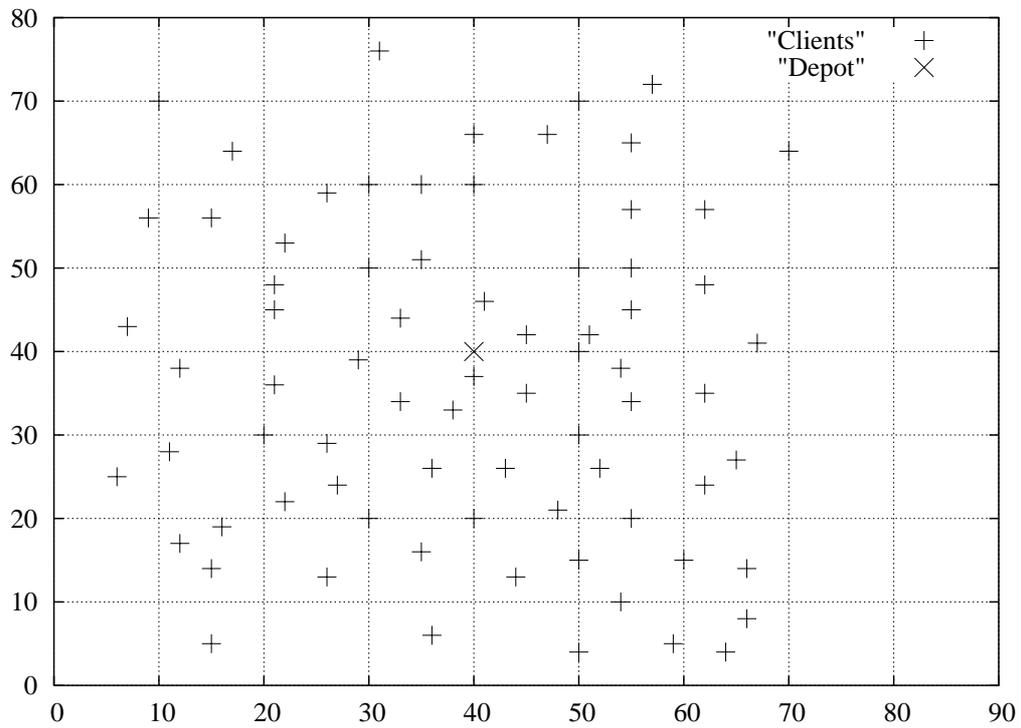
Versão 1

Neste anexo são apresentados os resultados obtidos para os problemas tratados na VERSÃO 1. Inicialmente, mostra-se a distribuição geográfica de nós desses problemas e em seguida os resultados dos problemas cujos clientes têm frequência 1 (problemas (a) e (c)) e frequência maior que 1 (problemas (b) e (d)).

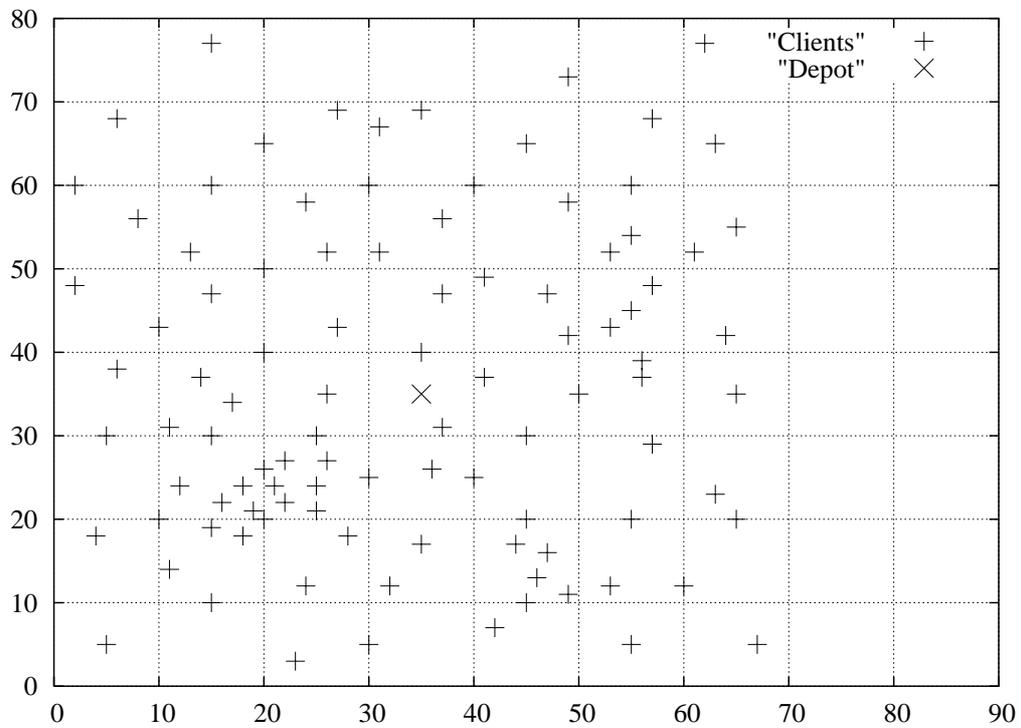
Problema com 50 nós:



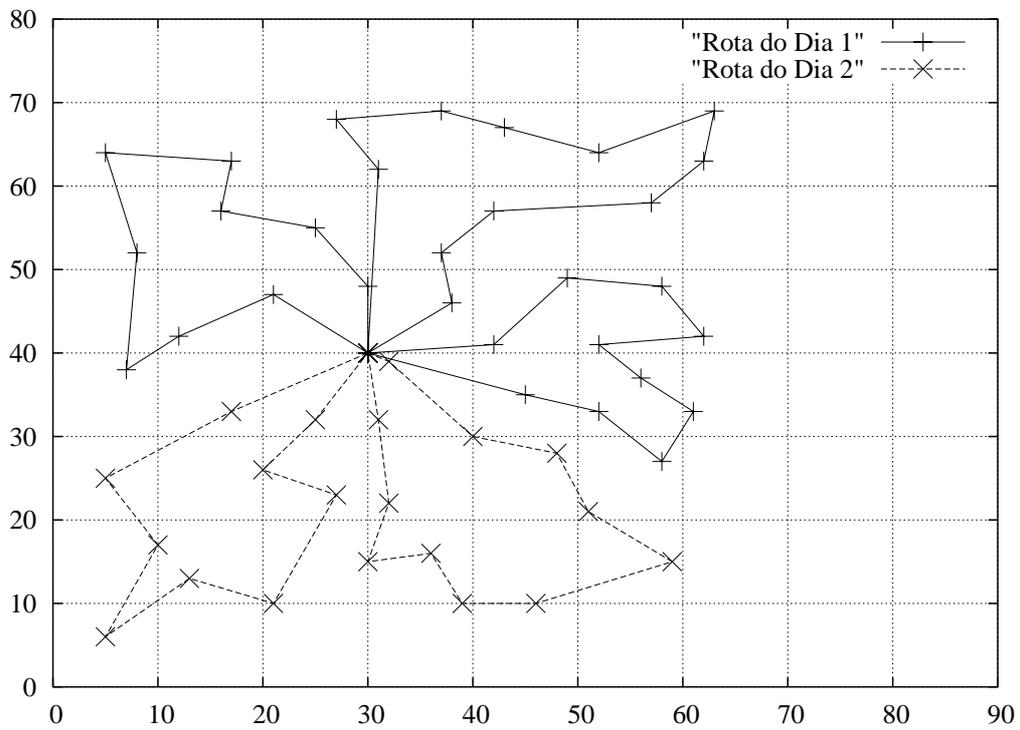
Problema com 75 nós:



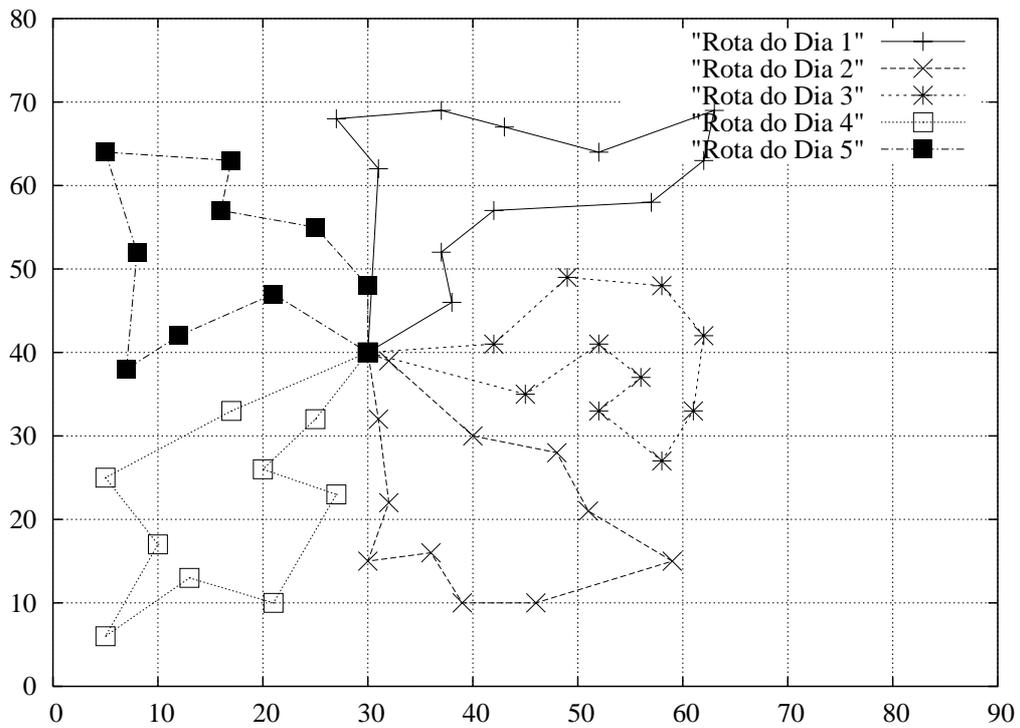
Problema com 100 nós:



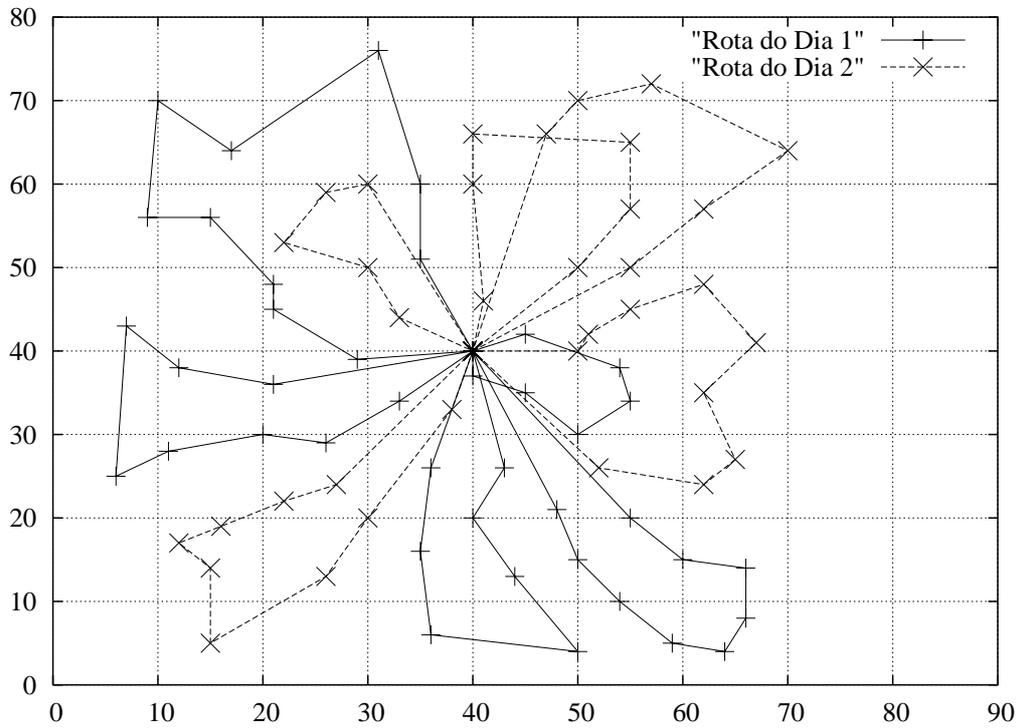
Rotas obtidas para o problema 50a (custo de 524,61):



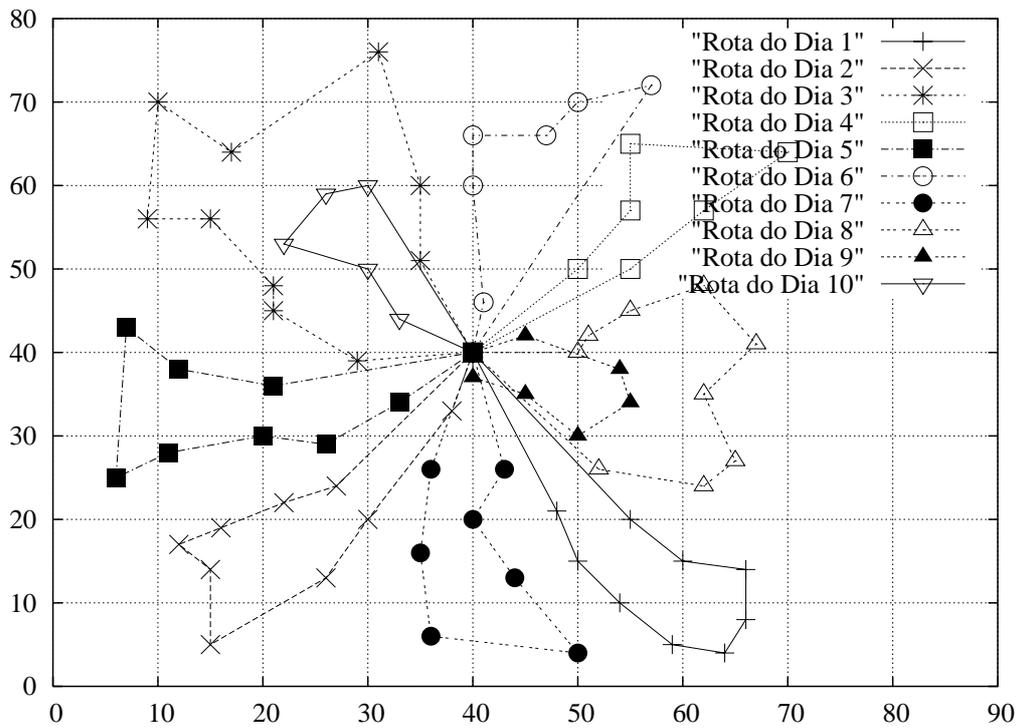
Rotas obtidas para o problema 50c (custo de 524,81):



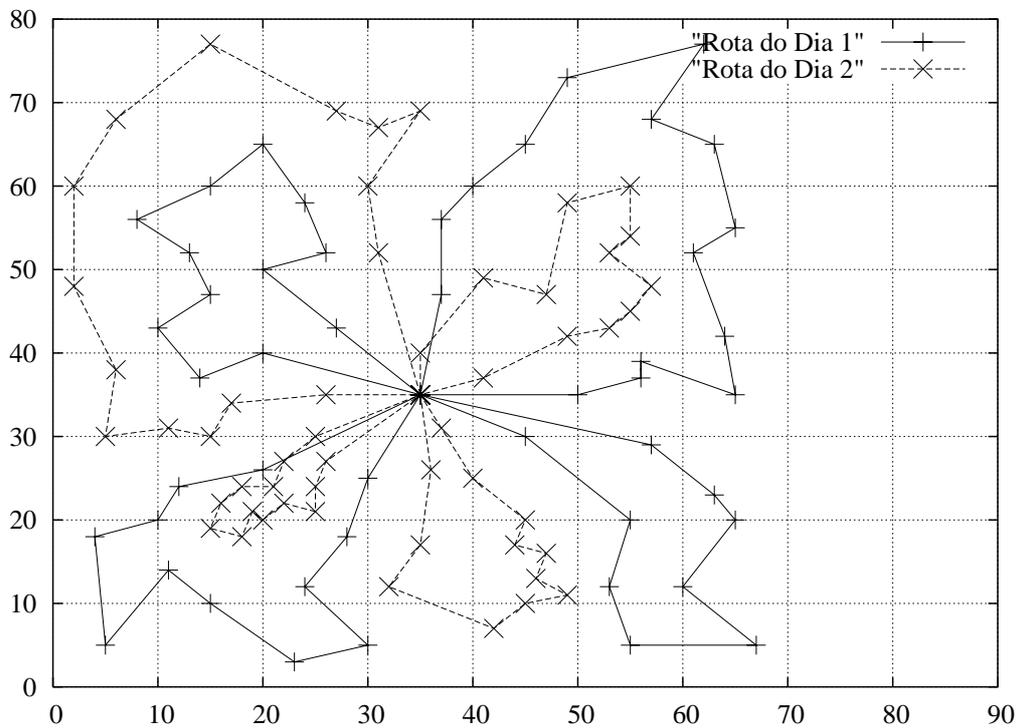
Rotas obtidas para o problema 75a (custo de 835,32):



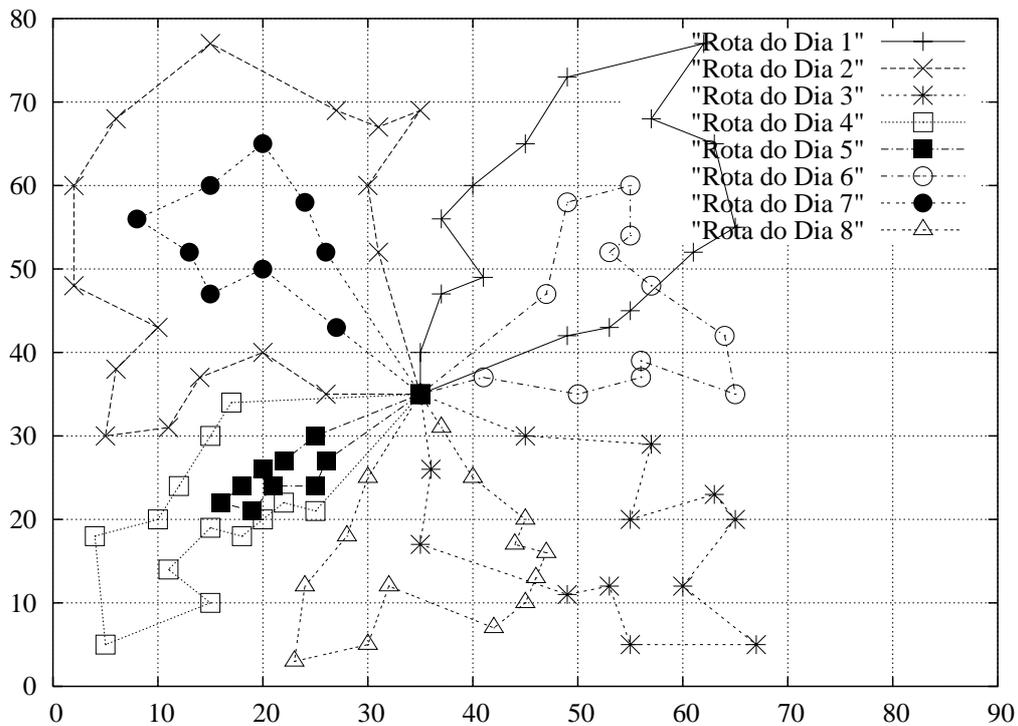
Rotas obtidas para o problema 75c (custo de 836,36):



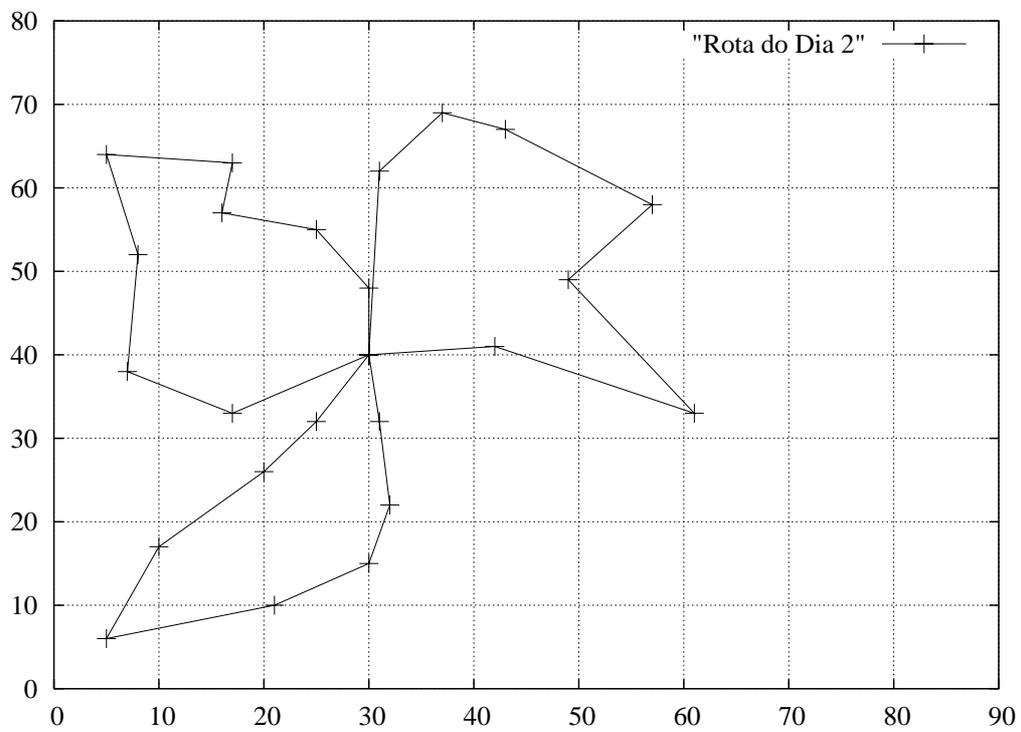
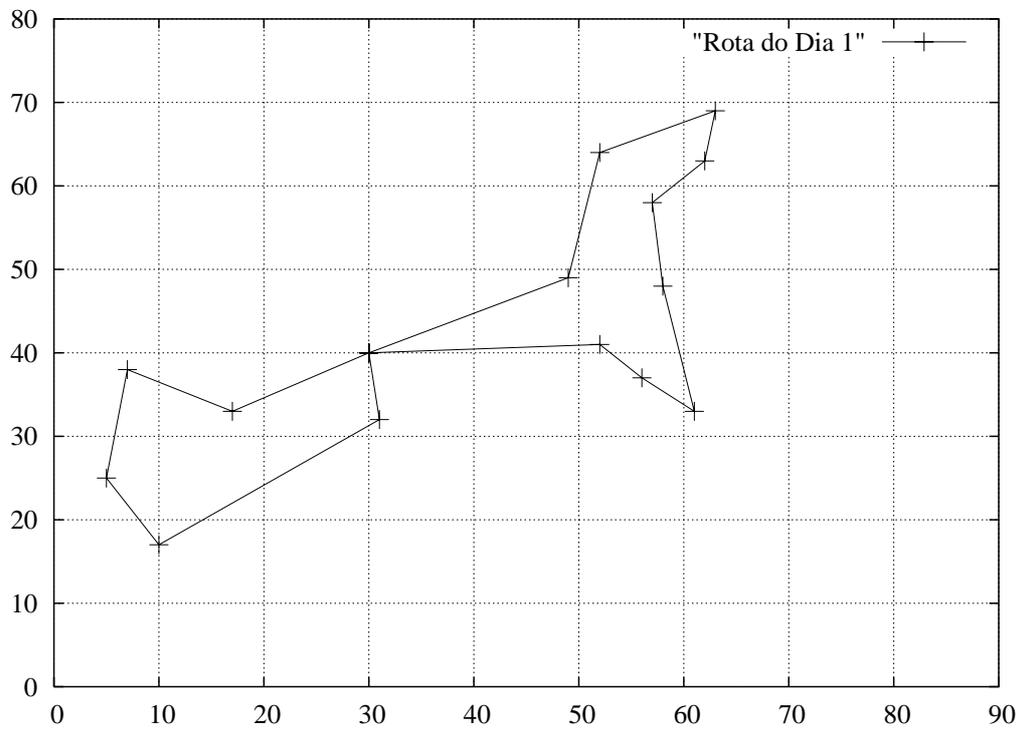
Rotas obtidas para o problema 100a (custo de 828,78):

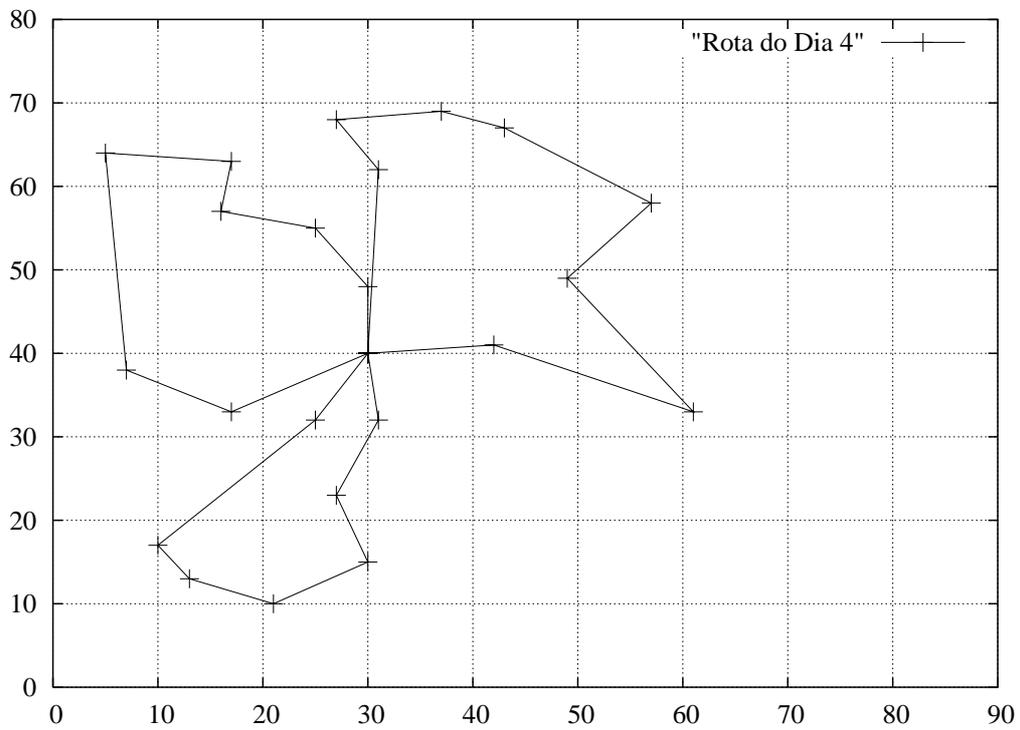
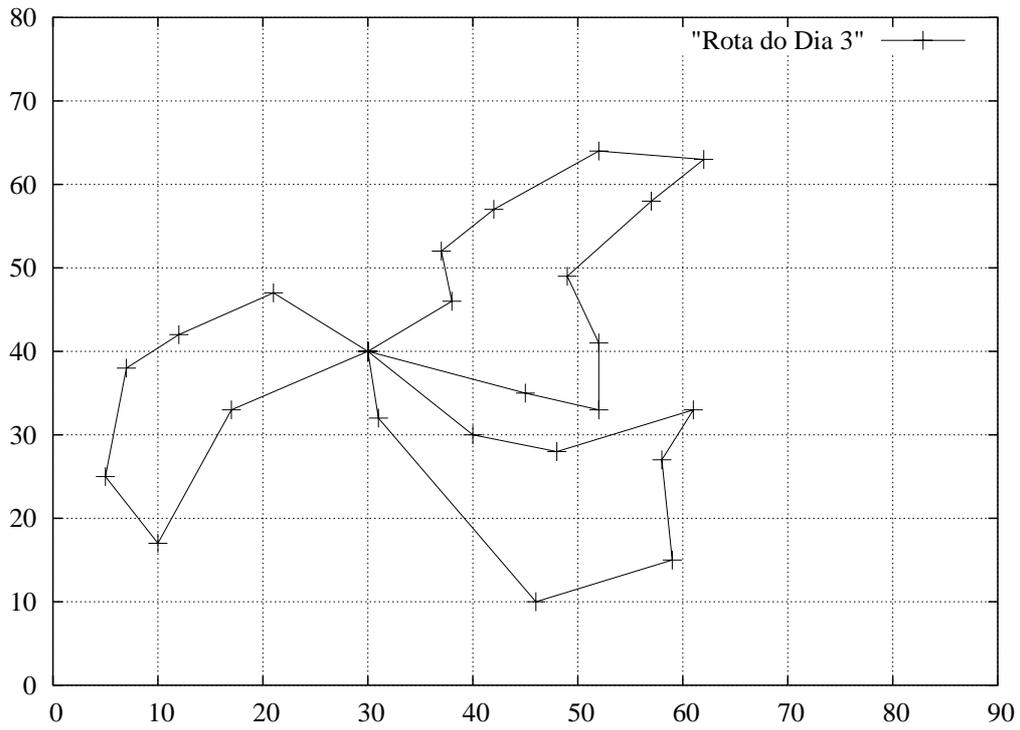


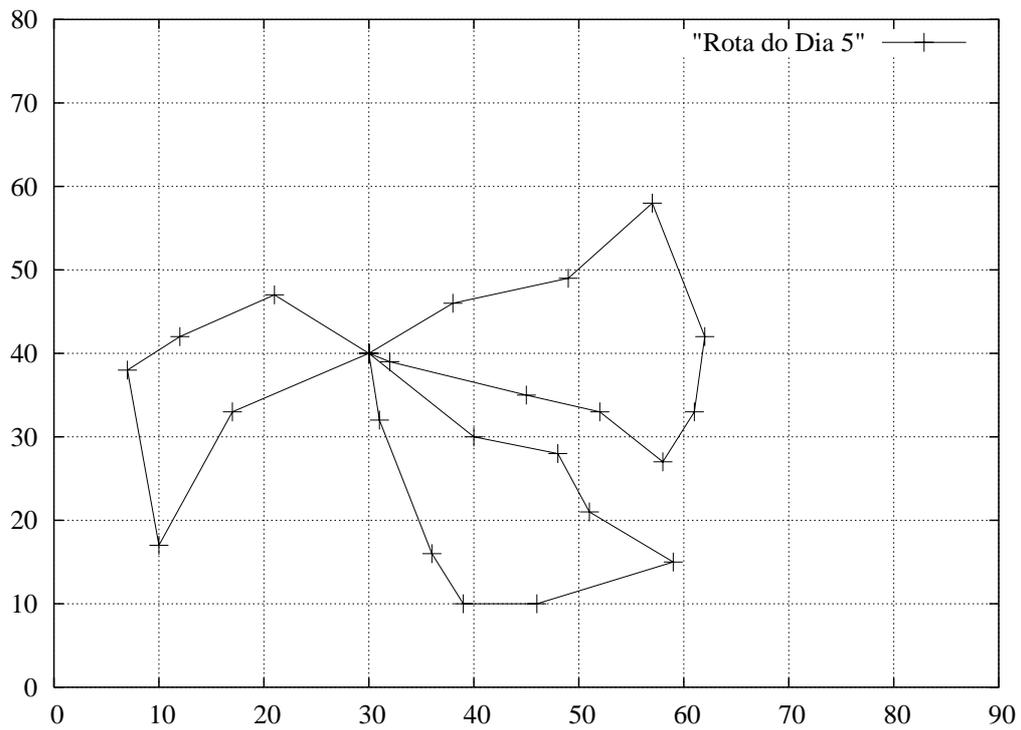
Rotas obtidas para o problema 100c (custo de 834,12):



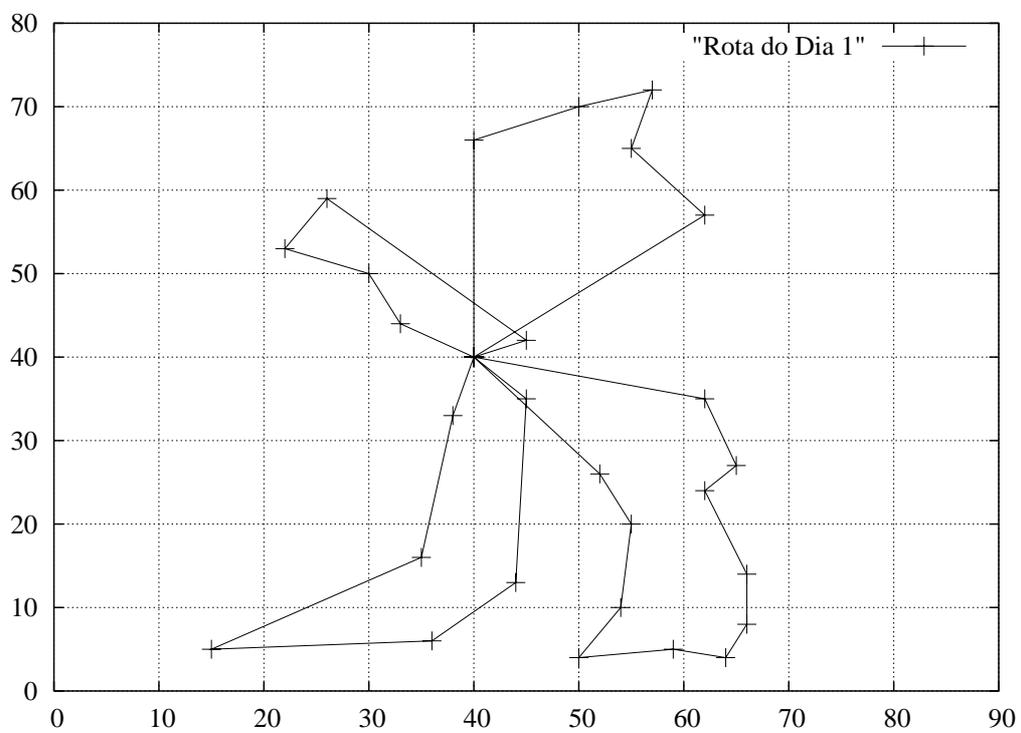
Rotas obtidas para o problema 50b (custo de 1379,75):

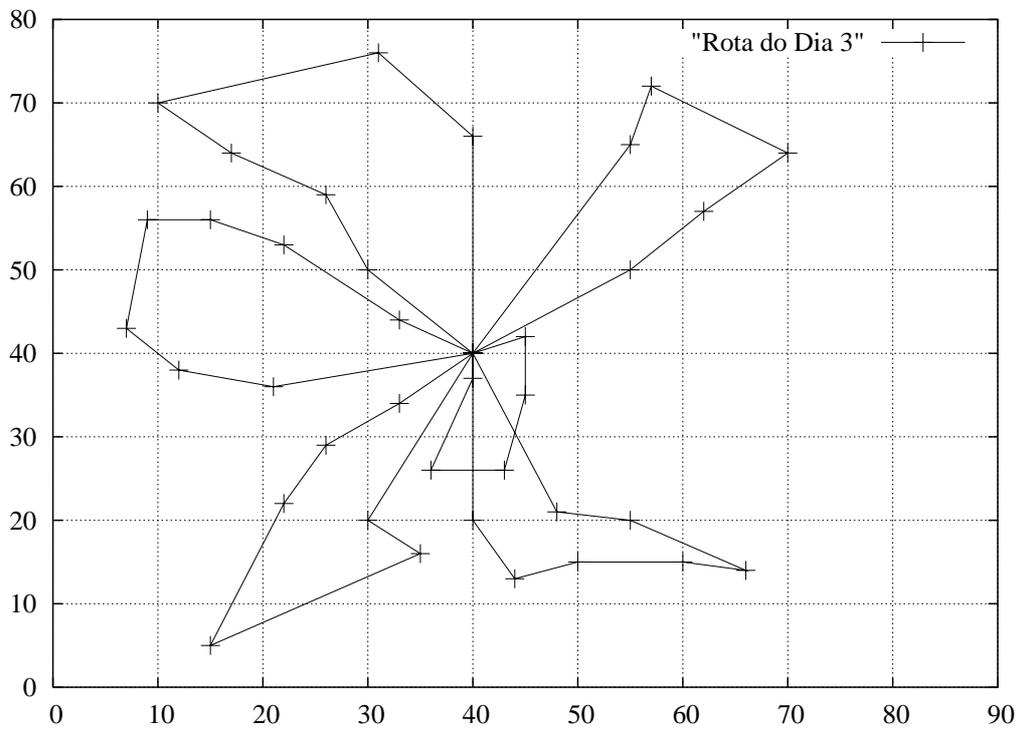
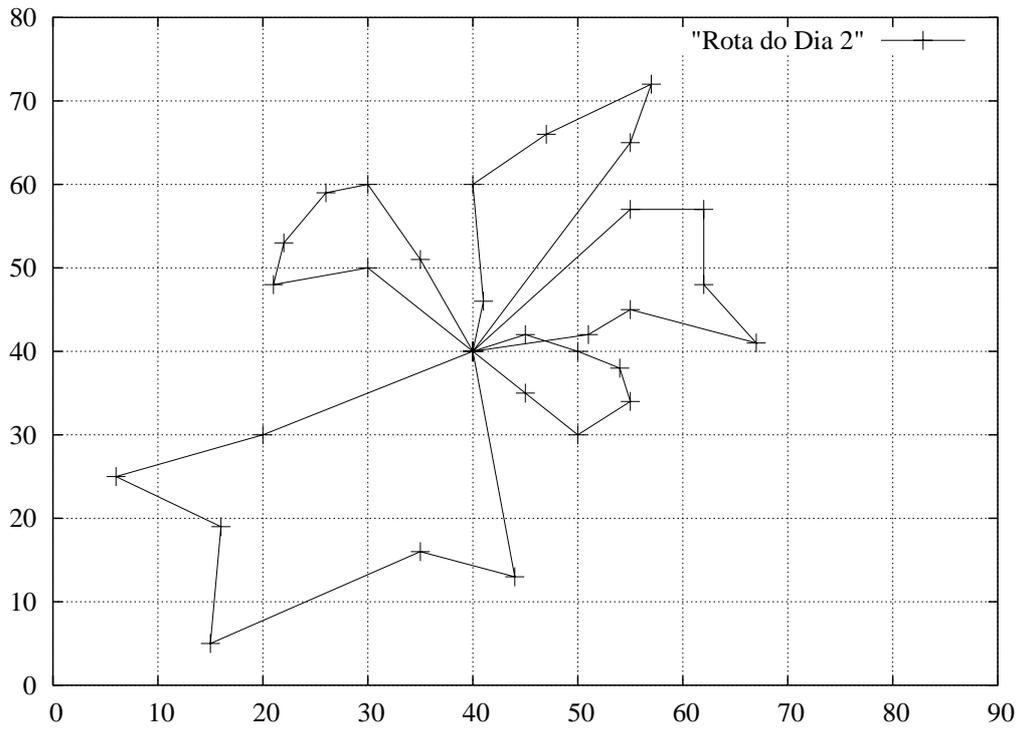


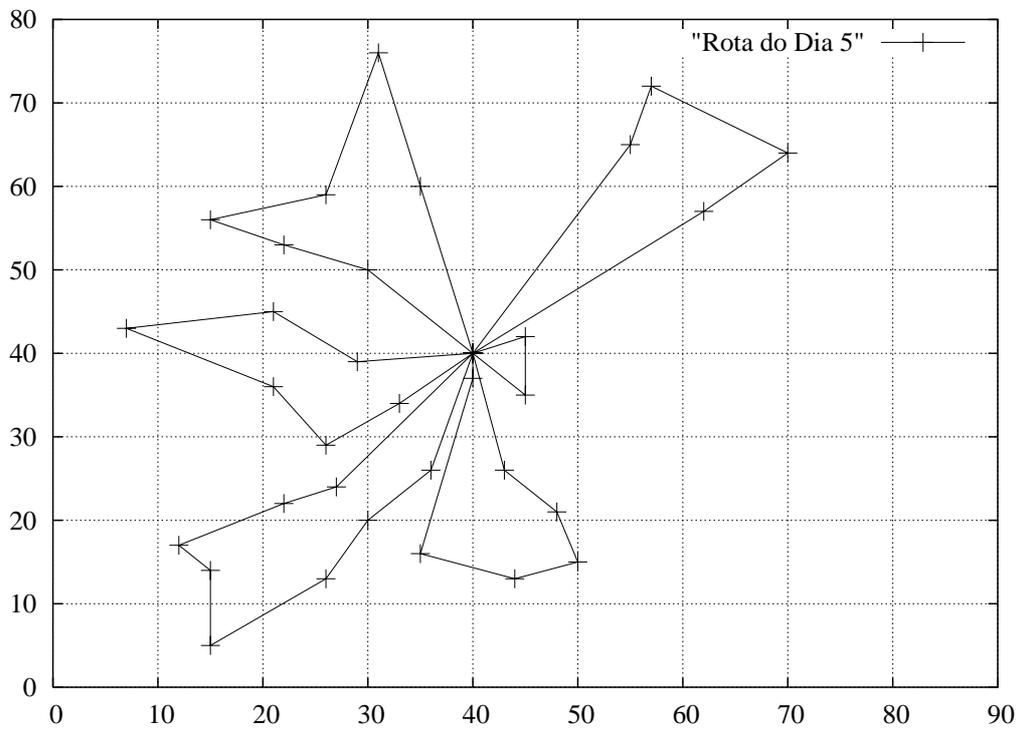
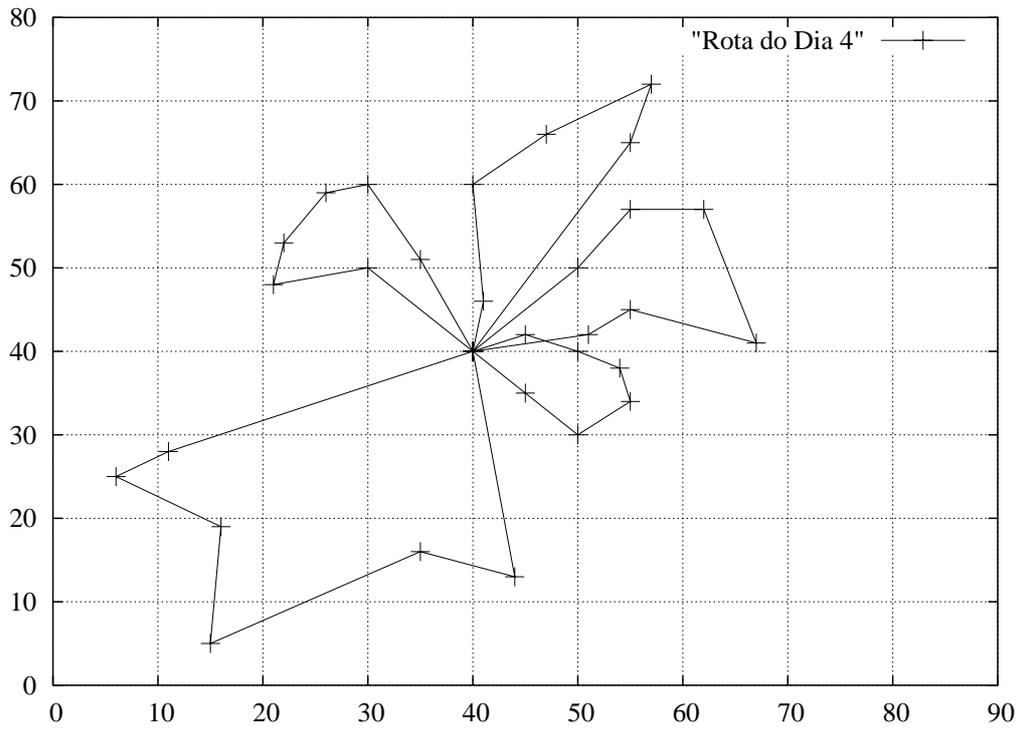




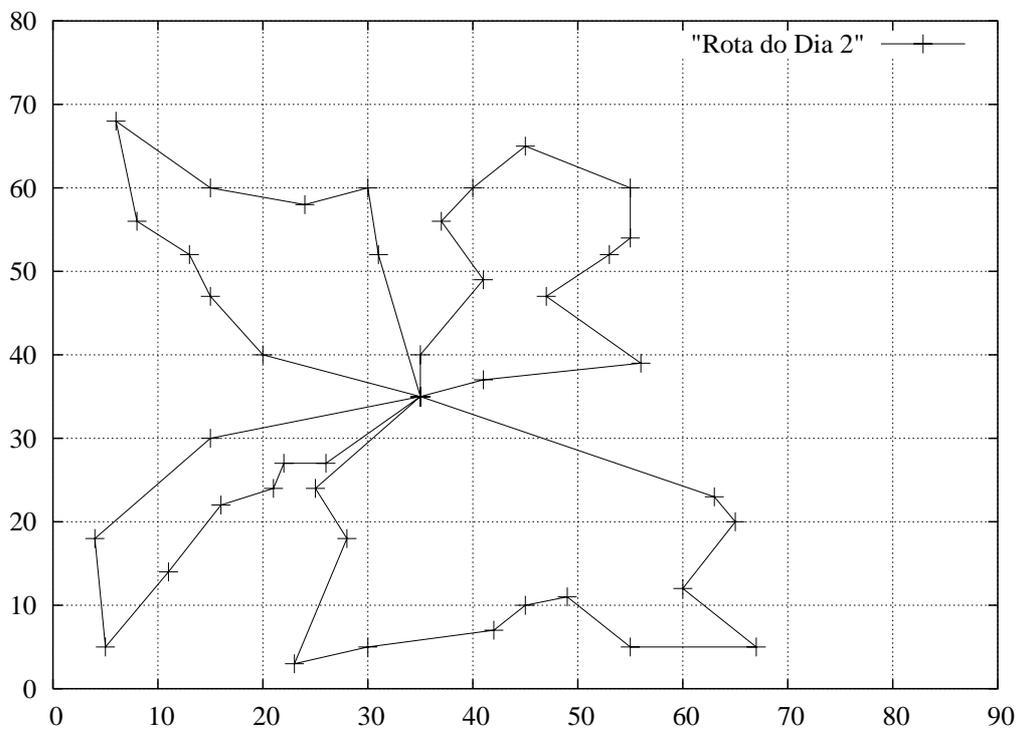
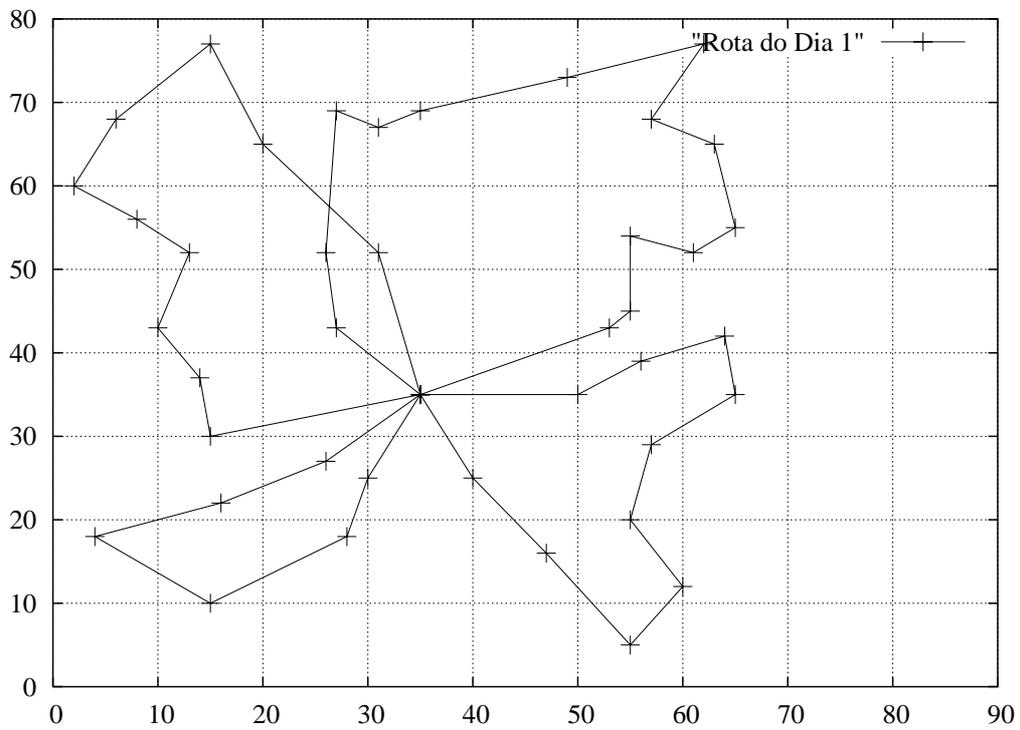
Rotas obtidas para o problema 75b (custo de 2084,25):

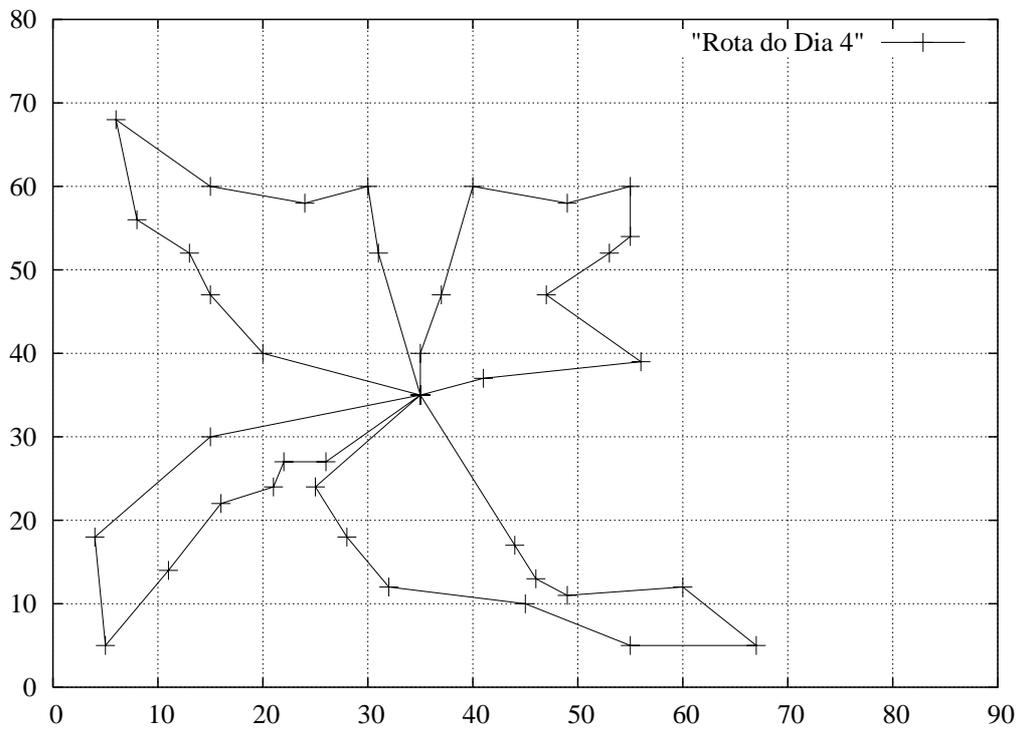
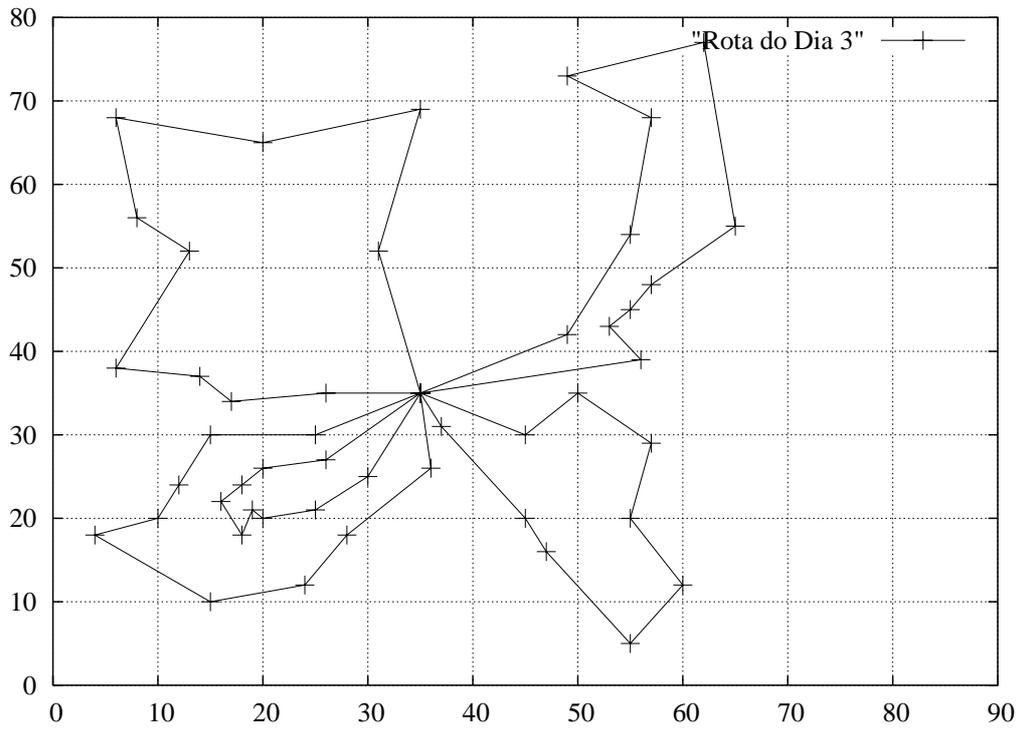


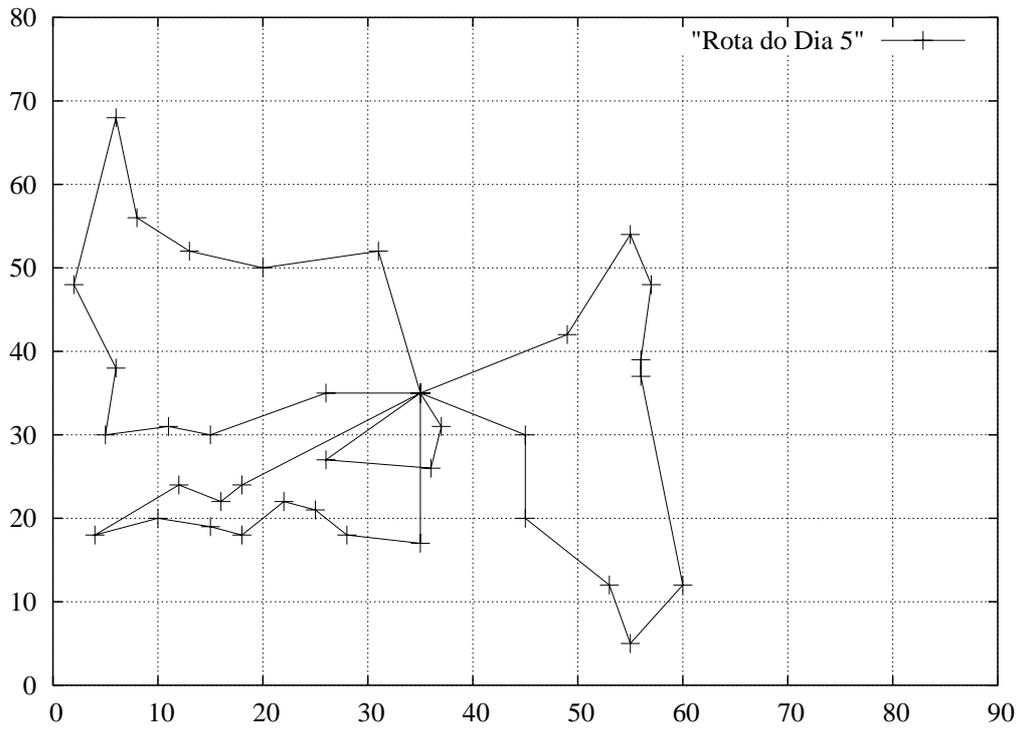




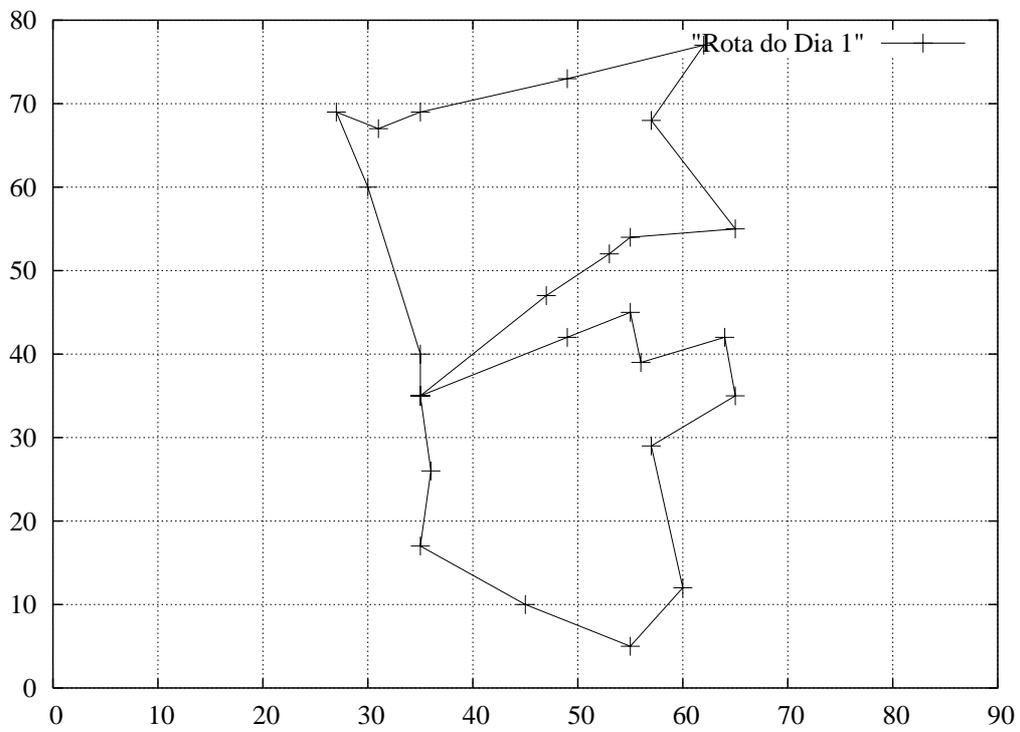
Rotas obtidas para o problema 100b (custo de 2164,41):

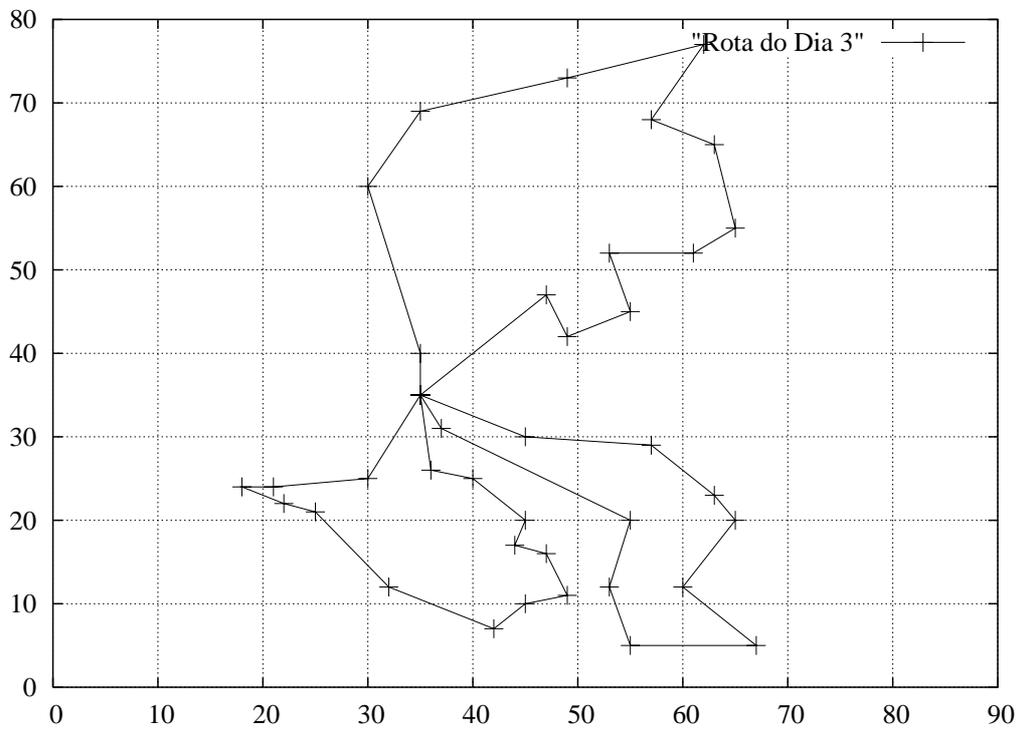
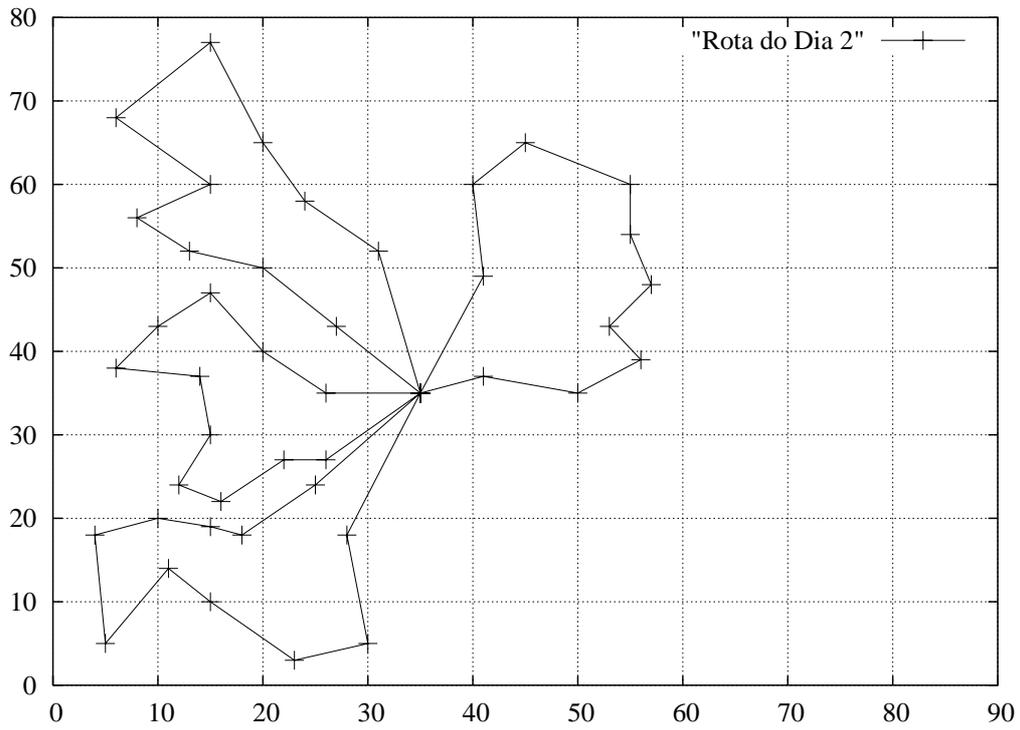


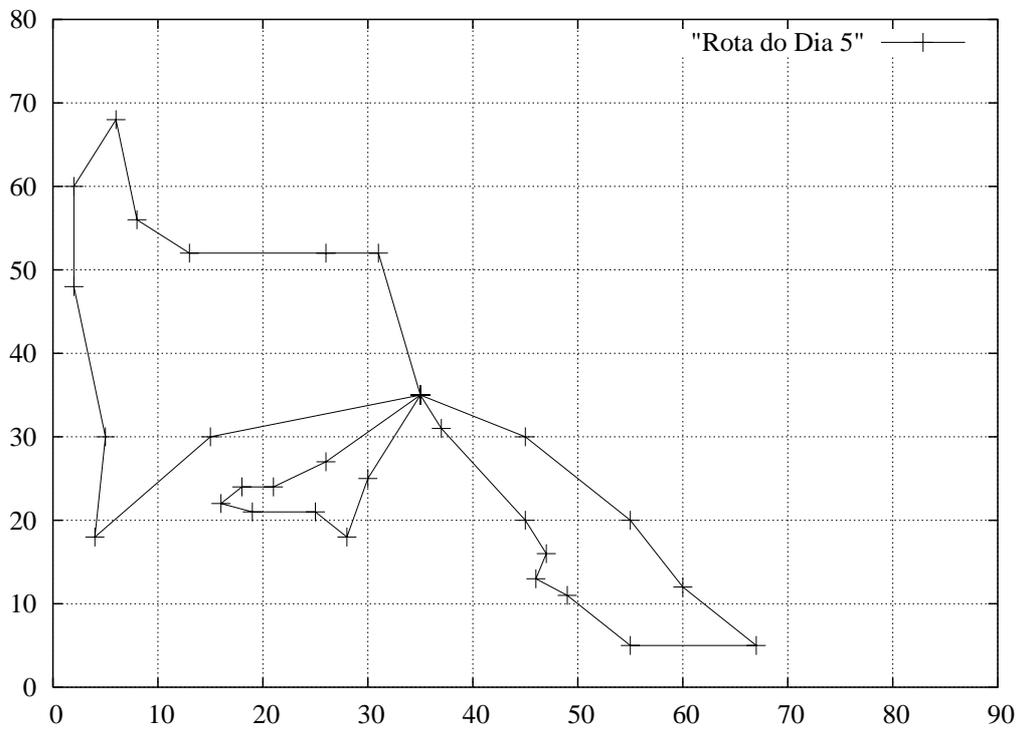
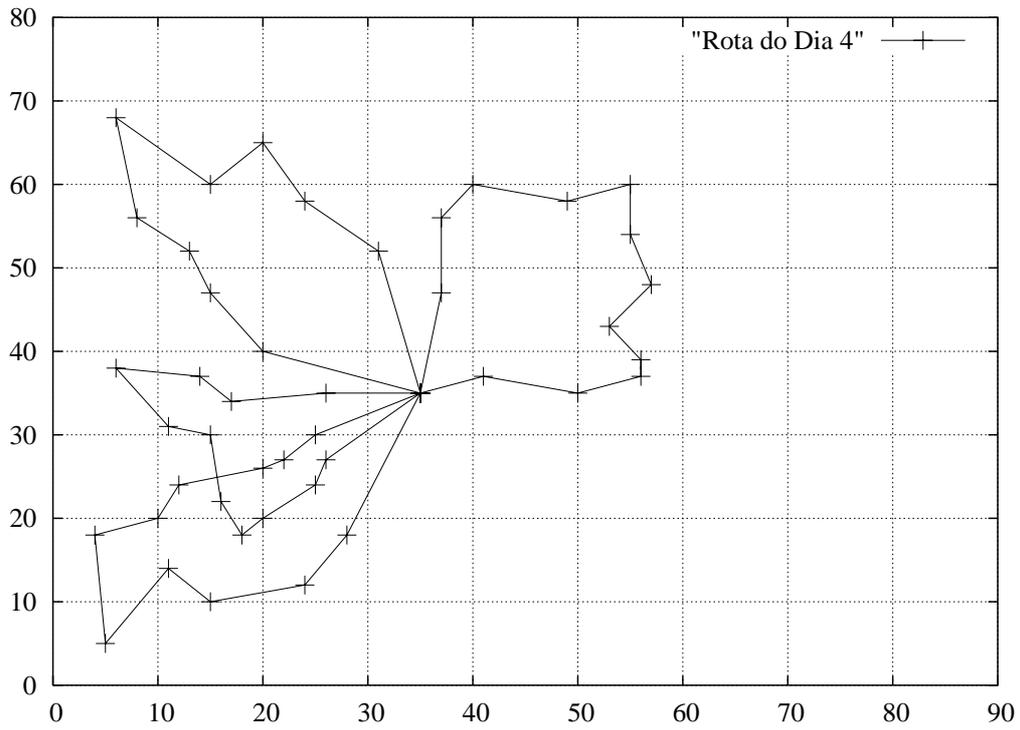




Rotas obtidas para o problema 100d (custo de 1676.28):





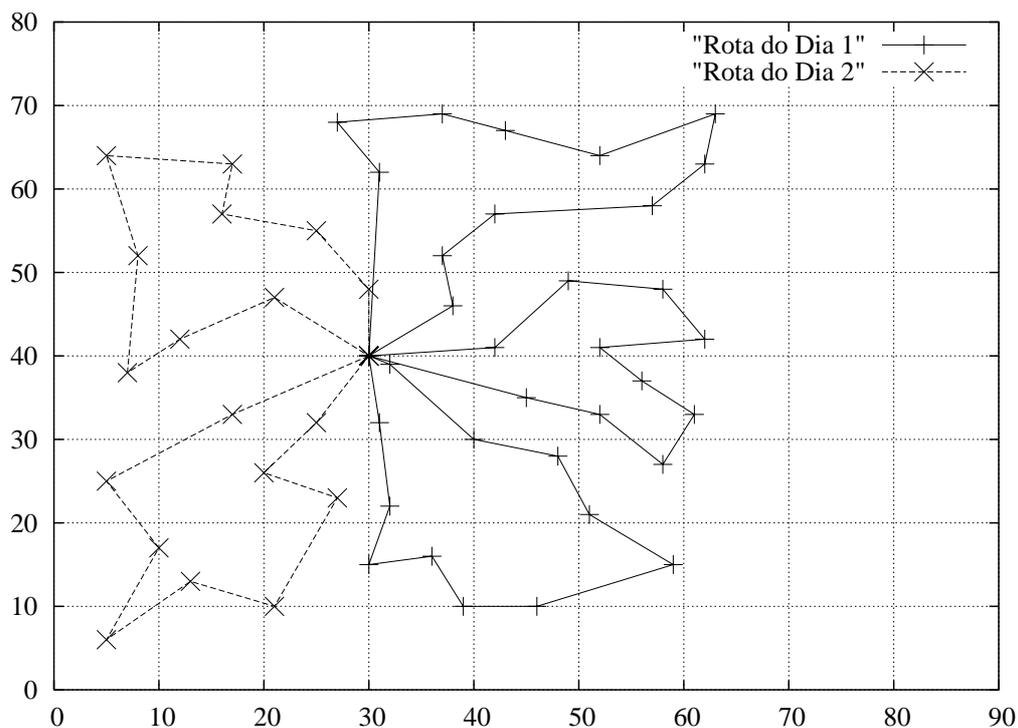


Anexo B: Rotas geradas pelo programa

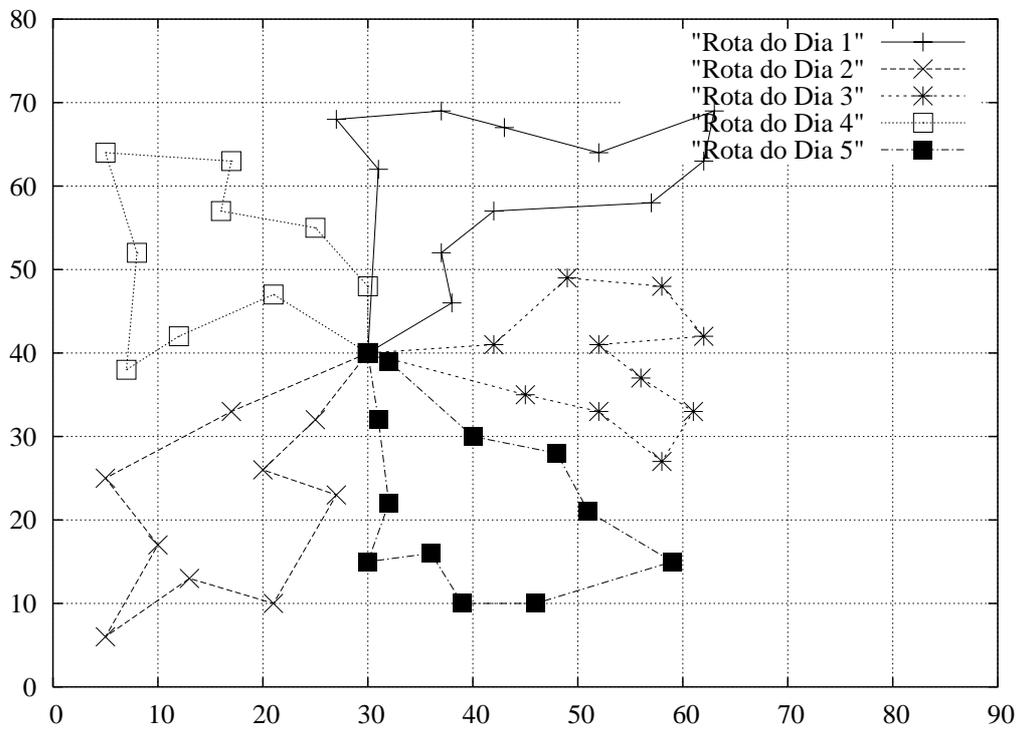
Versão 2

Neste anexo são apresentados os resultados obtidos para os problemas tratados na VERSÃO 2. São mostrados inicialmente os resultados dos problemas cujos clientes têm frequência 1 (problemas (a) e (c)) e em seguida os resultados dos problemas com frequência maior que 1 (problemas (b) e (d)). A distribuição geográfica dos clientes é a mesma da apresentada no anexo A.

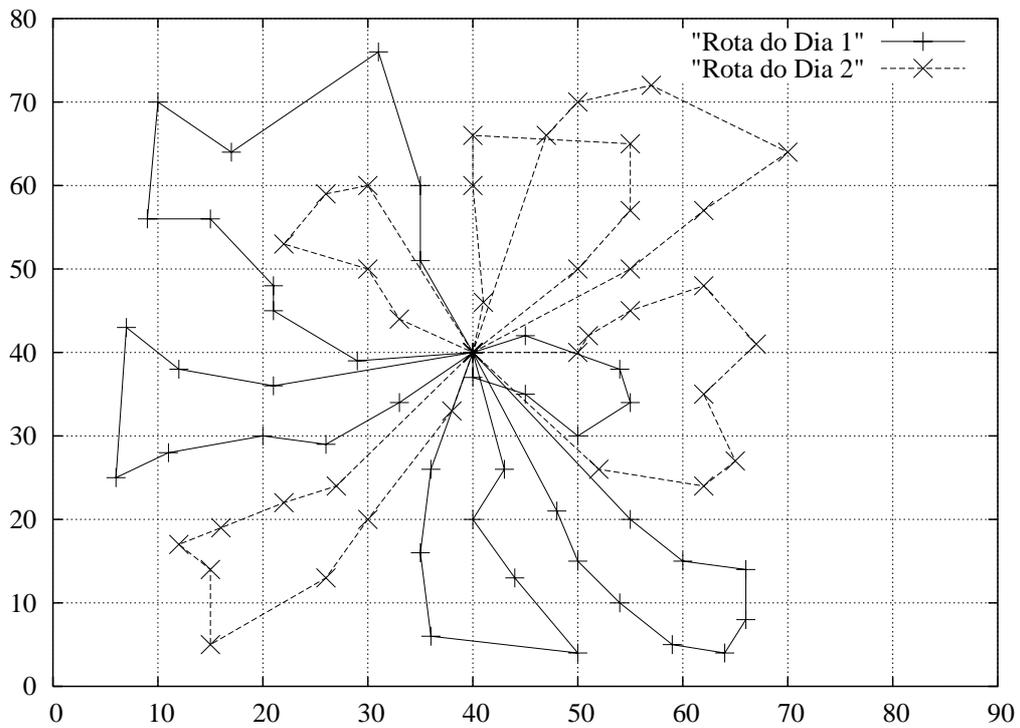
Rotas obtidas para o problema 50a (custo de 524,61):



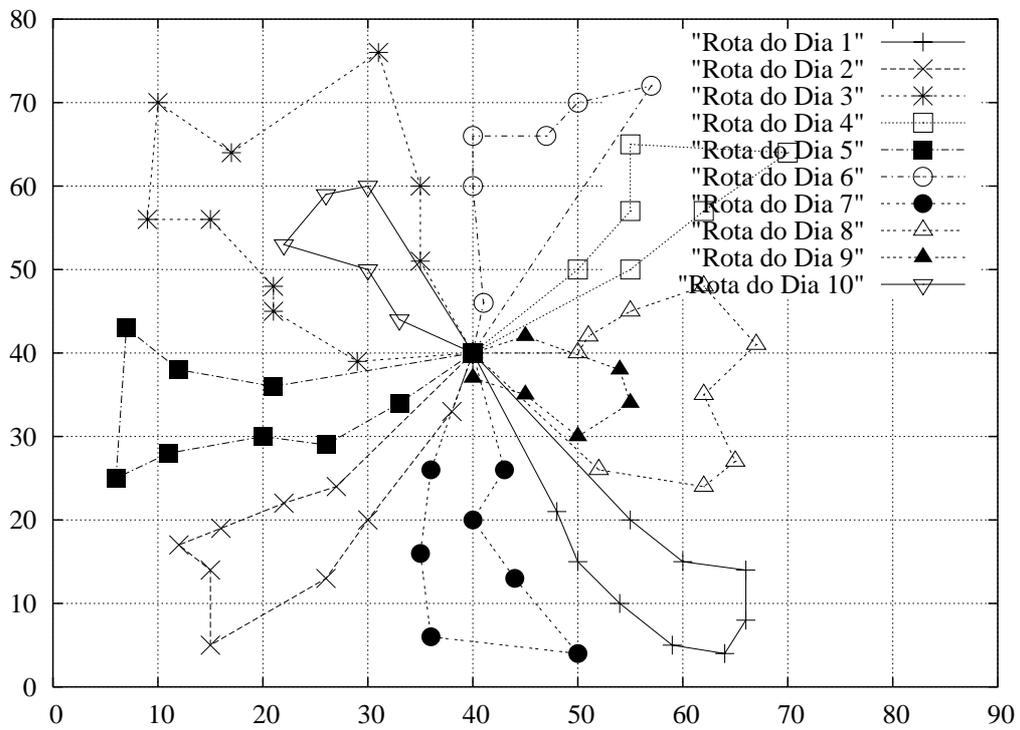
Rotas obtidas para o problema 50c (custo de 524,61):



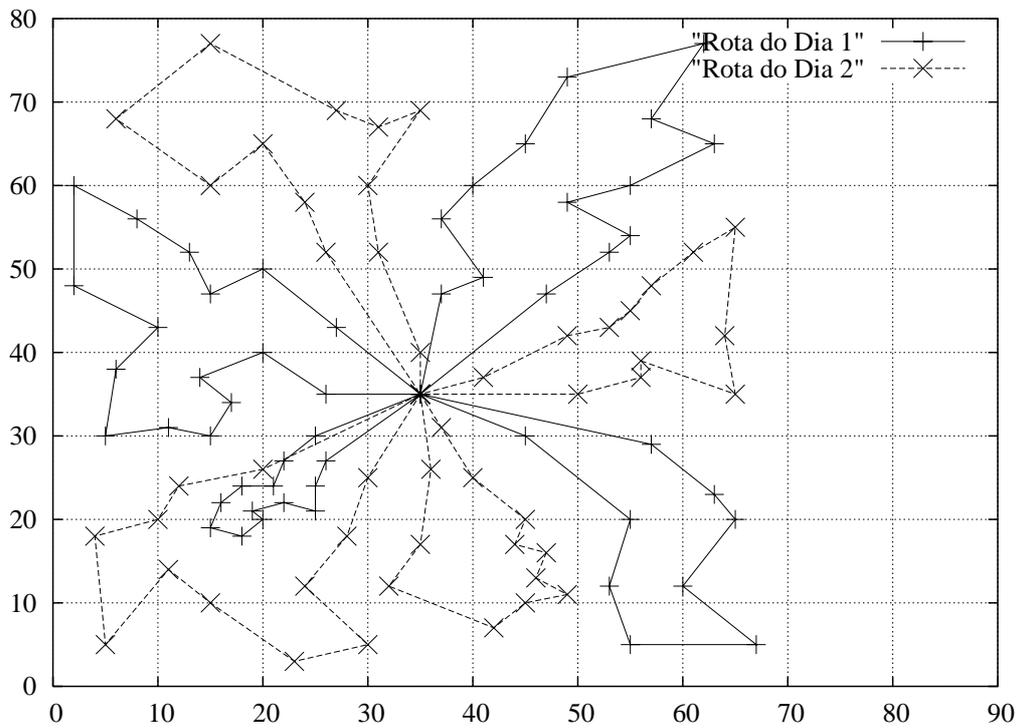
Rotas obtidas para o problema 75a (custo de 835,32):



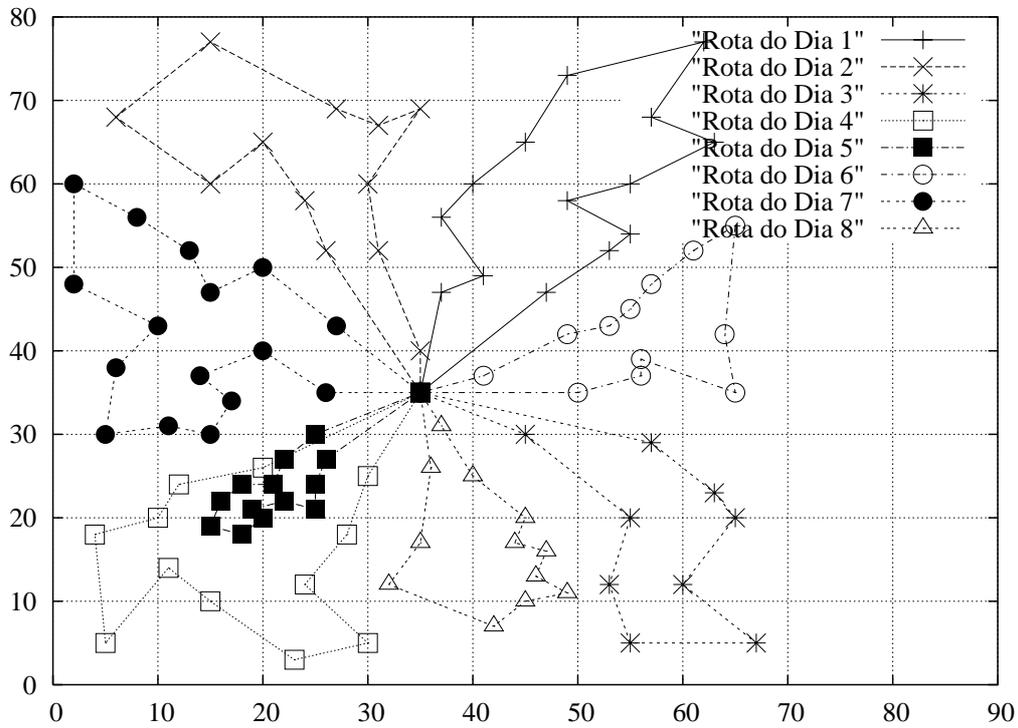
Rotas obtidas para o problema 75c (custo de 836,36):



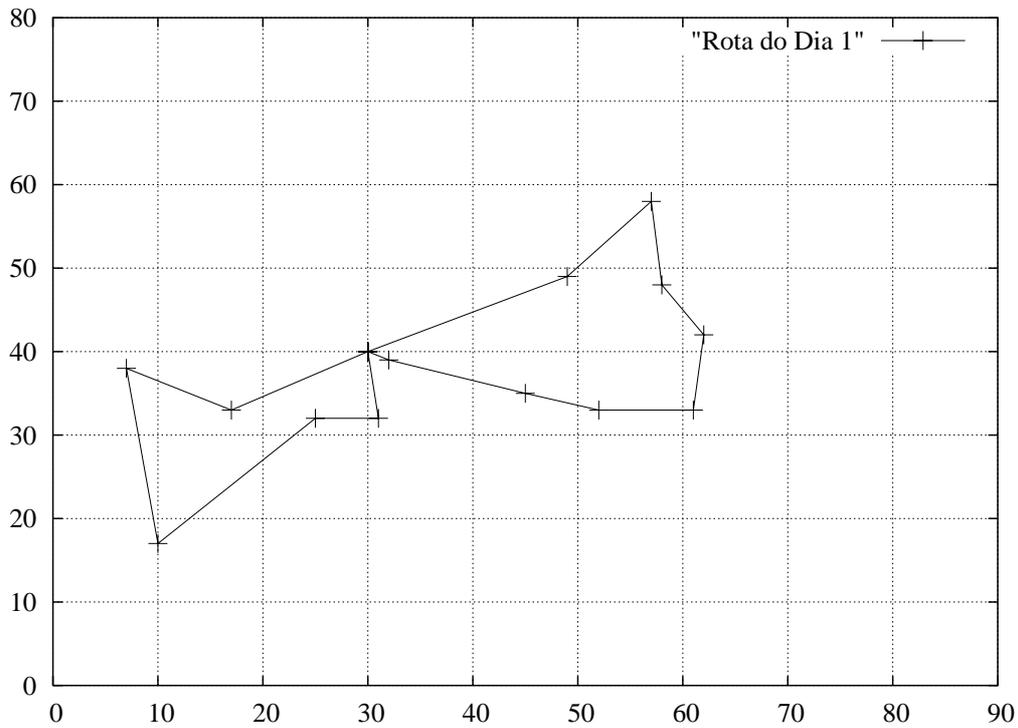
Rotas obtidas para o problema 100a (custo de 826,63):

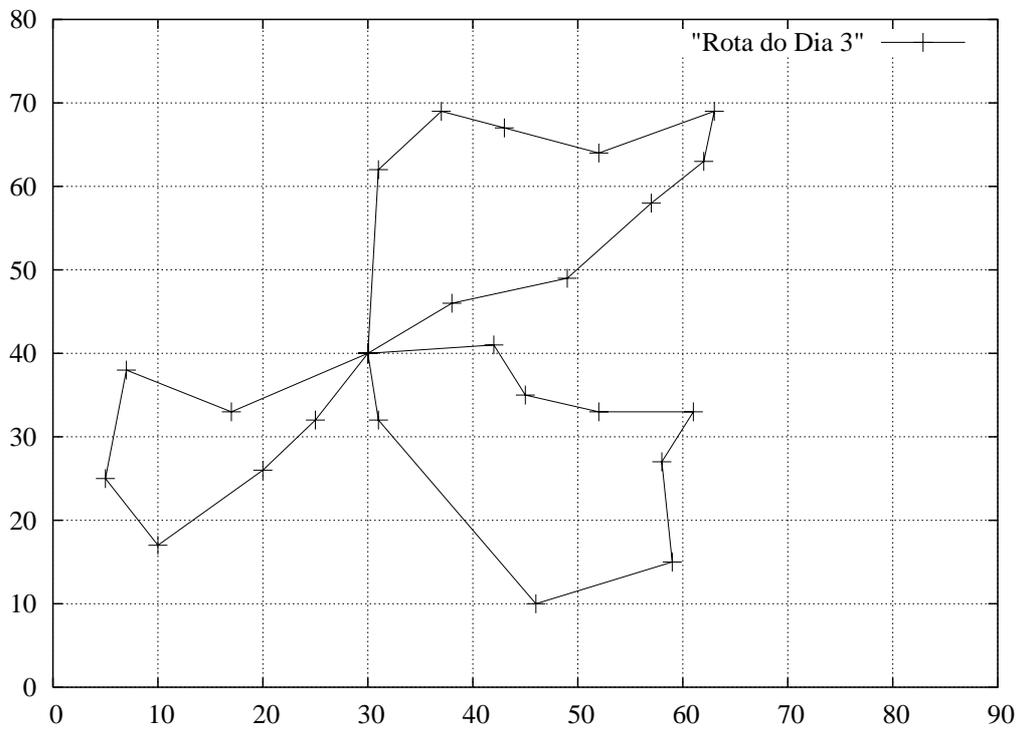
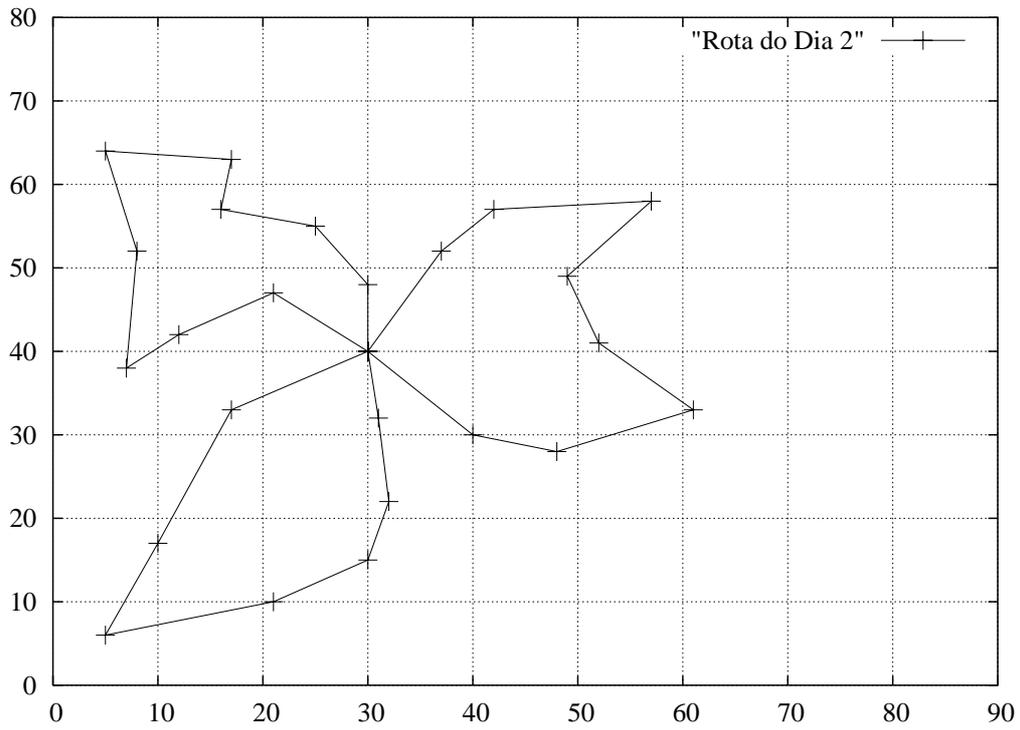


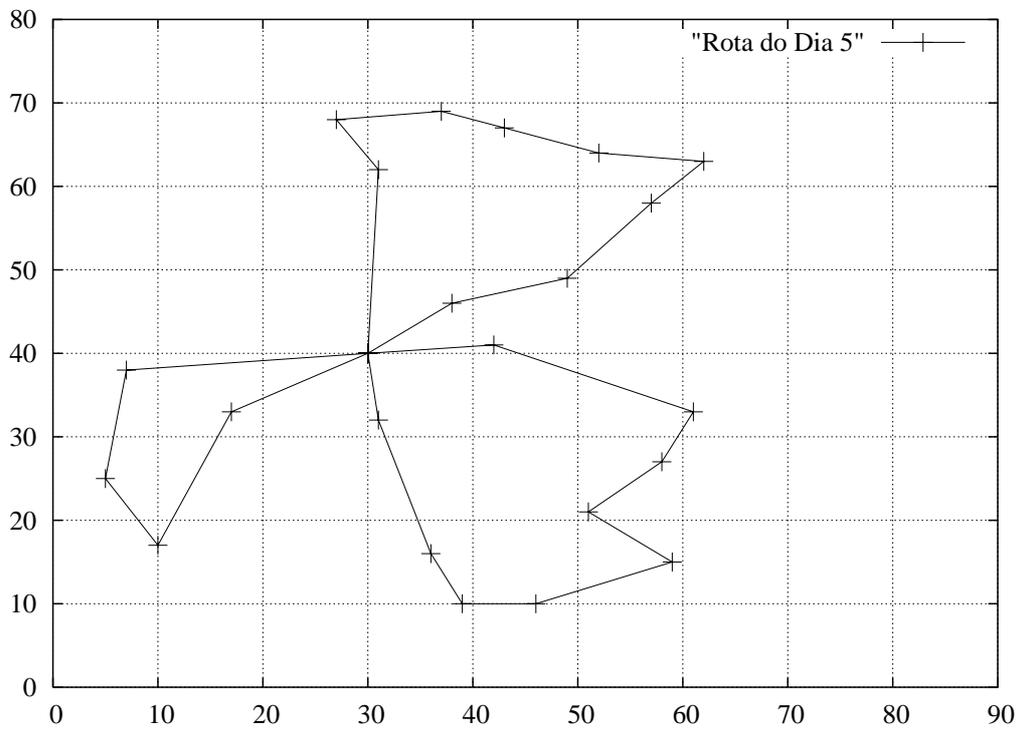
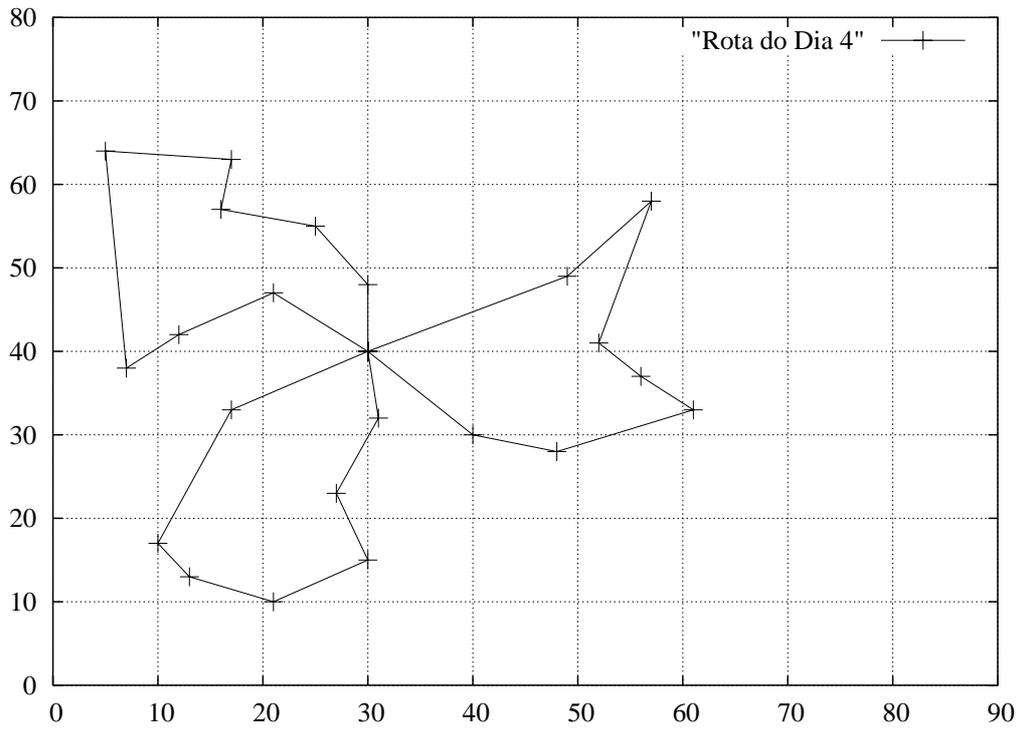
Rotas obtidas para o problema 100c (custo de 826,63):



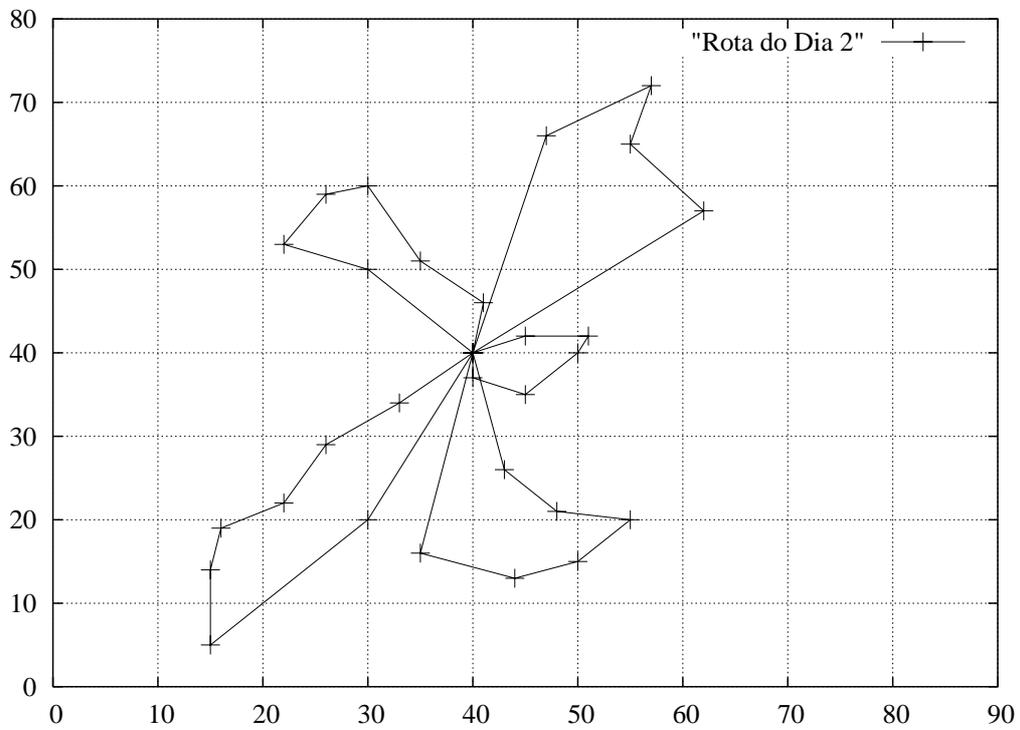
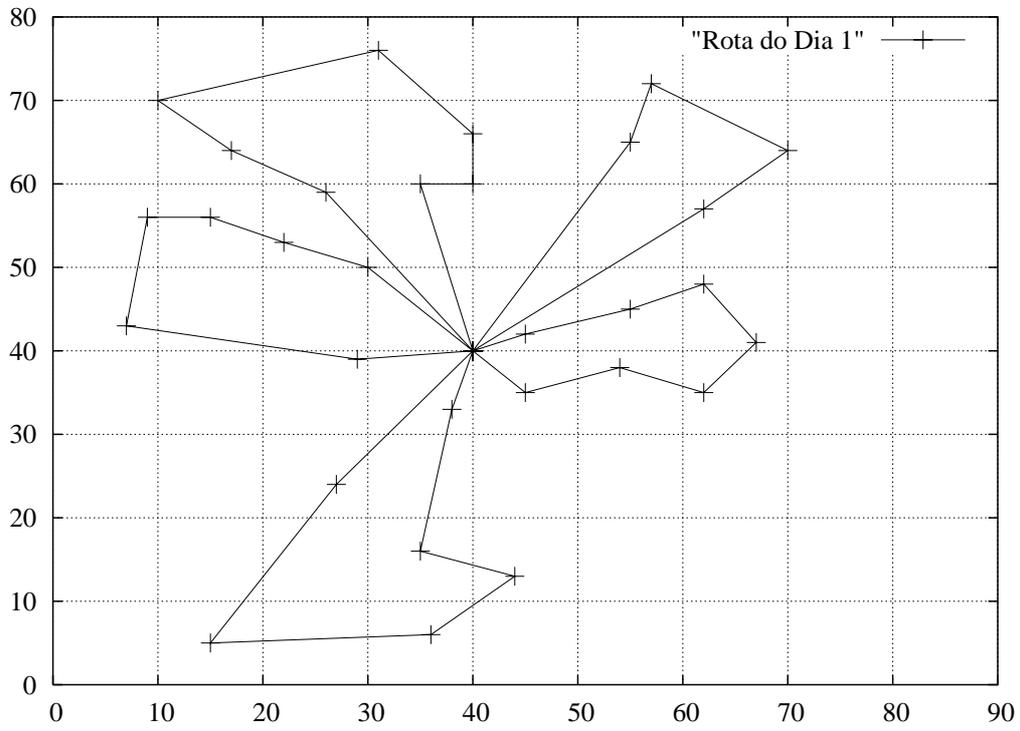
Rotas obtidas para o problema 50b (custo de 1335,59):

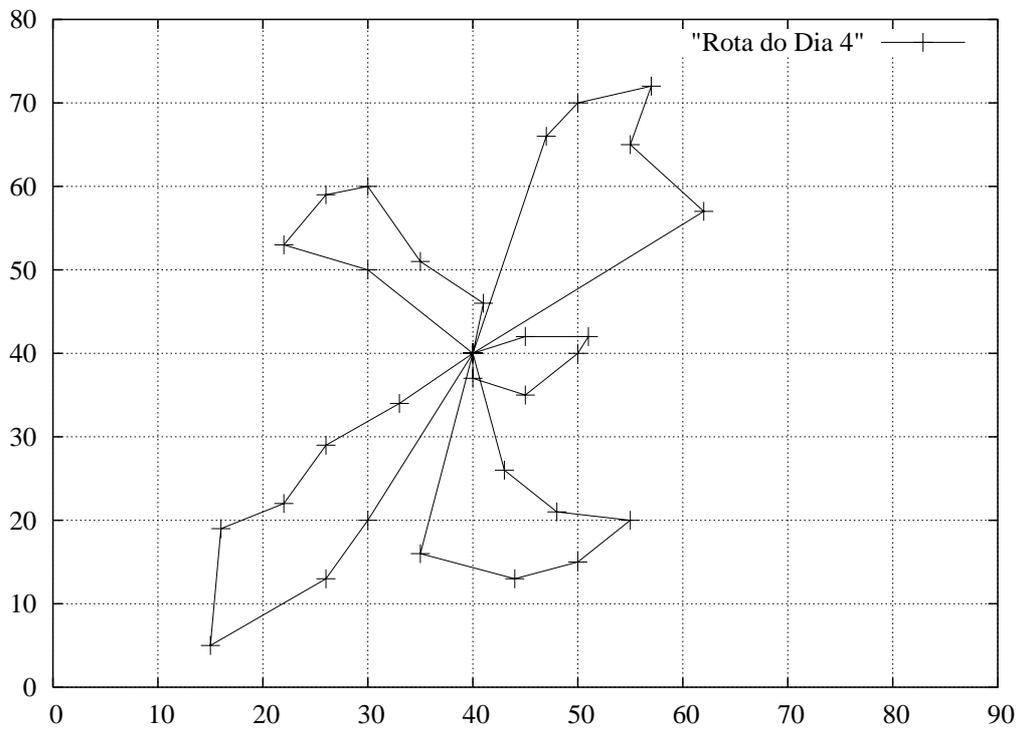
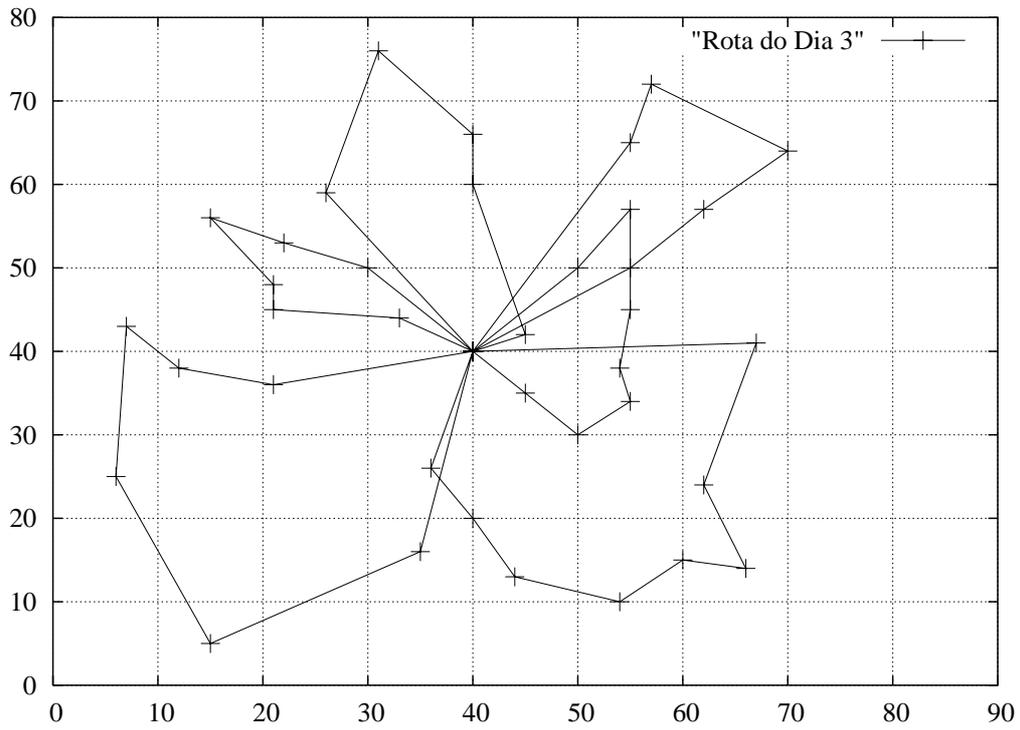


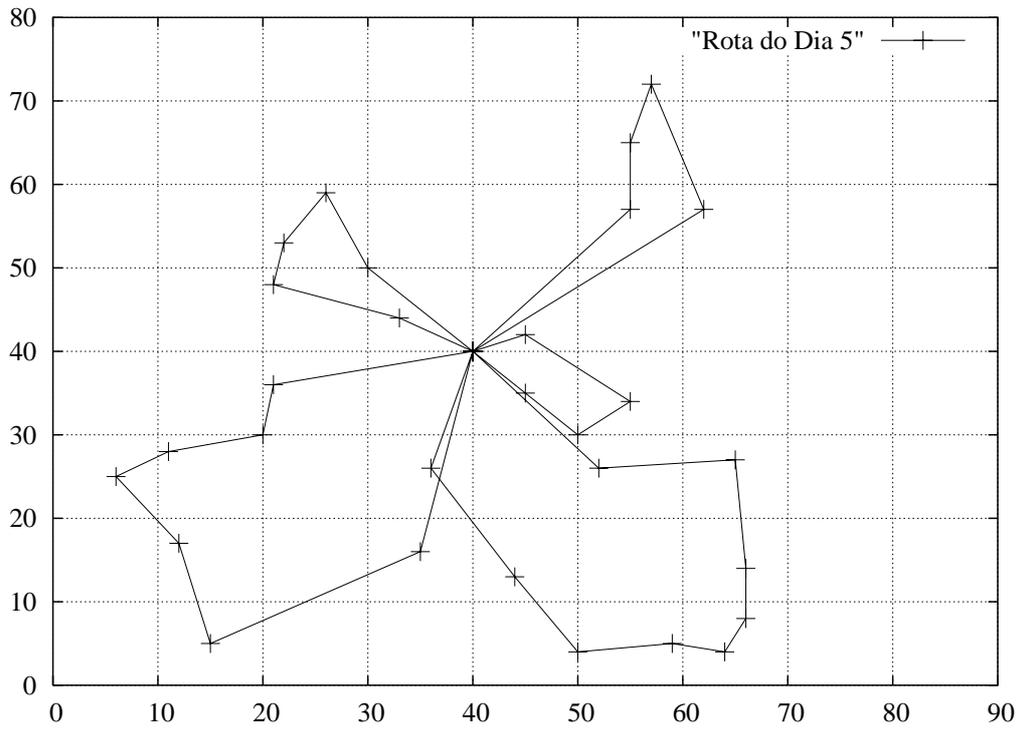




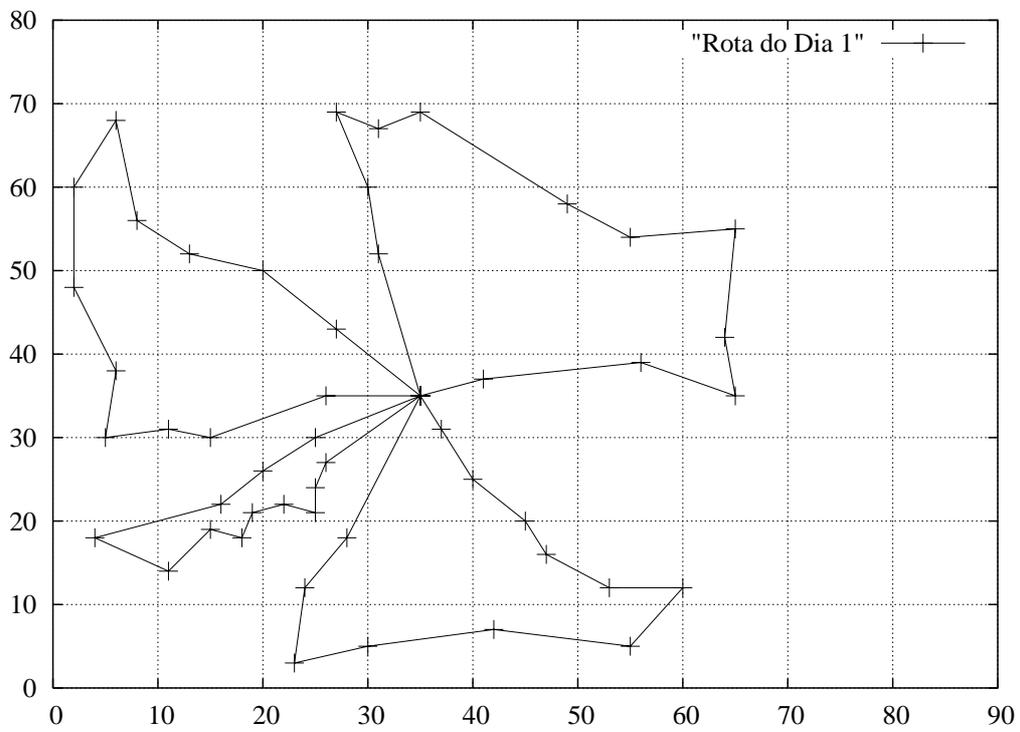
Rotas obtidas para o problema 75b (custo de 2068,2):

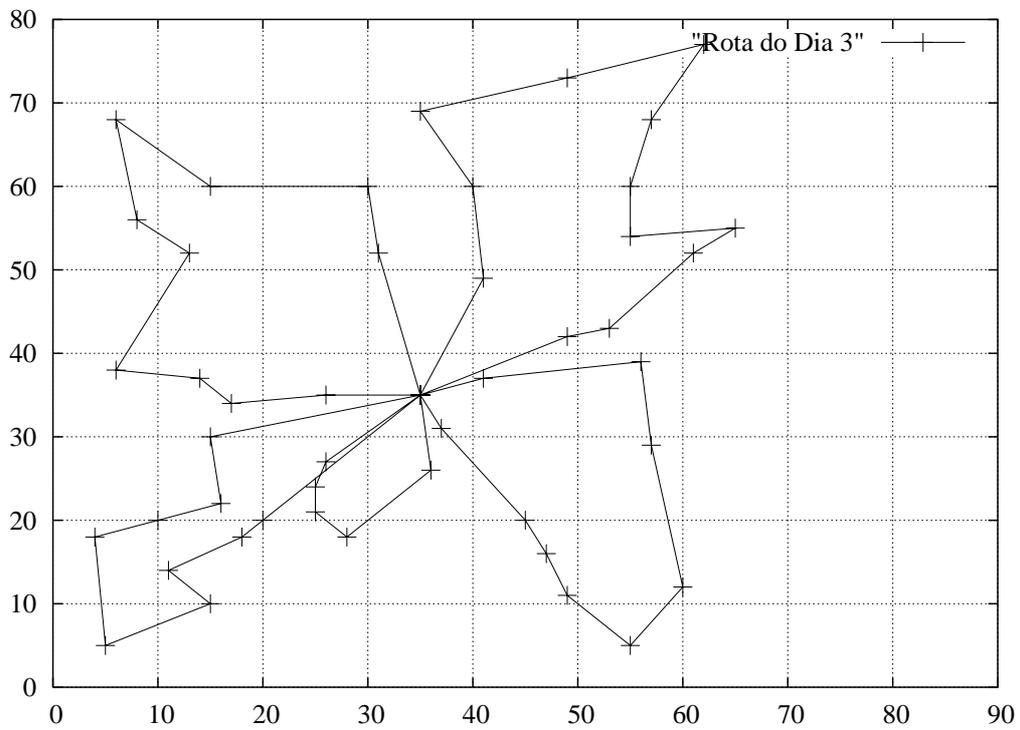
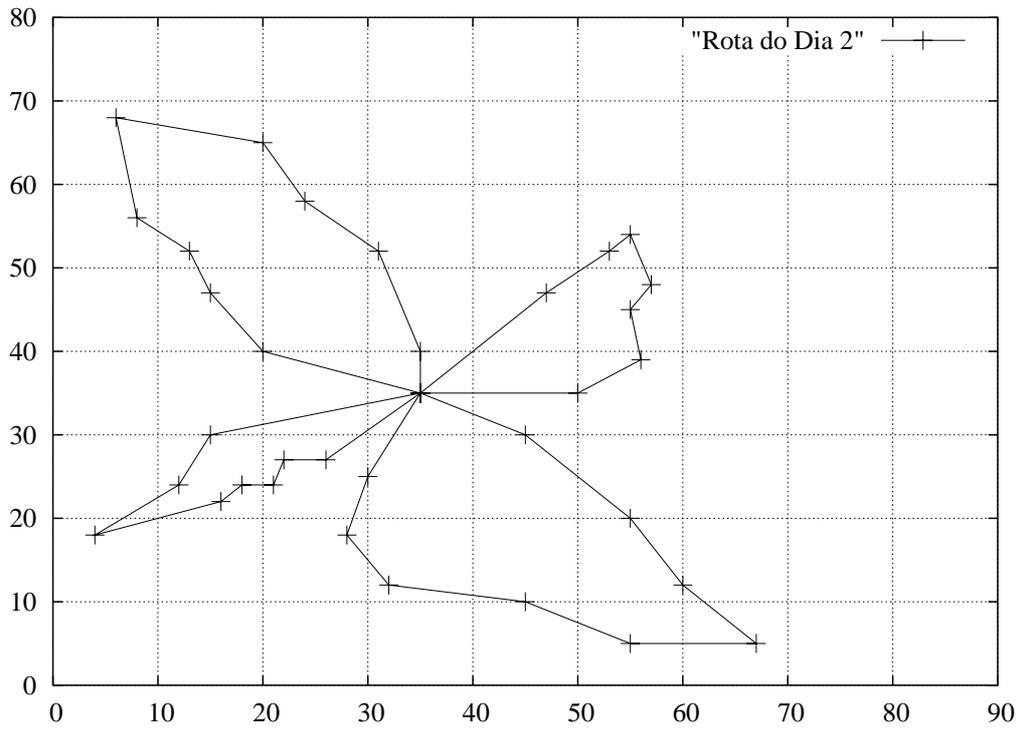


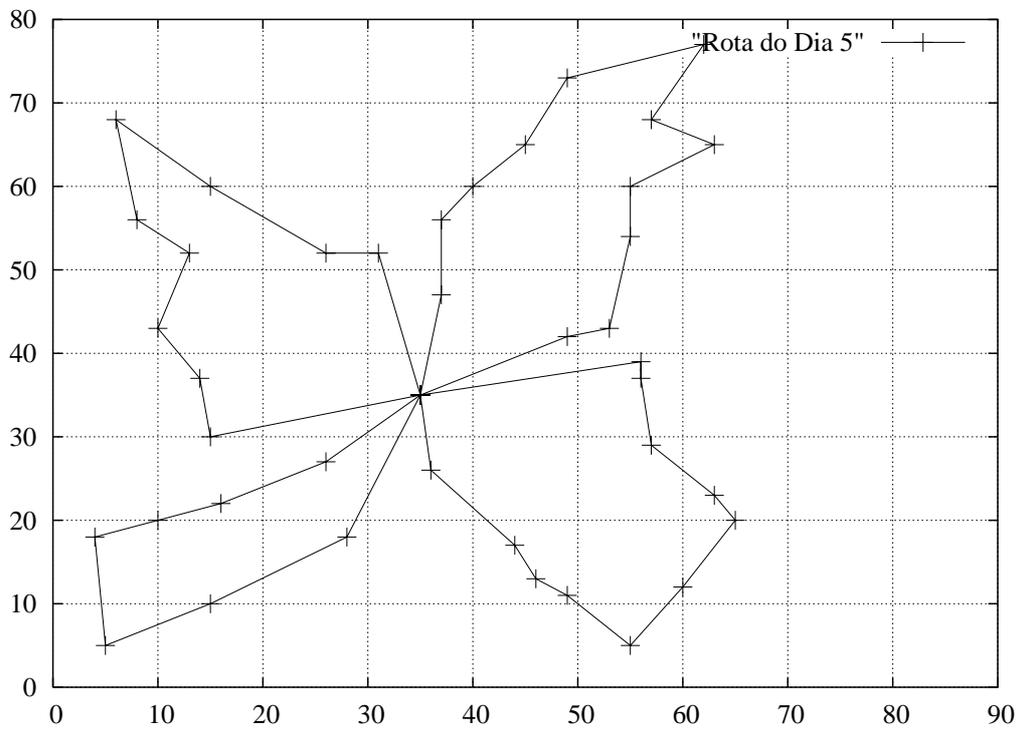
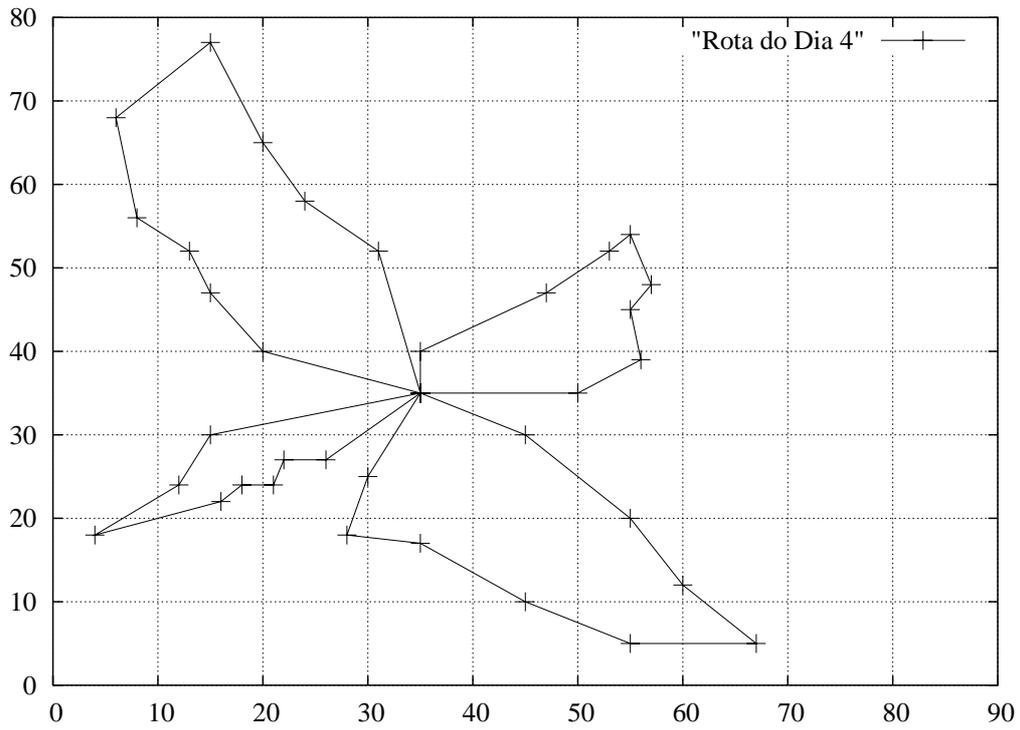




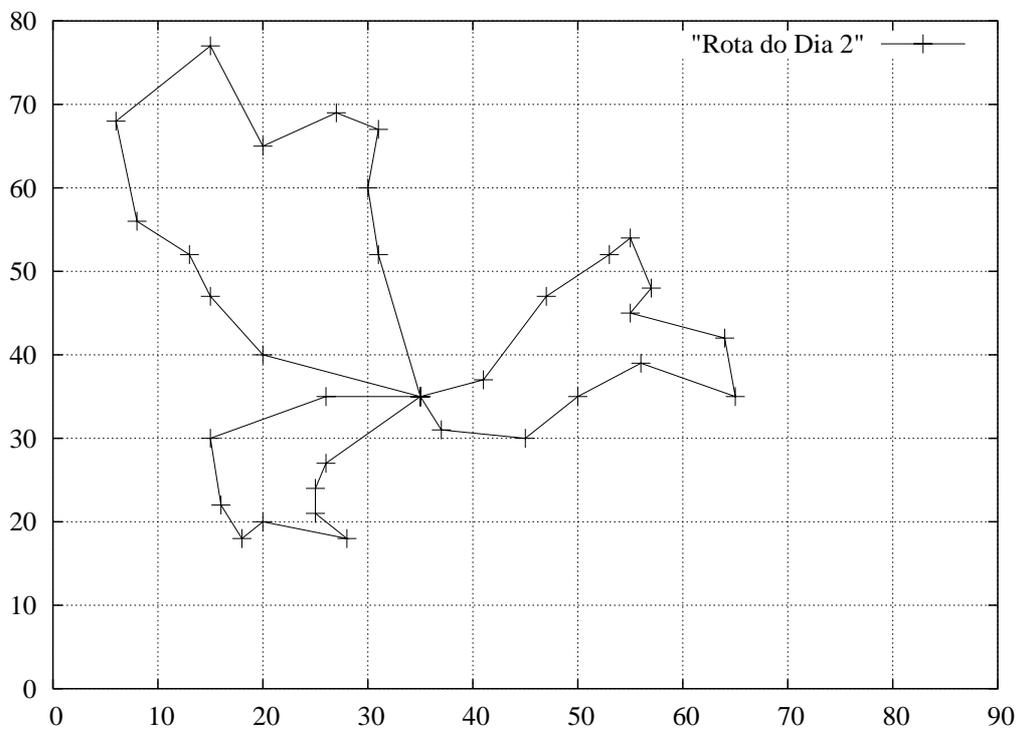
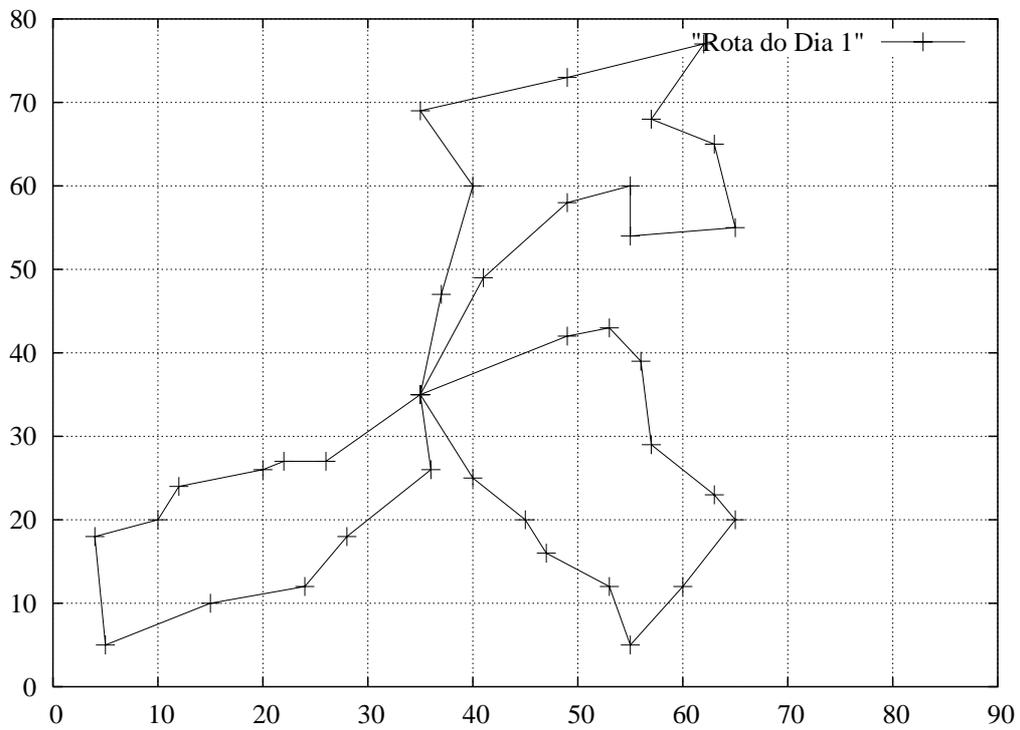
Rotas obtidas para o problema 100b (custo de 2061,25):

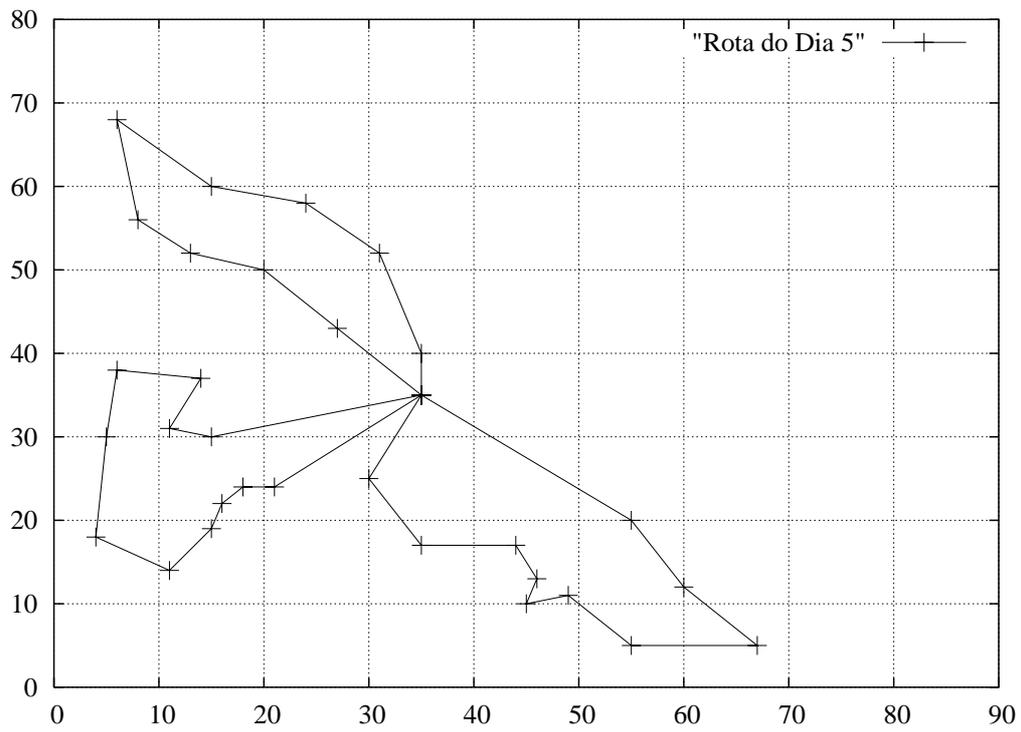






Rotas obtidas para o problema 100d (custo de 1652.88):





Anexo C: Dados para os problemas com 50, 75 e 100 clientes

As Tabelas 1, 2 e 3 apresentam os dados para os problemas que envolvem 50, 75 e 100 clientes, respectivamente.

Tabela 1: Dados para os problemas com 50 clientes.

Cientes	Coordenada x	Coordenada y	Demanda
depósito	30	40	0
1	37	52	7
2	49	49	30
3	52	64	16
4	20	26	9
5	40	30	21
6	21	47	15
7	17	63	19
8	31	62	23
9	52	33	11
10	51	21	5
11	42	41	19
12	31	32	29
13	5	25	23
14	12	42	21

Tabela 1: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
15	36	16	10
16	52	41	15
17	27	23	3
18	17	33	41
19	13	13	9
20	57	58	28
21	62	42	8
22	42	57	8
23	16	57	16
24	8	52	10
25	7	38	28
26	27	68	7
27	30	48	15
28	43	67	14
29	58	48	6
30	58	27	19
31	37	69	11
32	38	46	12
33	46	10	23
34	61	33	26
35	62	63	17
36	63	69	6
37	32	22	9
38	45	35	15
39	59	15	14
40	5	6	7
41	10	17	27

Tabela 1: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
42	21	10	13
43	5	64	11
44	30	15	16
45	39	10	10
46	32	39	5
47	25	32	25
48	25	55	17
49	48	28	18
50	56	37	10

Tabela 2: Dados para os problemas com 75 clientes.

Cientes	Coordenada x	Coordenada y	Demanda
depósito	40	40	0
1	22	22	18
2	36	26	26
3	21	45	11
4	45	35	30
5	55	20	21
6	33	34	19
7	50	50	15
8	55	45	16
9	26	59	29
10	40	66	26
11	55	65	37
12	35	51	16
13	62	35	12

Tabela 2: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
14	62	57	31
15	62	24	8
16	21	36	19
17	33	44	20
18	9	56	13
19	62	48	15
20	66	14	22
21	44	13	28
22	26	13	12
23	11	28	6
24	7	43	27
25	17	64	14
26	41	46	18
27	55	34	17
28	35	16	29
29	52	26	13
30	43	26	22
31	31	76	25
32	22	53	28
33	26	29	27
34	50	40	19
35	55	50	10
36	54	10	12
37	60	15	14
38	47	66	24
39	30	60	16
40	30	50	33

Tabela 2: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
41	12	17	15
42	15	14	11
43	16	19	18
44	21	48	17
45	50	30	21
46	51	42	27
47	50	15	19
48	48	21	20
49	12	38	5
50	15	56	22
51	29	39	12
52	54	38	19
53	55	57	22
54	67	41	16
55	10	70	7
56	6	25	26
57	65	27	14
58	40	60	21
59	70	64	24
60	64	4	13
61	36	6	15
62	30	20	18
63	20	30	11
64	15	5	28
65	50	70	9
66	57	72	37
67	45	42	30

Tabela 2: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
68	38	33	10
69	50	4	8
70	66	8	11
71	59	5	3
72	35	60	1
73	27	24	6
74	40	20	10
75	40	37	20

Tabela 3: Dados para os problemas com 100 clientes.

Cientes	Coordenada x	Coordenada y	Demanda
depósito	35	35	0
1	41	49	10
2	35	17	7
3	55	45	13
4	55	20	19
5	15	30	26
6	25	30	3
7	20	50	5
8	10	43	9
9	55	60	16
10	30	60	16
11	20	65	12
12	50	35	19
13	30	25	23
14	15	10	20

Tabela 3: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
15	30	5	8
16	10	20	19
17	5	30	2
18	20	40	12
19	15	60	17
20	45	65	9
21	45	20	11
22	45	10	18
23	55	5	29
24	65	35	3
25	65	20	6
26	45	30	17
27	35	40	16
28	41	37	16
29	64	42	9
30	40	60	21
31	31	52	27
32	35	69	23
33	53	52	11
34	65	55	14
35	63	65	8
36	2	60	5
37	20	20	8
38	5	5	16
39	60	12	31
40	40	25	9
41	42	7	5

Tabela 3: (continuação)

Clientes	Coordenada x	Coordenada y	Demanda
42	24	12	5
43	23	3	7
44	11	14	18
45	6	38	16
46	2	48	1
47	8	56	27
48	13	52	36
49	6	68	30
50	47	47	13
51	49	58	10
52	27	43	9
53	37	31	14
54	57	29	18
55	63	23	2
56	53	12	6
57	32	12	7
58	36	26	18
59	21	24	18
60	17	34	3
61	12	24	13
62	24	58	19
63	27	69	10
64	15	77	9
65	62	77	20
66	49	73	25
67	67	5	25
68	56	39	36

Tabela 3: (continuação)

Cientes	Coordenada x	Coordenada y	Demanda
69	37	47	6
70	37	56	5
71	57	68	15
72	47	16	25
73	44	17	9
74	46	13	8
75	49	11	18
76	49	42	13
77	53	43	14
78	61	52	3
79	57	48	23
80	56	37	6
81	55	54	26
82	15	47	16
83	14	37	11
84	11	31	7
85	16	22	41
86	4	18	35
87	28	18	26
88	26	52	9
89	26	35	15
90	31	67	3
91	15	19	1
92	22	22	2
93	18	24	22
94	26	27	27
95	25	24	20

Tabela 3: (continuação)

Clientes	Coordenada x	Coordenada y	Demanda
96	22	27	11
97	25	21	12
98	19	21	10
99	20	26	9
100	18	18	17